MATHEMATICAL AND EXPERIMENTAL STUDIES OF MICROBIAL PROCESSES WITH LAG EFFECTS

Thesis by

Nam Sun Wang

In Partial Fulfillment of the Requirement

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California 91125

1989

(Submitted November 8, 1985)

Very special thanks
to Greg
for his insight
and to Patty
for her love,
always and forever.

ABSTRACT

Unlike most chemical reaction dynamics, microbial behavior depends not only on the present state of the environment surrounding a microorganism but, more importantly, on its past history as well. Herein lies a major obstacle in the modeling of a biological process with a simple set of equations. By incorporating a culture's past history in the form of a time-lag kernel, a novel approach to bioprocess identification and modeling is formulated. A time-lag kernel is included in the state equations, and a generalized method of mathematical simplification via the transformation of an integro-differential equation to a set of first-order ODE's is developed. The time-lag convolution integral arises during the process of transforming a structured, mechanistic model into an equivalent unstructured model as a result of lumping. The resulting model possesses the combined advantages of the simplicity of an unstructured, lumped-parameter model and the predictive power of a complex structured model. The experimental determination of the kernel is performed by cultivating *Saccharomyces cerevisiae* in a chemically defined medium of either glucose or ethanol as the limiting carbon source and in a tightly controlled environment of temperature and pH. All the model parameters can be feasibly resolved with a simple set of experiments. The validity of the time-lag modeling approach is clearly demonstrated experimentally by its superior capability in predicting the various transient responses under different modes of operation. Seemingly unreproducible experiments are shown to be united when time-lag effects are taken into consideration. This modeling work fits within the general framework of on-line computer parameter estimation, control, and optimization of a biochemical reactor. As such, the proposed modeling approach to biological systems identifies the cause-effect relationship more clearly and is well suited for process control purposes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

It has been pointed out that two important problems in the optimal design and operation of a biological reactor are the lack of reliable biological sensors and the lack of simple mathematical models with satisfactory predictive capability (Cooney, 1979; Wang and Stephanopoulos, 1984). The sensor inadequacy is especially acute in the areas of continuous measurement of cell mass and substrate/product concentrations, which, incidentally, are among the most fundamental state variables in nearly all fermentation systems. The relatively poor state of instrumentation means that the current measurements are discrete in time and frequently contain a high level of noise that must be filtered out before they are to be used to control a bioreactor (Stephanopoulos and San, 1984).

The objectives of developing a mathematical model are twofold. First, mathematical models represent a measure of one's comprehension about the system under investigation. It is really a concise way of summarizing one's knowledge so that the behavior of a system can be predicted, given the environmental conditions and any disturbances thereof. Only when the understanding is firm can one translate the concept into a mathematical model; the translation is at best a fuzzy one when the understanding is vague. However, one should be cautioned that the converse is not always true; a seemingly clearly stated model on paper may not always have an equally clearly demonstrated valid basis in practice. In other words, one understands no more than what a model states, but not all models accurately reflect the reality that one observes. Consequently, a model that contains unverifiable assumptions should be suspected.

Secondly, mathematical models are needed both for control purposes and in bioreactor design. They are the condensed versions of our knowledge about a system, and their sophistication can vary widely. A useful model should be properly

balanced with respect to its mathematical complexity and its ability in capturing the essential features for the intended purpose. It should also be simple enough to help one better understand the process and to permit improvement or optimization of the process; not vice versa. It should allow the direct determination of at least its key parameters, if not all parameters, by following some feasible experimental procedures. A well designed model itself should guide one in designing such experiments, as well as suggesting new ones so that the understanding of the system can be further deepened and the model itself can be improved. In this respect, the validity of a complex model is especially questionable when it contains a large number of parameters whose values cannot be experimentally evaluated individually. The success of models in engineering has always depended on the valid use of approximations and assumptions in reducing the complexity of the real world to simple and manageable mathematical abstraction, and biochemical engineering is no exception in this respect.

Most of the modeling concepts widely used in biochemical engineering have not changed for more than two decades. This is especially true for lumped parameter microbial cell growth models. Many newly proposed models are derivatives of these basic concepts (Fredrickson and Tsuchiya, 1977). On the other hand, single-cell models that incorporate the details of gene replication, translation, and transcription have only recently been proposed (Shuler *et al.*, 1979; Domach and Shuler, 1984; Domach *et al.*,, 1984; Ataai and Shuler, 1985); these models are not widely used because their validity and utility have not yet been fully established. Thus, as far as the existing models are concerned, they are either inadequate during transient operation (lumped parameter models), too complicated for control and optimization purposes because the calculation is too time consuming (single-cell models), or contain a large number of parameters that cannot be determined directly within a

reasonable time constraint (structured models) for most practical applications. Despite significant modeling efforts, simple, descriptive, and easy to construct models are not yet available. This research effort is partially aimed at answering some of these problems.

1.2 AIM AND SCOPE OF THIS WORK

Shown schematically in Figure 1.2.1 is an interactive estimation-control-optimization scheme in which the on-line measurement on a bioreactor is passed through an estimation-filter block to reduce the noise and to yield a set of on-line estimates for the state variables and growth parameters. The presence of gross measurement errors can be detected by applying the material balance constraints, and the source of errors can be identified, followed by error rectification itself. One such methodology based on the χ^2 hypothesis is presented in Appendix E. The corresponding program source codes are listed in Appendix F. These estimates are used as the basis for feedback control as well as for on-line process modeling. The biochemical process is continuously modeled, new values of the model parameters are estimated, and the biological model itself, including the shape of the kernel, is constantly updated. This can be accomplished by tracking the control history and comparing the deviation of the actual state away from the predicted values. Although such an ideal scheme does not presently exist in a completed form that can be directly implemented in an actual process, the state and parameter estimation and the new approach to modeling proposed herein is a step toward the realization of such a scheme.

This study is motivated by the general observation that time-lag effects frequently exist in a biochemical reactor system. Its existence has long been recognized, for example, at the beginning of a batch fermentation in the form of a lag phase. It is also present when a culture is transferred into a new richer medium

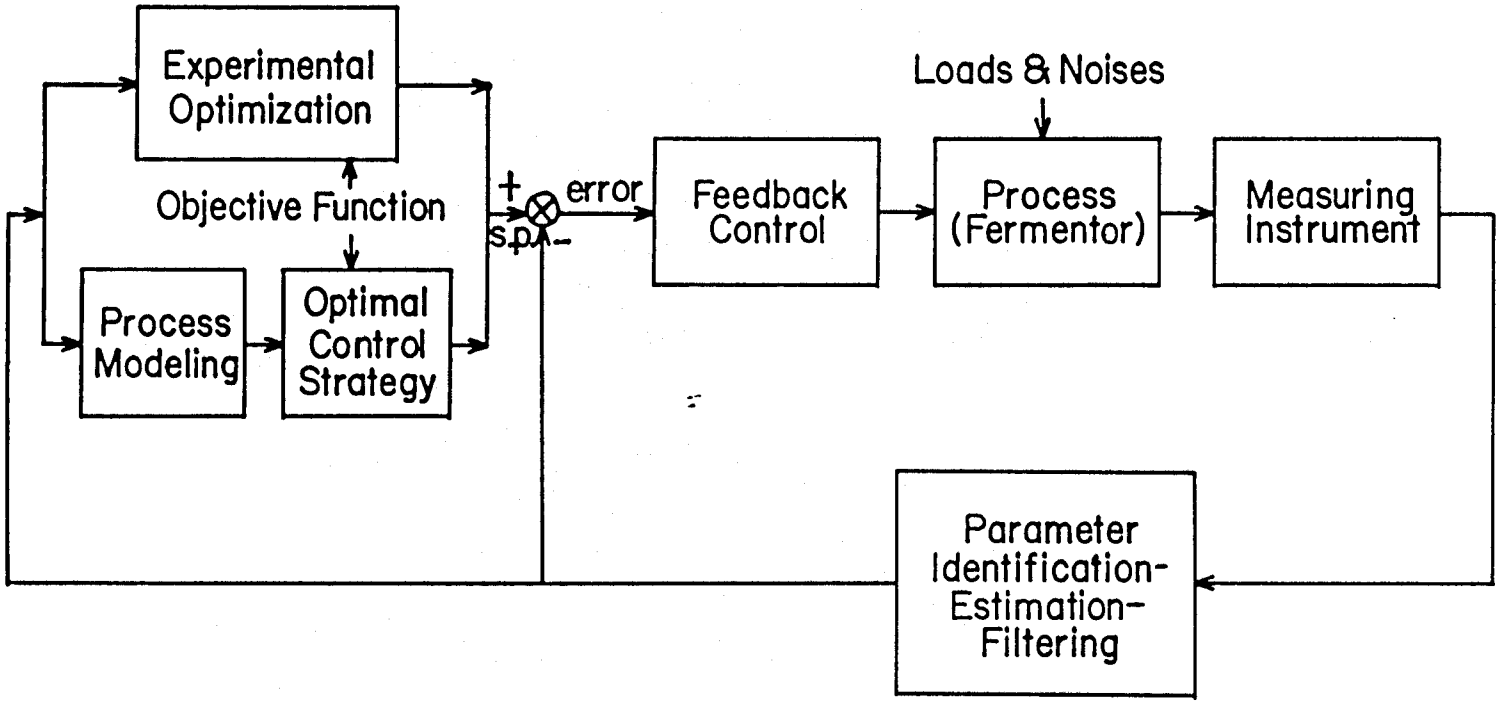


Figure 1.2.1. Block diagram of the measurement- estimation- modeling- optimization- control configuration.

that is capable of supporting a higher growth rate than the original one. The recognition of the inadequacy of the unstructured models in predicting the lag behavior has prompted the proposal of a range of structured models to explain these time-lag observations. However, most of these models are too complicated and are unsuitable for process control purposes.

The objective of this study is to develop a simple model that can predict a variety of transient as well as steady-state behaviors commonly encountered in a biological reactor. Some of these behaviors include lag phases, diauxic growth, asymmetric responses, hysteresis effects, and damped and sustained oscillations. The model shall express the cause-effect relationship in a form appropriate for practical use in a process design and control environment.

This study attempts to address mainly the second problem regarding the modeling of a biochemical reactor. A new approach to bioprocess identification and modeling is outlined. The proposed approach considers the effect on rates and yields of not only the present state of the system but also the previous history through the concept of a kernel integral. The resulting set of integro-differential equations are then shown to be equivalent to a set of first-order ordinary differential equations representing a generalized structured model. These simple ordinary differential equations can then be relatively easily manipulated with the well developed mathematical techniques to yield insightful information on the dynamics of the system, including the analysis of the stability of steady states, etc. Furthermore, size reduction techniques are outlined, which can lead to a directly observable model of a lower dimension while preserving simultaneously the biological significance of various parameters. Finally, it is demonstrated experimentally that a time-lag model can be used to predict correctly transient behaviors based on parameters that have

been determined from steady-state fermentation runs. The experiments were conducted in accordance with the state-of-the-art on-line monitoring techniques, and on-line and off-line data were analyzed with advanced parameter estimation algorithms.

CHAPTER 2

THEORIES
ON THE TIME-LAG KERNEL

2.1 INTRODUCTION

It is a well established fact that human reaction to environment depends not only on the nature of the immediate surrounding environment but, to a large degree, on past human experience as well. The same statement also holds true for a microbial system. The dynamic behavior of a biological system is intimately controlled both by the prevailing environmental conditions and by the past histories. It is not uncommon for the response to some external stimuli to appear at a later time. Quite often, this dependence on past history is totally neglected in modeling a microbial system. As a result, it is sometimes difficult to reproduce or interpret biological experiments. Similarly, it is difficult to extract definite information from a set of experiments that seem inconsistent, when time-lag effects are not properly considered.

Time-lag effects are frequently encountered in microbial systems. For example, they are present at the beginning of a batch fermentation in the form of a lag phase. Figures 2.1.1-2.1.6 show some of the measurements recorded for a batch fermentation of 5.0 g/l of glucose by *Saccharomyces cerevisiae*. The continuous lines represent the on-line measurement of the corresponding variables, and the circles indicate off-line measurements. The fermentor was inoculated at 0 hour, and microbial growth did not begin to accelerate until one hour later. Glucose consumption and cell growth were accompanied by ethanol production. As glucose became exhausted, microbial activity, as indicated by the oxygen uptake rate (OUR), carbon dioxide evolution rate (CER), and total growth rate, dropped to an insignificant level. The growth rate rested for approximately one hour before ethanol was finally utilized for further growth. As ethanol became completely depleted, the level of CER rose suddenly due to the uptake of a small amount of acetic acid formed earlier. These two distinct growth phases can be easily inferred from the OUR, CER,

and total growth rate curves. In fact, the leveling off of the biomass concentration curve during metabolic shifts in a batch fermentation has long been used to detect graphically the existence of diauxic growth.

For example, the diauxic batch fermentation of glucose in which ethanol is formed as the intermediate product exhibits a lag phase before glucose begins to be consumed; another lag phase is present before ethanol is taken up, as shown in Figure 2.1.5. The dependence of the first lag phase on the inoculum condition has been extensively studied in the past. It is the second lag that is most often used to detect the presence of more than one metabolizable substrate in a fermentor.

A drop in the growth rate in the middle of a fermentation run is due to the fact that a microorganism needs time to undergo a metabolic adjustment before it can shift from the utilization of glucose to that of ethanol. The production of enzymes and cofactors needed for the metabolization of ethanol is induced only when the glucose concentration drops to a critically low value. Synthesis of these enzymes, in turn, requires additional intermediates that must first be constructed from still lower intermediates, and so on.

Similarly, when the nutrient concentration is increased, the level of growth enzymes present in a cell must be increased before a higher rate of biomass synthesis can be achieved. Again, the metabolic mechanism in this case must be adjusted, and a factory with a higher capacity has to be assembled so that a larger amount of chemicals can flow through the anabolic and catabolic pathways. For example, when the limiting substrate concentration is increased as a result of a step change in the dilution rate in a continuous fermentor, the specific growth rate of the microorganism does not start to increase immediately, as does the limiting substrate concentration. This point is illustrated in Figure 2.1.7, which is recaptured from

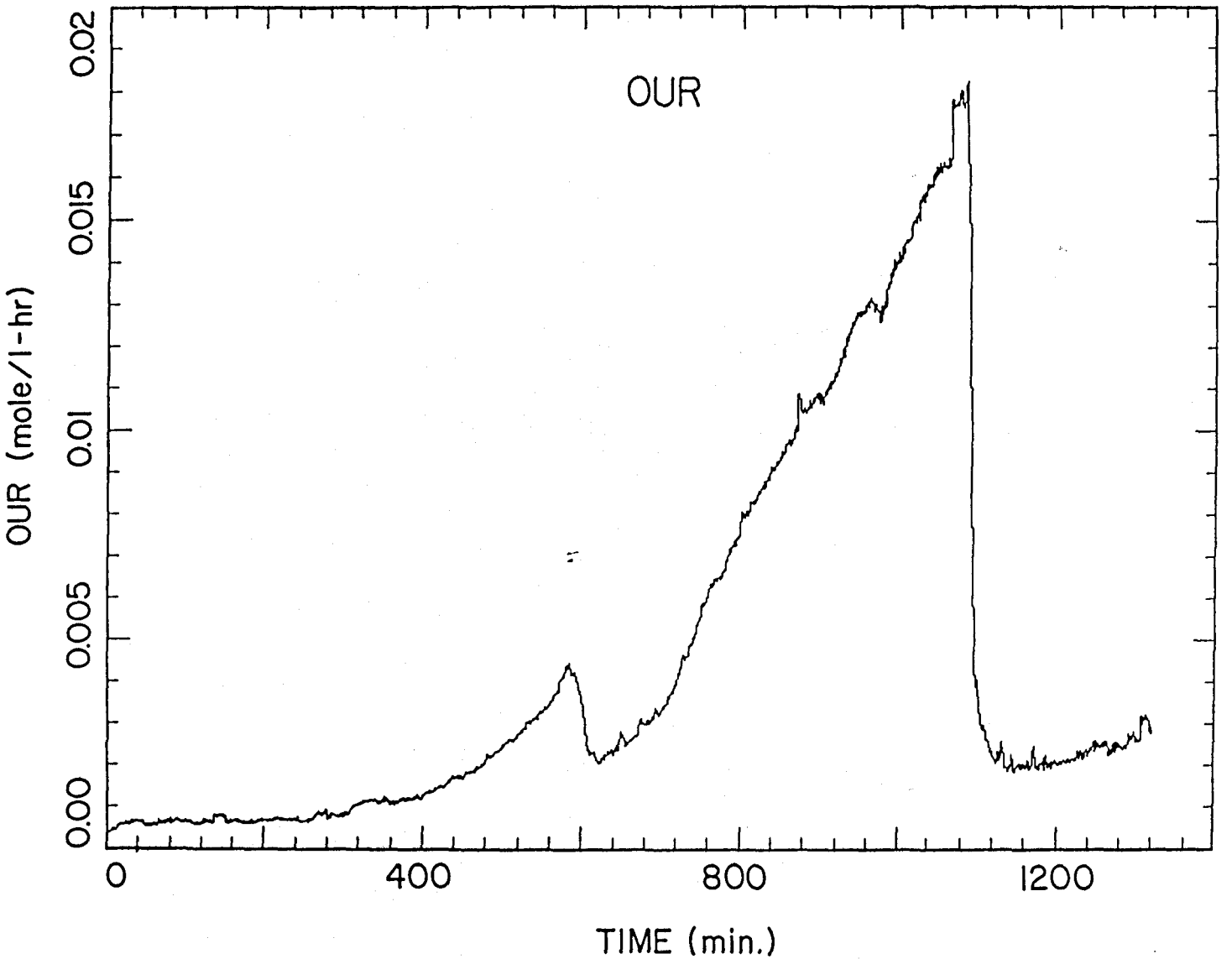


Figure 2.1.1. Oxygen uptake rate (OUR) as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. Note that there are two distinct regions of high growth activities.

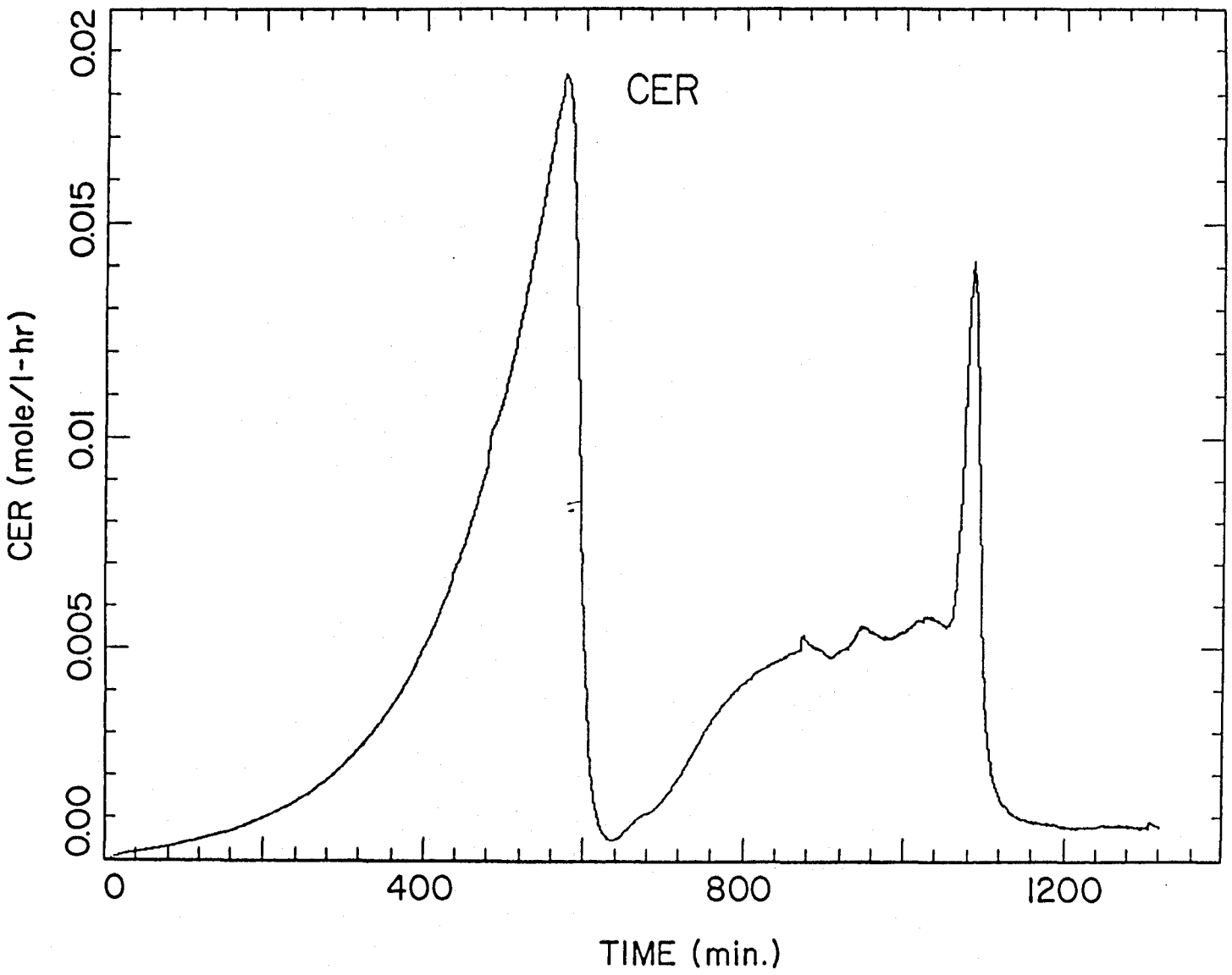


Figure 2.1.2. Carbon dioxide evolution rate (CER) as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose.

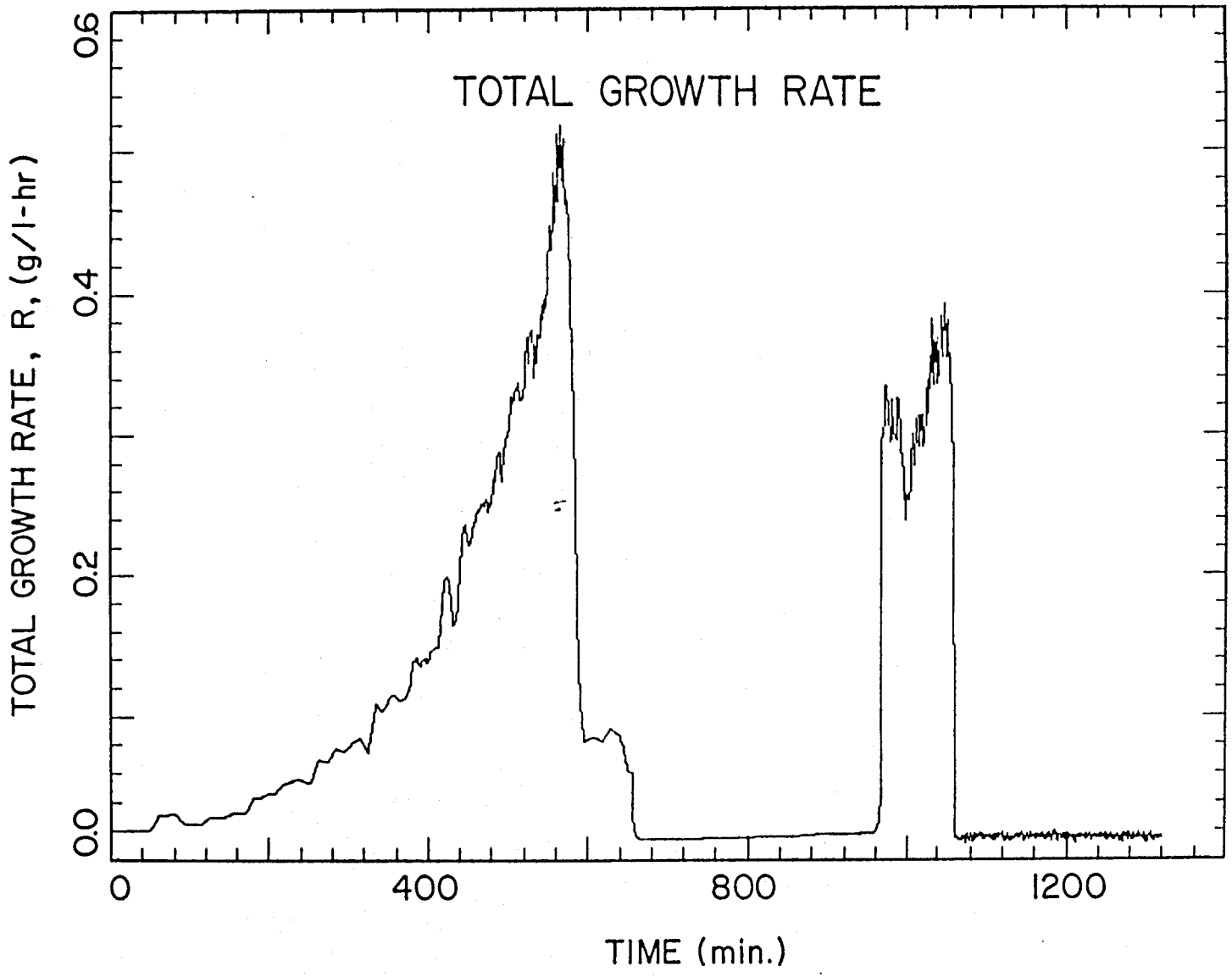


Figure 2.1.3. Total growth rate as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose.

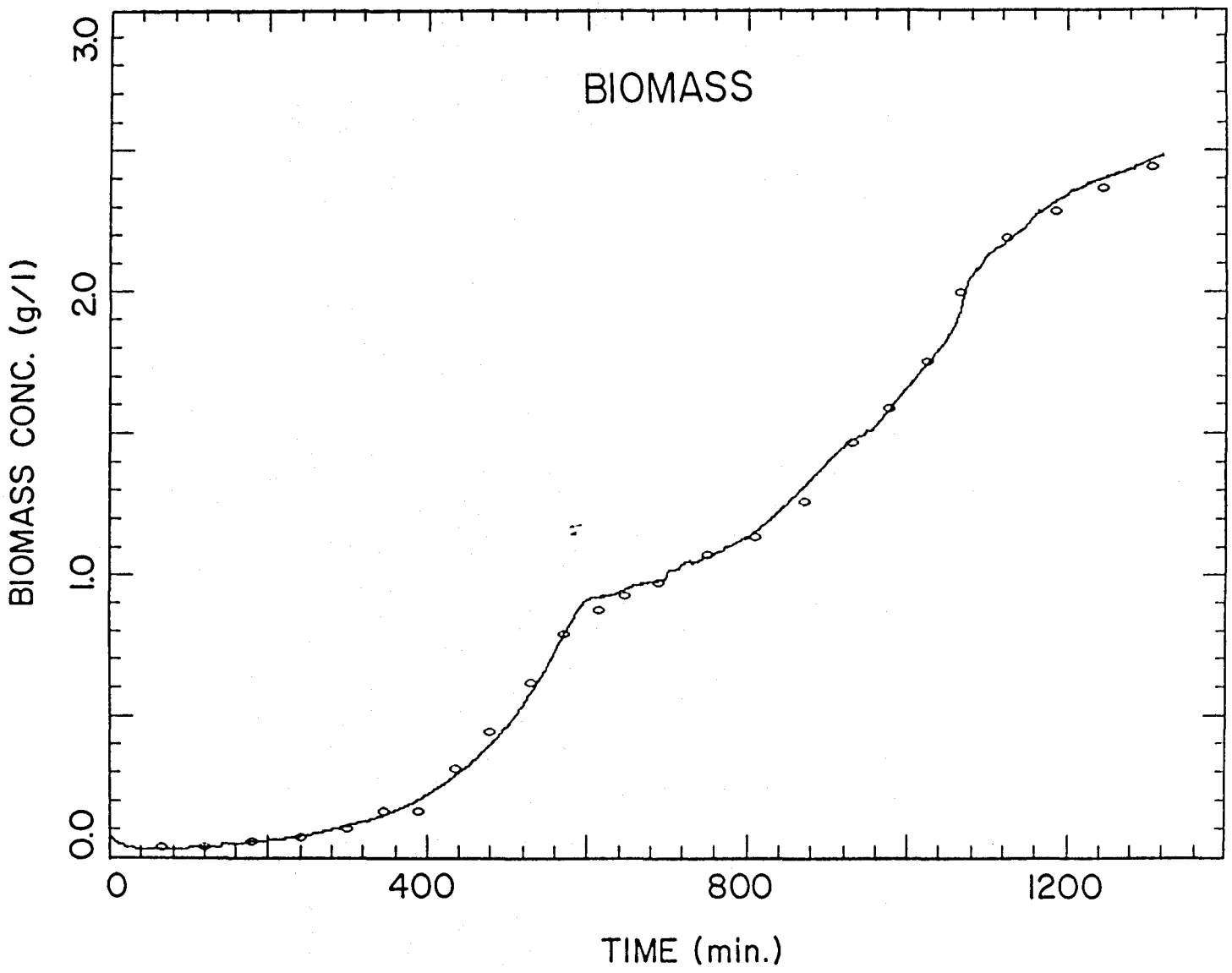


Figure 2.1.4. Cell biomass concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

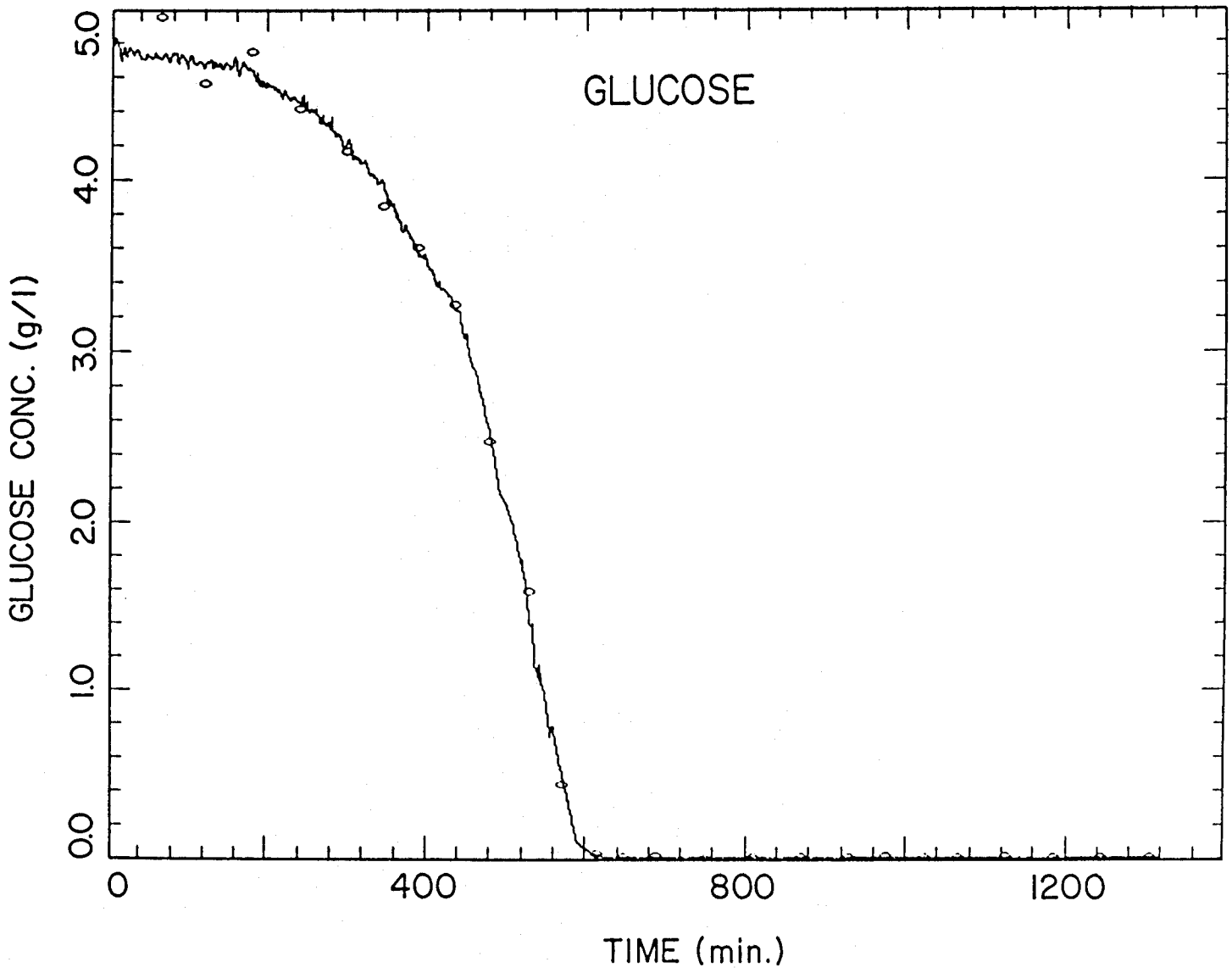


Figure 2.1.5. Glucose concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

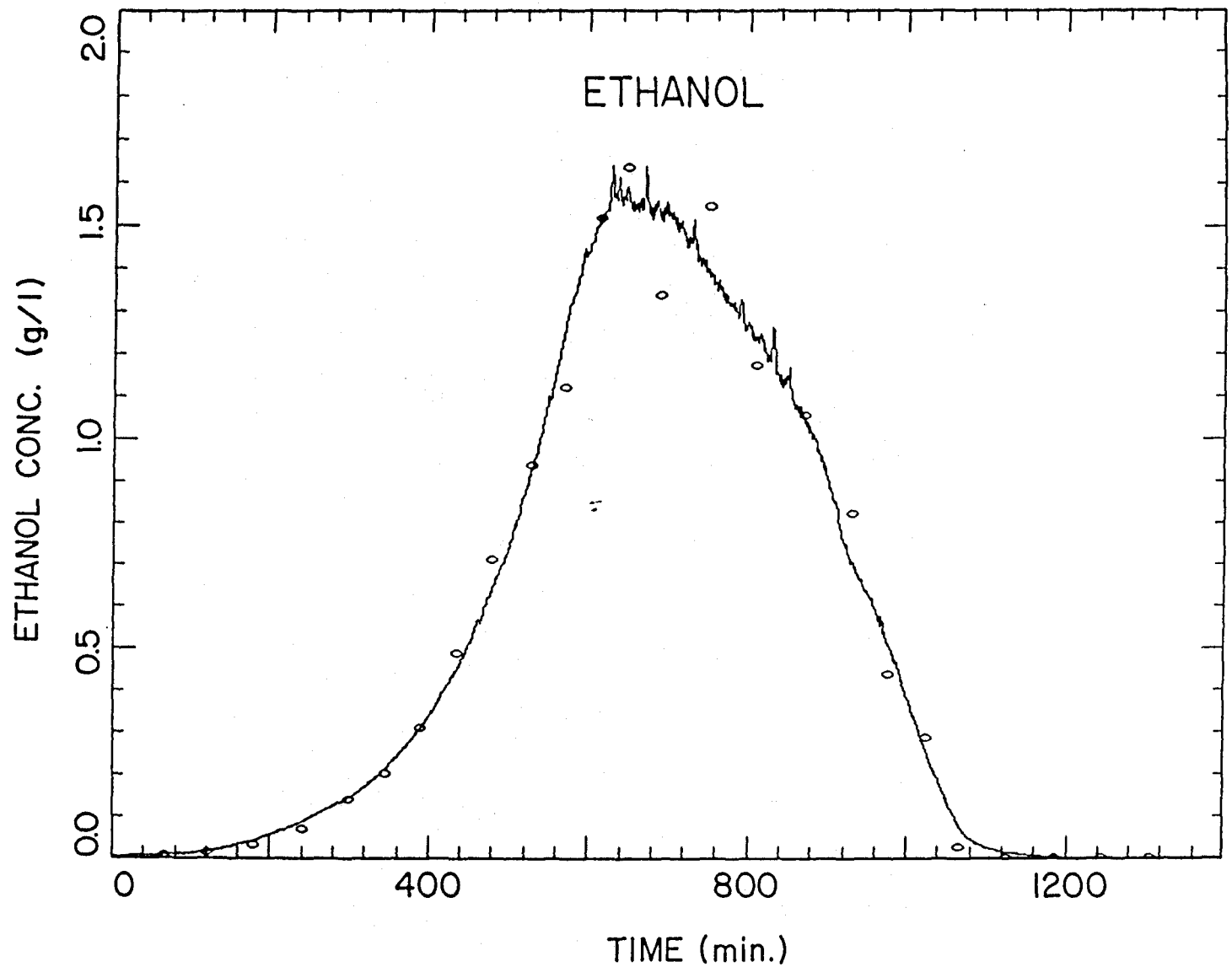


Figure 2.1.6. Ethanol concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

San (1984). In this figure, the dotted lines that represent the Kalman filter estimates of the corresponding variables are more reliable than the solid lines that represent the results of applying a moving average. The figure shows that although the glucose concentration was increasing after a sudden shift in the dilution rate from 0.123 hr^{-1} to 0.258 hr^{-1} at 0.0 hour, the observed apparent specific growth remained the same until 1.1 hours later.

In addition to the aforementioned time-lag phenomena, time-lag effects are also manifested in various observed oscillatory behaviors due to the response's dependence on the past history. It is well known that time-lags in control variables can destabilize a system and render it oscillatory. Similarly, the presence of time-lags in state variables can also profoundly influence the stability of the system and cause an otherwise stable system to oscillate. Oscillations in biological systems are quite prevalent and are frequently reported in literature. One such example is the continuous ethanol production from glucose by using *Zymomonas mobilis*. The literature data shown in Figure 2.1.8 are taken from Lee *et al.* Note that when the dilution rate is shifted up from 0.05 hr^{-1} to 0.10 hr^{-1}, the system oscillates in a damped manner. When the glucose concentration in the feed is increased from 150 g/l to 200 g/l, the oscillation is much more sustained at the same dilution rate of 0.10 hr^{-1}. As shown in later sections, a model can exhibit this type of oscillatory behavior, including the effect of increased limiting substrate concentrations in the feed on the oscillation tendency, as a result of incorporating a simple time-lag term in the mathematical model.

This adaptive characteristic of the biological system minimizes the waste and is essential for the survival of a living organism. It is one of the main factors that differentiate a biological process from a more traditional, purely physical or

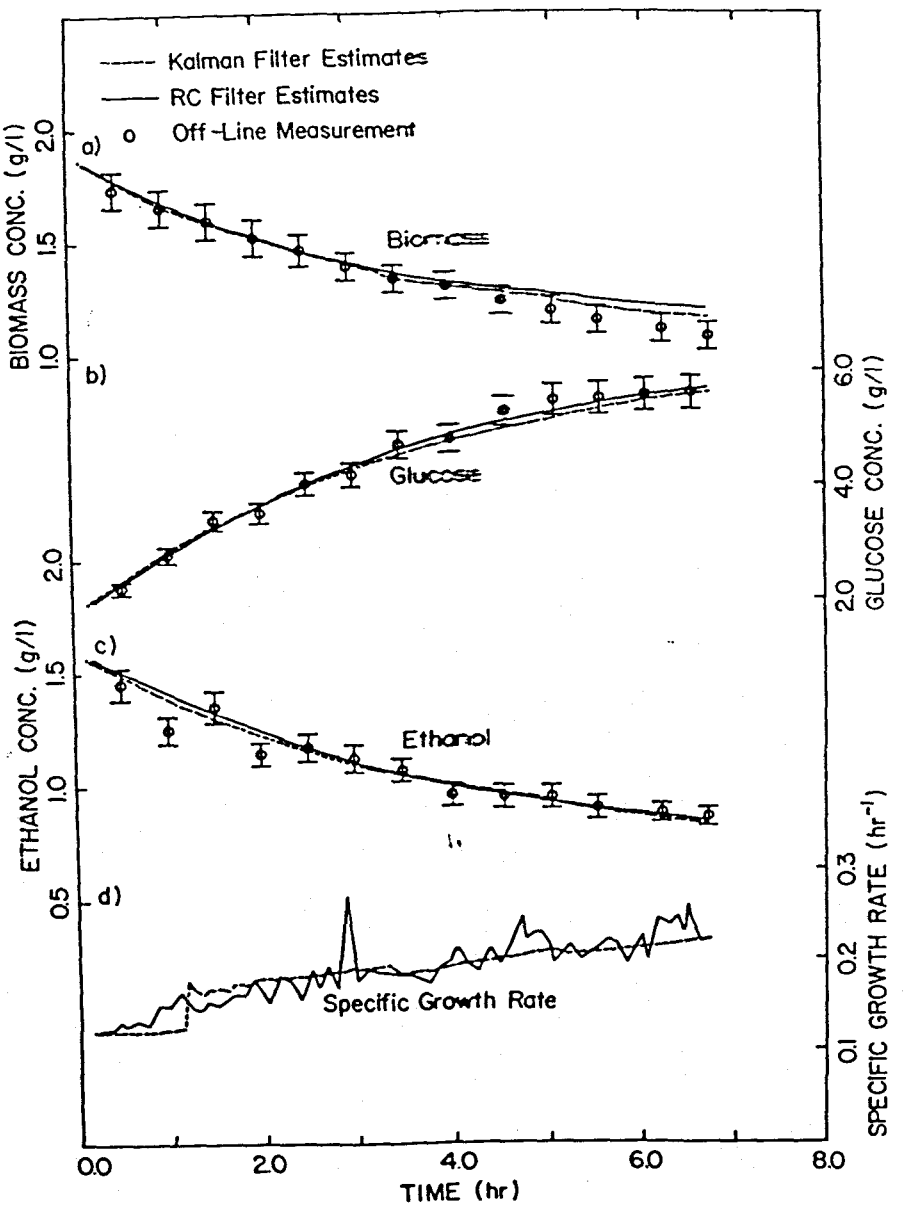
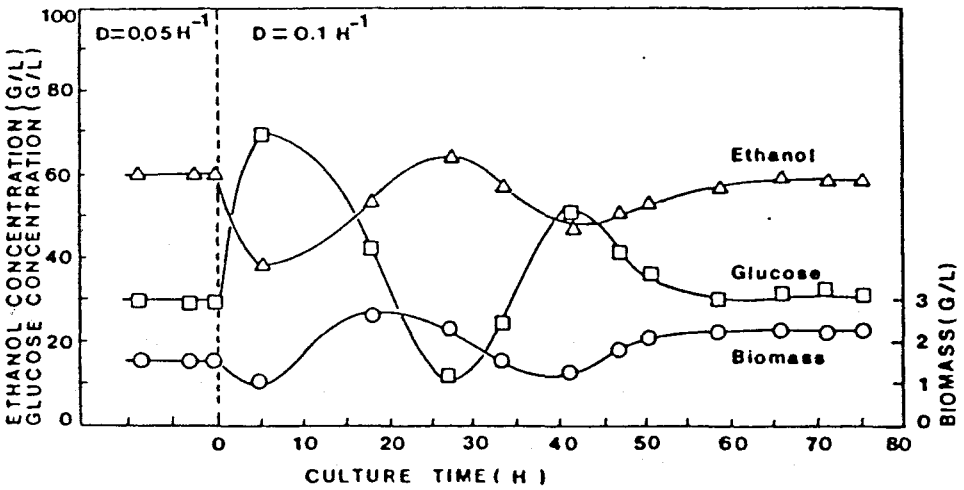
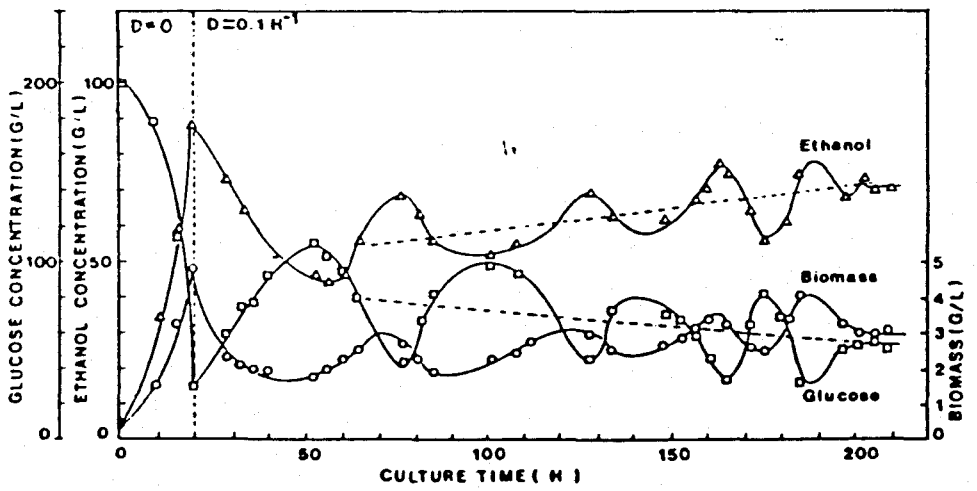


Figure 2.1.7. Time course of a) cell biomass concentration, b) glucose concentration, c) ethanol concentration, and d) specific growth rate in a dilution rate step-up (from D=0.123 hr^{-1} to D=0.258 hr^{-1} at t=0 hr) experiment of continuous glucose-limited cultivation of *S. cerevisiae* with ethanol formation. The solid lines represent the moving average of the on-line measurement, the dotted lines represent the on-line Kalman filter estimates of the corresponding variables, and the open circles indicate the off-line measurements.



Oscillations in Z.mobilis culture with 15% glucose medium following a dilution rate change.



Oscillations in Z.mobilis culture with 20% glucose medium following change from batch to continuous culture.

From Lee, Tribe, & Rogers, Biotechnology Letters,1, 421 (1979)

Figure 2.1.8. a) Damped oscillations of a continuous *Z. mobilis* culture with 150 g/l glucose in the feed following a dilution rate change at $t = 0$. b) Sustained oscillations of *Z. mobilis* culture with 200 g/l glucose in the feed following a change at $t = 20$ hr from batch to continuous operation at a dilution rate of 0.1 hr^{-1}.

chemical process. Thus, an attempt at modeling a biological process must consider the adaptive properties that cause history, memory, or time-lag effects.

As there are quite a number of models in biochemical engineering, there are also many ways of characterizing a model. Discussion on this topic can be found in the section on modeling in Appendix G. One fundamental way to classify models is to divide them according to the degree of structure that is incorporated in each model. As in other disciplines, a biological model of growth and product formation can be either unstructured or structured. However, regardless of what approach one takes, the purpose of modeling is to express one's knowledge of the system under study in explicit, concise terms, usually mathematically. One also wishes to gain the ability to predict the output of a system for a given input. This concept is graphically expressed in Figure 2.1.9. In modeling a biological reactor, the input to the system is composed of such variables as the limiting substrate concentration in the fermentation broth, the temperature, and the pH; the output variables of interest from the system consist of such variables as the specific growth rate, substrate to cell yield coefficients, and a variety of specific product formation rates.

In the unstructured modeling approach, the relationship between the output of the system and input of the system is described by a constitutive equation relating the former to the latter. For example, the specific growth rate can be expressed as a direct function of the limiting substrate concentration. The famous Monod model falls within this category. The advantage of the unstructured approach is that the resulting model is relatively simple and is usually adequate in describing steady-state behaviors. The main criticism of an unstructured model is that the relationship between, for example, μ and s is largely empirical in nature and there is often the unstated assumption of balanced growth. Although balanced growth processes have attracted much attention within the last few decades, it is a highly

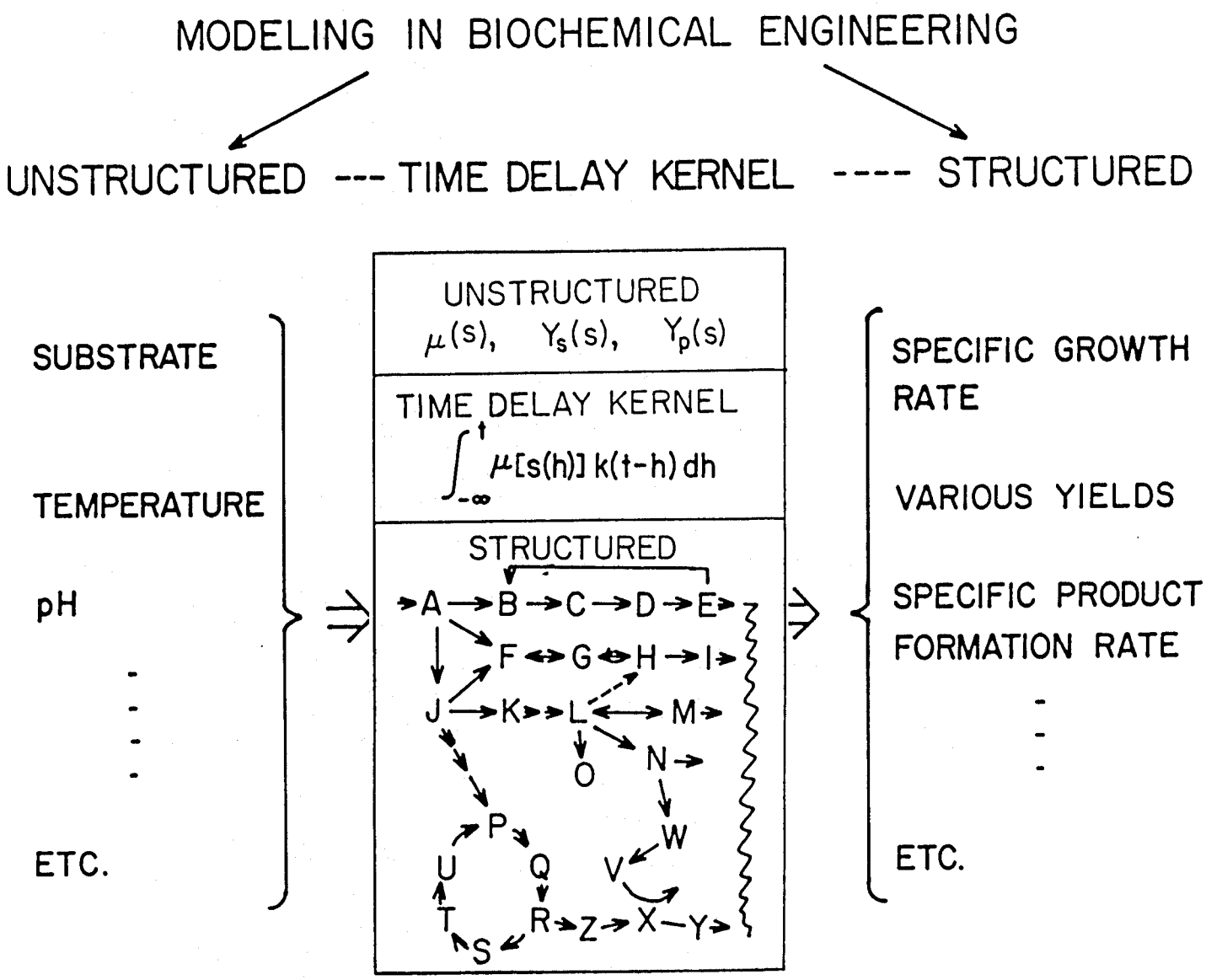


Figure 2.1.9. Graphical representation of two different traditional modeling approaches in biochemical engineering. The time-lag kernel model lies in between the unstructured model and the structured model.

idealistic concept. Balanced growth requires every component in the cell to be increasing at exactly the same ratio, such that the composition of the cell remains unchanged as it grows and expands. As we now know, this state is rarely reached during a transient; it is an exception rather than a rule. Because of this oversimplification, the model cannot predict various phenomena observed in transient situations.

To overcome this problem, one has traditionally resorted to structured models to describe both steady-state and, especially, transient observations. The structured modeling approach is graphically represented in Figure 2.1.9. In this approach, a microbial cell is basically viewed as an expanding chemical reactor in which an extremely large number of enzymatically catalyzed parallel and sequential reactions occur simultaneously in a complicated reaction network. Related models of nutrient and ion diffusion are generated to describe how the transport of chemical species across the cell membrane is accomplished, and how transport rates are affected by the membrane composition. Anabolic and catabolic pathways are usually included in a reasonably respectable model to show the fate of reaction intermediates. Biological feedforward and feedback mechanisms are also inevitably considered. So included in structured models are the branching and rejoining of reactions, the generation and deactivation of different important enzymes and cofactors, the replication process such as DNA synthesis initiation and cell division, and the transcription and translation processes of protein formation. In the structured modeling approach, ideally, one differential equation is written for each of the chemical species under consideration. However, it is neither possible nor desirable to consider literally thousands of chemical compounds. Although the process of lumping is more closely associated with the unstructured approach, it is also widely implemented in the structured approach to narrow the number of species under consideration to a manageable level. Therefore, any modeling attempts in the biological field, strictly

speaking, result in lumped models, unless *every* chemical species present in a cell is accounted for.

If performed correctly, the structured approach yields a high degree of insight into the workings of metabolic pathways and internal control mechanisms. However, the unavailability of sensors to monitor the course of these reactions inside a cell often makes it extremely tedious or impossible, depending on the type of models, to verify the validity of the proposed detailed mechanisms. A structured model is, by its nature, complicated. It usually contains many parameters and variables that are difficult to measure or estimate. In such cases, the proposed mechanism can at best be viewed as a logical educated hypothesis and should not be confused with experimental facts. The inherent, but often hidden, lumping process results in the drift of these model parameters as environmental conditions change. Because it is currently not possible to track the changes in these model parameters or to estimate continuously all the states, a complicated structured model cannot yet be practically used in a process control environment.

In addition, a structured model does not clearly show the cause and effect relationship that is so vital in the formulation of a control strategy in which one must consider the relationship between the manipulated variables and the controlled variables. To make matters worse, the reaction rates in a structured model are often expressed in terms of discontinuous functions; thus, the mathematical analysis of a structured model is not at all routine. Its solution is not even guaranteed by the Existence and Uniqueness Theorem in the theoretical studies of ordinary differential equations. Predicting the response to a simple input by inspecting the rate expressions in such a model is also beyond an average person's mathematical foresight. The cause and effect relationship is often obscured behind a large number of equations. Lengthy computer simulations are always called for when it is necessary

to predict the model response. Heavy computational overload also excludes a complicated structured model from being utilized in feedback or feedforward control.

Furthermore, the metabolic mechanisms proposed in these models are largely based on our current understanding of *Saccharomyces cerevisiae* or *Escherichia coli* as a result of decades of intensive research by many investigators. One must invest a considerable amount of time and energy to reach the same level of knowledge for other microbial systems as one now has on these two species. One would find it difficult to conceive a valid, working structured model for a new organism that possesses different metabolic mechanisms, if one was under time constraints. For various economically competitive reasons, in practice one frequently encounters situations in which quick results are urgently needed to manufacture certain microbially derived products.

Thus, there is clearly a need for a model that is simple and can explain and predict not only the steady-state results but also the transient behaviors. In this thesis, a modeling approach is proposed that incorporates the past history of the microorganism in the form of a time-lag integral in the system's dynamic equation. This new modeling technique attempts to lump all the internal steps that are of no interest to a process controller in a time-lag kernel. The proposed approach aims to keep the structure of the model as simple as possible so that each term in the entire set of dynamic equations can be easily comprehended. At the same time, the dynamic behavior can be predicted without the voracious consumption of computer time. In this proposed approach, the cause-effect relationship is defined through a time-lag integral, whose mathematical meaning can be better understood by a process engineer. Specifically, the cause and the effect are related through a time-lag kernel that summarizes the dynamic response in a very compact manner. As a result, a more accurate and realistic representation of the actual biological system

is obtained and is ready for use in fermentor design and control. The aim in our modeling approach is to merge the advantages of both structured and unstructured modeling approaches and simultaneously to suppress the objectionable shortcomings of the respective approaches. In the following section, a novel approach to bioreactor modeling is introduced that, through the use of time-lag kernels, combines the simplicity of an unstructured model with the power of a complex structured model.

2.2 FORMULATION OF TIME-LAG EQUATIONS

The concept of a variable's dependence on its past history has been in existence for quite some time (May, 1973; Cushing, 1977; MacDonald, 1982). In ecological studies, the interaction of prey-predator has been described by the Volterra model, which includes a kernel associated with one of the states of the system. It might be interesting to note that the Volterra model was originally proposed to compensate for the duration for animals to reach maturity and to show the oscillatory trajectory of a prey-predator system (Volterra, 1931). It is also quite possible that some of the oscillatory behaviors often observed in a fermentor may be due to the instability caused by the presence of time-lags. An approach somewhat similar to the one proposed herein was taken by Powell (1969).

However, the model never really gained much significant following and had been largely forgotten over the years because no attempt was made to answer the question of what shape the integral should take or what types of biological significance the integral had. Even more damaging to the survival of the model has been the incorrect formulation of the integral and the misinterpretation of the kernel's meaning. For example, the integral has been expressed as:

$$\int_0^t k(t,h)n(h)dh, \tag{2.2.1}$$

which is not strictly valid. The effect before time zero plays the most important role in explaining the existence of a lag phase at the beginning of a batch fermentation, and it is this crucial effect that has been totally disregarded in the above formulation when the lower integration limit is set to 0. Another inadequacy of the above integral is that $n(t)$ in that equation is the number of cells in the system (*i.e.*, the population density), and the growth rate of a population, expressed as $\frac{dn(t)}{dt}$, is thought to depend on the history of the population density, instead of the environment. Furthermore, the usage of the integral is not quite correct.

For example, let us consider the famous logistic equation of growth in population theory,

$$\frac{dn(t)}{dt} = \alpha n(t)\left[1 - \beta n(t)\right], \tag{2.2.2}$$

where α is a growth constant, β is the reciprocal of the asymptotic cell concentration in the stationary phase, and $n(t)$ again is the population or cell density. In the misguided approach, the integral is simply added to the end of the originally existing equation that describes the reproduction rate of a population. It is frequently expressed as:

$$\frac{dn(t)}{dt} = \alpha n(t)\left[1 - \beta n(t)\right] + \int_0^t k(t,h)n(h)dh. \tag{2.2.3}$$

This treatment results in no time lag. On the contrary, it yields quite absurd results if the function for $k(t,h)$ is not chosen carefully. For example, if a constant value of K is assigned to $k(t,h)$, as has been routinely done in the past, the equation

$$\frac{dn(t)}{dt} = \alpha n(t)\left[1 - \beta n(t)\right] + K\int_0^t n(h)dh \tag{2.2.4}$$

indicates that the integral term $\int_0^t n(h)dh$ increases without bound if K is positive. Consequently, $n(t)$ also increases almost exponentially without limit. Alternatively, if K is negative, then $\frac{dn(t)}{dt}$ remains negative as $n(t)$ becomes negative because the integral term $\int_0^t n(h)dh$ is finite and positive at that point. See Bailey and Ollis

(1977). Furthermore, the kernel has been interpreted as no more than a crude memory function or an empirical weighing factor. The biological significance of the kernel was not realized, nor was its power or potential, and it is the author's opinion that one's knowledge of the microbial metabolism and cellular regulation was not being fully utilized. Thus, one sees that, unfortunately, the time-lag kernel integral has, over the years, been formulated incorrectly, applied inappropriately, and interpreted inaccurately. It is the purpose of this section to dispel some of the criticisms raised earlier concerning the time-lag approach to modeling a biological process.

In this section, we will formulate a model using the time-lag kernel approach and answer questions regarding the kernel's functional form. The essence of the approach is the inclusion of kernel functions in the equations that describes the dynamics of a bioreactor. However, the method used to apply the concept of time-lag kernels is quite different from Volterra's. The development of the model is based on the original presentation of Appendix H.

As an introductory example, first consider the familiar case of a continuous bioreactor of Figure 2.2.1 modeled by a lumped-parameter two-state-variable model, namely:

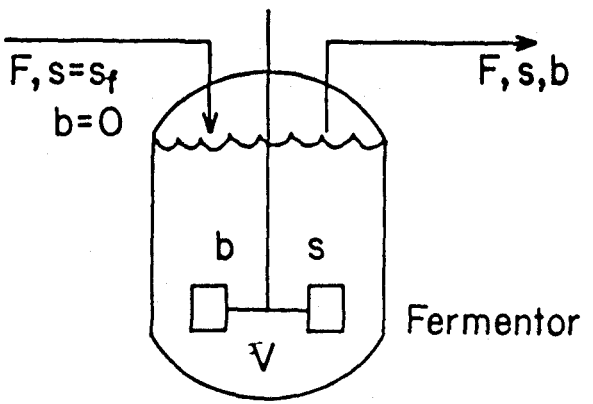
$$\frac{dx(t)}{dt} = -Dx(t) + \mu(s)x(t) \tag{2.2.5}$$

$$\frac{ds(t)}{dt} = D\left[s_f - s(t)\right] - \frac{1}{Y_s}\mu(s)x(t), \tag{2.2.6}$$

where we assume that the specific growth rate, μ, of biomass, x, is a function of the limiting substrate, s.

Many different forms have been proposed in the past to relate μ to environmental variables, including s. Of course, one of the most widely used relationships

Dynamic Equation of a Continuous Bioreactor
— An Unstructured Modeling Approach



$$\begin{cases} \frac{db}{dt} = -Db + \mu(s)b \\ \frac{ds}{dt} = D(s_f - s) - \frac{1}{Y_s}\mu(s)b \end{cases}$$

where $\mu(s) = \frac{\mu_m s}{K_s + s}$

Y_s = constant

Figure 2.2.1. The simplest set of dynamic equations for a continuous biological reactor in an unstructured modeling approach.

is that described by a Monod model:

$$\mu(s) = \frac{\mu_m s}{K_s + s}, \tag{2.2.7}$$

where μ_m is the maximum specific growth rate and K_s is the half-saturation constant, and Y_s is the substrate to cell yield coefficient, that is usually assumed to be constant. Various modifications to the above basic Monod model have been proposed. For example, a maintenance term can be included in either the biomass dynamic equation (Herbert, 1958) or the substrate dynamic equation (Pirt, 1965) or both; a substrate or production inhibition term may appear in the expression for μ in a variety of ways (Fredrickson and Tsuchiya, 1977), or the yield coefficient may be treated as a function of other states (Tanner, 1979).

The above model and its variations, and for that matter almost all other unstructured models presently in use, state that the behavior of the biomass-substrate system depends only on the present state, and there is no provision for the past history of the microorganism. It is implicitly assumed that the quality of the cell biomass, *i.e.*, the cell composition, does not affect the growth and the reproduction of the microorganism. However, it is known that the observed response of a cell population at a certain time instant is the composite result of various biological processes that were initiated at different time instants in the past, as a response to the instantaneous environmental conditions prevailing at each particular time. These various processes result in a present overall specific growth rate that can be described with the introduction of a time-lag kernel, $k(t,h)$, in the specific growth rate:

$$\frac{dx(t)}{dt} = -Dx(t) + \left[\int_{-\infty}^{t} \mu[s(h)]k(t,h)dh\right] x \tag{2.2.8}$$

$$\frac{ds(t)}{dt} = D\left[s_f - s(t)\right] - \frac{1}{Y_s}\left[\int_{-\infty}^{t} \mu[s(h)]k(t,h)dh\right] x(t). \tag{2.2.9}$$

Note that in this formulation, the microorganism's growth rate is assumed to be affected by the chemical environment, namely the limiting substrate concentration in the fermentation broth surrounding a cell. Lag effects due to other environmental factors such as temperature, pressure, pH, ionic strength, and nutrient composition can also be handled similarly.

In order for this time-lag model to be more than just an exercise in mathematical manipulation, we need to consider the following questions: What is the functional form of the time-lag kernel and its proper mathematical expression, and what procedure should be followed if one is to determine this function experimentally, and can the functional determination indeed be carried out feasibly with the presently available techniques and sensors? When the function is found, one is apt to ask what biological or physical significance does the kernel have. As mentioned previously our stated objective is to develop a simple but adequate model for use in process design and control. However, the introduction of a kernel results in a set of integro-differential equations for which mathematical theories are not well developed. Therefore, we need to demonstrate that the introduction of time-lag integrals does not really complicate the analysis of the system. These are some of the points that will be addressed later in this thesis. First, we will proceed to develop a scheme to represent the kernel function. As will be shown later, our handling of the kernel is more general in the sense that the shape of the kernel is not restricted to some pre-defined set of functions.

We will first examine a more specific but simpler problem to familiarize ourselves with the notion of time-lag kernel and to acquire a better understanding of it. Afterwards, a more general matrix representation of the time-lag kernel will be presented for a vector of dynamic equations in later sections. For a linearized

time-invariant system, k no longer depends on t and the integration variables h separately but on the difference $t - h$.

$$\frac{dx(t)}{dt} = -Dx(t) + \left[\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right] x(t) \tag{2.2.10}$$

$$\frac{ds(t)}{dt} = D\left[s_f - s(t)\right] - \frac{1}{Y_s}\left[\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right] x(t). \tag{2.2.11}$$

Non-dimensionalization can be carried out to simplify above equations without any loss of generality. If time is scaled with reference to D^{-1} and concentrations are scaled with reference to s_f, then we have:

$$\frac{dx(t)}{dt} = \left[-1 + \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right] x(t) \tag{2.2.12}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s}\left[\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right] x(t). \tag{2.2.13}$$

The kernel $k(t)$ is usually referred to as the impulse response function in process control terminology and can be interpreted as a weighing factor. Shown schematically in Figure 2.2.2 is the graphical interpretation of a time-invariant kernel integral that relates the input to the output of a linear system. The input to the system is divided into a string of impulses. The response of the system to the impulse input is described by the impulse transfer function. Thus, the output of the system corresponding to a single impulse is determined by the shape of the impulse transfer function multiplied by the magnitude of the impulse input. The overall output of the system as a response to the entire string of impulse inputs is simply the summation of all the similarly shaped individual impulse responses of different magnitudes. The output integral can be conceptualized as the response to a string of impulses, each of which is felt by the system over a period of time according to

the impulse transfer function. In summary, one adds together all the responses to each individual stimuli to find the overall response to a specified input.

It should be noted that the interpretation is straight-forward only for a linear time-invariant system. The shape of the impulse will change in time if the system is time variant, and the superposition principle does not hold if the system is nonlinear.

The integral $y(t) = \int_{-\infty}^{t} \mu(h)k(t-h)dh$ is known in mathematics as a convolution integral or a Faltung integral, a German term meaning *folding*. Another interpretation of the convolution integral is presented in Figure 2.2.3. The right part of the paper containing the kernel function is folded over so that the origin of the kernel function matches the current time in the input curve. The dotted lines in the figure connect the weighing factor from the kernel function to the input of the system. The response of the system is simply the summation of all the input values weighed by the corresponding values of the kernel. As time passes, the folding is continually shifted toward the right side of the paper to match the origin of the kernel function with the current time, and the same process of calculating the output is repeated.

In our kernel integral associated with the specific growth rate of the population, the input to the system described in the above equations is what we will call the intrinsic specific growth rate of the microorganism, the output from the system is what we will denote as the observed specific growth rate, and the system is a collection of microbial cells themselves, each being viewed as a minute chemical factory. Since it can be generally assumed that future state have no effect on the present, $k(t)$ can be implicitly set to zero for $t < 0$. A non-zero kernel value for $t < 0$ means response before excitation or the determination of the present state based on future events, which is not physically realizable. For $t > 0$, a kernel function that is not identically zero implies the dependence of the response on

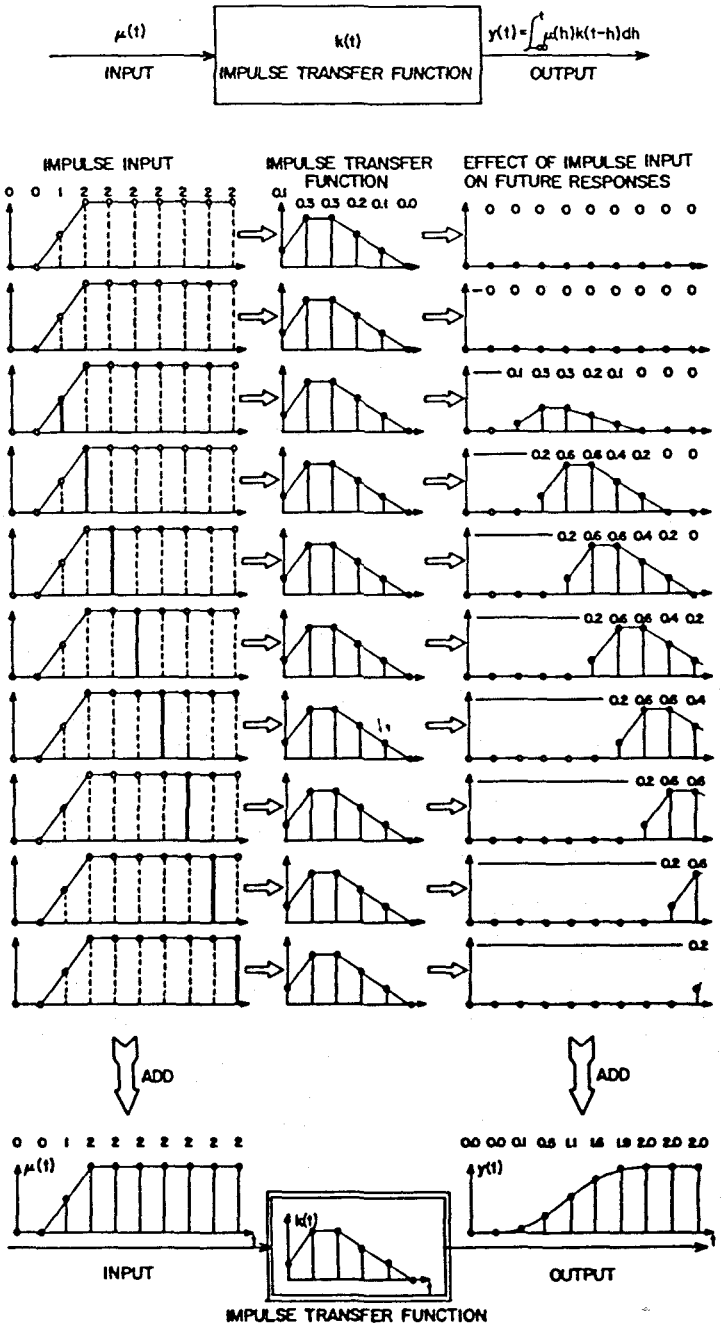


Figure 2.2.2. Interpretation of a time-invariant kernel integral which relates the input to the output of a linear system.

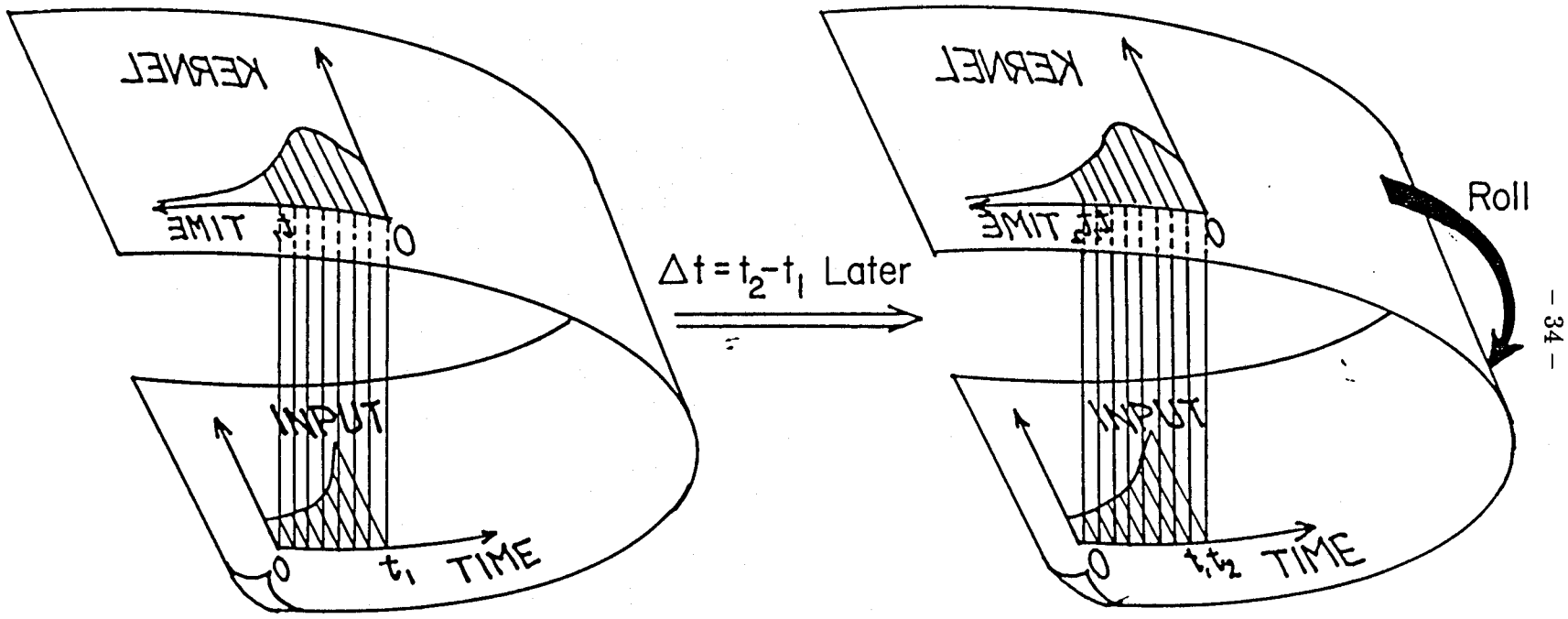


Figure 2.2.3. A convolution integral can be calculated by folding the kernel function over the input and by continually matching the origin of the kernel with the current time.

the past histories. Note that, strictly speaking, $k(t)$ is not a time-lag probability distribution function and $k(t) < 0$, *i.e.*, negative weighing, is possible; whereas, it is not possible to have $k(t) < 0$ if $k(t)$ were interpreted as a probability distribution function. Correspondingly, it is not strictly mathematically correct to interpret the kernel as a memory function or a forgetting factor, although doing so may help the mental visualization of the time-lag concept.

The μ in the integrand of Equations (2.2.10) and (2.2.11) is the specific growth rate that would have been realized if the system operated at a steady state characterized by the corresponding value of s for a prolonged period of time; it is the true specific growth rate in the absence of time-lag effects; thus, the term *intrinsic specific growth rate* is reserved for this variable. The presently observed *apparent* value of the specific growth rate is given by the integral of Equations (2.2.10) or (2.2.11). It can be viewed as a string of instantaneous intrinsic specific growth rates, each of which affects the microorganism's actually realized growth rate according to the specifications of the time-lag kernel. The future course of action for the population is, thus, explicitly guided by the kernel function.

What kind of shape does a kernel take, and how do different shapes of the kernel affect the ultimate dynamic behavior of the microbial behavior? Various possibilities exist for the functional form of $k(t)$. Figure 2.2.4 lists three of the many possibilities. First, one can set $k(t)$ to be a simple delta function centered at the origin, $\delta(t)$, meaning that both the future and the past have absolutely no effect on the specific growth rate and that the immediate present carries all the weight. The integral $\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh$ reduces to $\mu[s(t)]$ in this case, and Equations (2.2.10) and (2.2.11) reduce to the conventional unstructured model of Equations (2.2.5) and (2.2.6). In other words, this represents the unstructured model in which there is

no time-lag. Thus, the limiting case of the absence of time-lag is automatically included in our formulation.

Another possibility is to assume that there is a fixed time lag in the response of the system, *i.e.*, $k(t) = \delta(t-\tau)$, meaning that the specific growth rate depends on the substrate concentration at a discrete time instant τ units before the present. Only two scalar parameters are needed to completely specify a delta function, namely the magnitude and the time lag, τ. In actuality, only τ is needed since the area under the kernel function must be unity, as shown in a later section. The state equations in this case are reduced to:

$$\frac{dx(t)}{dt} = \Big[-1 + \mu[s(t-\tau)]\Big] x(t) \tag{2.2.14}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s}\mu[s(t-\tau)]x(t). \tag{2.2.15}$$

Although this function can be easily specified without generating a vast number of unknown parameters to be evaluated experimentally, one criticism of this type of function is that it is not easy to justify why the specific growth rate should depend on the environment at a specified time instant in the past based our current knowledge on the biology of growth and reproduction. One of the cases where it may be reasonable to assume so is when a certain amount of delay is needed for the organism to mature before reproduction is possible. Another case is the eucaryotic cell cycle where different events must proceed in sequence before a cell can undergo mitosis and divide. For most commonly encountered situations, a discrete time delay is obviously a quite drastic assumption. Nonetheless, it can be considered as a special limiting case in our formulation.

This type of approach, employing a discrete constant time lag, is frequently used in population dynamics, although it is generally formulated in such a way that

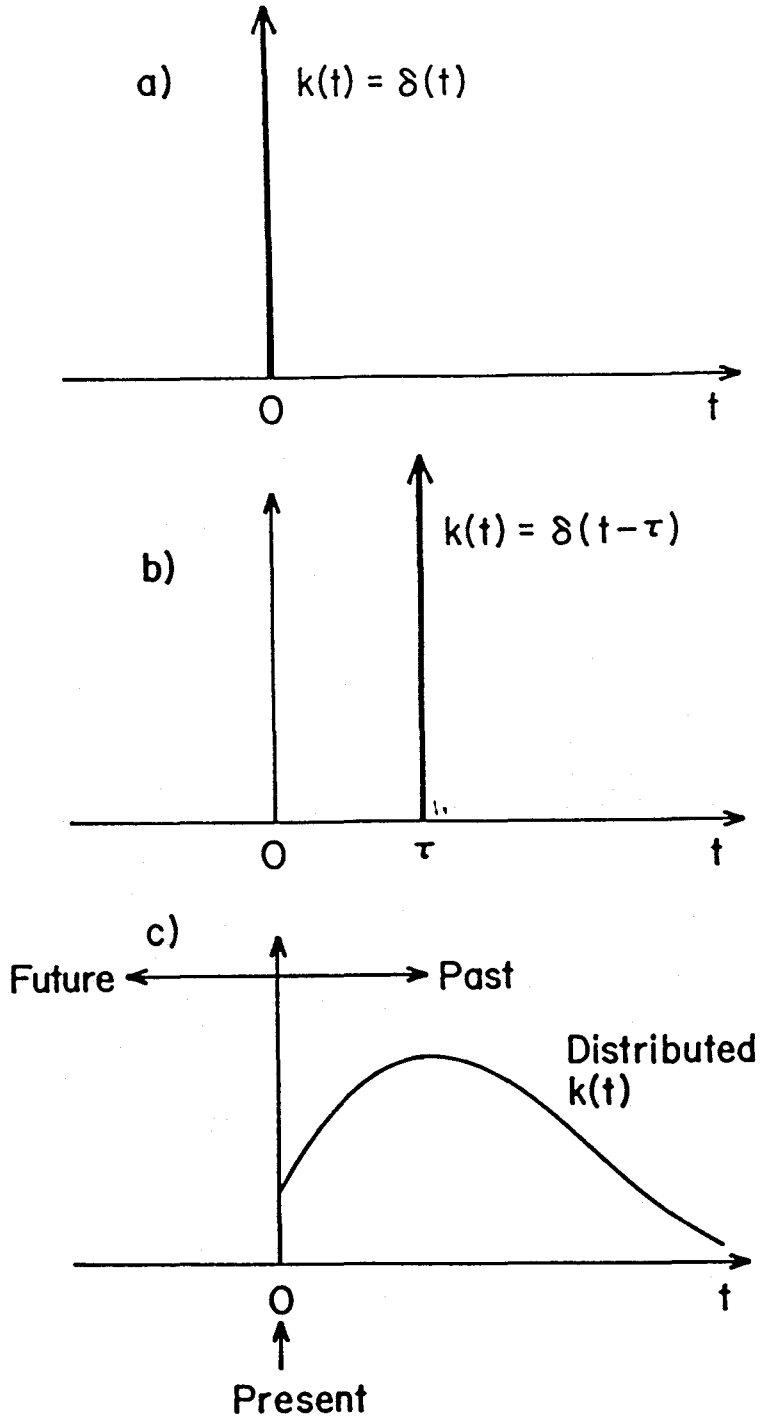


Figure 2.2.4. Three frequently used functional forms of $k(t)$: (a) delta function without time-lag; (b) delta function with a discrete time-lag, τ, *i.e.*, time-delay; (c) general distributed time-lag. Note the direction of past and future are the reverse of the conventional time plots.

a state variable's response depends on its own past histories rather than on the environment. Similar time delay, sometimes called dead-time, is also used in process control, although the time delay in process control is often associated with one or more of the controlled variables or measurements instead of the state of the system. This subtle difference makes the mathematical problem statement completely different, and the theories developed to deal with the time delay problems in the field of control or signal processing are relatively helpless in analyzing the above set of equations. Mathematical analysis of this discrete time delay case can be performed by using the theories of differential-difference equations (Bellman and Cooke, 1963). As briefly shown in the following section, the relatively simple system of Equations (2.2.14) and (2.2.15) can be successfully analyzed. However, because mathematical theories of differential-difference equations are not as well developed as those dealing with ordinary differential equations, some problems may arise in integrating and analyzing this type of differential-difference equation in general, especially when the set of equations is slightly more complicated than Equations (2.2.14) and (2.2.15).

The third choice presented in Figure 2.2.4 is an arbitrary continuous and differentiable function for $t > 0$. For $t < 0$ the function is identically equal to zero. Note that the influence of the present state of the environment on the system is proportional to the magnitude of the kernel at $t = 0$. The influence of the environment at τ units into the past is given by the value of the kernel at $t = \tau$, *i.e.*, $k(\tau)$. Similarly, the effect of the future state at a time τ units from now is given by $k(-\tau)$, which is zero because the future should not affect the present. Therefore, the direction of the flow of time is the reverse of what one is accustomed to, due to the convolution nature of the impulse transfer function.

To specify such a general function theoretically requires that an infinite number of points along the curve be given. In actuality, the number of points specified is

finite, and the quality of the curve will be in some way proportional to the number of points given to specify the curve, with the rest of the curve being generated by some interpolation schemes. Traditionally, such an approach is neither convenient nor practical, and the function specified in such a manner is not fit for an extensive mathematical analysis.

A more general approach is to express an arbitrary function $k(t)$ in terms of a series of base functions. Such approximation of functions by other well studied base functions has always been adopted not only in the field of engineering but also in mathematical science. Taylor's series expansion with polynomials and Fourier series expansion with sine and cosine functions are some of the most popular ones. However, in this work we shall take a noteworthy exception in the choice for the base function. Neither polynomials nor sine/cosine functions will be indiscriminately used to approximate the kernel. The approximation of an analytical function $k(t)$ is accomplished by expressing it as a summation of exponential distribution functions of order m or less, and the formidable task of specifying a complete function with an infinite number of points is reduced to that of finding a small number of coefficients such as a_0, a_1, etc.

$$k(t) = a_0 k_0(t) + a_1 k_1(t) + a_2 k_2(t) + \ldots + a_m k_m(t), \tag{2.2.16}$$

where the general expression for the nth exponential distribution function is:

$$k_n(t) = \begin{cases} \dfrac{1}{Tn!}\left(\dfrac{t}{T}\right)^n e^{-\frac{t}{T}} & \text{for } t \geq 0 \\ 0 & \text{for } t < 0. \end{cases} \tag{2.2.17}$$

The reason for choosing this relatively unknown set of exponential distribution functions will become apparent in later sections. In summary, these base functions are chosen over the traditional power series or trigonometric functions because they permit the transformation of the integro-differential equations into a set of simple

first-order equations to be carried out with care. The first two exponential distribution functions are sometimes used in ecological studies of population dynamics, and they have over the years earned special names. The 0th-order exponential distribution function, $k_0(t)$, and the 1st-order exponential distribution function, $k_1(y)$, are called *weak* and *strong* generic delay, respectively.

$$\text{n=0} \qquad k_0 = \tfrac{1}{T} e^{-\frac{t}{T}} \quad \ldots \text{ weak generic delay} \tag{2.2.18}$$

$$\text{n=1} \qquad k_1 = \tfrac{t}{T^2} e^{-\frac{t}{T}} \quad \ldots \text{ strong generic delay.} \tag{2.2.19}$$

Some of the properties of the exponential distribution functions are shown in Figure 2.2.5. Specifically, the function has a maximum value of $\frac{1}{T n!} n^n e^{-n}$ at $t = nT$. Furthermore, the ith moment of $k_n(t)$ is:

$$i\text{th moment of } k_n(t) = \int_{-\infty}^{\infty} t^i k_n(t) dt = \frac{(n+i)!}{n!} T^i. \tag{2.2.20}$$

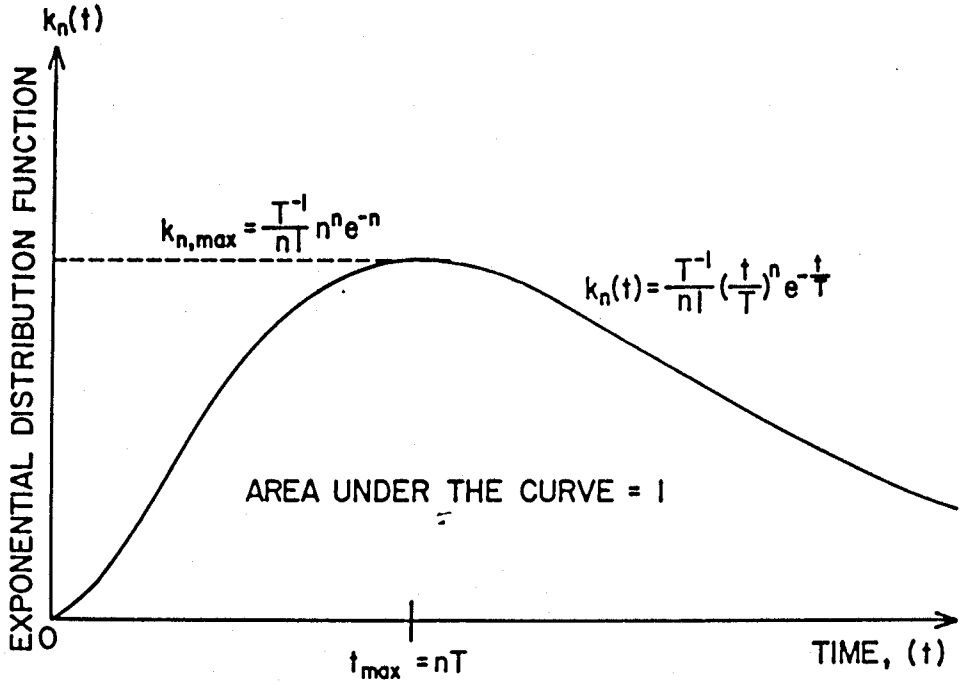
Note that the functions are normalized such that the 0th moment of the function, *i.e.*, the area under the curve, is unity. Since the kernel function relates the intrinsic specific growth rate to the observed specific growth rate in the presence of past history effects, the two quantities should be equal in the absence of history effects. As a result, the area under the kernel function in this case must be unity. This need for normalization explains the factor $\frac{1}{T n!} T^n$ in Equation (2.2.17). The first moment, which corresponds to the average delay, is proportional to both the lag time-constant, T, and the order of the exponential distribution function, $(n+1)$:

$$\tau = (n+1)T. \tag{2.2.21}$$

Thus, the functions both flatten out and shift toward the right as n increases. The first few of these exponential distribution functions are shown in Figure 2.2.6.

The variance, which is indicative of the degree of spread of the function from its average delay, is calculated from the first moment and the second moment:

$$\sigma^2 = (\text{2nd Moment}) - (\text{1st Moment})^2 = (n+1)T^2 = \frac{\tau^2}{n+1}. \tag{2.2.22}$$



$$\text{ith Moment of } k_n(t) = \int_0^\infty t^i k_n(t)dt = \frac{(n+i)!}{n!} T^i$$

0th Moment	= Area under the Curve	=	1	
1st Moment	= Average Delay	= τ	= $(n+1)T$	
2nd Moment	=		= $(n+1)(n+2)T^2$	
Variance of k_n	= Average Spread =	σ^2	= $(n+1)T^2$	= $\frac{\tau^2}{n+1}$

Figure 2.2.5. Some properties of the exponential distribution functions.

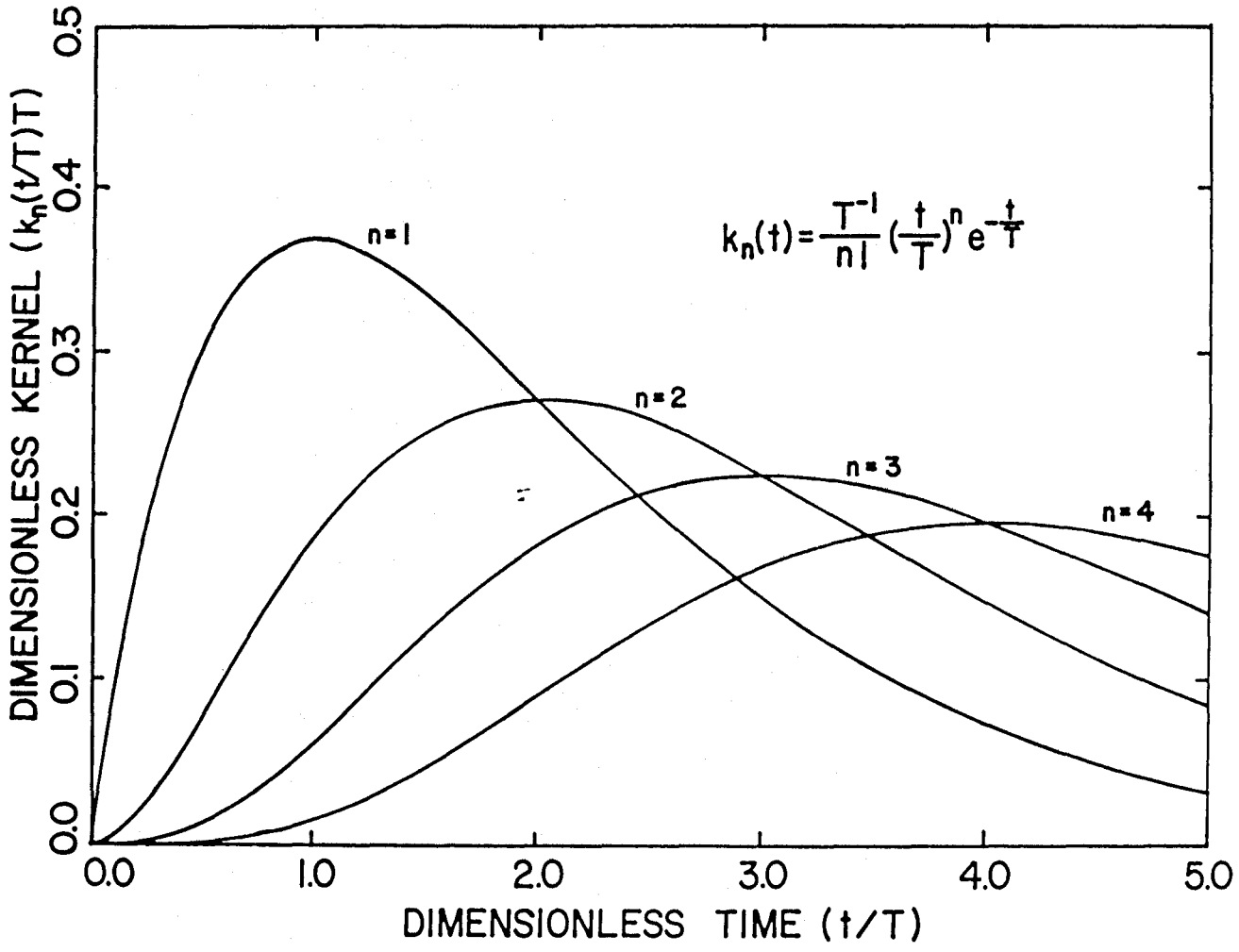


Figure 2.2.6. Exponential distribution function of order n.

Thus, one sees that the average spread is proportional to the average delay but inversely proportional to the "order" of the kernel function.

If these functions shown in Figure 2.2.6 are normalized with respect to the average lag, $(n+1)T$, then one can see that the peak at the average lag becomes higher and narrows as n increases, as indicated in Figure 2.2.7. It can be shown that as $n \longrightarrow \infty$, $k_n(t) \longrightarrow \delta(t-\tau)$, where τ is the average lag. In this limiting case, the state equations are again reduced to Equations (2.2.14) and (2.2.15). Therefore, the discrete time-lag is included in this general treatment of the kernel function as a limiting special case, if exponential distribution functions described in Equation (2.2.17) are chosen as the base functions for series expansion.

Another interesting property of the exponential distribution function can be derived by differentiating Equation (2.2.17) with respect to t:

$$\begin{aligned} T\frac{dk_n(t)}{dt} &= \frac{1}{T(n-1)!}\left(\frac{t}{T}\right)^{n-1}e^{-\frac{t}{T}} - \frac{1}{Tn!}\left(\frac{t}{T}\right)^{n}e^{-\frac{t}{T}} \\ &= k_{n-1}(t) - k_n(t). \end{aligned} \tag{2.2.23}$$

Thus, the additional property is that:

$$T\frac{dk_n(t)}{dt} + k_n(t) = k_{n-1}(t) \quad \text{for } n = 0, 1, 2, \ldots, \tag{2.2.24}$$

where

$$k_{-1}(t) = \delta(t) = \begin{cases} 0 & \text{for } t < 0 \\ \infty & \text{for } t = 0 \\ 0 & \text{for } t > 0. \end{cases} \tag{2.2.25}$$

The above equation can also be regarded as a generating function for $k_n(t)$. The dynamics of such a system is graphically described in Figure 2.2.8, where each increase in the order of the kernel function represents one first-order time-lag in the dynamics.

At close inspection, the nth-order exponential distribution function is identical to the residence time distribution function of a system of n-CSTRs in series in the modeling of a chemical reactor. This is not at all surprising, if one views each

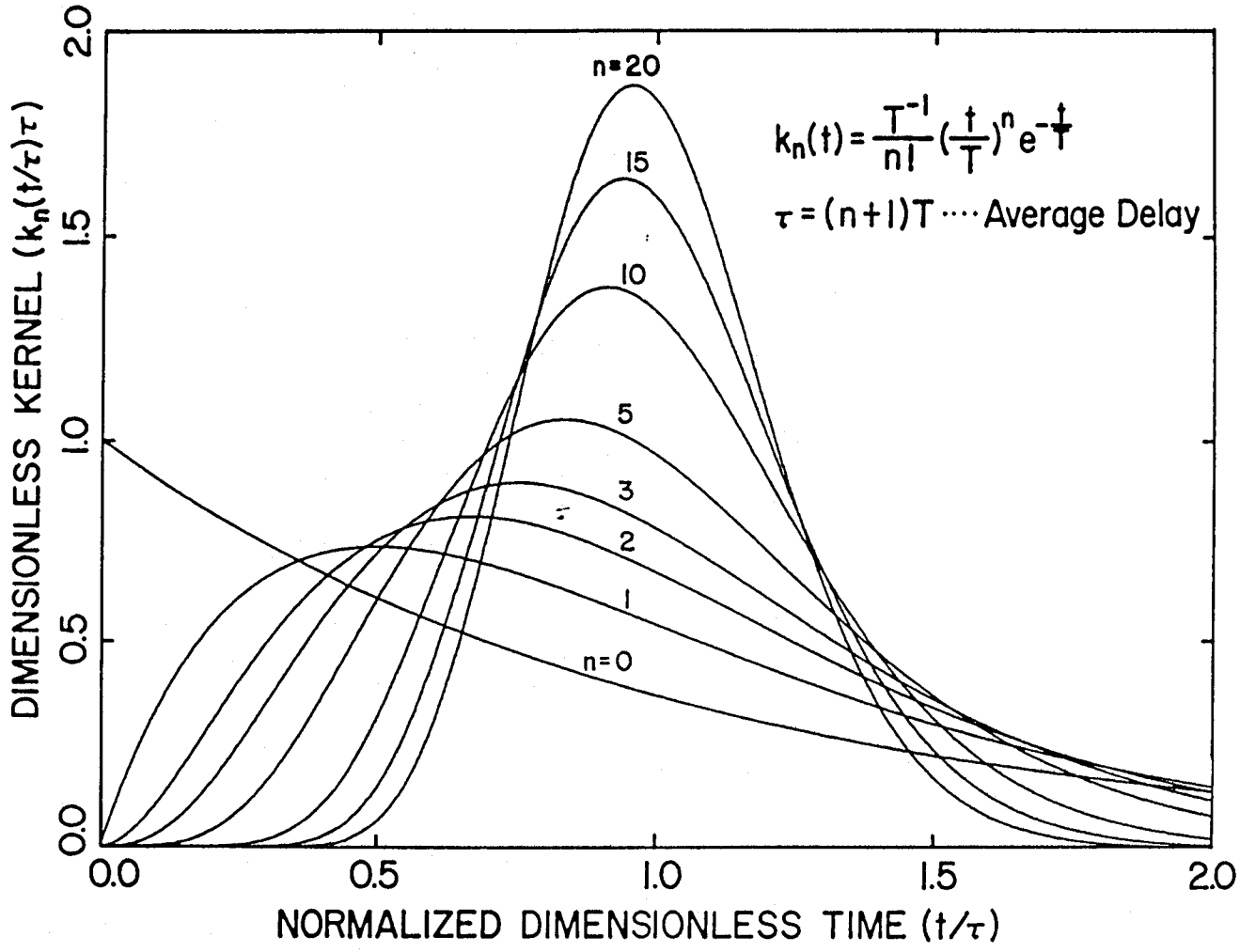


Figure 2.2.7. Exponential distribution function of order n normalized with respect to the average lag.

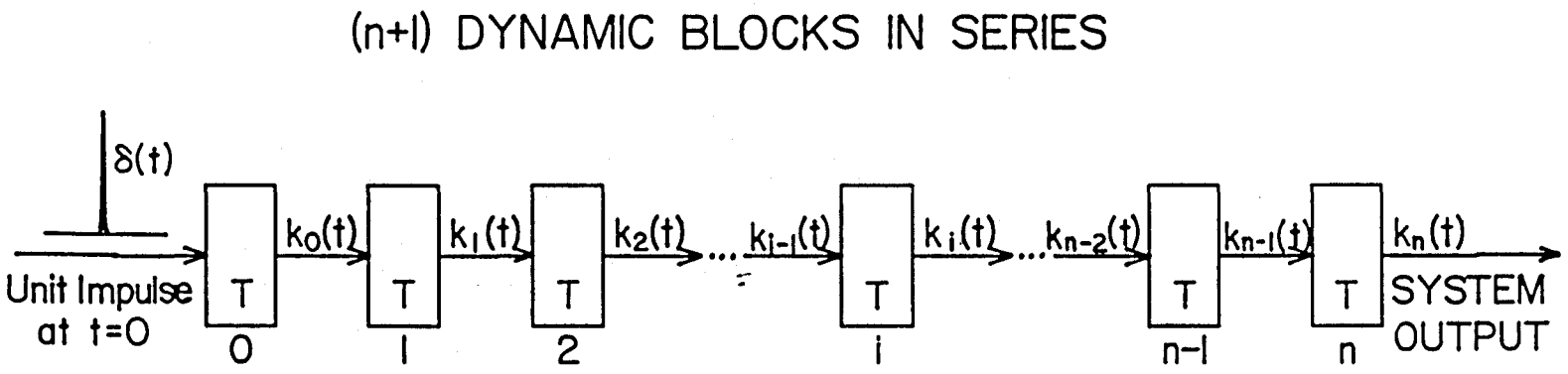


Figure 2.2.8. A series of first-order system blocks where the input to the ith dynamic block is $k_{i-1}(t)$ and the output from the ith block is $k_i(t)$. Note that the overall input to the entire system is a unit impulse and the overall output of the system exiting from the last block is described by $k_n(t)$.

block in Figure 2.2.8 as a homogeneous tank of equal volume. In this case, the unit impulse entering the system is analogous to the addition of an impulse of tracing dyes dumped into the first stirred tank at time zero to raise its initial dye concentration to unity. The dyes are then carried by the flow to the subsequent tanks and, eventually, through the entire system. The dye concentration appearing at the exit of the last tank can be described by $k_n(t)$.

If tracing dyes are added at $t = 0$ to m separate tanks as indicated in Figure 2.2.9, the dye concentration at the exit can be described by m exponential distribution functions. The number of exponential distribution functions needed to describe a system is the same as the number of distinctive entry points of impulses.

The general expression for the case where impulse disturbances (*e.g.*, tracing dyes) enter into each tank can be derived rigorously.

$$\begin{aligned}
\frac{dk_0(\tilde{t})}{d\tilde{t}} &= -k_0(\tilde{t}) \\
\frac{dk_1(\tilde{t})}{d\tilde{t}} &= -k_1(\tilde{t}) + k_0(\tilde{t}) \\
&\vdots \\
\frac{dk_{n-1}(\tilde{t})}{d\tilde{t}} &= -k_{n-1}(\tilde{t}) + k_{n-2}(\tilde{t}) \\
\frac{dk_n(\tilde{t})}{d\tilde{t}} &= -k_n(\tilde{t}) + k_{n-1}(\tilde{t})
\end{aligned} \tag{2.2.26}$$

where $\tilde{t} = \frac{t}{T}$. The above set of equations can be rewritten in a matrix form as:

$$\underbrace{\begin{bmatrix} \frac{dk_0(\tilde{t})}{d\tilde{t}} \\ \frac{dk_1(\tilde{t})}{d\tilde{t}} \\ \vdots \\ \frac{dk_{n-1}(\tilde{t})}{d\tilde{t}} \\ \frac{dk_n(\tilde{t})}{d\tilde{t}} \end{bmatrix}}_{\frac{d\mathbf{k}(\tilde{t})}{d\tilde{t}}} = \underbrace{\begin{bmatrix} -1 & 0 & \dots & 0 & 0 \\ 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 0 \\ 0 & 0 & \dots & 1 & 1 \end{bmatrix}}_{\mathbf{J}} \cdot \underbrace{\begin{bmatrix} k_0(\tilde{t}) \\ k_1(\tilde{t}) \\ \vdots \\ k_{n-1}(\tilde{t}) \\ k_n(\tilde{t}) \end{bmatrix}}_{\mathbf{k}(\tilde{t})} \tag{2.2.27a}$$

$$\frac{d\mathbf{k}(\tilde{t})}{d\tilde{t}} = \mathbf{J}\mathbf{k}(\tilde{t}). \tag{2.2.27b}$$

Note that the transition matrix for $\mathbf{k}(\tilde{t})$, $\mathbf{J}$, is already in a block-diagonal Jordan normal form whose matrix exponential expression can be especially easily obtained. The solution to the above first-order vector-matrix differential equation is simply the exponential of $\mathbf{J}$ multiplied by the initial condition of $\mathbf{k}$.

$$\begin{aligned}\mathbf{k}(\tilde{t}) &= \exp(\mathbf{J}\tilde{t})\mathbf{k}(0) \\ &= \begin{bmatrix} e^{-\tilde{t}} & 0 & \dots & 0 & 0 \\ \tilde{t}e^{-\tilde{t}} & e^{-\tilde{t}} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{t}^{n-1}e^{-\tilde{t}} & \tilde{t}^{n-2}e^{-\tilde{t}} & \dots & e^{-\tilde{t}} & 0 \\ \tilde{t}^{n}e^{-\tilde{t}} & \tilde{t}^{n-1}e^{-\tilde{t}} & \dots & \tilde{t}e^{-\tilde{t}} & e^{-\tilde{t}} \end{bmatrix} \begin{bmatrix} k_0(0) \\ k_1(0) \\ \vdots \\ k_{n-1}(0) \\ k_n(0) \end{bmatrix}\end{aligned} \tag{2.2.28}$$

Similar to the introduction of an impulse to the 0th tank in our original treatment, the presence of the impulse at points other than the 0th tank is signified by the nonzero initial condition for $k_i(0)$. As a demonstration, if the impulse is allowed to enter the 2nd tank and the ith tank with magnitudes of 0.3 and 0.7, the output from the system would be:

$$\mathbf{k}(\tilde{t}) = \begin{bmatrix} e^{-\tilde{t}} & 0 & \dots & 0 & 0 \\ \tilde{t}e^{-\tilde{t}} & e^{-\tilde{t}} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{t}^{n-1}e^{-\tilde{t}} & \tilde{t}^{n-2}e^{-\tilde{t}} & \dots & e^{-\tilde{t}} & 0 \\ \tilde{t}^{n}e^{-\tilde{t}} & \tilde{t}^{n-1}e^{-\tilde{t}} & \dots & \tilde{t}e^{-\tilde{t}} & e^{-\tilde{t}} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ k_2(0) = \frac{0.3}{T} \\ \vdots \\ k_i(0) = \frac{0.7}{T} \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{2.2.29}$$

Expanding the matrix multiplication, one obtains the output from the end of the system:

$$k_n(t) = \frac{0.3}{T}\left(\frac{t}{T}\right)^{n-2}e^{-\frac{t}{T}} + \frac{0.7}{T}\left(\frac{t}{T}\right)^{n-i}e^{-\frac{t}{T}} \tag{2.2.30}$$

In addition, the intermediate values along various parts of this specialized example system are also available as individual columns in the above matrix expression. This slight detour further clarifies the relationship between a time-lag kernel and n CSTRs in series. Note that the resulting kernel is merely a linear combination of the

original set of base exponential distribution functions. Also note that the original case where an impulse is introduced only at one end of n first-order systems in series is contained in this general treatment as a specilized case. However, for simplicity, only this specilized case will be emphasized hereforth.

Accordingly, if $k(t)$ is expressed as the sum of m exponential distribution functions, the observed specific growth rate at time t, expressed as a functional $y(t) \equiv \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh$, will be the weighed sum of m integrals, each of which being $y_j(t) \equiv \int_{-\infty}^{t} \mu[s(h)]k_j(t-h)dh$ $j = 1, 2, ..., m$. This argument can be shown to be true by the following equation, starting with $k(t) = \sum_{j=0}^{m} a_j k_j(t)$:

$$\begin{aligned} y(t) &= \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh \\ &= \int_{-\infty}^{t} \mu[s(h)] \overbrace{\left[\sum_{j=0}^{m} a_j k_j(t-h)\right]}^{k(t-h)} dh \\ &= \sum_{j=0}^{m} a_j \underbrace{\left[\int_{-\infty}^{t} \mu[s(h)]k_j(t-h)dh\right]}_{y_j(t)} \\ &= \sum_{j=0}^{m} a_j y_j(t). \end{aligned} \tag{2.2.31}$$

In actuality, the weighing factors a_j's and the lag time-constant T are chosen in such a way as to fit the observed transient of the specific system in a shift-up or shift-down experiment. A small value of m varying from one to three usually gives a very satisfactory fit, and it is rarely necessary to employ m with orders larger than four.

Quite significantly, we are not bounded by the limited functional shapes of each individual exponential distribution function. By expressing the kernel as a linear combination of these base functions, any sufficiently smooth continuously differentiable function can be represented if a sufficiently large number of base

POSSIBLE POINTS OF DISTURBANCE ENTRY

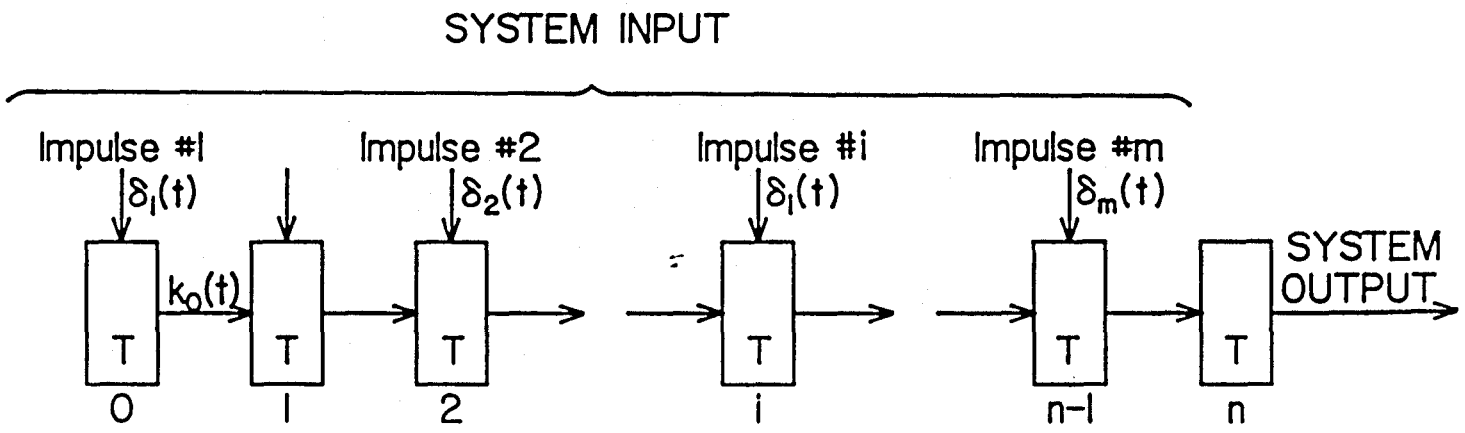


Figure 2.2.9. Introduction of impulses or addition of m dyes to a series of $n+1$ tanks at various points.

functions are used. This is because the approach is essentially the same as expanding the function $g(t) = e^{\frac{t}{T}}k(t)$ by a power series. Multiplying the resulting power series by $e^{-\frac{t}{T}}$ yields the series expansion of function $k(t)$ in terms of the exponential distribution functions as the base functions.

It may be rigorously proved that an analytical function $k(t)$ can be expressed by a series of exponential distribution functions if the order is sufficiently large. Basically, the proof proceeds as the following. Given that $k(t)$ is an analytical function and that $e^{\frac{t}{T}}$ is certainly an analytical function, it follows that the product of these two analytical functions, $e^{\frac{t}{T}}k(t)$, is also an analytical function. Since any analytical function can be expended in Taylor's power series, it suffices to state that $e^{\frac{t}{T}}k(t)$ can be approximated by:

$$e^{\frac{t}{T}}k(t) = a_0 + a_1\left(\frac{t}{T}\right) + a_2\left(\frac{t}{T}\right)^2 + \ldots + a_m\left(\frac{t}{T}\right)^m + \ldots, \tag{2.2.32}$$

which, when multiplied by $e^{-\frac{t}{T}}$, yields:

$$\begin{aligned} k(t) &= a_0e^{-\frac{t}{T}} + a_1\left(\frac{t}{T}\right)e^{-\frac{t}{T}} + a_2\left(\frac{t}{T}\right)^2 e^{-\frac{t}{T}} + \ldots + a_m\left(\frac{t}{T}\right)^m e^{-\frac{t}{T}} + \ldots \\ &= a_0k_0(t) + a_1k_1(t) + a_2k_2(t) + \ldots + a_mk_m(t) + \ldots \end{aligned} \tag{2.2.33}$$

Theoretically, m could be extended to ∞, but two or three terms should be sufficient under most circumstances in practice. For example, a linear combination of $k_0(t)$ and $k_1(t)$ results in:

$$k(t) = \left(a_0\frac{1}{T} + a_1\frac{t}{T^2}\right)e^{-\frac{t}{T}}, \tag{2.2.34}$$

where $a_0 + a_1 = 1$ so that the area under the kernel is normalized to unity. Some of the shapes of $k(t)$ generated by a combination of these two base functions are shown in Figure 2.2.10. The block process diagram for this simple linear combination is shown in Figure 2.2.11. Note that the first term corresponds to the system response at the exit as a result of the impulse stimulus introduced at the last block, and the

second term represents the response due to the impulse stimulus imposed on the first block. The component associated with a_1 must propagate through two blocks before exiting; whereas, that associated with a_0 travels through only a single block. Furthermore, it is also possible to generate many other shapes for the kernel with different combinations of higher-order base functions that are linearly independent. This is equivalent to having impulses directed at various blocks as indicated in Figure 2.2.9.

For example, a linear combination of two or more exponential distribution functions of different time constants is also possible if the process responsible for the time-lag is known to possess more than one characteristic time constant. The simplest two block combination can be described by:

$$k(t) = a_a \frac{1}{T_a} e^{-\frac{t}{T_a}} + a_b \frac{1}{T_b} e^{-\frac{t}{T_b}}, \tag{2.2.35}$$

where $a_a + a_b = 1$ as seen before. Schematically, this corresponds to two parallel tanks of unequal volumes, thus, unequal residence time constants. See Figure 2.2.12. For an impulse going through two system blocks of unequal time constants in series in a noninteracting sequential manner, dynamic equations for each block can be written as:

$$T_a \frac{dk_a(t)}{dt} + k_a(t) = \delta(t) \tag{2.2.36a}$$

$$T_b \frac{dk_b(t)}{dt} + k_b(t) = k_a(t). \tag{2.2.36b}$$

The output from the first block $k_a(t)$ is simply the 0th-order kernel function:

$$\begin{aligned} k_a(t) &= \int_{-\infty}^{t} \delta(h) \cdot \frac{1}{T_a} e^{-\frac{t-h}{T_a}} dh \\ &= \frac{1}{T_a} e^{-\frac{t}{T_a}}. \end{aligned} \tag{2.2.37a}$$

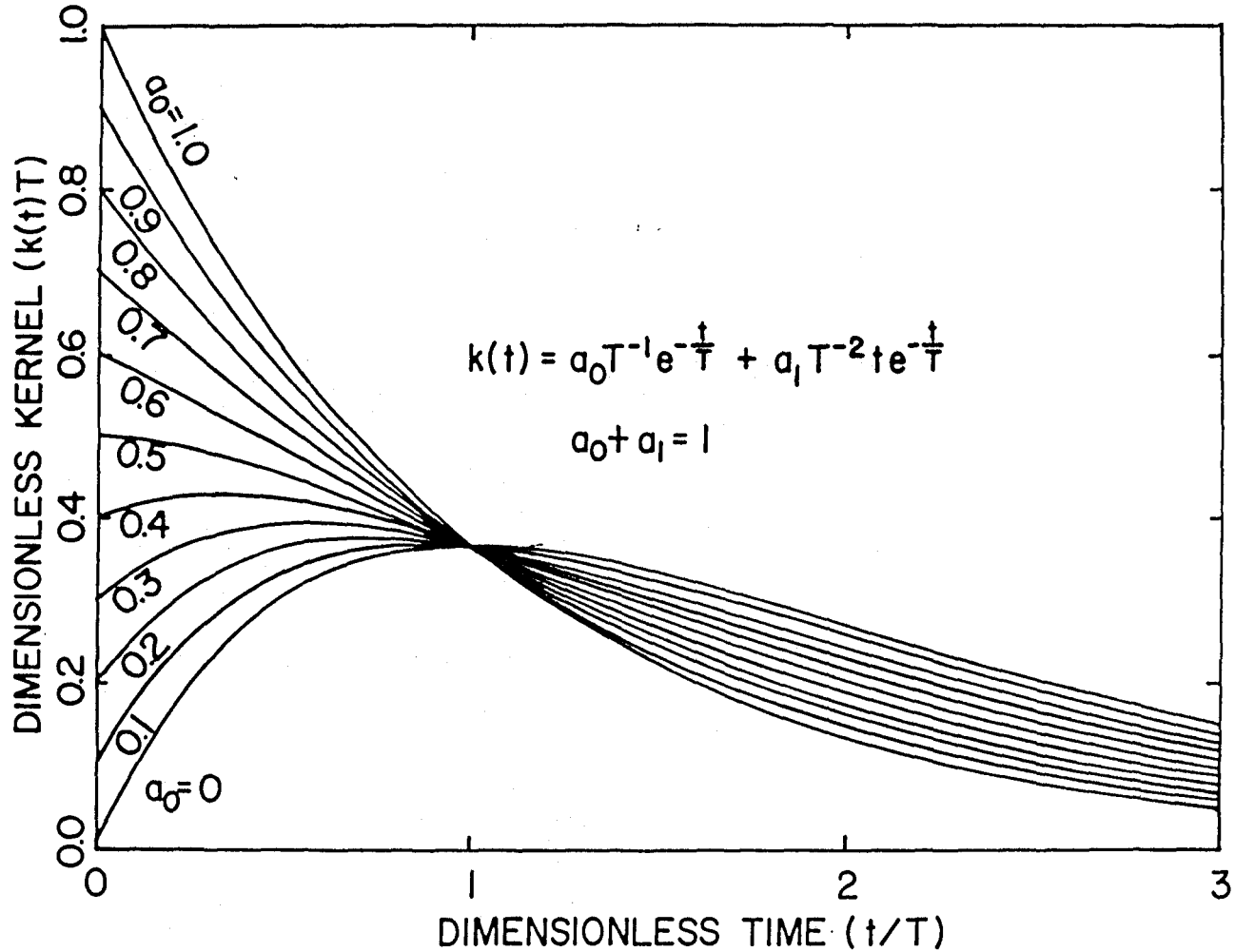


Figure 2.2.10. Linear combination of the 0th-order exponential distribution function, $k_0(t)$, and the 1st-order exponential distribution function, $k_1(t)$;

$$k(t) = a_0 k_0(t) + a_1 k_1(t), \quad a_0 + a_1 = 1$$

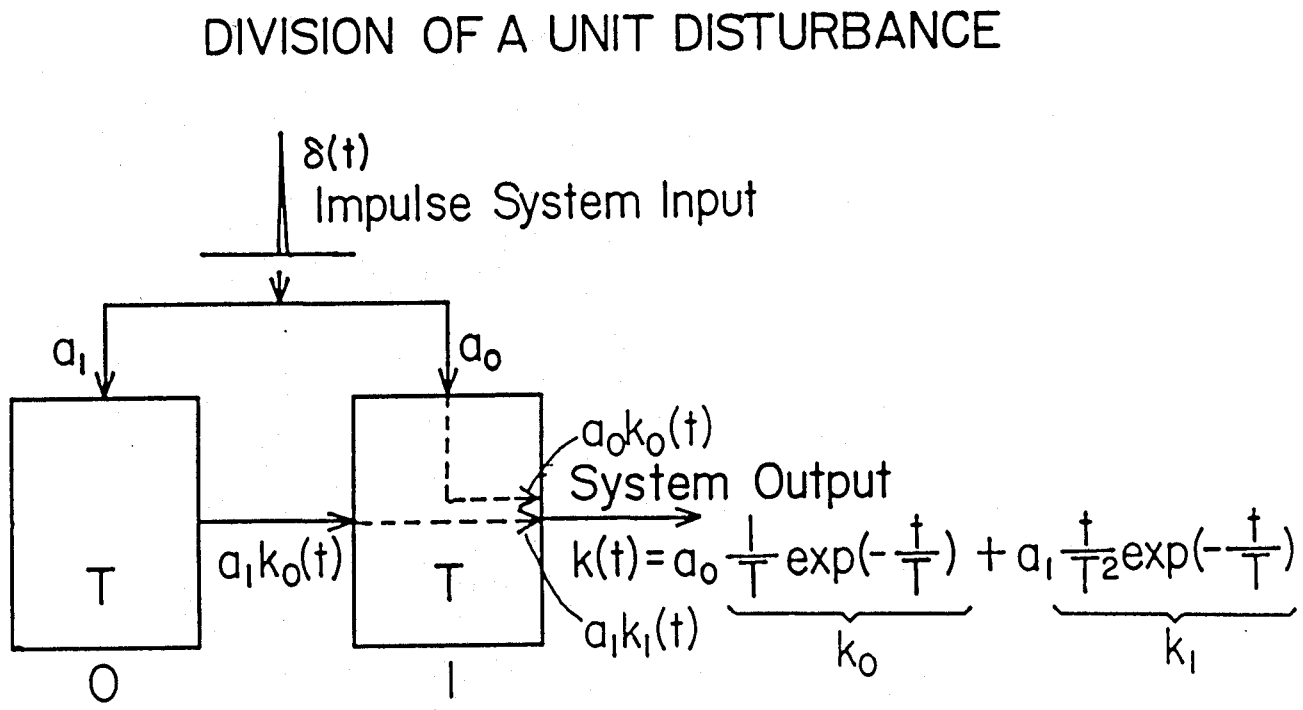


Figure 2.2.11. Simultaneous introduction of a unit impulse to two system blocks. The magnitude of the impulse is divided between the first block and the second block as indicated by a_1 and a_0, respectively.

And the output from the second block $k_b(t)$ is:

$$\begin{aligned} k_b(t) &= \int_{-\infty}^{t} k_a(h) \cdot \frac{1}{T_b} e^{-\frac{t-h}{T_b}} dh \\ &= \int_{-\infty}^{t} \frac{1}{T_a} e^{-\frac{h}{T_a}} \frac{1}{T_b} e^{-\frac{t-h}{T_b}} dh \\ &= \frac{1}{T_a - T_b} \left(e^{-\frac{t}{T_a}} - e^{-\frac{t}{T_b}} \right). \end{aligned} \tag{2.2.37b}$$

These results can be more easily obtained by using transfer functions:

$$(T_a s + 1) K_a(s) = Q(s) \tag{2.2.38b}$$

$$(T_b s + 1) K_b(s) = K_a(s). \tag{2.2.38b}$$

Which may be easily solved algebraically to yield:

$$K_b(s) = \frac{1}{(T_a s + 1)(T_b s + 1)} Q(s) = \frac{1}{T_a - T_b} \left(\frac{T_a}{(T_a s + 1)} + \frac{T_b}{(T_b s + 1)} \right). \tag{2.2.38c}$$

Or in time domain, it is equivalent to:

$$k_b(t) = \begin{cases} \frac{1}{T_a - T_b} \left(e^{-\frac{t}{T_a}} + e^{-\frac{t}{T_b}} \right) & \text{for } T_a \neq T_b \\ \frac{1}{T} e^{-\frac{t}{T}} + \frac{t}{T^2} e^{-\frac{t}{T}} & \text{for } T_a = T_b = T \, . \end{cases} \tag{2.2.38d}$$

Similar to passing through two dynamic systems in parallel, passing through two different dynamic systems in series results in an overall kernel that is a linear combination of two exponential functions, each possessing the characteristic time constant of the respective dynamic system block. The only exception is when two dynamic blocks have exactly the same time constant, in which case the solution reverts back to the original first-order kernel base function.

When there is interaction between the two dynamic systems in series as shown in Figure 2.2.13, Equation (2.2.36) is modified slightly.

$$T_a \frac{dk_a(t)}{dt} + \alpha k_a(t) = L_{ab} k_b(t) + q_a(t) \tag{2.2.39a}$$

$$T_b \frac{dk_a(t)}{dt} + \beta k_b(t) = L_{ba} k_a(t) + q_b(t). \tag{2.2.39b}$$

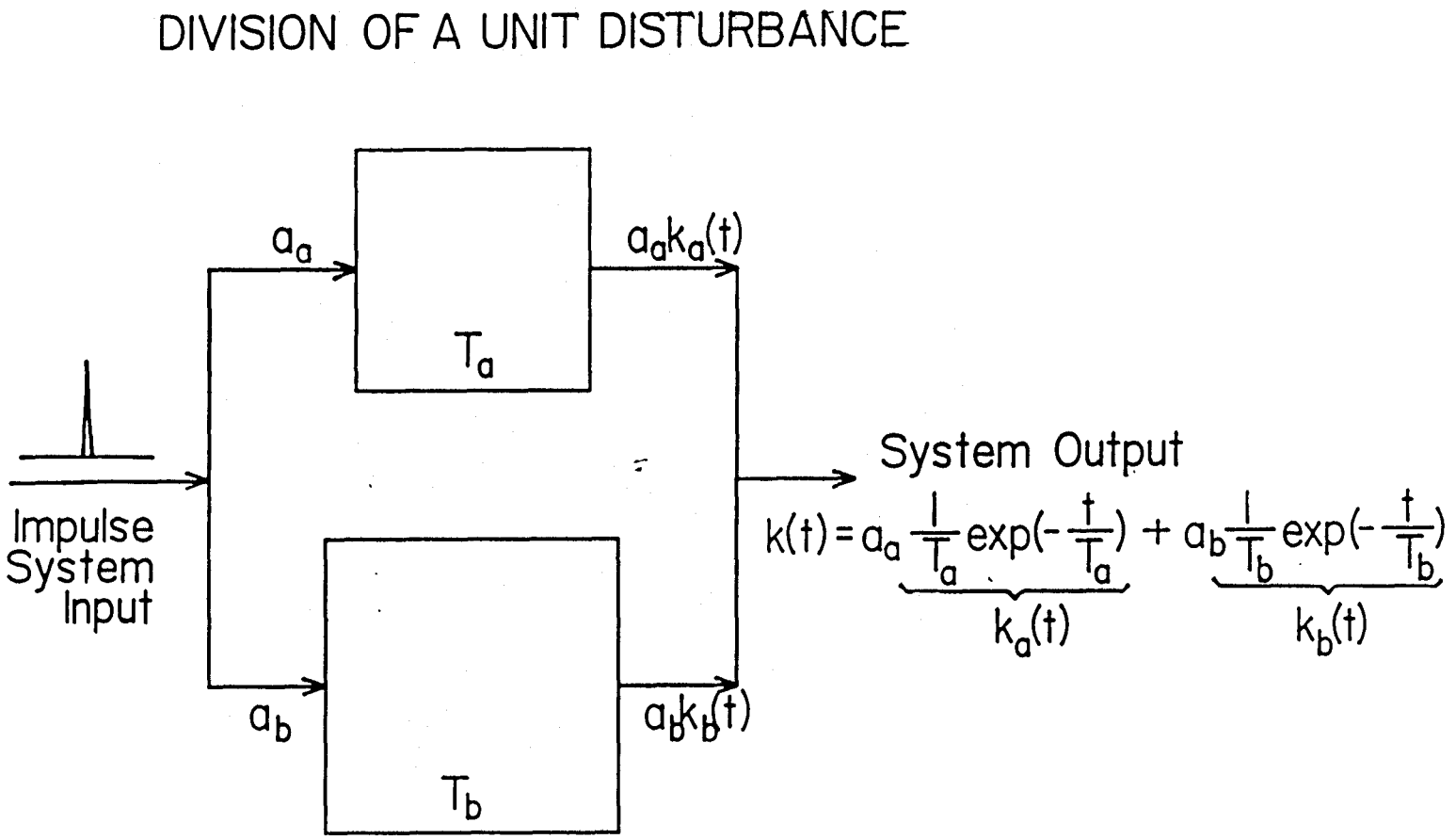


Figure 2.2.12. Simultaneous introduction of a unit impulse to two parallel system blocks of unequal dynamic time constants.

In the Laplace domain, they are equivalent to:

$$(T_a s + \alpha) K_a(s) = L_{ab} K_b(s) + Q_a(s) \tag{2.2.40a}$$

$$(T_b s + \beta) K_b(s) = L_{ba} K_a(s) + Q_b(s). \tag{2.2.40b}$$

The overall transfer function for this second-order system is:

$$K_b(s) = \frac{L_{ba}}{(T_a s + \alpha)(T_b s + \beta) - L_{ba} L_{ab}} Q_a(s) + \frac{(T_a s + \alpha)}{(T_a s + \alpha)(T_b s + \beta) - L_{ba} L_{ab}} Q_b(s). \tag{2.2.41}$$

For example, when $T_a = 1$, $T_b = 2$, $\alpha = 1$, $\beta = 1$, $L_{ab} = 0.5$, and $L_{ba} = 1$, the characteristic equation of the system is:

$$\begin{aligned} (T_a s + \alpha)(T_b s + \beta) - L_{ba} L_{ab} &= 0 \\ (s + 1)(2s + 1) - 0.5 = 2s^2 + 3s + 0.5 &= 0, \end{aligned} \tag{2.2.42}$$

which has the roots $s_1 = -1.309$ and $s_2 = -0.1909$. Thus, the new characteristic time constants are $T_1 = \frac{1}{1.309} = 0.7639$ and $T_2 = \frac{1}{0.1909} = 5.236$. Therefore, this interacting second-order system can also be described by an equivalent two first-order systems in parallel. Note that the effect of interaction is reflected mainly in the modified time constants, with one being shortened and the other being lengthened. In general, the effect of interaction is to retard the system response. This interacting example just considered is shown in Figure 2.2.14, which may sometimes be viewed as a positive feedback or recycle.

A general second-order system subject to an impulse can be described by:

$$T^2 \frac{d^2 k(t)}{dt^2} + 2\xi T \frac{dk(t)}{dt} + k(t) = \delta(t). \tag{2.2.43}$$

The solution to this general form depends on the magnitude of the parameter ξ.

$$k(t) = \begin{cases} \frac{1}{T\sqrt{1-\xi^2}} e^{-\frac{\xi t}{T}} \sin(\sqrt{1-\xi^2}\frac{t}{T}) & \text{for } 0 < \xi < 1 \\ \frac{t}{T^2} e^{-\frac{t}{T}} & \text{for } \xi = 1 \\ \frac{1}{T\sqrt{\xi^2-1}} e^{-\frac{\xi t}{T}} \sinh(\sqrt{\xi^2-1}\frac{t}{T}) & \text{for } \xi > 1. \end{cases} \tag{2.2.44}$$

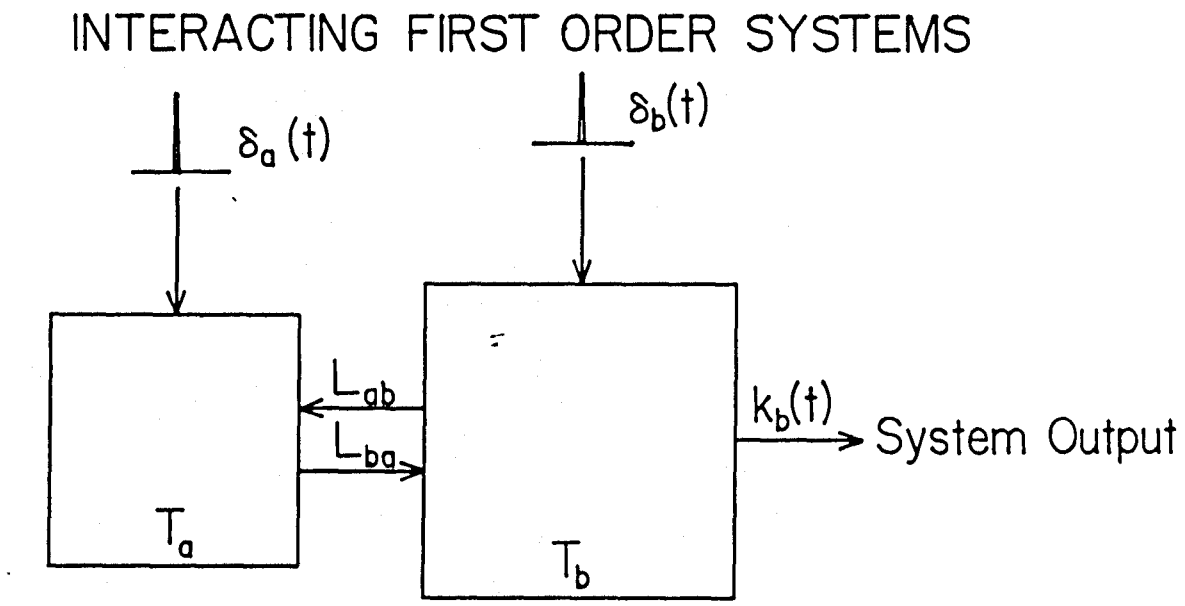


Figure 2.2.13. Introduction of unit impulses to two interacting system blocks of unequal dynamic time constants.

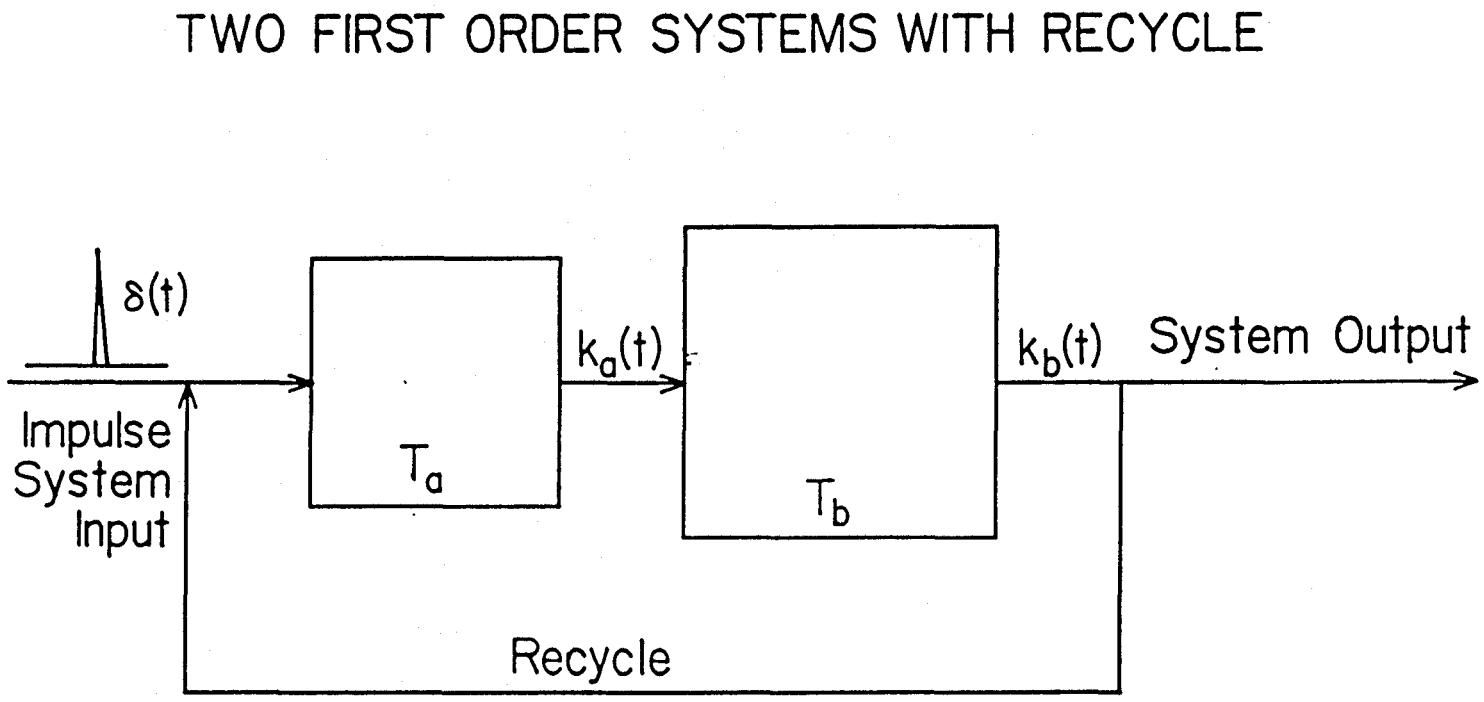


Figure 2.2.14. System with a positive feedback or recycle structure.

Note that the case $\xi = 1$ is the general kernel used throughout this thesis; whereas, the case $\xi > 1$ gives rise to two system time constants and has just been treated. The last case with $\xi < 1$ is under-damped and yields oscillatory responses. Thus, negative values are permissible for a kernel. This confirms the earlier contention that although a kernel may be considered as a weighing function, strictly speaking, it is not a probability distribution function for which negative values are mathematically undefined.

The more general treatment of a two-dimensional linear dynamic system can be facilitated in a matrix notation.

$$\underbrace{\begin{bmatrix} \frac{dx_1(t)}{dt} \\ \frac{dx_2(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}(t)}{dt}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}}_{\mathbf{q}(t)} \tag{2.2.45}$$

The important parameters for this general set of dynamic equations can be calculated from the fundamental matrix **A** as follows.

$$T = \frac{1}{(\mathrm{detA})^{\frac{1}{2}}} \tag{2.2.46}$$

$$\xi = \frac{-\mathrm{trA}}{2(\mathrm{detA})^{\frac{1}{2}}} \tag{2.2.47}$$

The system response's dependence on the parameter ξ is the same before.

In summary, a mixture of the above two methods of combination with equal and/or unequal system time constants results in:

$$k(t) = a_0 \frac{1}{T_0} e^{-\frac{t}{T_0}} + a_1 \frac{t}{T_1^2} e^{-\frac{t}{T_1}}, \tag{2.2.48}$$

where, again, $a_0 + a_1 = 1$.

One can extend this manipulation further to a general third-order system such as Figure 2.2.15, and so on. Of course, theoretically, one need not be restricted to the straight flow-through format of Figure 2.2.8 or Figure 2.2.9. The dynamic

blocks and flow streams may be arranged in an infinite number of ways. To demonstrate some possible configurations, a hypothetical systems block diagram is shown in Figure 2.2.16. As in classical process control, the flow diagram may contain feedforward streams, feedback/recycle streams, and cross interactions; the impulse disturbance may also be introduced at various points. Furthermore, a group of blocks may be combined together to give a single equivalent block. Or, conversely, a block with complicated system dynamics may be decomposed into a set of simpler, preferably first-order, blocks that display the same dynamics as the original.

Since the performance of a thorough and exhaustive systems analysis of any or all possible combinations of such complicated configurations is not the immediate objective of this study, it is not pursued further. It suffices to state that this thesis recognizes the presence of these dynamic blocks that prevent the effect of an input to a system to be observed instantaneously at the exit. The exact structure of the lag or delay is closely related to the configuration of these dynamic blocks; therefore, the nature of the blocks can be inferred from the observation of lag behaviors. The later part of this thesis is devoted to the experimental determination of the number and the time constant of the dynamic blocks for a biochemical reactor. The fundamental approach taken in this thesis is to describe a system as a series of simple first-order dynamic blocks represented graphically in Figure 2.2.8. This thesis also attempts to approximate a system with complicated dynamics, *e.g.*, Figure 2.2.16, into an equivalent but simpler one.

2.3 SOLUTIONS TO GENERAL TIME-LAG EQUATIONS

The reason for choosing exponential probability distribution functions is that they permit the easy and elegant transformation of a set of integro-differential equations into a set of simple ordinary differential equations. These exponential

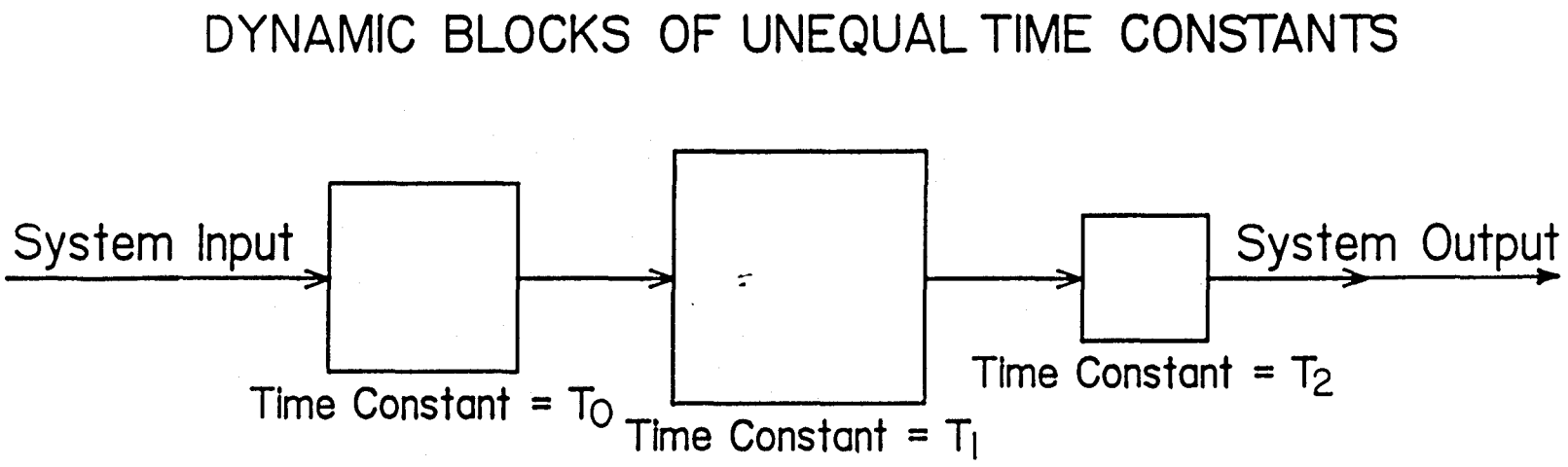


Figure 2.2.15. Simultaneous introduction of a unit impulse to three system blocks of unequal dynamic time constants in series.

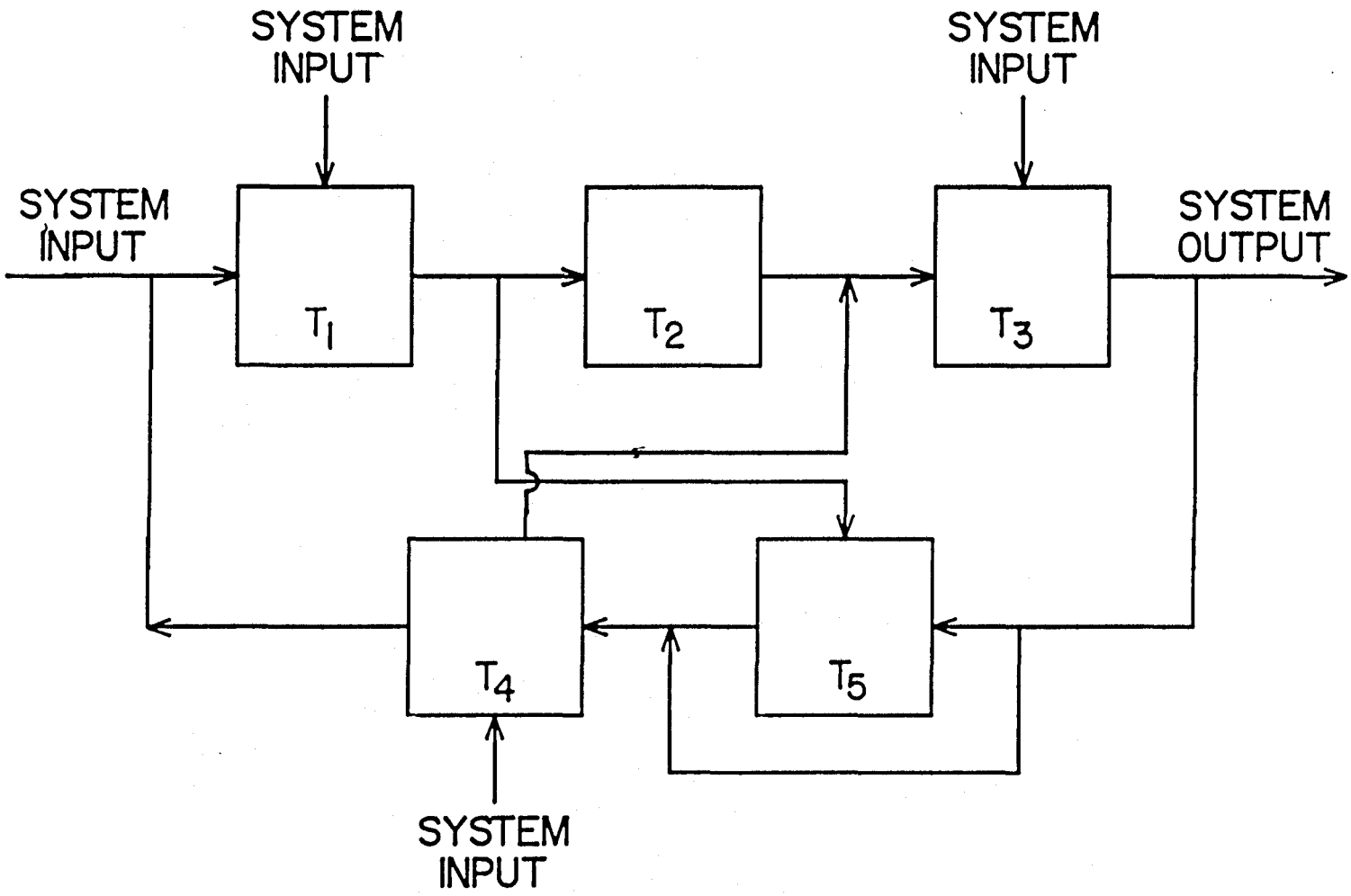


Figure 2.2.16. A hypothetical general system block diagram with feedforward, feedback, and cross interactions.

distribution functions possess the property that each and every one of them is the solution to the following differential equation:

$$\boxed{\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i k_n(t)}{dt^i} = 0}, \tag{2.3.1}$$

with the initial conditions:

$$\frac{d^i k_n(0)}{dt^i} = 0 \qquad \text{for } i = 0, 1, 2, \ldots, n-1 \tag{2.3.2}$$

$$\frac{d^n k_n(0)}{dt^n} = \frac{1}{T^{n+1}} \qquad \text{for } i = n. \tag{2.3.3}$$

For example, $k_0(t) = \frac{t}{T} e^{-\frac{t}{T}}$ satisfies:

$$T \frac{dk_0(t)}{dt} + k_0(t) = 0 \tag{2.3.4}$$

with the initial condition:

$$k_0(0) = \frac{1}{T}. \tag{2.3.5}$$

Similarly, $k_1(t) = \frac{t}{T^2} e^{-\frac{t}{T}}$ satisfies:

$$T^2 \frac{d^2 k_1(t)}{dt^2} + 2T \frac{dk_1(t)}{dt} + k_1(t) = 0 \tag{2.3.6}$$

with the set of initial conditions:

$$k_1(0) = 0 \tag{2.3.7}$$

$$\frac{dk_1(0)}{dt} = \frac{1}{T^2}. \tag{2.3.8}$$

The above properties of the exponential distribution functions can be used to eliminate the kernel from the integro-differential Equations (2.2.12) and (2.2.13) and convert them into a larger, but mathematically identical, set of first-order ordinary differential equations.

If we treat the integral containing the kernel as a new function $y_n(t)$, defined as:

$$y_n(t) \equiv \int_{-\infty}^{t} \mu(h) k_n(t-h) dh, \tag{2.3.9.0}$$

then repeatedly differentiating $y_n(t)$ with respect to t with the help of Leibnitz's rule yields:

$$\frac{dy_n(t)}{dt} = \int_{-\infty}^{t} \mu(h)\frac{dk_n(t-h)}{dt}dh + \mu(t)k(0) \qquad (2.3.9.1)$$

$$\frac{d^2y_n(t)}{dt^2} = \int_{-\infty}^{t} \mu(h)\frac{d^2k_n(t-h)}{dt^2}dh + \frac{d\mu(t)}{dt}k(0) + \mu(t)\frac{dk_n(0)}{dt} \qquad (2.3.9.2)$$

$$\frac{d^3y_n(t)}{dt^3} = \int_{-\infty}^{t} \mu(h)\frac{d^3k_n(t-h)}{dt^3}dh + \frac{d^2\mu(t)}{dt^2}k(0) + \frac{d\mu(t)}{dt}\frac{dk_n(0)}{dt} + \mu(t)\frac{d^2k_n(0)}{dt^2} \qquad (2.3.9.3)$$

$$\vdots \qquad (2.3.9)$$

$$\frac{d^{n-1}y_n(t)}{dt^{n-1}} = \int_{-\infty}^{t} \mu(h)\frac{d^{n-1}k_n(t-h)}{dt^{n-1}}dh + \frac{d^{n-2}\mu(t)}{dt^{n-2}}k(0) + \frac{d^{n-3}\mu(t)}{dt^{n-3}}\frac{dk_n(0)}{dt} + \ldots + \frac{d\mu(t)}{dt}\frac{d^{n-3}k_n(0)}{dt^{n-3}} + \mu(t)\frac{d^{n-2}k_n(0)}{dt^{n-2}} \qquad (2.3.9.n-1)$$

$$\frac{d^{n}y_n(t)}{dt^{n}} = \int_{-\infty}^{t} \mu(h)\frac{d^{n}k_n(t-h)}{dt^{n}}dh + \frac{d^{n-1}\mu(t)}{dt^{n-1}}k(0) + \frac{d^{n-2}\mu(t)}{dt^{n-2}}\frac{dk_n(0)}{dt} + \ldots + \frac{d\mu(t)}{dt}\frac{d^{n-2}k_n(0)}{dt^{n-2}} + \mu(t)\frac{d^{n-1}k_n(0)}{dt^{n-1}} \qquad (2.3.9.n)$$

$$\frac{d^{n+1}y_n(t)}{dt^{n+1}} = \int_{-\infty}^{t} \mu(h)\frac{d^{n+1}k_n(t-h)}{dt^{n+1}}dh + \frac{d^{n}\mu(t)}{dt^{n}}k(0) + \frac{d^{n-1}\mu(t)}{dt^{n-1}}\frac{dk_n(0)}{dt} + \ldots + \frac{d\mu(t)}{dt}\frac{d^{n-1}k_n(0)}{dt^{n-1}} + \mu(t)\frac{d^{n}k_n(0)}{dt^{n}}. \qquad (2.3.9.n+1)$$

Each of the above equations is first multiplied by $\binom{n+1}{i}T^i$, where $\binom{n+1}{i}$ is the ith binomial coefficient that corresponds to the order of the derivative in $y(t)$, as indicated by the last number of the equation label (2.3.9). Adding the resulting

equations and collecting terms that are common in $\frac{d^i k_n(0)}{dt^i}$ gives:

$$\begin{aligned}\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_n(t)}{dt^i} &= \int_{-\infty}^{t}\mu(h)\left[\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i k_n(t-h)}{dt^i}\right]dh \\ &+ \left[\sum_{i=1}^{n+1}\binom{n+1}{i}T^i\frac{d^{i-1}\mu(t)}{dt^{i-1}}\right]k(0) \\ &+ \left[\sum_{i=2}^{n+1}\binom{n+1}{i}T^i\frac{d^{i-2}\mu(t)}{dt^{i-2}}\right]\frac{dk_n(0)}{dt} \\ &+ \left[\sum_{i=3}^{n+1}\binom{n+1}{i}T^i\frac{d^{i-3}\mu(t)}{dt^{i-3}}\right]\frac{d^2k_n(0)}{dt^2} \\ &+ \cdots \\ &+ \left[(n+1)T^n\mu(t)+T^{n+1}\frac{d\mu(t)}{dt}\right]\frac{d^{n-1}k_n(0)}{dt^{n-1}} \\ &+ T^{n+1}\mu(t)\frac{d^n k_n(0)}{dt^n}.\end{aligned} \tag{2.3.10}$$

As indicated in Equation (2.3.1), one of the powerful properties of the exponential distribution functions is that $\sum_{i=0}^{n+1} T^i\binom{n+1}{i}\frac{d^i k_n(t)}{dt^i} = 0$. Thus, with the integrand of the above equation being identically equal to zero, the above equation is reduced to one without the integral:

$$\begin{aligned}\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_n(t)}{dt^i} &= \left[\sum_{i=1}^{n+1}\binom{n+1}{i}T^i\frac{d^{i-1}\mu(t)}{dt^{i-1}}\right]k(0) \\ &+ \left[\sum_{i=2}^{n+1}\binom{n+1}{i}T^i\frac{d^{i-2}\mu(t)}{dt^{i-2}}\right]\frac{dk_n(0)}{dt} \\ &+ \left[\sum_{i=3}^{n+1}\binom{n+1}{i}T^i\frac{d^{i-3}\mu(t)}{dt^{i-3}}\right]\frac{d^2k_n(0)}{dt^2} \\ &+ \cdots \\ &+ \left[(n+1)T^n\mu(t)+T^{n+1}\frac{d\mu(t)}{dt}\right]\frac{d^{n-1}k_n(0)}{dt^{n-1}} \\ &+ T^{n+1}\mu(t)\frac{d^n k_n(0)}{dt^n}.\end{aligned} \tag{2.3.11}$$

Since $\frac{d^i k_n(0)}{dt^i} = 0$ for $i = 0, 1, 2, \ldots, n-1$ from Equation (2.3.2), all the $\frac{d^i k_n(0)}{dt^i}$ terms on the right-hand side of the above equation, except for the last one, are

identically equal to zero. Thus, with the additional help of Equation (2.3.3) in evaluating $\frac{d^n k_n(0)}{dt^n}$, which is equal to $\frac{1}{T^{n+1}}$, Equation (2.3.10) is further reduced to a much simpler form:

$$\boxed{\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y_n(t)}{dt^i} = \mu(t)}. \tag{2.3.12}$$

The simplicity of the last equation is not at all surprising nor coincidental, for simplicity is what we are striving to achieve. As a matter of fact, the original theoretical basis for choosing the exponential distribution functions as the base functions stems from the desire to replace the integral expression of Equation (2.3.9.0) with a highly simplified differential equation such as Equation (2.3.12). Specifically, the base function $k_n(t)$ has been chosen to enable just such a simplification. Note that the choice of the base function $k_n(t)$ is not unique; it need not be exactly the same exponential distribution function as we have adopted, as shown by the following derivation. However, other forms of simplified equations that are different from Equation (2.3.12) will result if other types of functions are employed as the base kernel functions. For example, if another base kernel function is the solution to the same differential equation as Equation (2.3.1):

$$\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i k_n(t)}{dt^i} = 0, \tag{2.3.13}$$

but with a slightly different set of initial conditions:

$$k(0) \neq 0 \qquad \text{for } i = 0 \tag{2.3.14}$$

$$\frac{d^i k_n(0)}{dt^i} = 0 \qquad \text{for } i = 1, 2, 3 \ldots, n, \tag{2.3.15}$$

then the new equivalent differential equation that is satisfied by $y_n(t)$ is:

$$\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y_n(t)}{dt^i} = \left[\sum_{i=1}^{n+1} \binom{n+1}{i} T^i \frac{d^{i-1} \mu(t)}{dt^{i-1}} \right] k_n(0), \tag{2.3.16}$$

which is similar to Equation (2.3.12) but definitely not exactly the same. In general, any of the terms appearing in Equation (2.3.11) may be retained by specifying a slightly different set of initial conditions for $\frac{d^i k_n(0)}{dt^i}$.

Appreciation for the meaning of the kernel can be further enhanced by tracing the steps followed to derive the base functions. The choice of a "nice" base kernel function can be arrived at by the following argument. Starting with Equation (2.3.9.0), one wishes to eliminate the integral first by taking the derivatives of Equation (2.3.9.0) repeatedly, then somehow by adding these derivatives with different combinations, and finally by choosing the kernel base function that enables the integrand to be set to zero after similar integral terms have been collected. Adding the derivatives of Equation (2.3.9) linearly yields the following equation that is similar to Equation (2.3.10):

$$\begin{aligned}\sum_{i=0}^{n+1} b_i \frac{d^i y_n(t)}{dt^i} &= \int_{-\infty}^{t} \mu(h) \sum_{i=0}^{n+1} b_i \frac{d^i k_n(t-h)}{dt^i} dh \\ &+ \left[\sum_{i=1}^{n+1} b_i \frac{d^{i-1}\mu(t)}{dt^{i-1}}\right] k(0) \\ &+ \left[\sum_{i=2}^{n+1} b_i \frac{d^{i-2}\mu(t)}{dt^{i-2}}\right] \frac{dk_n(0)}{dt} \\ &+ \left[\sum_{i=3}^{n+1} b_i \frac{d^{i-3}\mu(t)}{dt^{i-3}}\right] \frac{d^2 k_n(0)}{dt^2} \\ &+ \cdots \\ &+ \left[\sum_{i=n}^{n+1} b_i \frac{d^{i-n}\mu(t)}{dt^{i-n}}\right] \frac{d^2 k_n(0)}{dt^2} \\ &+ b_{n+1}\mu(t) \frac{d^n k_n(0)}{dt^n}.\end{aligned} \tag{2.3.17}$$

Since $k_n(t)$ is still undefined at this point in the derivation, theoretically one can choose any set of functions as long as they are linearly independent. Forcing the

integrand in the above equation to be equal to zero yields the first condition that must be satisfied by the kernel base functions:

$$\begin{aligned}
\mathcal{L}\{k_n(t)\} &= \sum_{i=0}^{n+1} b_i \frac{d^i k_n(t)}{dt^i} \\
&= \left(b_0 + b_1 \frac{d}{dt} + b_2 \frac{d^2}{dt^2} + \ldots + b_n \frac{d^n}{dt^n} + b_{n+1} \frac{d^{n+1}}{dt^{n+1}} \right) k_n(t) \\
&= b_0 k(t) + b_1 \frac{dk_n(t)}{dt} + b_2 \frac{d^2 k_n(t)}{dt^2} + \ldots \\
&\qquad + b_n \frac{d^n k_n(t)}{dt^n} + b_{n+1} \frac{d^{n+1} k_n(t)}{dt^{n+1}} \\
&= 0,
\end{aligned} \tag{2.3.18}$$

where the linear differential operator $\mathcal{L}\{\cdot\}$ is defined as:

$$\mathcal{L}\{\cdot\} \equiv \left(b_0 + b_1 \frac{d}{dt} + b_2 \frac{d^2}{dt^2} + \ldots + b_n \frac{d^n}{dt^n} + b_{n+1} \frac{d^{n+1}}{dt^{n+1}} \right). \tag{2.3.19}$$

Thus, Equation (2.3.18) can be rewritten in a simplified manner as:

$$\mathcal{L}\{k_n(t)\} = 0. \tag{2.3.20}$$

Imposition of this condition reduces Equation (2.3.17) to:

$$\begin{aligned}
\sum_{i=0}^{n+1} b_i \frac{d^i y_n(t)}{dt^i} &= \left[\sum_{i=1}^{n+1} b_i \frac{d^{i-1} \mu(t)}{dt^{i-1}} \right] k(0) \\
&\quad + \left[\sum_{i=2}^{n+1} b_i \frac{d^{i-2} \mu(t)}{dt^{i-2}} \right] \frac{dk_n(0)}{dt} \\
&\quad + \left[\sum_{i=3}^{n+1} b_i \frac{d^{i-3} \mu(t)}{dt^{i-3}} \right] \frac{d^2 k_n(0)}{dt^2} \\
&\quad + \ldots \\
&\quad + \left[\sum_{i=n}^{n+1} b_i \frac{d^{i-n} \mu(t)}{dt^{i-n}} \right] \frac{d^2 k_n(0)}{dt^2} \\
&\quad + b_{n+1} \mu(t) \frac{d^n k_n(0)}{dt^n}.
\end{aligned} \tag{2.3.21}$$

In this way, the integral involving $k_n(t)$ is replaced by a differential equation in terms of $y_n(t)$. Keeping this overall scheme in mind, a linear combination of

Equations (2.3.9) leads to a differential equation in $y_n(t)$ whose homogeneous form is exactly the same as that satisfied by $k_n(t)$:

$$\begin{aligned}\mathcal{L}\{y_n(t)\} &\equiv \left(b_0 + b_1\frac{d}{dt} + b_2\frac{d^2}{dt^2} + \ldots + b_n\frac{d^n}{dt^n} + b_{n+1}\frac{d^{n+1}}{dt^{n+1}}\right) y_n(t) \\ &= b_0 y_n(t) + b_1\frac{dy_n(t)}{dt} + b_2\frac{d^2y_n(t)}{dt^2} + \ldots + b_n\frac{d^n y_n(t)}{dt^n} + b_{n+1}\frac{d^{n+1}y_n(t)}{dt^{n+1}} \\ &= 0.\end{aligned} \tag{2.3.22}$$

Because the homogeneous differential equation involving $y_n(t)$ satisfies exactly the same equation as that which describes $k_n(t)$, $k_n(t)$ is basically the solution of $y_n(t)$ in the absence of forcing functions on the right-hand side. Thus, the output from a nth-order system, $y_n(t)$, is described by the nth-order kernel function, $k_n(t)$, and the specific solution. The homogeneous solution to the system dynamic equation, $k_n(t)$, is an intrinsic property of the system under study, and the specific solution depends on the nature of the forcing input on the right-hand side of Equation (2.3.11). The final solution to the system dynamic equation is the linear combination of the homogeneous solution and the specific solution.

From well known theories of calculus, the general solution to the constant coefficient homogeneous linear equation of (2.3.18) is:

$$k_n(t) = \alpha e^{rt}, \tag{2.3.23}$$

where α is a constant determined from the initial conditions. Substituting this into Equation (2.3.18) yields:

$$\mathcal{L}\{e^{rt}\} = \left(b_0 + b_1 r + b_2 r^2 + \ldots + b_n r^n + b_{n+1} r^{n+1}\right) e^{rt}. \tag{2.3.24}$$

In order for Equation (2.3.24) to be zero, the terms preceeding e^{rt} must be set to zero, because e^{rt} itself is never zero if the solution is not a trivial one. Thus, we

obtain the characteristic (or auxiliary) equation associated with both Equations (2.3.18) for $k_n(t)$ and (2.3.22) for $y_n(t)$:

$$b_0 + b_1 r + b_2 r^2 + \ldots + b_n r^n + b_{n+1} r^{n+1} = 0. \qquad (2.3.25)$$

The above algebraic equation can be solved to give the kernel base functions. If there exist $n+1$ distinct roots, then the $n+1$ base kernel functions will be simple exponential functions, each with different time constants. When the roots are repeated, the fundamental solutions to Equation (2.3.18) corresponding to these repeated roots r_0 are $e^{r_0 t}$ and $te^{r_0 t}$. A very special case exists where the roots to the above equation are repeated $n+1$ times.

$$b_0 + b_1 r + b_2 r^2 + \ldots + b_n r^n + b_{n+1} r^{n+1} = (r - r_0)^{n+1} = 0 \qquad (2.3.26)$$

In this case, the solution to Equation (2.3.18) are $e^{r_0 t}$, $te^{r_0 t}$, $t^2 e^{r_0 t}$, $t^3 e^{r_0 t}$, ..., $t^{n-1} e^{r_0 t}$, and $t^n e^{r_0 t}$. In our notation, we have set $r_0 = \frac{1}{T}$, and there is only one time constant because there is only one root. Other cases where roots are distinct are not considered in detail here.

With the integral in Equation (2.3.17) now gone, if one wants to really simplify that equation, it is only logical that $k_n(t)$ should have the property that all n initial values of $\frac{d^i k_n(0)}{dt^i}$ being 0 except for one single $\frac{d^i k_n(0)}{dt^i}$ which should be allowed a finite value to prevent the trivial solution from taking over. An inspection of Equation (2.3.17) indicates that the resulting differential equation in $y_n(t)$ is the simplest when $\frac{d^n k_n(0)}{dt^n}$ is finite, because that term appears only once in Equation (2.3.17). This corresponds precisely to a condition in which the roots to the characteristic equation (2.3.25) are repeated $n+1$ times. Consequently, in order for Equation (2.3.26) to be true, the coefficients in the characteristic equation are simply binomial coefficients, *i.e.*, $b_i = \binom{n+1}{i}$. Thus, after considering the time constant, it follows that each of the Equations (2.3.9) should be multiplied by the respective binomial

coefficient $T^i\binom{n+1}{i}$, and we have effectively forced $k_n(t)$ to satisfy the constant coefficient linear differential equation (2.3.18).

A similar procedure as the one just outlined above may be used to obtain the appropriate form for the base functions in other circumstances. By now, it should be clear as to how the constant coefficients preceeding the derivatives of $k_n(t)$ in Equation (2.3.21) are determined. Different choices of $k(t)$ may leave a few nonzero derivatives of $k(t)$ at $t = 0$. This, in turn, generally will not yield binomial coefficients as the constant coefficients preceeding the derivatives of $k_n(t)$ in Equation (2.3.21).

In general, the resulting equivalent differential equation is one order higher than the kernel originally contained inside the integral in Equation (2.3.9.0), including Equation (2.3.12), which is the result of a special case of repeated roots for the characteristic equation. For example, for the 0th-order kernel $k_0(t)$, the equivalent differential equation is:

$$T\frac{dy_0(t)}{dt} + y_0(t) = \mu(t). \tag{2.3.27.0}$$

Similarly, for $n = 1$, $n = 2$, ..., $n - 1$, and n, the integrals involving $k_1(t)$, $k_2(t)$, ..., $k_{n-1}(t)$, and $k_n(t)$ are transformed respectively to:

$$T^2\frac{d^2y_1(t)}{dt^2} + 2T\frac{dy_1(t)}{dt} + y_1(t) = \mu(t) \tag{2.3.27.1}$$

$$T^3\frac{d^3y_2(t)}{dt^3} + 3T^2\frac{d^2y_2(t)}{dt^2} + 3T\frac{dy_2(t)}{dt} + y_2(t) = \mu(t) \tag{2.3.27.2}$$

$$\vdots \tag{2.3.27}$$

$$T^n\frac{d^ny_{n-1}(t)}{dt^n} + nT^{n-1}\frac{d^{n-1}y_{n-1}(t)}{dt^{n-1}} + \ldots + nT\frac{dy_{n-1}(t)}{dt} + y_{n-1}(t) = \mu(t) \tag{2.3.27.n-1}$$

$$T^{n+1}\frac{d^{n+1}y_n(t)}{dt^{n+1}} + (n+1)T^n\frac{d^ny_n(t)}{dt^n} + \ldots + (n+1)T\frac{dy_n(t)}{dt} + y_n(t) = \mu(t). \tag{2.3.27.n}$$

Notice the similarity between these equations, (*e.g.*, Equations (2.3.27.0), (2.3.27.1), and (2.3.12)), and those equations that are satisfied by the corresponding kernel functions, (*e.g.*, Equations (2.3.4), (2.3.6), and (2.3.1)). A higher-order differential equation such as Equation (2.3.27.0) can be easily transformed into a set of first-order differential equations through some well known canonical transformations. Thus, for a 0th-order kernel, the set of integral state equations of (2.2.12) and (2.2.13) is reduced to the following:

$$\frac{dx}{dt} = (y-1)x \tag{2.3.28}$$

$$\frac{ds}{dt} = 1 - s - \frac{1}{Y_s}yx \tag{2.3.29}$$

$$\frac{dy}{dt} = \frac{1}{T}(-y+\mu). \tag{2.3.30}$$

As shown in later sections, the biological significance of the 0th-order kernel can be extracted from the above equivalent set of equations. One of the possible interpretations of the above set of equations is that the rate of reproduction of the biomass is autocatalytically proportional to the biomass itself and to a new variable y. If one so wishes, this variable y can be interpreted as the concentration or level of some "critical enzyme" that limits the growth of the microorganism. Furthermore, Equation (2.3.30) indicates that this "critical enzyme" follows a first-order deactivation kinetics, and its rate of formation is described by the function $\mu(s)$. If μ has a Monod form, then the production of the enzyme follows that of Michaelis-Menton kinetics, which, incidentally, is originally derived to describe enzyme kinetics. Thus, time-lag formulation can partially explain the relationship between the enzyme kinetics and microbial specific growth rate in terms of the equivalence of the integral form and the differential form.

Similarly, for a simple nth-order kernel $k_n(t)$, the integral is transformed to the following set of equations:

$$\begin{aligned}
\frac{dy}{dt} &= z_1 \\
\frac{dz_1}{dt} &= z_2 \\
&\vdots \\
\frac{dz_{n-1}}{dt} &= z_n \\
\frac{dz_n}{dt} &= \frac{1}{T^{n+1}}\left[-\sum_{i=0}^{n}\binom{n+1}{i}T^i z_i + \mu(t)\right].
\end{aligned} \tag{2.3.31}$$

Linear Combination of Base Functions

This approach would not have been very useful if one is restricted only to the individual exponential distribution functions. For more complicated dynamics, one would expect to employ a set of these base functions in order to describe the behavior of the system. Because of the linear properties of the differential and integration operators, a linear combination of more than one base function of $k_n(t)$ will leave the approach unchanged. For example, if $k(t)$ is expressed as a linear combination of n exponential distribution functions:

$$k(t) = a_0k_0(t) + a_1k_1(t) + a_2k_2(t) + \ldots + a_{n-1}k_{n-1}(t) + a_nk_n(t), \tag{2.3.32}$$

the general observed specific growth rate $y(t) \equiv \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh$ also has the same linear combination of the individual observed specific growth rates $y_i(t)$s, each of which describes the dynamic behavior of a purely ith-order subsystem governed by $k_i(t)$:

$$y_i(t) \equiv \int_{-\infty}^{t} \mu[s(h)]k_i(t-h)dh \qquad i = 1, 2, ..., n. \tag{2.3.33}$$

The fact that $k(t)$ and $y(t)$ both comply with the same linear combination can be proved by the following steps:

$$\begin{aligned}
y(t) &= \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh \\
&= \int_{-\infty}^{t} \mu[s(h)]\cdot \\
&\quad \cdot \underbrace{\Big[a_0k_0(t-h) + a_1k_1(t-h) + \ldots + a_{n+1}k_{n+1}(t-h) + a_nk_n(t-h)\Big]}_{k(t-h)} dh \\
&= a_0 \underbrace{\int_{-\infty}^{t} \mu[s(h)]k_0(t-h)dh}_{y_0(t)} + a_1 \underbrace{\int_{-\infty}^{t} \mu[s(h)]k_1(t-h)dh}_{y_1(t)} + \ldots \\
&\quad + a_{n-1} \underbrace{\int_{-\infty}^{t} \mu[s(h)]k_{n-1}(t-h)dh}_{y_{n-1}(t)} + a_n \underbrace{\int_{-\infty}^{t} \mu[s(h)]k_n(t-h)dh}_{y_n(t)} \\
&= a_0y_0(t) + a_1y_1(t) + \ldots + a_{n+1}y_{n-1}(t) + a_ny_n(t).
\end{aligned} \tag{2.3.34}$$

The remainder of this section is devoted to the derivation of the differential equation satisfied by $y(t)$ for a system governed by a general kernel function that can be described by Equation (2.3.32). Again, for simplicity, it is our objective to reach an equation that describes a general $y(t)$ in a form similar to Equation (2.3.27.n), which describes $y_n(t)$, where n is the order of $y(t)$. This is accomplished by repeatedly differentiating Equations (2.3.27.0)-(2.3.27.n) that describe the relationships satisfied by each individual output component and by adding the resulting equations in order to obtain an equation similar in form to Equation (2.3.27.n). The form similar to Equation (2.3.27) is desired because it has the same order as the general $y(t)$.

First, Equation (2.3.27.0) is differentiated and then multiplied by appropriate constants.

$$\binom{n}{0} \times : \; T\frac{dy_0(t)}{dt} + y_0(t) = \mu(t)$$

$$\binom{n}{1}T\frac{d}{dt}\times:\ nT^2\frac{d^2y_0(t)}{dt^2}+nT\frac{dy_0(t)}{dt}=nT\frac{d\mu(t)}{dt}$$

$$\binom{n}{2}T^2\frac{d^2}{dt^2}\times:\ \frac{n(n-1)}{2}T^3\frac{d^3y_0(t)}{dt^3}+\frac{n(n-1)}{2}T^2\frac{d^2y_0(t)}{dt^2}=\frac{n(n-1)}{2}T^2\frac{d^2\mu(t)}{dt^2}$$

$$\vdots$$

$$\binom{n}{i-1}T^{i-1}\frac{d^{i-1}}{dt^{i-1}}\times:\ \binom{n}{i-1}T^i\frac{d^iy_0(t)}{dt^i}+\binom{n}{i-1}T^{i-1}\frac{d^{i-1}y_0(t)}{dt^{i-1}}=\binom{n}{i-1}T^{i-1}\frac{d^{i-1}\mu(t)}{dt^{i-1}}$$

$$\binom{n}{i}T^i\frac{d^i}{dt^i}\times:\ \binom{n}{i}T^{i+1}\frac{d^{i+1}y_0(t)}{dt^{i+1}}+\binom{n}{i}T^i\frac{d^iy_0(t)}{dt^i}=\binom{n}{i}T^i\frac{d^i\mu(t)}{dt^i}$$

$$\vdots$$

$$\binom{n}{n-2}T^{n-2}\frac{d^{n-2}}{dt^{n-2}}\times:\ \binom{n}{n-2}T^{n-1}\frac{d^{n-1}y_0(t)}{dt^{n-1}}+\binom{n}{n-2}T^{n-2}\frac{d^{n-2}y_0(t)}{dt^{n-2}}=\binom{n}{n-2}T^{n-2}\frac{d^{n-2}\mu(t)}{dt^{n-2}}$$

$$\binom{n}{n-1}T^{n-1}\frac{d^{n-1}}{dt^{n-1}}\times:\ nT^n\frac{d^ny_0(t)}{dt^n}+nT^{n-1}\frac{d^{n-1}y_0(t)}{dt^{n-1}}=nT^{n-1}\frac{d^{n-1}\mu(t)}{dt^{n-1}}$$

$$\binom{n}{n}T^n\frac{d^n}{dt^n}\times:\ T^{n+1}\frac{d^{n+1}y_0(t)}{dt^{n+1}}+T^n\frac{d^ny_0(t)}{dt^n}=T^n\frac{d^n\mu(t)}{dt^n}\qquad(2.3.35)$$

The factors applied to Equation (2.3.27.0) to obtain the preceeding equations are identified on the left hand side of each equation. Adding these questions together, one obtains:

$$\begin{aligned}
&T^{n+1}\frac{d^{n+1}y_0(t)}{dt^{n+1}}+(n+1)T^n\frac{d^ny_0(t)}{dt^n}+\frac{(n+1)n}{2}T^{n-1}\frac{d^{n-1}y_0(t)}{dt^{n-1}}\\
&\quad+\ldots+\binom{n+1}{i}T^i\frac{d^iy_0(t)}{dt^i}+\ldots\\
&\quad+\frac{(n+1)n}{2}T^2\frac{d^2y_0(t)}{dt^2}+(n+1)T\frac{dy_0(t)}{dt}+y_0(t)\\
&\qquad=T^n\frac{d^n\mu(t)}{dt^n}+nT^{n-1}\frac{d^{n-1}\mu(t)}{dt^{n-1}}+\frac{n(n-1)}{2}T^{n-2}\frac{d^{n-2}\mu(t)}{dt^{n-2}}\\
&\qquad\quad+\ldots+\binom{n}{i}T^i\frac{d^i\mu(t)}{dt^i}+\ldots\\
&\qquad\quad+\frac{n(n-1)}{2}T^2\frac{d^2\mu(t)}{dt^2}+nT\frac{d\mu(t)}{dt}+\mu(t),
\end{aligned}\qquad(2.3.36)$$

which can be further expressed in a simple summation notation:

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_0(t)}{dt^i} = \sum_{i=0}^{n}\binom{n}{i}T^i\frac{d^i\mu(t)}{dt^i}. \tag{2.3.37}$$

In deriving this general expression, the special property of adding two neighboring binomial coefficients has been used.

$$\binom{n}{i-1}+\binom{n}{i}=\binom{n+1}{i} \tag{2.3.38}$$

Note that the left-hand side of the resulting Equation (2.3.37) has exactly the same format as that of Equation (2.3.27).

Similarly, Equation (2.3.27.1), when multiplied by the left hand quantities yields:

$$\begin{aligned}
\binom{n-1}{0}\times: \quad & T^2\frac{d^2y_1(t)}{dt^2}+2T\frac{dy_1(t)}{dt}+y_1(t)=\mu(t)\\
\binom{n-1}{1}T\frac{d}{dt}\times: \quad & (n-1)T^3\frac{d^3y_1(t)}{dt^3}+2(n-1)T^2\frac{d^2y_1(t)}{dt^2}\\
& +(n-1)T\frac{dy_1(t)}{dt}=(n-1)T\frac{d\mu(t)}{dt}\\
\binom{n-1}{2}T^2\frac{d^2}{dt^2}\times: \quad & \frac{(n-1)(n-2)}{2}T^4\frac{d^4y_1(t)}{dt^4}+(n-1)(n-2)T^3\frac{d^3y_1(t)}{dt^3}\\
& +\frac{(n-1)(n-2)}{2}T^2\frac{d^2y_1(t)}{dt^2}=\frac{(n-1)(n-2)}{2}T^2\frac{d^2\mu(t)}{dt^2}\\
& \vdots \qquad\qquad (2.3.39)\\
\binom{n-1}{i-1}T^{i-1}\frac{d^{i-1}}{dt^{i-1}}\times: \quad & \binom{n-1}{i-2}T^i\frac{d^iy_1(t)}{dt^i}+2\binom{n-1}{i-2}T^{i-1}\frac{d^{i-1}y_1(t)}{dt^{i-1}}\\
& +\binom{n-1}{i-2}T^{i-2}\frac{d^{i-2}y_1(t)}{dt^{i-2}}=\binom{n-1}{i-2}T^{i-2}\frac{d^{i-2}\mu(t)}{dt^{i-2}}\\
\binom{n-1}{i}T^i\frac{d^i}{dt^i}\times: \quad & \binom{n-1}{i-1}T^{i+1}\frac{d^{i+1}y_1(t)}{dt^{i+1}}+2\binom{n-1}{i-1}T^i\frac{d^iy_1(t)}{dt^i}\\
& +\binom{n-1}{i-1}T^{i-1}\frac{d^{i-1}y_1(t)}{dt^{i-1}}=\binom{n-1}{i-1}T^{i-1}\frac{d^{i-1}\mu(t)}{dt^{i-1}}\\
\binom{n-1}{i+1}T^{i+1}\frac{d^{i+1}}{dt^{i+1}}\times: \quad & \binom{n-1}{i}T^{i+2}\frac{d^{i+2}y_1(t)}{dt^{i+2}}+2\binom{n-1}{i}T^{i+1}\frac{d^{i+1}y_1(t)}{dt^{i+1}}\\
& +\binom{n-1}{i}T^i\frac{d^iy_1(t)}{dt^i}=\binom{n-1}{i}T^i\frac{d^i\mu(t)}{dt^i}
\end{aligned}$$

$$\vdots \qquad (2.3.39)$$

$$\binom{n-1}{n-2}T^{n-2}\frac{d^{n-3}}{dt^{n-3}}\times: \quad \binom{n-1}{n-3}T^{n-1}\frac{d^{n-1}y_1(t)}{dt^{n-1}}+\binom{n-1}{n-3}T^{n-2}\frac{d^{n-2}y_1(t)}{dt^{n-2}}$$
$$+\binom{n-1}{n-2}T^{n-3}\frac{d^{n-3}y_1(t)}{dt^{n-3}}=\binom{n-1}{n-2}T^{n-3}\frac{d^{n-3}\mu(t)}{dt^{n-3}}$$

$$\binom{n-1}{n-1}T^{n-1}\frac{d^{n-2}}{dt^{n-2}}\times: \quad (n-1)T^n\frac{d^n y_1(t)}{dt^n}+2(n-1)T^{n-1}\frac{d^{n-1}y_1(t)}{dt^{n-1}}$$
$$+(n-1)T^{n-2}\frac{d^{n-2}y_1(t)}{dt^{n-2}}=(n-1)T^{n-2}\frac{d^{n-2}\mu(t)}{dt^{n-2}}$$

$$\binom{n-1}{n}T^n\frac{d^{n-1}}{dt^{n-1}}\times: \quad T^{n+1}\frac{d^{n+1}y_1(t)}{dt^{n+1}}+2T^n\frac{d^n y_1(t)}{dt^n}$$
$$+T^{n-1}\frac{d^{n-1}y_1(t)}{dt^{n-1}}=T^{n-1}\frac{d^{n-1}\mu(t)}{dt^{n-1}}$$

In the same way as for $y_0(t)$, adding the above equations of $y_1(t)$ together, one obtains:

$$\begin{aligned}
&T^{n+1}\frac{d^{n+1}y_0(t)}{dt^{n+1}}+(n+1)T^n\frac{d^n y_0(t)}{dt^n}+\frac{(n+1)n}{2}T^{n-1}\frac{d^{n-1}y_0(t)}{dt^{n-1}}\\
&\quad+\ldots+\binom{n+1}{i}T^i\frac{d^i y_0(t)}{dt^i}+\ldots\\
&\qquad+\frac{(n+1)n}{2}T^2\frac{d^2y_0(t)}{dt^2}+(n+1)T\frac{dy_0(t)}{dt}+y_0(t)\\
&\qquad=T^{n-1}\frac{d^{n-1}\mu(t)}{dt^{n-1}}+(n-1)T^{n-2}\frac{d^{n-2}\mu(t)}{dt^{n-2}}+\frac{(n-1)(n-2)}{2}T^{n-3}\frac{d^{n-3}\mu(t)}{dt^{n-3}}\\
&\qquad\quad+\ldots+\binom{n}{i}T^i\frac{d^i\mu(t)}{dt^i}+\ldots\\
&\qquad\qquad+\frac{(n-1)(n-2)}{2}T^2\frac{d^2\mu(t)}{dt^2}+(n-1)T\frac{d\mu(t)}{dt}+\mu(t),
\end{aligned} \qquad (2.3.40)$$

which can be rewritten compactly as:

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_1(t)}{dt^i}=\sum_{i=0}^{n-1}\binom{n-1}{i}T^i\frac{d^i\mu(t)}{dt^i}. \qquad (2.3.41)$$

Similar to Equation (2.3.38), the following special property of adding three neighboring binomial coefficients has been used in deriving the above compact equation.

$$\begin{aligned}\binom{n-1}{i-2}+2\binom{n-1}{i-1}+\binom{n-1}{i} &= \left[\binom{n-1}{i-2}+\binom{n-1}{i-1}\right]+\left[\binom{n-1}{i-1}+\binom{n-1}{i}\right] \\ &= \binom{n}{i-1}+\binom{n}{i} \\ &= \binom{n+1}{i}\end{aligned} \tag{2.3.42}$$

Continuing the same process further for $y_2(t)$, $y_3(t)$, ..., $y_j(t)$, ..., $y_{n-2}(t)$, $y_{n-1}(t)$, and finally for $y_n(t)$, one arrives at:

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_2(t)}{dt^i} = \sum_{i=0}^{n-2}\binom{n-2}{i}T^i\frac{d^i\mu(t)}{dt^i} \tag{2.3.41.2}$$

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_3(t)}{dt^i} = \sum_{i=0}^{n-3}\binom{n-3}{i}T^i\frac{d^i\mu(t)}{dt^i} \tag{2.3.41.3}$$

$$\vdots \tag{2.3.41}$$

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_j(t)}{dt^i} = \sum_{i=0}^{n-j}\binom{n-j}{i}T^i\frac{d^i\mu(t)}{dt^i} \tag{2.3.41.j}$$

$$\vdots \tag{2.3.41}$$

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_{n-2}(t)}{dt^i} = \sum_{i=0}^{2}\binom{2}{i}T^i\frac{d^i\mu(t)}{dt^i} \tag{2.3.41.n-2}$$

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_{n-1}(t)}{dt^i} = \sum_{i=0}^{1}\binom{1}{i}T^i\frac{d^i\mu(t)}{dt^i} \tag{2.3.41.n-1}$$

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^i y_n(t)}{dt^i} = \sum_{i=0}^{0}\binom{0}{i}T^i\frac{d^i\mu(t)}{dt^i} \tag{2.3.41.n}$$

The right-hand quantities in the last three equations are simply $\mu(t)+2T\frac{d\mu(t)}{dt}+T^2\frac{d^2\mu(t)}{dt^2}$, $\mu(t)+T\frac{d\mu(t)}{dt}$, and $\mu(t)$, respectively. Adding all the above equations

together once more, with each equation being weighed by a_j, gives:

$$\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\left[\sum_{j=0}^{n}a_j\frac{d^iy_j(t)}{dt^i}\right] = \left[\sum_{j=0}^{n}\binom{j}{0}a_{n-j}\right]\mu(t) + \left[\sum_{j=1}^{n}\binom{j}{1}a_{n-j}\right]T\frac{d\mu(t)}{dt}$$
$$+\left[\sum_{j=2}^{n}\binom{j}{2}a_{n-j}\right]T^2\frac{d^2\mu(t)}{dt^2}+\dots$$
$$+\left[\sum_{j=i}^{n}\binom{j}{i}a_{n-j}\right]T^i\frac{d^i\mu(t)}{dt^i}+\dots$$
$$+\left[\sum_{j=n-2}^{n}\binom{j}{n-2}a_{n-j}\right]T^{n-2}\frac{d^{n-2}\mu(t)}{dt^{n-2}}$$
$$+\left[\sum_{j=n-1}^{n}\binom{j}{n-1}a_{n-j}\right]T^{n-1}\frac{d^{n-1}\mu(t)}{dt^{n-1}}$$
$$+\left[\sum_{j=n}^{n}\binom{j}{n}a_{n-j}\right]T^n\frac{d^n\mu(t)}{dt^n}. \qquad (2.3.43)$$

Finally, with the following relationship

$$\sum_{j=0}^{n}a_j\frac{d^iy_j(t)}{dt^i} = \frac{d^i}{dt^i}\left[\sum_{j=0}^{n}a_jy_j(t)\right] = \frac{d^i}{dt^i}y(t), \qquad (2.3.44)$$

Equation (2.3.43) is ultimately reduced to:

$$\begin{aligned}\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^iy(t)}{dt^i} &= \left[\sum_{j=0}^{n}\binom{j}{0}a_{n-j}\right]\mu(t)+\left[\sum_{j=1}^{n}\binom{j}{1}a_{n-j}\right]T\frac{d\mu(t)}{dt}\\ &+\left[\sum_{j=2}^{n}\binom{j}{2}a_{n-j}\right]T^2\frac{d^2\mu(t)}{dt^2}+\cdots\\ &+\left[\sum_{j=i}^{n}\binom{j}{i}a_{n-j}\right]T^i\frac{d^i\mu(t)}{dt^i}+\cdots\\ &+\left[\frac{n(n-1)}{2}a_0+(n-1)a_1+a_2\right]T^{n-2}\frac{d^{n-2}\mu(t)}{dt^{n-2}}\\ &+\left[na_0+a_1\right]T^{n-1}\frac{d^{n-1}\mu(t)}{dt^{n-1}}\\ &++a_0T^n\frac{d^n\mu(t)}{dt^n}\\ &=\sum_{i=0}^{n}\underbrace{\left[\sum_{j=i}^{n}\binom{j}{i}a_{n-j}\right]}_{\alpha_i}T^i\frac{d^i\mu(t)}{dt^i}\end{aligned} \tag{2.3.45a}$$

Or, more compactly,

$$\boxed{\sum_{i=0}^{n+1}\binom{n+1}{i}T^i\frac{d^iy(t)}{dt^i}=\sum_{i=0}^{n}\alpha_iT^i\frac{d^i\mu(t)}{dt^i};\qquad \text{where } \alpha_i=\sum_{j=i}^{n}\binom{j}{i}a_{n-j},} \tag{2.3.45b}$$

which is the final general result sought as a counterpart of Equation (2.3.12) when $k(t)$ is a linear combination of base functions described by Equation (2.3.32).

General First-Order Kernel: Specifically, for a general first-order kernel that has the form:

$$\begin{aligned}k(t) &= a_0k_0(t)+a_1k_1(t)\\ &=\left(a_0\frac{1}{T}+a_1\frac{t}{T^2}\right)e^{-\frac{t}{T}}\qquad (a_0+a_1=1),\end{aligned} \tag{2.3.46}$$

Linear manipulation according to the differentiation and multiplication operations indicated at the left-hand side of the following equations yields:

$$a_0 \times (2.3.27.0): \quad a_0\Big[\quad T\frac{dy_0(t)}{dt} + y_0(t) = \mu(t) \quad\Big] \tag{2.3.47.1}$$

$$a_0 T\frac{d}{dt} \times (2.3.27.0): \quad a_0\Big[T^2\frac{d^2y_0(t)}{dt^2} + T\frac{dy_0(t)}{dt} \quad = T\frac{d\mu(t)}{dt}\Big] \tag{2.3.47.2}$$

$$a_1 \times (2.3.27.1): \quad a_1\Big[T^2\frac{d^2y_1(t)}{dt^2} + 2T\frac{dy_1(t)}{dt} + y_1(t) = \mu(t) \quad\Big] \tag{2.3.47.3}$$

Summation of the above three equations yields:

$$T^2\left[a_0\frac{d^2y_0(t)}{dt^2} + a_1\frac{d^2y_1(t)}{dt^2}\right] + 2T\left[a_0\frac{dy_0(t)}{dt} + a_1\frac{dy_1(t)}{dt}\right] + [a_0y_0(t) + a_1y_1(t)] = [a_0 + a_1]\mu(t) + a_0T\frac{d\mu(t)}{dt}. \tag{2.3.48}$$

Consequently, substituting $y(t) = a_0y_0(t) + a_1y_1(t)$, $\frac{dy(t)}{dt} = a_0\frac{dy_0(t)}{dt} + a_1\frac{dy_1(t)}{dt}$, and $\frac{d^2y(t)}{dt^2} = a_0\frac{d^2y_0(t)}{dt^2} + a_1\frac{d^2y_1(t)}{dt^2}$ and utilizing the unity condition on the coefficients, *i.e.*, $a_0 + a_1 = 1$, one can convert the integral $y(t) = \int_{-\infty}^{t}\mu[s(h)]k(t-h)dh$ to a second-order ordinary differential equation:

$$\boxed{T^2\frac{d^2y(t)}{dt^2} + 2T\frac{dy(t)}{dt} + y(t) = \mu(t) + a_0T\frac{d\mu(t)}{dt}}, \tag{2.3.49}$$

which can be further transformed to a mathematically equivalent set of two first-order ordinary differential equations.

$$\frac{dy(t)}{dt} = z \tag{2.3.50}$$

$$\frac{dz(t)}{dt} = -\frac{2}{T}z - \frac{1}{T^2}y + \frac{1}{T^2}\mu(t) + a_0\frac{1}{T}\frac{d\mu(t)}{dt}. \tag{2.3.51}$$

For example, with the kernel of Equation (2.3.46), the system dynamic equations (2.2.12) and (2.2.13) are now:

$$\frac{dx}{dt} = (y-1)x \tag{2.3.52}$$

$$\frac{ds}{dt} = 1 - s - \frac{1}{Y_s}yx \tag{2.3.53}$$

$$\frac{dy}{dt} = z \tag{2.3.54}$$

$$\begin{aligned}\frac{dz}{dt} &= -\frac{2}{T}z - \frac{1}{T^2}y + \frac{1}{T^2}\mu[s(t)] + a_0\frac{1}{T}\frac{d\mu}{ds}\frac{ds}{dt} \\ &= -\frac{2}{T}z - \frac{1}{T^2}y + \frac{1}{T^2}\mu(s) \\ &\quad + a_0\frac{1}{T}\frac{d\mu(s)}{ds} - a_0\frac{1}{T}\frac{d\mu(s)}{ds}s - a_0\frac{1}{T}\frac{1}{Y_s}\frac{d\mu(s)}{ds}yx.\end{aligned} \tag{2.3.55}$$

General Second-Order Kernel: The same procedure can be carried to a general second-order kernel function of the following form:

$$\begin{aligned}k(t) &= a_0k_0(t) + a_1k_1(t) + a_2k_2(t) \\ &= \left(a_0\frac{1}{T} + a_1\frac{t}{T^2} + a_2\frac{t^2}{T^3}\right)e^{-\frac{t}{T}} \qquad (a_0 + a_1 + a_2 = 1).\end{aligned} \tag{2.3.56}$$

The equations to be summed are indicated below.

$$\begin{aligned}
&a_0 \times (2.3.27.0): && a_0\Big[&& T\frac{dy_0(t)}{dt} && +y_0(t) && =\mu(t) && \Big] \\
&a_0 2T\frac{d}{dt} \times (2.3.27.0): && a_0\Big[&& +2T^2\frac{d^2y_0(t)}{dt^2} && +2T\frac{dy_0(t)}{dt} && =2T\frac{d\mu(t)}{dt} && \Big] \\
&a_0T^2\frac{d^2}{dt^2} \times (2.3.27.0): && a_0\Big[&& T^3\frac{d^3y_0(t)}{dt^3} && +T^2\frac{d^2y_0(t)}{dt^2} && =T^2\frac{d^2\mu(t)}{dt^2} && \Big] \\
&a_1 \times (2.3.27.1): && a_1\Big[&& T^2\frac{d^2y_1(t)}{dt^2} && +2T\frac{dy_1(t)}{dt} && +y_1(t) && =\mu(t) && \Big] \\
&a_1T\frac{d}{dt} \times (2.3.27.1): && a_1\Big[&& T^3\frac{d^3y_1(t)}{dt^3} && +2T^2\frac{d^2y_1(t)}{dt^2} && +T\frac{dy_1(t)}{dt} && =T\frac{d\mu(t)}{dt} && \Big] \\
&a_2 \times (2.3.27.2): && a_2\Big[&& T^3\frac{d^3y_2(t)}{dt^3} && +3T^2\frac{d^2y_2(t)}{dt^2} && +3T\frac{dy_2(t)}{dt} && +y_2(t) && =\mu(t) && \Big]
\end{aligned} \tag{2.3.57}$$

Summing up these six equations yields:

$$\boxed{T^3\frac{d^3y(t)}{dt^3} + 3T^2\frac{d^2y(t)}{dt^2} + 3T\frac{dy(t)}{dt} + y(t) = \mu(t) + (2a_0 + a_1)T\frac{d\mu(t)}{dt} + a_0T^2\frac{d^2\mu(t)}{dt^2}}. \tag{2.3.58}$$

Since, as mentioned previously, a first-order kernel is usually sufficient in describing bioreactor dynamics, the dependence of the specific growth rate (and other similar culture parameters) on the past history of the culture can thus be described

with only two additional differential equations. This modest increase in the dimension of the system is a small price to pay considering the significantly enhanced predictive capabilities of the model.

2.4 KERNALS AND KALMAN FILTERS

Filter versus Kernel: In an attempt to invert the convolution integral from simulated system input and output, one sees that the noise in the system input is effectively filtered by the integral operation, and only the slow moving trends pass through the integral. Thus, one can deliberately impose a system dynamic equation on the noise and force it though a dynamic system block, where the noise will be dampened. Various filter designs are based on the concept of adding a dynamic system block between the source of noise and the smoothed signal. This concept is graphically illustrated in Figure 2.4.1, where T is the time constant of the dynamic system block corresponding to the noise filter.

One may visualize the working of such a filter by considering the random injection of dyes in a CSTR, where the dyes are the visualized representation of noise. If the stirred tank is infinitesimally small, *i.e.*, the dynamic system block or the filter having a small time constant, the dye concentration in the tank outlet will fluctuate in about the same way as that in the inlet. On the other hand, if the tank is large, the extra buffering capacity will absorb most fluctuations in the inlet dye concentration, and the outlet dye concentration will only vary gradually and smoothly despite the quick fluctuation in the inlet. The same argument and theories developed for the kernel can also be applied to noise filtering, except that the system input is now substituted with noise, the dynamic system block is viewed as the filter, and the system output is equivalent to the smoothed output. The relationship between kernel and noise filtering in parameter/state estimation will be briefly discussed in this section.

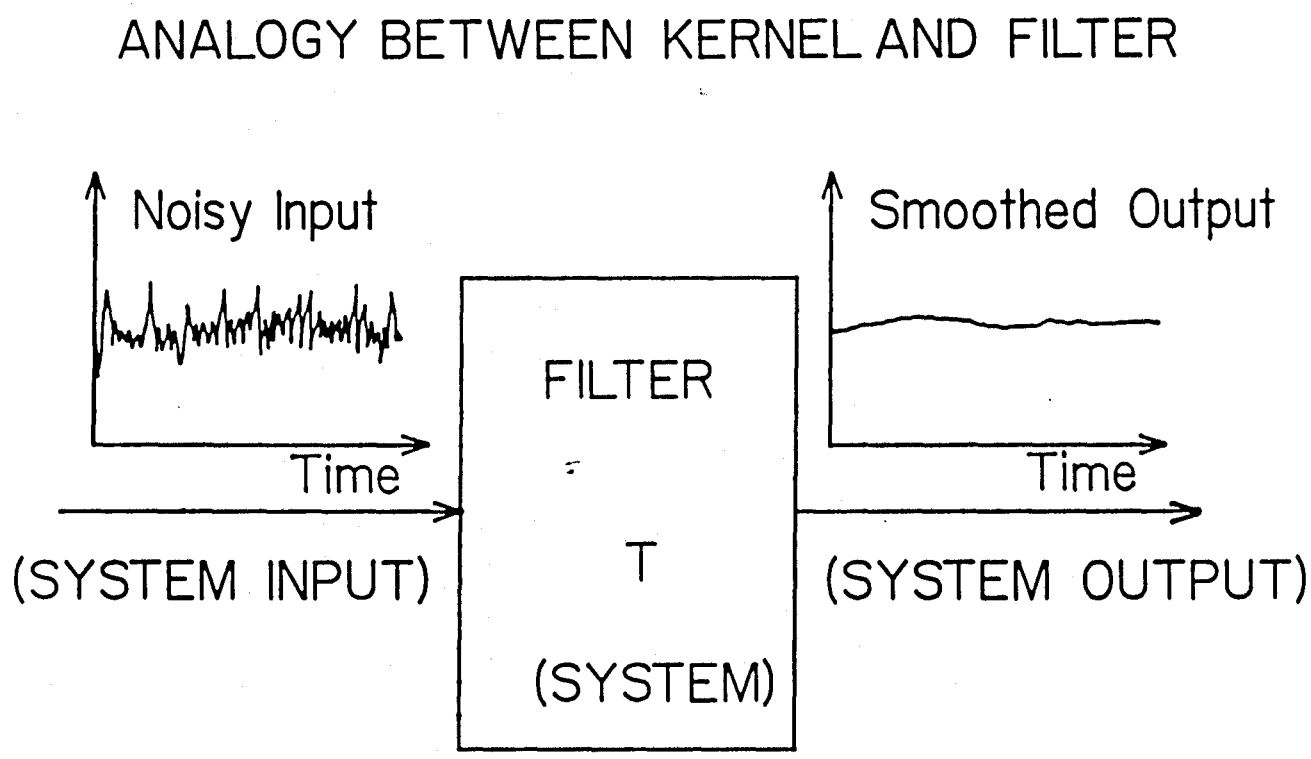


Figure 2.4.1. A kernel that describes the relationship between the system input and output can be effectively used for noise filtering.

As an example, for a set of general nonlinear system dynamic equations:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}) + \boldsymbol{\varsigma}(\mathbf{t}) \tag{2.4.1}$$

and measurement equations:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\xi}(\mathbf{t}), \tag{2.4.2}$$

the following extended Kalman filter has often been used.

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = \mathbf{f}(\hat{\mathbf{x}}) + \mathbf{K}(\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}})), \tag{2.4.3}$$

where $\mathbf{K}$ is the Kalman filter gain defined by:

$$\mathbf{K} = \mathbf{P}\mathbf{h}_{\mathbf{x}}^{\mathbf{T}}(\hat{\mathbf{x}})\mathbf{S}^{-1}. \tag{2.4.4}$$

The corresponding variance dynamic equation is:

$$\frac{d\hat{\mathbf{P}}(t)}{dt} = \mathbf{f}_{\mathbf{x}}(\hat{\mathbf{x}})\mathbf{P} + \mathbf{P}\mathbf{f}_{\mathbf{x}}^{\mathbf{T}}(\hat{\mathbf{x}}) + \mathbf{Q} - \mathbf{P}\mathbf{h}_{\mathbf{x}}^{\mathbf{T}}(\hat{\mathbf{x}})\mathbf{S}^{-1}\mathbf{h}_{\mathbf{x}}(\hat{\mathbf{x}})\mathbf{P}. \tag{2.4.5}$$

The details of this scheme will not be deliberated here, since the Kalman filter is also reviewed in Appendix G.

An adaptive extended Kalman filter has been applied by Stephanopoulos and San (1984) in bioreactor estimation. Their most significant contribution is the addition of the system dynamic equations for states or parameters whose dynamics are not well known. In general, time invariant parameters can be estimated by setting the dynamic equation to zero. The estimates are quite accurate if these parameters are indeed time invariant. However, setting the dynamic equation to zero does not work well if these parameters/states change with time. For example, the specific growth rate of a culture is one such variable that changes during the course of fermentation and yet for which there exists no suitable dynamic equation.

One approach is to assume that it is driven purely by a white noise ς of variable intensity σ^2.

$$\frac{d\mu(t)}{dt} = 0 + \varsigma(t) \tag{2.4.6}$$

Doing so, in accordance with one's lack of knowledge on the dynamic process of $\mu(t)$, avoids wrongly giving any structure to $\mu(t)$.

Another way of looking at this equation, in light of the previous discussion on CSTRs in series, is that the above equation describes a system input/output relationship that is separated by a first-order system block of an infinitely long time constant.

$$\lim_{T\to\infty} \quad \frac{d\mu(t)}{dt} = -\frac{\mu(t)}{T} + \varsigma(t) \quad \Longrightarrow \quad \frac{d\mu(t)}{dt} = \varsigma(t) \tag{2.4.7a}$$

More precise formulation would place the time constant T in the right place.

$$T\frac{d\mu(t)}{dt} + \mu(t) = \varsigma(t) \tag{2.4.8}$$

Graphically, this equation is shown in Figure 2.4.2. Hence, it is not surprising that the adaptive filter is sluggish in response, in view of the long time-constant employed.

To improve the estimation algorithm, Stephanopoulos and San first suggested the use of the following equations:

$$\frac{d\mu(t)}{dt} = C(t) + \eta(t) \tag{2.4.9a}$$

$$\frac{dC(t)}{dt} = \eta'(t) \tag{2.4.9b}$$

The reason behind the use of the above equations is to convert white noise $\eta(t)$ to colored noise $\varsigma(t)$. This is indeed true as shown later, but the dynamic representation of these equations is not quite conventional. This author believes that their intention is really as two dynamic filter blocks in series, as shown in Figure

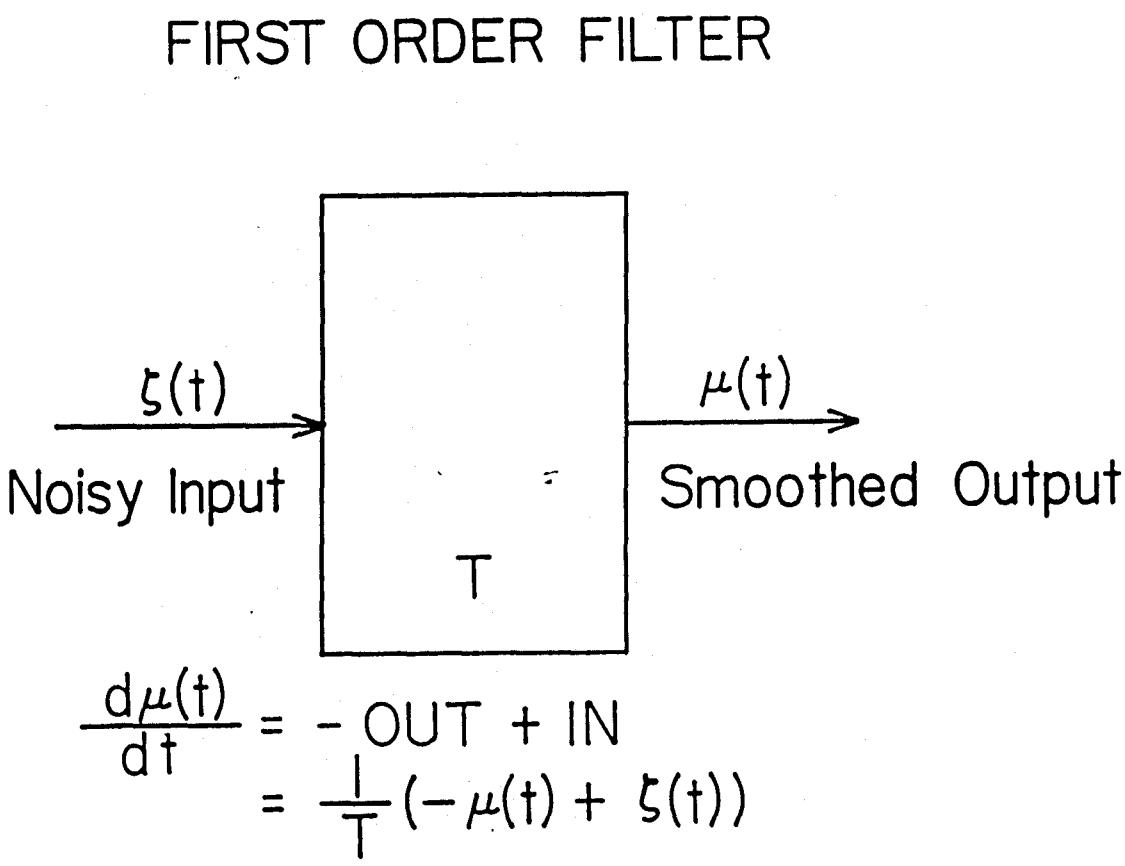


Figure 2.4.2. Noise filtering with a first-order dynamic system block.

2.4.3, where each noise enters into separate dynamic blocks. (An analogous example of how kernels tie the system output to the system input is treated in Section 2.2, where the possibility of a system input entering into more than one first-order dynamic system block is also briefly discussed.) In that case, the filter equations should have been:

$$T_\mu \frac{d\mu(t)}{dt} = -\mu(t) + C(t) + \eta(t) \qquad (2.4.10a)$$

$$T_c \frac{dC(t)}{dt} = -C + \eta'(t) \qquad (2.4.10b)$$

The first of these equations represents the last dynamic block and the second equation represents the first dynamic block on the left.

This slight modification only clarifies the use of these filter equations and gives them some physical meaning, *e.g.*, CSTRs in series; it does not invalidate the original treatment of parameter/state estimation. When viewed as imposing two first-order filters in series and, thus, providing more buffer between the noisy input and the ultimate estimate, it is clear why the estimate should be less noisy as compared to the original treatment where the variable $C(t)$ is not considered. However, the inclusion of the variable $C(t)$ in itself does not improve the sluggish response, although they filter the incoming noise quite well. The reason is that although the two filter equations originally proposed by Stephanopoulos and San do provide the means for the two noises $\eta(t)$ and $\eta'(t)$ to pass through two system blocks, they are implicitly assigned an infinitely long time constant. (These equations have T misplaced; Equations (2.4.10a) and (2.4.10b) are believed to be the best ones.)

$$\lim_{T_\mu \to \infty} \frac{d\mu(t)}{dt} = -\frac{\mu(t)}{T_\mu} + C(t) + \eta(t) \quad \Longrightarrow \quad \frac{d\mu(t)}{dt} = C(t) + \eta(t) \qquad (2.4.11a)$$

$$\lim_{T_c \to \infty} \frac{dC(t)}{dt} = -\frac{C(t)}{T_c} + \eta'(t) \quad \Longrightarrow \quad \frac{dC(t)}{dt} = \eta'(t) \qquad (2.4.11b)$$

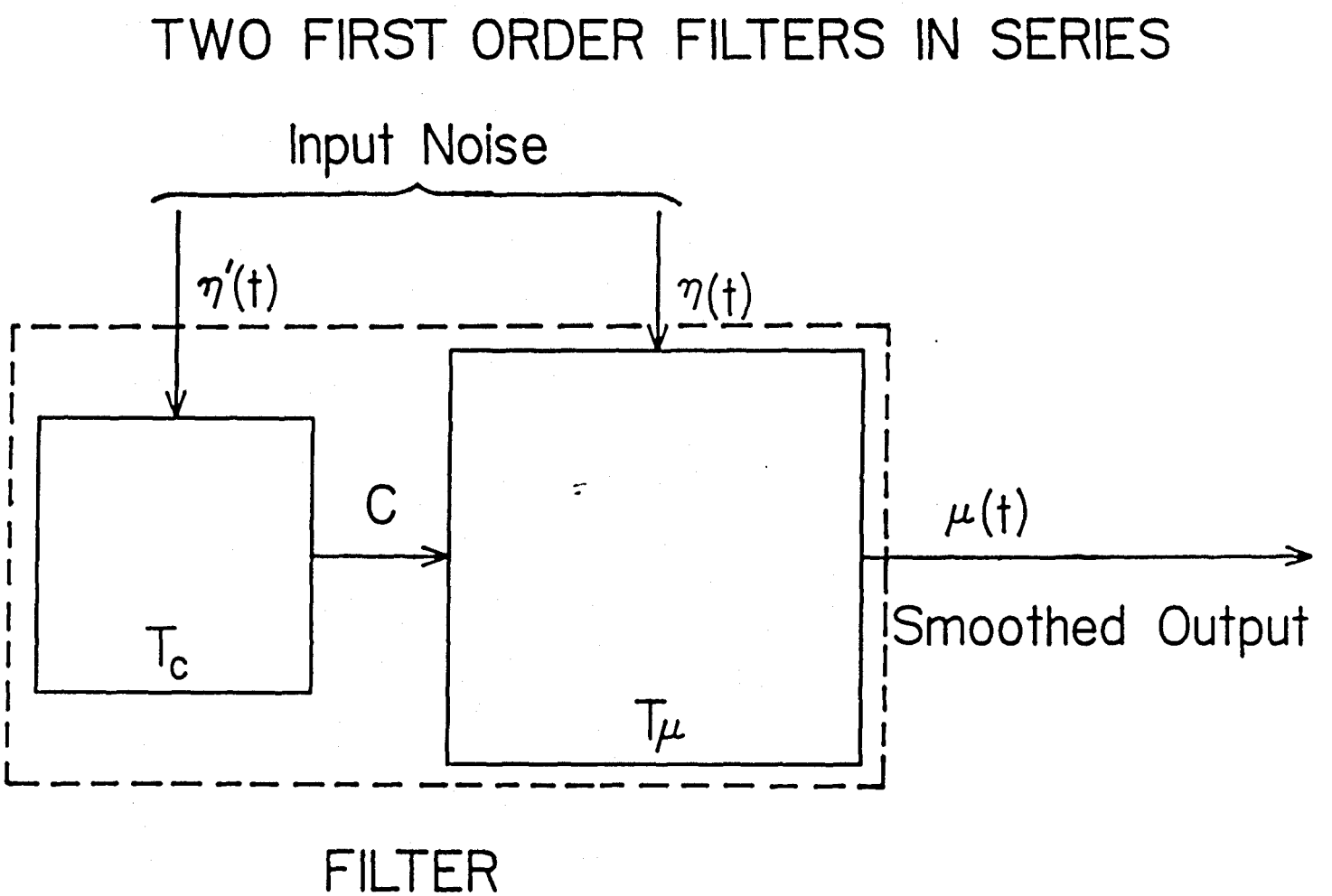


Figure 2.4.3. Noise filtering with two first-order dynamic system blocks in series.

To correct for this sluggish response, Stephanopoulos and San further suggested the use of a finite time constant that has the same order as the typical system response time represented by the dilution rate, D.

$$\frac{d\mu(t)}{dt} = C(t) + \eta(t) \tag{2.4.12a}$$

$$\frac{dC(t)}{dt} = -\frac{C(t)}{T} + \eta'(t) \tag{2.4.12b}$$

Because these equations are missing the term involving $\mu(t)$ itself, a possible exponential decay term in the characteristic equation, $\frac{1}{(Ts+1)}$, is substituted with a constant $\frac{1}{s}$. It is not optimal and is a special case of the modified formulation of Equations (2.4.10a) and (2.4.10b) presented earlier. Our modified set of filter equations have an overall second-order transfer function:

$$(T_\mu s + 1)U(s) = C(s) + Q_\mu(s) \tag{2.4.13b}$$

$$(T_c s + 1)C(s) = Q_c(s), \tag{2.4.13b}$$

which can be combined to give the final transfer function:

$$U(s) = \frac{1}{(T_\mu s + 1)(T_c s + 1)} Q_c(s) + \frac{1}{(T_\mu s + 1)} Q_\mu(s). \tag{2.4.13c}$$

In the time domain, the correlated noises $\eta(t)$ and $\eta'(t)$ are filtered through a convolution integral:

$$\mu(t) = \int_{-\infty}^{t} \eta'(h) k_c(t-h)dh + \int_{-\infty}^{t} \eta(h) k_\mu(t-h)dh, \tag{2.4.14}$$

where $k_c(t)$ and $k_\mu(t)$ are the filter kernel equations.

$$k_\mu(t) = \frac{1}{T_\mu} e^{-\frac{t}{T_\mu}} \tag{2.4.15a}$$

$$k_c(t) = \begin{cases} \frac{1}{T_c - T_\mu}\left(e^{-\frac{t}{T_c}} - e^{-\frac{t}{T_\mu}}\right) & \text{for } T_c \neq T_\mu \\ \frac{1}{T} e^{-\frac{t}{T}} + \frac{t}{T^2} e^{-\frac{t}{T}} & \text{for } T_c = T_\mu = T\ . \end{cases} \tag{2.4.15b}$$

A more general form of the second-order filter equations may be expressed as:

$$T_\mu \frac{d\mu(t)}{dt} + \beta\mu(t) = \gamma_{\mu c} C(t) + q_\mu(t) \tag{2.4.16a}$$

$$T_c \frac{dC(t)}{dt} + \alpha C(t) = \gamma_{c\mu} \mu(t) + q_c(t) \tag{2.4.16b}$$

The filter transfer function for these equations is:

$$U(s) = \frac{\gamma_{\mu c}}{(T_c s + \alpha)(T_\mu s + \beta) - \gamma_{\mu c}\gamma_{c\mu}} Q_c(s) + \frac{(T_c s + \alpha)}{(T_c s + \alpha)(T_\mu s + \beta) - \gamma_{\mu c}\gamma_{c\mu}} Q_\mu(s). \tag{2.4.17}$$

The corresponding characteristic equation is:

$$(T_c s + \alpha)(T_\mu s + \beta) - \gamma_{\mu c}\gamma_{c\mu} = 0, \tag{2.4.18}$$

which can be expressed in an even more general form:

$$T^2 s^2 + 2\xi T s + 1 = 0. \tag{2.4.19}$$

Equating the above two equations, one obtains:

$$T = \sqrt{\frac{T_c T_\mu}{\alpha\beta - \gamma_{\mu c}\gamma_{c\mu}}} \tag{2.4.20}$$

$$\frac{\alpha T_\mu + \beta T_c}{\alpha\beta - \gamma_{\mu c}\gamma_{c\mu}} = 2\xi T \tag{2.4.21}$$

A second-order filter with the desired properties can be designed by assigning proper values to the filter parameters. A similar equation has been discussed in the section entitled "Formulation of Time-Lag Equations". The time-domain filter equation corresponding to the above characteristic equation is:

$$k(t) = \begin{cases} \frac{1}{T\sqrt{1-\xi^2}} e^{-\frac{\xi t}{T}} \sin(\sqrt{1-\xi^2}\frac{t}{T}) & \text{for } 0 < \xi < 1 \\ \frac{t}{T^2} e^{-\frac{t}{T}} & \text{for } \xi = 1 \\ \frac{1}{T\sqrt{\xi^2-1}} e^{-\frac{\xi t}{T}} \sinh(\sqrt{\xi^2-1}\frac{t}{T}) & \text{for } \xi > 1. \end{cases} \tag{2.4.22}$$

Usually, the critically damped case of $\xi = 1$ is chosen. The underdamped case of $0 < \xi < 1$ is to be avoided due to the oscillations, although the response is fast. The overdamped case of $\xi > 1$ is not desirable because the response is sluggish.

These equations used for adaptive parameter identification and state estimation can be viewed as imparting weighing functions to the noise that drives the system. In effect, the noise is passed through a separate pre-filter before being subjected to Kalman filtering, although only one integrated program is usually written for the combined adaptive filtering and Kalman filtering. Figure 2.4.4 shows the application of the pre-filter developed in this section.

Colored Noise: Another interesting point provided by the above set of equations is the transformation of white noise to colored noise. White noise is one where the autocorrelation function is a delta function. The autocorrelation function $\phi_{xx}(t)$ is defined as:

$$\phi_{xx}(t) = \mathbf{E}[x(t_1)x(t_1 + t)], \tag{2.4.23}$$

where $\mathbf{E}$ is the expected value operator. Because averaging a member of the ensemble over time gives the equivalent result as averaging all members of an ensemble at a fixed time due to the *ergodic hypothesis*, the above equation can be expressed alternatively as:

$$\phi_{xx}(t) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t + \tau)dt \tag{2.4.24}$$

The autocorrelation function $\phi_{xx}(t)$ in the frequency domain is called the power spectral density function, denoted as $\Phi_{xx}(\omega)$. Thus, with our earlier definition of Fourier transform, it is expressed as:

$$\Phi_{xx}(\omega) \;=\; \mathcal{F}\{\phi_{xx}(t)\} \;=\; \int_{-\infty}^{\infty} e^{i2\pi\omega t}\phi_{xx}(t)dt. \tag{2.4.25}$$

As mentioned previously, various other common forms of definitions for Fourier transform exist, and it is advised to use the transform in a consistent manner.

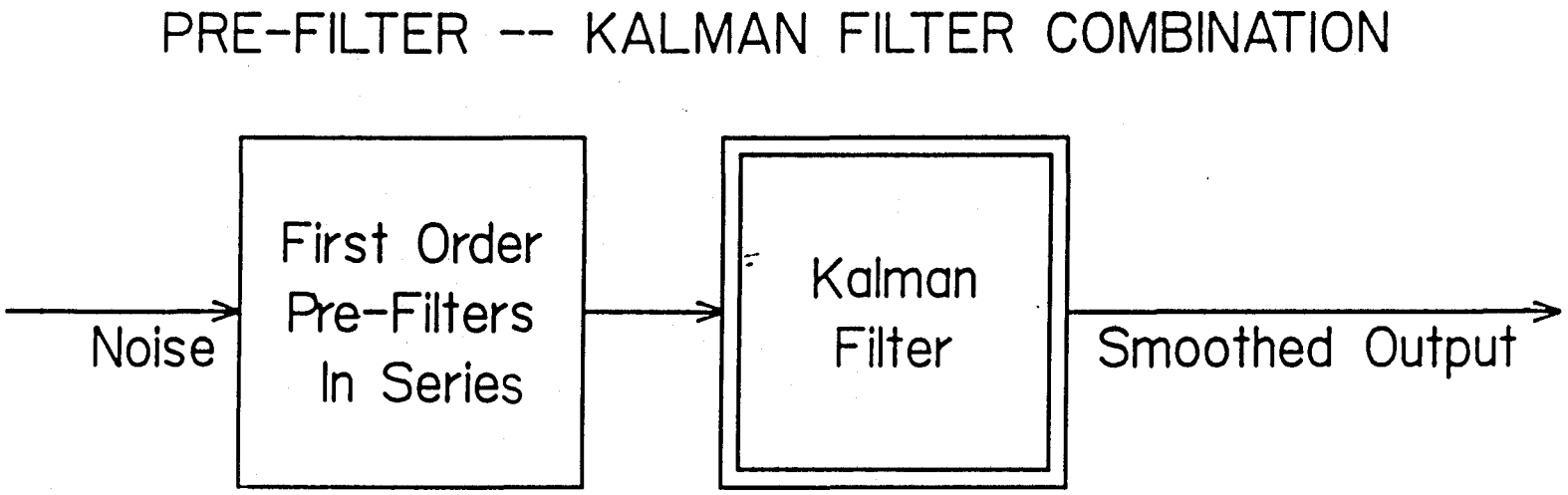


Figure 2.4.4. Pre-filter is often used to supplement the Kalman filter in bioreactor parameter identification and state estimation.

Because the filter equation and the kernel satisfies exactly the same mathematical relationship, the convolution theorem of Fourier transform can be used to see how the noise structure is transformed by the filter as it reduces the noise level. For example, the following convolution integral describes how the noise $\eta(t)$ can be reduced by the filter $k(t)$.

$$y_{\text{smooth}}(t) = \int_{-\infty}^{t} k(t-h)\eta(h)dh \tag{2.4.26}$$

In the frequency domain, the following relation holds.

$$Y_{\text{smooth}}(\omega) = K(\omega) \cdot \mathcal{E}(\omega) \tag{2.4.27}$$

The power spectral density function of a noisy variable can be calculated by multiplying the Fourier transform by its complex conjugate. The filter transfer function can also be obtained similarly. The objectives here are to investigate how a first-order dynamic filter transforms the power spectrum of a noisy input. This can be accomplished by simply taking the Fourier transform of the kernel (or filter equation).

$$K(\omega) \;=\; \mathcal{F}\{k(t)\} \;=\; \int_{-\infty}^{\infty} e^{i2\pi\omega t}k(t)dt \tag{2.4.28}$$

Substituting a 0th-order kernel, one obtains:

$$K_0(\omega) \;=\; \mathcal{F}\{k_0(t)\} \;=\; \int_{-\infty}^{\infty} e^{i2\pi\omega t}\frac{1}{T}e^{-\frac{t}{T}}dt \;=\; \frac{2}{1+i2\pi\omega T}. \tag{2.4.29}$$

The above equation is basically one of $\frac{1}{1+i\omega T}$, factors such as 2π being dependent on the definition of a Fourier transform. The power spectral density of a first-order kernel is:

$$P = K_0(\omega) \cdot K_0^*(\omega) \Rightarrow \frac{1}{1+i\omega T} \cdot \frac{1}{1+i\omega T}^{*} = \frac{1}{1+\omega^2T^2}. \tag{2.4.30}$$

This function is plotted in Figure 2.4.5 along with a few other frequently encountered power spectral density functions.

Thus, given a white noise that has an evenly distributed power spectral density of Figure 2.4.5a, the filter modifies the noise structure by multiplying $\frac{1}{i\omega^2 T^2}$ to the original constant function. This process results in the smoothed estimates having the power spectral density of $\frac{1}{i\omega^2 T^2}$. Note that the power of high frequency components for the smoothed estimates is low, and high frequency noise is thus reduced by the filter. Because the power spectral density function of the resulting smoothed estimates is no longer even, the filter has rendered the original white noise into a "colored" one.

In summary, this section describes the use of a kernel as a filter equation to relate the system input and the system output. When the system is regarded as a filter, the system input consists of the noisy input, and the system output is the smoothed output. Thus, there exist many applications for the kernel concept outside of bioreactor modeling.

2.5 REDUCTION OF STRUCTURED MODELS TO UNSTRUCTURED MODELS

The difference between a complex structured model and a simple unstructured model is analogous to that between statistical and classical thermodynamics. Whereas a structured model attempts to explain the observed phenomena through a large set of differential equations in terms of the more fundamental variables such as the concentrations of various intermediates, unstructured models are usually composed of those variables that can be physically "seen" or "felt" more readily and are, thus, more comprehensible to human minds. The proposed modeling approach herein attempts to retain the general form of an unstructured model so as to facilitate simple physical interpretation of the variables by such familiar terms or concepts as the specific growth rate. At the same time, this modeling approach attempts to incorporate only those metabolic intermediates that are important to

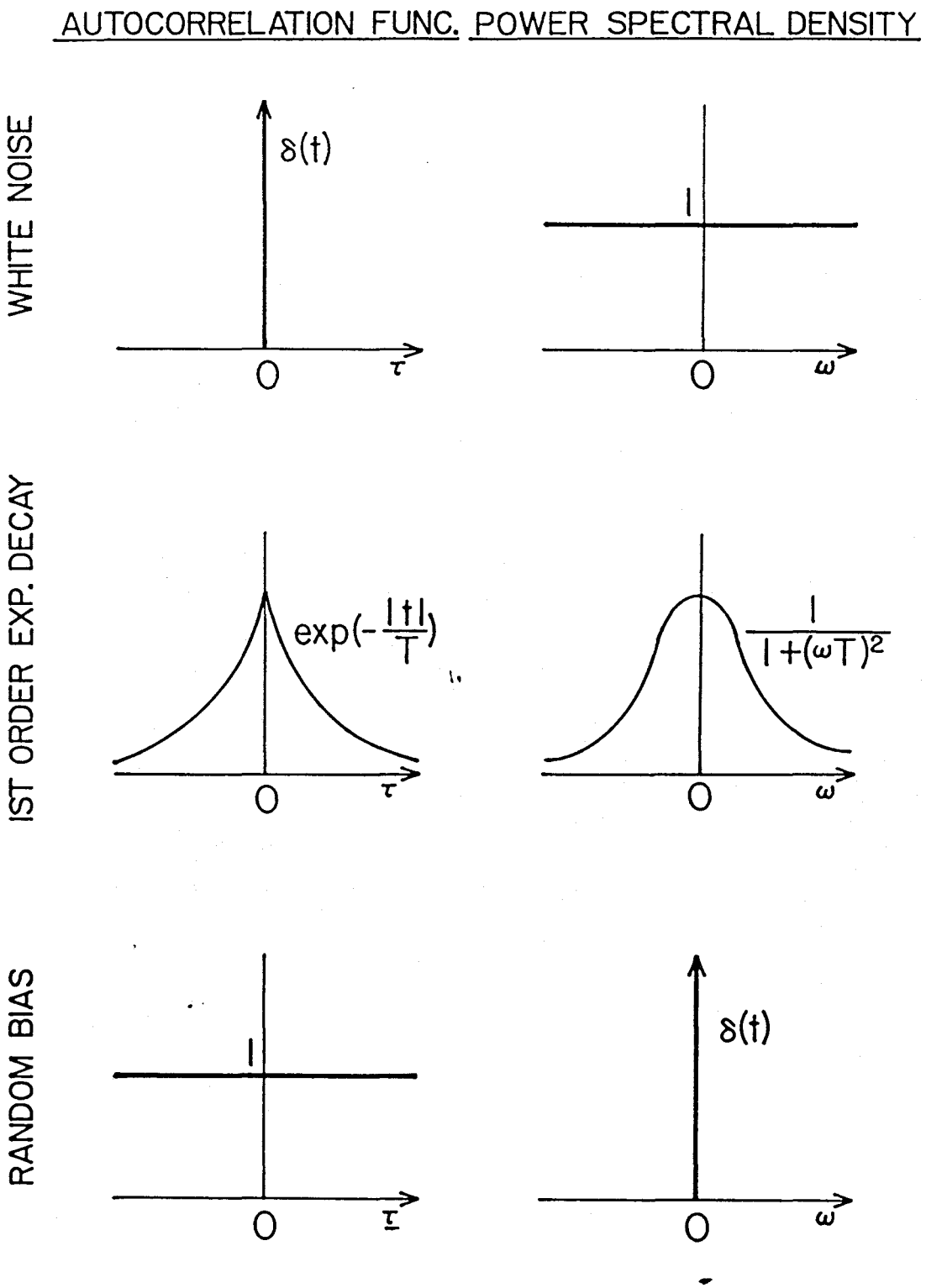


Figure 2.4.5. Autocorrelation functions and power spectral density functions for some frequently used random processes.

the dynamics of the system. It also attempts to reduce the order of a complicated structured model through the judicious process of lumping and the analysis of eigenvalue-eigenvector of a linearized system. How this can be accomplished is outlined below.

Origin of the Time-Lag Kernel: In general, a dynamic system (including a structured model) can be described by a set of first-order differential equations:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{2.5.1}$$

where $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the input to the system. Currently, there is no general established way of solving such a set of differential equations if they are nonlinear. As long as the nonlinearity is not too severe, one generally quasi-linearizes the nonlinear set of equations around the point of interest before attempting to solve them. For a system linear in the state variables, the above equation can be written as:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{g}(t). \tag{2.5.2}$$

Note that the dependence of $\mathbf{A}$ on time does not destroy the linearity. The fundamental-matrix solution to the above differential equation is expressed by the following Lagrange formula:

$$\begin{aligned} \mathbf{x}(t) &= \int_{-\infty}^{t} \mathbf{K}(t,h)\mathbf{g}(h)dh \\ &= \mathbf{K}(t,t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{K}(t,h)\mathbf{g}(h)dh, \end{aligned} \tag{2.5.3}$$

where $\mathbf{K}$ is the fundamental matrix of Equation (2.5.1).

$\mathbf{K}$ is also sometimes called the *transition matrix*. It has a few well known, extremely useful properties. The first one is that it satisfies the following matrix differential equation analogous to the homogeneous form of the state vector differential equation:

$$\frac{d\mathbf{K}(t,t_0)}{dt} = \mathbf{A}(t)\mathbf{K}(t,t_0) \qquad \text{for } t \geq t_0, \tag{2.5.4}$$

with the initial condition:

$$\mathbf{K}(t_0, t_0) = \mathbf{I}. \tag{2.5.5}$$

This dynamic relationship can be further generalized for any $h \geq t_0$:

$$\frac{d\mathbf{K}(t,h)}{dt} = \mathbf{A}(t)\mathbf{K}(t,h) \qquad \text{for } t \geq t_0, h \geq t_0. \tag{2.5.6}$$

$$\mathbf{K}(h,h) = \mathbf{I} \qquad \text{for } h \geq t_0 \tag{2.5.7}$$

The kernel matrix also satisfies the adjoint differential equation:

$$\frac{d\mathbf{K}^T(t,h)}{dh} = -\mathbf{A}^T(h)\mathbf{K}^T(t,h) \qquad \text{for } t \geq t_0, h \geq t_0. \tag{2.5.8}$$

For any t_1, t_2, $t_3 \geq t_0$, the kernel matrix can be chained:

$$\mathbf{K}(t_3, t_1) = \mathbf{K}(t_3, t_2) \cdot \mathbf{K}(t_2, t_1). \tag{2.5.9}$$

The above equation leads directly to the following identities:

$$\begin{aligned} \mathbf{I} = \mathbf{K}(t_1, t_1) &= \mathbf{K}(t_1, t_2) \cdot \mathbf{K}(t_2, t_1) \\ \Longrightarrow \mathbf{K}(t_1, t_2) &= \Big[\mathbf{K}(t_2, t_1)\Big]^{-1}. \end{aligned} \tag{2.5.10}$$

Thus, the kernel matrix is nonsingular for all $t \geq t_0$ and $h \geq t_0$. This can also be seen from the fact that exponential functions are never equal to zero. Furthermore,

$$\det\Big|\mathbf{K}(t_2, t_1)\Big| = \exp\left\{\int_{t_1}^{t_2} \mathrm{tr}\Big[F(t)\Big]\right\} \tag{2.5.11}$$

These properties are listed because they can be quite useful when handling a time-lag kernel.

Because the matrix $\mathbf{K}(t,h)$ depends on t and h separately and because the matrix itself may be monstrously dimensioned with numerous nonzero off-diagonal elements, it is difficult to be graphed and visualized in the traditional three-dimensional space that one is accustomed to. Thus, the meaning of each element of the matrix usually cannot be readily communicated with such conventional and easily comprehensible biochemical engineering terms as specific growth rate or yield, etc.

If the linearization matrix $\mathbf{A}(t)$ is constant, then this solution further reduces to:

$$\mathbf{x}(t) = \int_{-\infty}^{t} \mathbf{K}(t-h)\mathbf{g}(h)dh, \tag{2.5.12}$$

where

$$\mathbf{K}(t) = e^{\mathbf{A}t}. \tag{2.5.13}$$

Thus, the appearance of a kernel in Equations (2.2.8)-(2.2.11) is spontaneous; it arises mathematically during the process of solving a set of differential equations. As shown in the above derivation, what is called the kernel is mathematically equivalent to the fundamental matrix of a set of first-order ordinary differential equations. One also sees that the kernel inside the time-lag integral when expressed as $k(t-h)$ is actually the linearized time-invariant scalar representation of the more general form of $\mathbf{K}(t,h)$ of Equation (2.5.3).

Evaluation of a Multidimensional Time-Lag Kernel: There exist a number of techniques that can be used to evaluate the $n \times n$ matrix exponential $e^{\mathbf{A}t}$. The first method uses the infinite series expansion:

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2!}(\mathbf{A}t)^2 + \frac{1}{3!}(\mathbf{A}t)^3 + \ldots. \tag{2.5.14}$$

This method requires the repeated evaluation of matrix multiplications and is rather tedious. The second method makes use of the Cayley-Hamilton theorem. A consequence of this theorem is that a matrix exponential $e^{\mathbf{A}t}$ of dimension $n \times n$, can be expressed as an $(n-1)$th degree polynomial:

$$e^{\mathbf{A}t} = \sum_{j=0}^{n-1} \beta_j \mathbf{A}^j. \tag{2.5.15}$$

Thus, the problem is one of evaluating the n coefficients of β_j, $j = 0, 1, 2, \ldots, n-1$. The third method relies on the direct integration of

$$\frac{d}{dt}\left[e^{\mathbf{A}t}\right] = \mathbf{A}e^{\mathbf{A}t}, \tag{2.5.16}$$

with the initial condition:

$$e^0 = \mathbf{I}. \tag{2.5.17}$$

This is equivalent to integrating $(n \times n)$ simultaneous first-order ordinary differential equations.

Many other methods are available. The method used in this thesis converts the matrix $\mathbf{A}$ into a diagonal (or at least a Jordan normal) form via a similarity transformation.

$$\mathbf{J} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}, \tag{2.5.18}$$

where $\mathbf{T}$ is the transformation matrix composed of generalized eigenvectors, and $\mathbf{J}$ is a converted block-diagonal Jordan matrix that has the form:

$$\begin{bmatrix} \mathbf{J}_1 & 0 & \dots & 0 \\ 0 & \mathbf{J}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{J}_m \end{bmatrix}. \tag{2.5.19}$$

Each of the above Jordan blocks $\mathbf{J}_j$, dimensioned $n_j \times n_j$, has the form:

$$\mathbf{J}_j = \underbrace{\begin{bmatrix} \lambda_j & 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_j & 1 & \dots & 0 & 0 \\ 0 & 0 & \lambda_j & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_j & 1 \\ 0 & 0 & 0 & \dots & 0 & \lambda_j \end{bmatrix}}_{n_j \text{ columns}} n_j \text{ rows.} \tag{2.5.20}$$

The matrix exponential of $e^{\mathbf{J}t}$ is also block diagonal:

$$\begin{bmatrix} e^{\mathbf{J}_1 t} & 0 & \dots & 0 \\ 0 & e^{\mathbf{J}_2 t} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{\mathbf{J}_m t} \end{bmatrix}, \tag{2.5.21}$$

where the matrix exponential of the jth Jordan block $e^{\mathbf{J}_j t}$ is:

$$e^{\mathbf{J}_j t} = \underbrace{\begin{bmatrix} e^{\lambda_j t} & te^{\lambda_j t} & \frac{t^2}{2!}e^{\lambda_j t} & \frac{t^3}{3!}e^{\lambda_j t} & \dots \\ 0 & e^{\lambda_j t} & te^{\lambda_j t} & \frac{t^2}{2!}e^{\lambda_j t} & \dots \\ 0 & 0 & e^{\lambda_j t} & te^{\lambda_j t} & \dots \\ 0 & 0 & 0 & e^{\lambda_j t} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{n_j \text{ columns}} n_j \text{ rows.} \tag{2.5.22}$$

Finally the original matrix exponential can be evaluated by an inverse similarity transformation:

$$e^{\mathbf{A}t} = \mathbf{T}e^{\mathbf{J}t}\mathbf{T}^{-1}. \tag{2.5.23}$$

Time-Lag Kernel from a Structured Model: The structured model and the unstructured model are related in that a structured model can be reduced to an equivalent unstructured model. It will be shown that the connection between them is provided by the time-lag kernel.

The first step is to partition the vector of state variables, $\mathbf{x}(t)$, based on whether they appear in an unstructured model. Those variables that appear in the resulting unstructured model are grouped in $\mathbf{x}_1(t)$, and the remainder of the state vector $\mathbf{x}(t)$ that are included *only* in the structured model but not in the unstructured model are grouped in $\mathbf{x}_2(t)$. For example, the biomass, substrate, and product concentrations will be contained in $\mathbf{x}_1(t)$. All the intermediates and enzymes that are not considered as the product will be part of $\mathbf{x}_2(t)$. The result of this partition of $\mathbf{x}(t)$ is:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}. \tag{2.5.24}$$

The linearization matrix $\mathbf{A}(t)$ and the non-homogeneous forcing function $\mathbf{g}(t)$ can be partitioned similarly:

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{A}_{11}(t) & \mathbf{A}_{12}(t) \\ \mathbf{A}_{21}(t) & \mathbf{A}_{22}(t) \end{bmatrix} \tag{2.5.25}$$

$$\mathbf{g}(t) = \begin{bmatrix} \mathbf{g}_1(t) \\ \mathbf{g}_2(t) \end{bmatrix}. \tag{2.5.26}$$

With this partition, Equation (2.5.2) becomes:

$$\frac{d\mathbf{x}_1(t)}{dt} = \mathbf{A}_{11}(t)\mathbf{x}_1(t) + \mathbf{A}_{12}(t)\mathbf{x}_2(t) + \mathbf{g}_1(t) \tag{2.5.27}$$

$$\begin{aligned} \frac{d\mathbf{x}_2(t)}{dt} &= \mathbf{A}_{21}(t)\mathbf{x}_1(t) + \mathbf{A}_{22}(t)\mathbf{x}_2(t) + \mathbf{g}_2(t) \\ &= \mathbf{A}_{22}(t)\mathbf{x}_2(t) + \tilde{\mathbf{g}}(t), \end{aligned} \tag{2.5.28}$$

where $\tilde{\mathbf{g}}(t) = \mathbf{A}_{21}(t)\mathbf{x}_1(t) + \mathbf{g}_2(t)$ is the nonhomogeneous part of Equation (2.5.28), which has a fundamental matrix solution analogous to Equation (2.5.12) as described by:

$$\mathbf{x}_2 = \int_{-\infty}^{t} \mathbf{K}_{22}(t,h)\tilde{\mathbf{g}}(h)dh, \tag{2.5.29}$$

where $\mathbf{K}_{22}(t,h)$ is the fundamental matrix to $\mathbf{A}_{22}(t)$ of Equation (2.5.28).

Thus, the unstructured model's equivalent of the structured model of Equation (2.5.1) is now reduced to Equation (2.5.28), whose more general form is:

$$\frac{d\mathbf{x}_1(t)}{dt} = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t), \tag{2.5.30}$$

where $\mathbf{x}_2$ is the time-lag integral defined by Equation (2.5.29). If $\mathbf{x}_1(t)$ is composed of the biomass and limiting substrate concentrations as in our previous example, then $\mathbf{x}_2$ is simply the scalar observed specific growth rate, previously denoted y. Similarly, $\mathbf{u}$ is composed of control variables which, in our previous example, are the dilution rate, D, and the substrate concentration in the feed, s_f. $\mathbf{K}_{22}$ is the scalar time-lag kernel, k, and $\tilde{\mathbf{g}}$ is the scaler intrinsic specific growth rate, μ.

Note that Equation (2.5.30) alone is the unstructured model; the addition of the information provided by the integral in Equation (2.5.29) upgrades it to a structured model because these two combined equations are the *exact* equivalent of the original structured model described by Equation (2.5.2). The time-lag kernel matrix $\mathbf{K}_{22}$ is the relationship that ties these two traditional modeling approaches. As can be

seen from the preceeding equations, the time-lag kernel arises quite naturally as a consequence of reducing a larger set of dynamic equations of a structured model to a smaller set of dynamic equations of an unstructured model.

It is emphasized that the time-lag kernel is not simply an artificial mathematical concept; it is derived from a biological basis. Its presence can be explained by the fact that there exists a large collection of metabolic pathways and regulation steps. When a microorganism is subjected to a stimulus, it requires time for the cell to respond to the external stimuli as it adjusts its internal states one after another. For example, intermediate metabolites, precursors, enzymes, and various cofactors may be needed before the final product, which may be a specific enzyme, a chemical, or simply the cell biomass itself, can be assembled. Thus, the time-lag kernel compacts all our knowledge about the actual process of how a cell responds to the external stimuli into a simple functional form. It is the fundamental matrix to the missing dynamic equations. Conversely, a kernel has an equivalent representation in terms of structured dynamics. We see that a unique kernel can be constructed given the dynamics of the system, but the reconstruction of structure from a kernel function is not unique because more than one different process can be responsible for the same response and, thus, the same kernel function. In such a circumstance, the rule of modeling dictates that the simpler mechanism be chosen.

The kernel concept can be used to check the validity of the proposed mechanism in a structured model. For example, if the experimentally obtained kernel function does not agree with that directly derived from the structured model, one can conclude that the original hypothesis is perhaps erroneous, and, as mentioned earlier, the shape of the kernel function can give one some insight as to the type of mechanism that may be responsible for the observed kernel. One may then revise his hypothesis and recheck to verify if the kernel function now conforms to the

assumed mechanism. Alternatively, one may build a reasonably good structured model from some experimentally determined kernel functions. For example, if one were to find that the time-lag kernel could be approximated by a 0th-order exponential distribution function, then, based on the implied meaning of this kernel, one might plan further experiments aimed at identifying the rate limiting "critical enzyme." Thus, the kernel might be used to help suggest the type of experiment to be performed.

The link provided by the time-lag kernel is analogous to the relationship in statistical thermodynamics that translates such esoteric concepts as the number of possible quantum states (*i.e.*, the degeneracy) associated with the energy level, the canonical ensemble partition function, and the undetermined multiplier to such commonly understood concepts as temperature, pressure, entropy, energy, and chemical potential. The introduction of a time-lag kernel eliminates the need to maintain a close monitor on the complicated internal state of a microorganism, many of which cannot be easily monitored, just as a thermodynamic partition function cannot be easily measured. The use of a time-lag kernel allows a fermentation technologist to concentrate on those variables that he can intuitively feel and "see" through the available measurements, just as a chemical plant operator functions with temperature and pressure.

Examples of Structured Model Reduction:

The methodology on the reduction of a structured model to an unstructured model will be demonstrated by analyzing some typical structured models. Particular attention will be focused on the determination of whether a model indeed has structures and, if so, which state variables have structures. As mentioned previously, the appearance of a time-lag can be attributed to the process dynamics associated with the structure. The time-lag kernel associated with the neglected

kinetic information when a structured model is reduced to an unstructured model will be derived theoretically.

Imanaka's Model of Enzyme Production: (Imanaka *et al., J. Ferment. Technol.*, **50**, 633, 1972.) One of the structured examples is Imanaka's model of enzyme (α-galactosidase) production developed for *Monascus* sp. The kinetic expressions are derived based on the operon theory of enzyme production. For a batch fermentor, the model consists of the following set of eight dynamic equations:

Biomass: $$\frac{dX}{dt} = \mu X \tag{2.5.31.1}$$

Glucose: $$\frac{dS_A}{dt} = -\frac{1}{Y_A}\mu_A X \tag{2.5.31.2}$$

Galactose: $$\frac{dS_B}{dt} = -\frac{1}{Y_B}\mu_B X \tag{2.5.31.3}$$

Intra. Galactose: $$\frac{ds_{Bi}}{dt} = U[\frac{G_B S_B}{K_{mB} + S_B} - s_{Bi}] - k_1 s_{Bi} - \mu s_{Bi} \tag{2.5.31.4}$$

Repressor: $$\frac{dr}{dt} = k_2 - k_3 r - k_4 r \cdot s_{Bi} + k_5 \overline{r s_{Bi}} - \mu r \tag{2.5.31.5}$$

Rep/Ind Complex: $$\frac{d\overline{r s_{Bi}}}{dt} = k_4 r \cdot s_{Bi} - k_5 \overline{r s_{Bi}} - \mu \overline{r s_{Bi}} \tag{2.5.31.6}$$

mRNA: $$\frac{dm}{dt} = k_6(r_c - r) - k_7 m - \mu m \tag{2.5.31.7}$$

Galactosidase: $$\frac{de}{dt} = k_8 m - \mu e \tag{2.5.31.8}$$

In the above equations, the macroscopic variables X (biomass), S_A (glucose), and S_B (galactose) are expressed in concentration units of mg/(reactor volume); whereas, the intracellular components s_{Bi} (intracellular galactose), r (repressor), $\overline{r s_{Bi}}$ (repressor-inducer complex), m (messenger RNA), and e (α-galactosidase) are

expressed in concentration units of mg/(g cell). Note that each of the dynamic equations for the intracellular components has a term that contains μ multiplied by the component itself. This term represents the dilution factor due to the expanding cell volume as the microorganism grows.

Similarly, for a continuous fermentor, the above set of equations become:

Biomass: $$\frac{dX}{dt} = \mu_X - DX \tag{2.5.32.1}$$

Glucose: $$\frac{dS_A}{dt} = -\frac{1}{Y_A}\mu_A X + D(S_{Af} - S_A) \tag{2.5.32.2}$$

Galactose: $$\frac{dS_B}{dt} = -\frac{1}{Y_B}\mu_B X + D(S_{Bf} - S_B) \tag{2.5.32.3}$$

Intra. Galactose: $$\frac{dS_{Bi}}{dt} = U[\frac{G_B S_B X}{K_{mB} + S_B} - S_{Bi}] - k_1 S_{Bi} - DS_{Bi} \tag{2.5.32.4}$$

Repressor: $$\frac{dR}{dt} = k_2 X - k_3 R - k_4 \frac{R \cdot S_{Bi}}{X} + k_5 \overline{RS_{Bi}} - DR \tag{2.5.32.5}$$

Rep/Ind Complex: $$\frac{d\overline{RS_{Bi}}}{dt} = k_4 \frac{R \cdot S_{Bi}}{X} - k_5 \overline{RS_{Bi}} - D\overline{RS_{Bi}} \tag{2.5.32.6}$$

mRNA: $$\frac{dM}{dt} = k_6(R_c - R) - k_7 M - DM \tag{2.5.32.7}$$

Galactosidase: $$\frac{dE}{dt} = k_8 M - DE \tag{2.5.32.8}$$

As in the batch fermentor, X (biomass), S_A (glucose), and S_B (galactose) are expressed in concentration units of mg/(reactor volume). However, intracellular components S_{Bi} (intracellular galactose), R (repressor), $\overline{RS_{Bi}}$ (repressor-inducer complex), M (messenger RNA), and E (α-galactosidase) are expressed in concentration units of mg/(fermentor volume) in the above set of equations for a continuous fermentor. The symbols for the cellular volume-based variables are in lower

case letters, and those for the fermentor volume-based variables are in upper case letters. This change of units is used to keep the two sets of equations in similar forms. These two sets of variables differ from each other by a factor of X:

$$S_{Bi} = X \cdot s_{Bi} \tag{2.5.33.a}$$

$$R = X \cdot r \qquad R_c = r_c \cdot r_c \tag{2.5.33.b}$$

$$\overline{RS_{Bi}} = X \cdot \overline{rs_{Bi}} \tag{2.5.33.c}$$

$$M = X \cdot m \tag{2.5.33.d}$$

$$E = X \cdot e \tag{2.5.33.e}$$

The first three of these equations describe the concentration variation of the major macroscopic components in the fermentor. These expressions are directly obtained from the well-known unstructured dynamic equations, with Monod-type of specific growth rate constitutive relationships.

$$\mu = \mu_A + \mu_B \tag{2.5.34.a}$$

$$\mu_A = \frac{\mu_{mA} S_A}{K_{SA} + S_A + \frac{K_{SA}}{K_i} S_B} \tag{2.5.34.b}$$

$$\mu_B = \begin{cases} \frac{\mu_{mB} S_B}{K_{SB} + S_B} & \text{for } S_A < S_{Ac} \\ 0 & \text{for } S_{Ac} \leq S_A \end{cases} \tag{2.5.34.c}$$

Because these first three dynamic equations contain no state variables other than themselves, they are completely decoupled from the rest of the equations and can be solved independently of the enzyme production. Furthermore, because of this complete decoupling, there is no structure for the biomass, glucose, or galactose. Thus, there is no time-lag in the response of these variables.

However, the above statement does not hold true for the enzyme. As shown in Figure 2.5.1, there are five dynamic steps that separate the overall output (e) from the input (D, S_{Af}, and S_{Bf}). Thus, all the structure is contained in a sequence of

events that finally lead to the enzyme production. The enzyme production requires the presence of mRNA (m), which is produced when the repressor level is below the critical level of r_c. Mathematically, this on/off event can be expressed as:

$$k_6 = \begin{cases} k_6 & \text{for } r < r_c \\ 0 & \text{for } r_c \leq r \end{cases} \tag{2.5.35}$$

The repressor, in turn, is produced at a constant rate and is inactivated by combining with the inducer – the intracellular galactose (s_{Bi}). The transport of galactose into the cell is effectively turned off when the fermentor glucose concentration exceeds a critical value of S_{Ac}. This statement is expressed mathematically as:

$$U = \begin{cases} U & \text{for } S_A < S_{Ac} \\ 0 & \text{for } S_{Ac} \leq S_A \end{cases} \tag{2.5.36}$$

As the first example, the continuous fermentor with a step change in the dilution rate from D=1.40 hr^{-1} to D=1.42 hr^{-1} at t=0 hr is simulated with the above set of equations. The same set of model parameters claimed to be used by the original authors are employed to generate the concentration profiles. The values of the parameters used are reproduced below:

Experimental Model Values		
μ_{mA}	0.215	hr^{-1}
μ_{mA}	0.208	hr^{-1}
K_{SA}	0.154	g/l
K_{SB}	0.258	g/l
K_i	0.139	g/l
Y_A	0.530	g/g cell
Y_B	0.516	g/g cell

BLOCK DIAGRAM OF IMANAKA'S MODEL

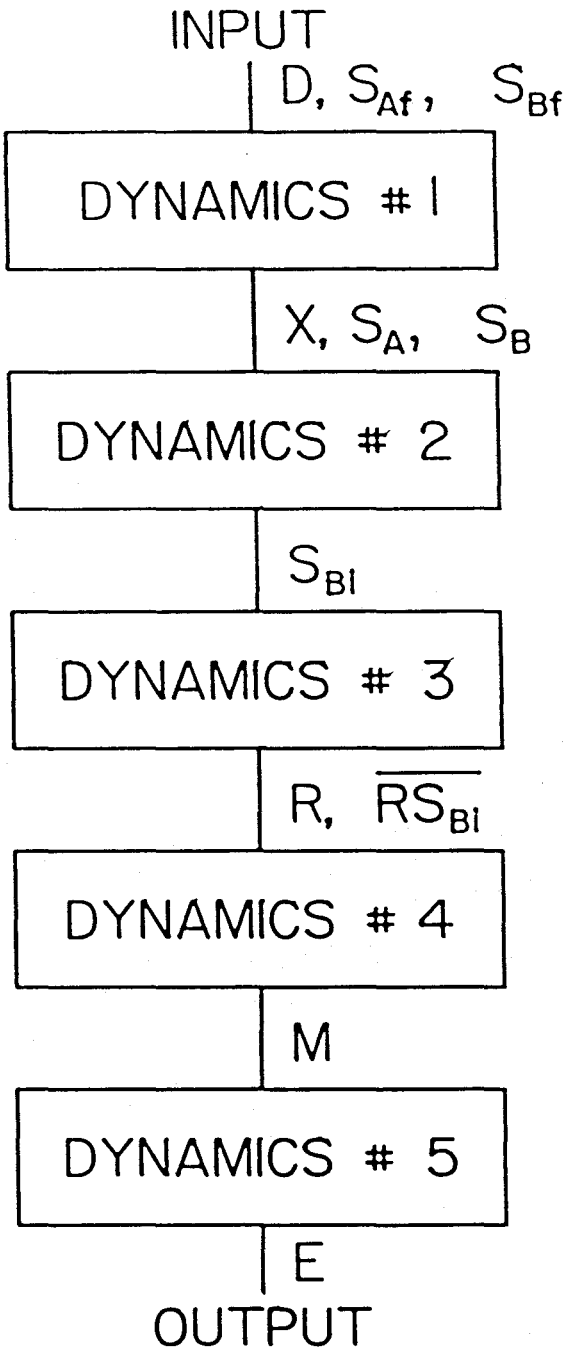


Figure 2.5.1. Steps leading to the formation of galactosidase.

Undetermined Model Values		
k_1	40.	hr^{-1}
k_2	1.	mg/(g cell)-hr
k_3	1.	hr^{-1}
k_4	0.1	(g cell)/mg-hr
k_5	0.0001	hr^{-1}
k_6	1.	hr^{-1}
k_7	8.	hr^{-1}
k_8	4.	K unit/(mg mRNA)-hr
U	100.	hr^{-1}
G_2	3.5	mg/(g cell)-hr
K_{mB}	0.00000001	g/l

Critical Values		
S_{Ac}	0.225	g/l
r_c	0.803	mg/(g cell)

Operating Conditions		
D	0.140 → 0.142	hr^{-1}
S_{fA}	20.	g/l
S_{fB}	5.	g/l

Initial Conditions		
X_0	13.	g/l
S_{A0}	0.223	g/l
S_{B0}	0.0501	g/l
s_{Bi0}	2.5	mg/(g cell)
r_0	0.718	mg/(g cell)
$\overline{rs_{Bi0}}$	1.28	mg/(g cell)
m_0	0.0104	mg/(g cell)
e_0	0.297	K units/(g cell)

The original authors' results are reproduced in Figure 2.5.2. The results of this author's simulation are shown in Figures 2.5.3 and 2.5.4. It should be noted that even after an extensive effort in experimenting with various sets of model parameters and the time of the dilution rate shift, the author was *unable* to obtain the type of behavior claimed by Imanaka *et al.* Because, as mentioned previously, the first three macroscopic variables are decoupled, these three dynamic equations can be analyzed independently of the rest of the equations. It is known that for a system described by Monod-style dynamics, a sudden increase in the glucose concentration shown in Figure 2.5.2 at t=7 hr is not possible unless some type of external disturbance is introduced at that instant. Alternatively, suppose the dilution rate were shifted at t=7 hr, there would also be sudden changes in the biomass and galactose concentrations at the same time, which are not present in the same figure.

The first step in expressing a structured model in a time-lag format is to identify the unstructured variables. In this example, the state vector for the unstructured

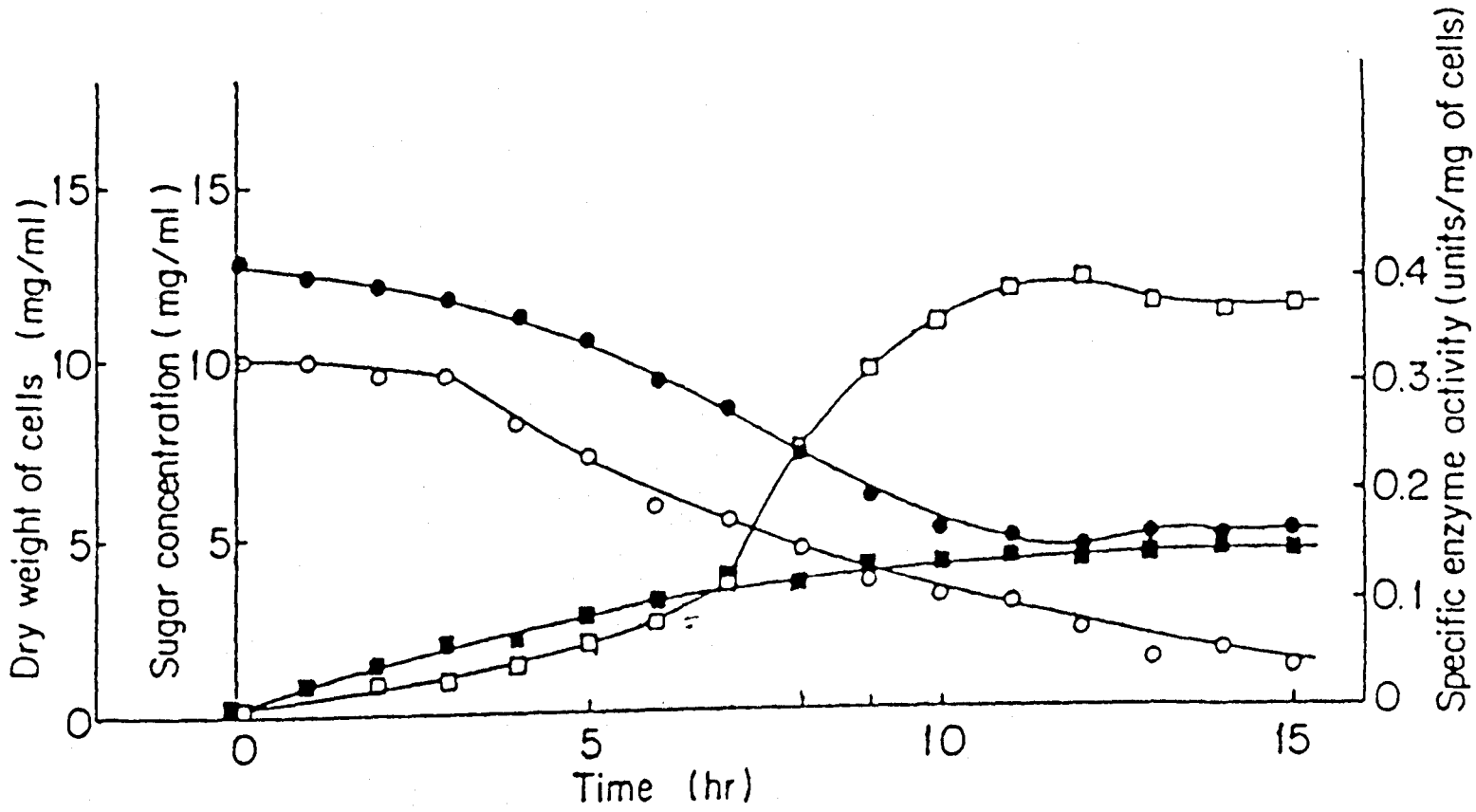


Figure 2.5.2. Transient behavior in a single-stage continuous culture. Glucose (2%) and galactose (0.5%) were used as carbon sources. The dilution rate was changed from 0.140 to 0.142 hr^{-1}. Reproduced from Imanaka *et al.* (1973).

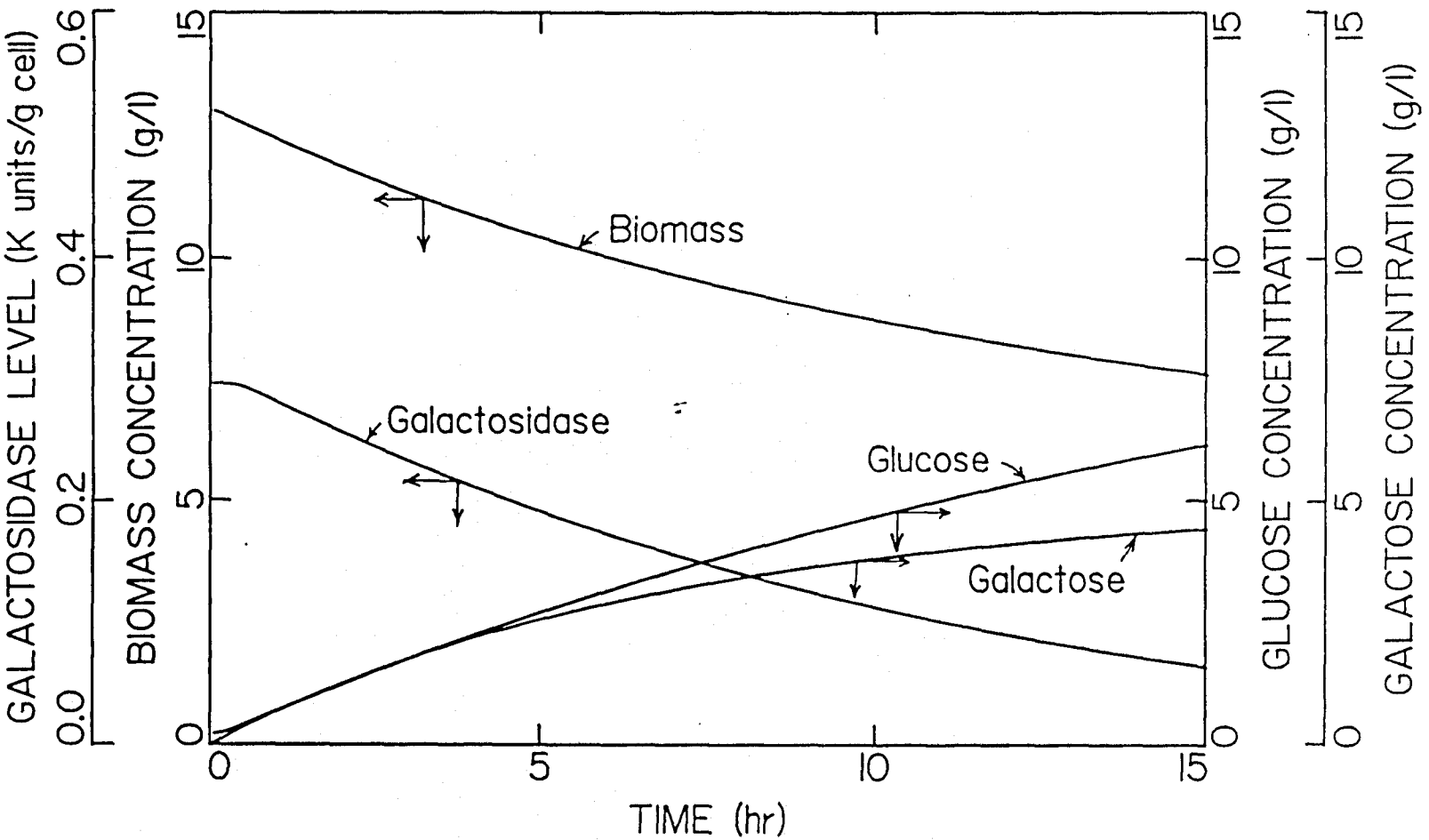


Figure 2.5.3. Time profile of macroscopic state variables (biomass, glucose, galactose, and enzyme) after a shift up in the dilution rate from 1.40 hr^{-1} to 1.42 hr^{-1} at t=0 hr. See text for the values of model parameters and initial conditions.

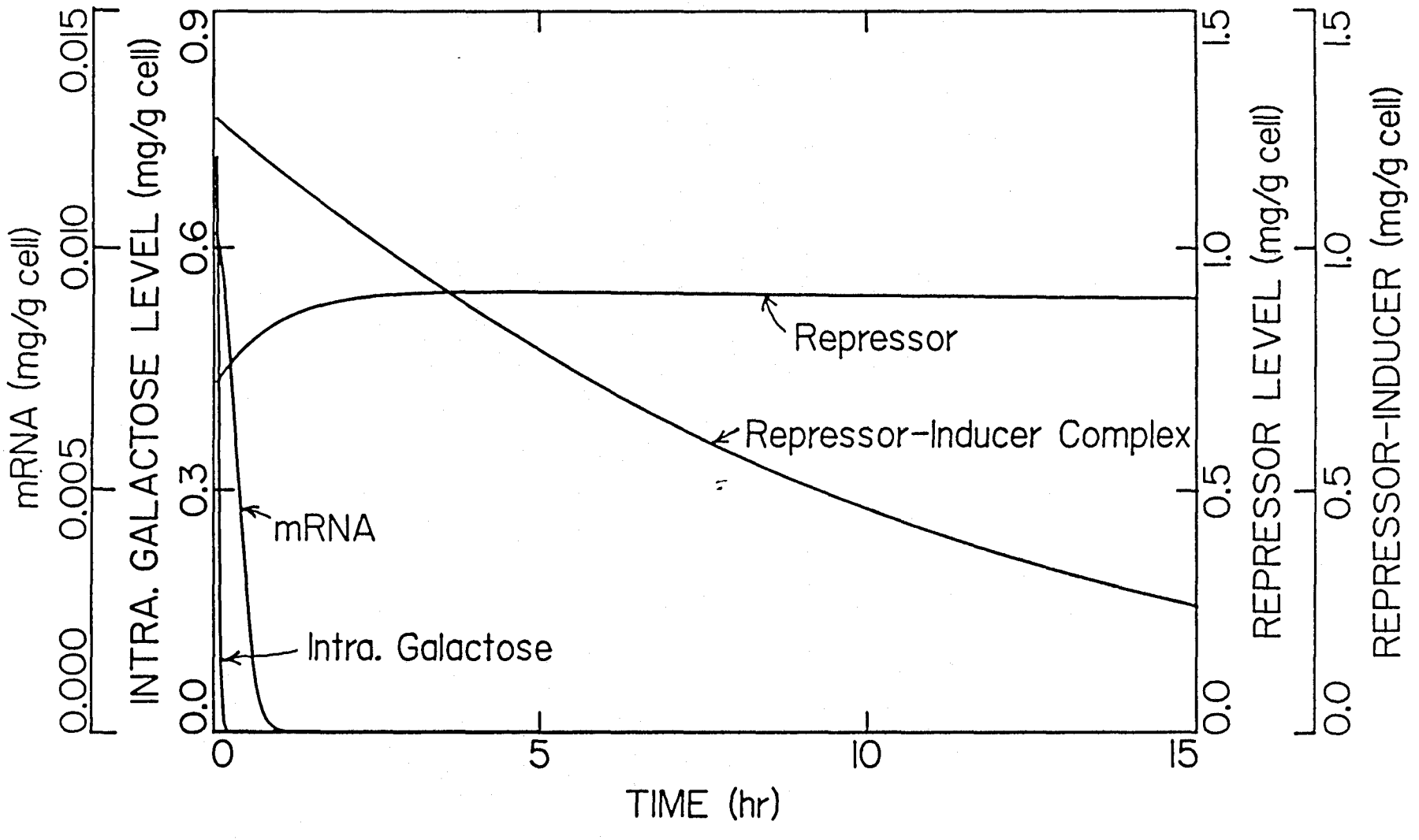


Figure 2.5.4. Time profile of intracellular state variables (intracellular galactose, repressor, repressor-inducer complex, and mRNA) after a shift up in the dilution rate from 1.40 hr^{-1} to 1.42 hr^{-1} at t=0 hr. See text for the values of model parameters and initial conditions.

variables is:

$$\mathbf{x}_1(t) = \begin{bmatrix} X(t) \\ S_A(t) \\ S_B(t) \\ E(t) \end{bmatrix} \tag{2.5.37}$$

The remaining intracellular variables are partitioned into $\mathbf{x}_2(t)$:

$$\mathbf{x}_2(t) = \begin{bmatrix} S_{Bi}(t) \\ R(t) \\ \overline{RS_{Bi}}(t) \\ M(t) \end{bmatrix} \tag{2.5.38}$$

Thus, the equivalent unstructured model presented in the form of Equation (2.5.30) is:

$$\underbrace{\begin{bmatrix} \frac{dX(t)}{dt} \\ \frac{dS_A(t)}{dt} \\ \frac{dS_B(t)}{dt} \\ \frac{dE(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}_1(t)}{dt}} = \underbrace{\begin{bmatrix} \mu_X - DX \\ -\frac{1}{Y_A}\mu_A X + D(S_{Af} - S_A) \\ -\frac{1}{Y_B}\mu_B X + D(S_{Bf} - S_B) \\ k_8 M - DE \end{bmatrix}}_{\mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t)} \tag{2.5.39}$$

The above equation indicates that there is a single time-lag variable M in the dynamics of enzyme formation. The time-lag kernel differential equation can be obtained for this multidimensional system from the dynamics of $\mathbf{x}_2(t)$. Following

Equation (2.5.28), the linearized dynamics of $\mathbf{x}_2(t)$ is expressed as:

$$\underbrace{\begin{bmatrix} \frac{dS_{Bi}(t)}{dt} \\ \frac{dR(t)}{dt} \\ \frac{d\overline{RS_{Bi}}}{dt} \\ \frac{dM(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}_2(t)}{dt}} =$$

$$\underbrace{\begin{bmatrix} -U-k_1-D & 0 & 0 & 0 \\ -\frac{k_4R_0}{X_0} & -\frac{k_4S_{Bi0}}{X_0}-k_3-D & k_5 & 0 \\ \frac{k_4R_0}{X_0} & \frac{k_4S_{Bi0}}{X_0} & -k_5-D & 0 \\ 0 & -k_6 & 0 & -k_7-D \end{bmatrix}}_{\mathbf{A}_{22}(t)} \underbrace{\begin{bmatrix} S_{Bi}(t) \\ R(t) \\ \overline{RS_{Bi}}(t) \\ M(t) \end{bmatrix}}_{\mathbf{x}_2(t)}$$

$$+ \underbrace{\begin{bmatrix} U\frac{G_BS_B(t)X(t)}{K_{mB}+S_B(t)} \\ \left[k_2+\frac{k_4R_0S_{Bi0}}{X_0^2}\right]X(t) \\ -\frac{k_4R_0S_{Bi0}}{X_0^2}X(t) \\ k_6r_cX(t) \end{bmatrix}}_{\tilde{\mathbf{g}}(t)} \tag{2.5.40}$$

The solution of the above set of dynamic equations is expressed in the time-lag kernel matrix form of Equation (2.5.29) as:

$$\begin{aligned} \mathbf{x}_2(t) &= \int_{-\infty}^{t} \underbrace{e^{\mathbf{A}_{22}(t-h)}}_{\mathbf{K}_{22}(t-h)} \tilde{\mathbf{g}}(h)dh \\ &= \int_{-\infty}^{t} \mathbf{K}_{22}(t-h)\tilde{\mathbf{g}}(h)dh. \end{aligned} \tag{2.5.41}$$

Thus, the time-lag variable M(t) in Equation (2.5.39) can be expressed in an equivalent kernel integral format as:

$$\begin{aligned} M(t) = \int_{-\infty}^{t} \Bigg[& k_{41}(t-h)U\left[\frac{G_BS_B(h)}{K_{mB}+S_B(h)}\right] + k_{42}(t-h)\left[k_2+\frac{k_4R_0S_{Bi0}}{X_0^2}\right] \\ & \cdot -k_{43}(t-h)\frac{k_4R_0S_{Bi0}}{X_0^2} + k_{44}(t-h)k_6r_c\Bigg] X(h)dh, \end{aligned} \tag{2.5.42}$$

where $k_{ij}(t)$ is the ijth element of the matrix exponential $e^{\mathbf{A}_{22}t}$.

The above equation can be simplified for the dilution rate shift-up example. Because of the large value of U, the dynamics of s_{Bi} is fast, and s_{Bi} is zero for the most part. Thus, its dynamics can be ignored to reduce the dimension of the system by 1 in order to facilitate greatly the mathematical analysis. With $s_{Bi} = 0$, Equation (2.5.40) becomes:

$$\underbrace{\begin{bmatrix} \frac{dR(t)}{dt} \\ \frac{d\overline{RS_{Bi}}}{dt} \\ \frac{dM(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}_2(t)}{dt}} = \underbrace{\begin{bmatrix} -k_3 - D & k_5 & 0 \\ 0 & -k_5 - D & 0 \\ -k_6 & 0 & -k_7 - D \end{bmatrix}}_{\mathbf{A}_{22}(t)} \underbrace{\begin{bmatrix} R(t) \\ \overline{RS_{Bi}}(t) \\ M(t) \end{bmatrix}}_{\mathbf{x}_2(t)} + \underbrace{\begin{bmatrix} k_2 X(t) \\ 0 \\ k_6 r_c X(t) \end{bmatrix}}_{\tilde{\mathbf{g}}(t)} \tag{2.5.43}$$

Note that linearization is not necessary with $s_{Bi} = 0$. At D=0.142 hr^{-1}, substituting the numerical values for the model parameters into $\mathbf{A}_{22}$ gives:

$$\mathbf{A}_{22} = \begin{bmatrix} -1.142 & 0.0001 & 0 \\ 0 & -0.1421 & 0 \\ -1 & 0 & -0.8142 \end{bmatrix} \tag{2.5.44}$$

$$\begin{aligned} 0 = \det|\mathbf{A}_{22} - \lambda\mathbf{I}| &= \begin{vmatrix} -1.142 - \lambda & 0.0001 & 0 \\ 0 & -0.1421 - \lambda & 0 \\ -1 & 0 & -8.142 - \lambda \end{vmatrix} \\ &= (-1.142 - \lambda)(-0.1421 - \lambda)(-8.142 - \lambda) \end{aligned} \tag{2.5.45}$$

$$\Longrightarrow \qquad \lambda_1 = -1.142 \qquad \lambda_2 = -0.1421 \qquad \lambda_3 = -8.142 \tag{2.5.46a}$$

$$\Longrightarrow \qquad \mathbf{T} = \begin{bmatrix} -7 & -7.9999 & 0 \\ 0 & -79991.0001 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{T}^{-1} = \begin{bmatrix} -0.1429 & 1.43 \times 10^{-5} & 0 \\ 0 & -1.25 \times 10^{-5} & 0 \\ 0.1429 & -1.79 \times 10^{-6} & 1 \end{bmatrix} \tag{2.5.46b}$$

$$\mathbf{K}(t) = e^{\mathbf{A}_{22}t} = \begin{bmatrix} k_{11}(t) & k_{12}(t) & k_{13}(t) \\ k_{21}(t) & k_{22}(t) & k_{23}(t) \\ k_{31}(t) & k_{32}(t) & k_{33}(t) \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} -7 & -7.9999 & 0 \\ 0 & -79991.0001 & 0 \\ 1 & 1 & 1 \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} e^{\lambda_1 t} & 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 \\ 0 & 0 & e^{\lambda_3 t} \end{bmatrix}}_{e^{\mathbf{J}t}} \underbrace{\begin{bmatrix} -0.1429 & 1.43 \times 10^{-5} & 0 \\ 0 & -1.25 \times 10^{-5} & 0 \\ 0.1429 & -1.79 \times 10^{-6} & 1 \end{bmatrix}}_{\mathbf{T}^{-1}}$$

$$= \begin{bmatrix} e^{\lambda_1 t} & \approx 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 \\ 0.1429(e^{\lambda_3 t} - e^{\lambda_1 t}) & \approx 0 & e^{\lambda_3 t} \end{bmatrix} \tag{2.5.46c}$$

With the given initial conditions, Equation (2.5.41) is now reduced to:

$$\underbrace{\begin{bmatrix} R(t) \\ \overline{RS_{Bi}}(t) \\ M(t) \end{bmatrix}}_{\mathbf{x}_2(t))} = \underbrace{\begin{bmatrix} k_{11}(t) & k_{12}(t) & k_{13}(t) \\ k_{21}(t) & k_{22}(t) & k_{23}(t) \\ k_{31}(t) & k_{32}(t) & k_{33}(t) \end{bmatrix}}_{\mathbf{K}_{22}(t)} \underbrace{\begin{bmatrix} R(0) \\ \overline{RS_{Bi}}(0) \\ M(0) \end{bmatrix}}_{\mathbf{x}_2(0)}$$

$$+ \int_0^t \underbrace{\begin{bmatrix} k_{11}(t-h) & k_{12}(t-h) & k_{13}(t-h) \\ k_{21}(t-h) & k_{22}(t-h) & k_{23}(t-h) \\ k_{31}(t-h) & k_{32}(t-h) & k_{33}(t-h) \end{bmatrix}}_{\mathbf{K}_{22}(t-h)} \underbrace{\begin{bmatrix} k_2 X(h) \\ 0 \\ k_6 r_c X(h) \end{bmatrix}}_{\tilde{\mathbf{g}}(h)} dh \tag{2.5.47}$$

Finally, the time-lag variable $M(t)$ is described by the following kernel integral:

$$M(t) = \int_{-\infty}^{t} \underbrace{\left\{0.1429\left(e^{\lambda_3(t-h)} - e^{\lambda_1(t-h)}\right) + 0.803e^{\lambda_3(t-h)}\right\}}_{k(t-h)} X(h)dh$$

$$= 0.1429\left(e^{\lambda_3 t} - e^{\lambda_1 t}\right)R(0) + e^{\lambda_3 t}M(0) \tag{2.5.48}$$

$$+ \int_0^t \left\{0.1429\left(e^{\lambda_3(t-h)} - e^{\lambda_1(t-h)}\right) + 0.803e^{\lambda_3(t-h)}\right\} X(h)dh.$$

Note that for $k_6 = 0$, the above equation is reduced to:

$$M(t) = e^{\lambda_3 t}M(0), \tag{2.5.49}$$

and there is no time-lag in the response of the enzyme because all the terms in the time-lag integral are practically equal to zero.

In Figure 2.5.5, the enzyme profile calculated with the time-lag approach, with a kernel described by Equation (2.5.48), is contrasted with that calculated by integrating the full set of eight dynamic equations. The kernel is turned on when $S_A < S_{Ac}$. Note that this critical point can be determined with the unstructured variables, and no knowledge of the process structure of enzyme induction and repression is used in the time-lag calculation. It is apparent that there is no visible difference between the two curves. Also shown in the same figure is the enzyme level as a function of time, calculated with Equation (2.5.49) for the entire duration. Since the dynamic equations for biomass, glucose, and galactose concentrations are unstructured, there is absolutely no difference in calculated results of these variables, whether a full set of equations is used or not.

The next example is the simulation of enzyme production in a batch mode. For reference purposes, Imanaka *et al.*'s experimental results and model prediction are reproduced in Figure 2.5.6. As in the continuous mode of operation, this author was unable to reproduce the same curves with the model and parameters provided by Imanaka *et al.*, although the overall features of the obtainable batch fermentation are comparable. The following is a list of the model parameters that were used that differ from the ones used in the previous example of continuous fermentation, due to the different operating conditions (temperature, nutrient composition, etc.)

Experimental Model Values		
μ_{mA}	0.190	hr^{-1}
μ_{mB}	0.162	hr^{-1}
K_{SA}	0.145	g/l
K_{SB}	0.307	g/l
Y_A	0.377	g/g cell
Y_B	0.361	g/g cell

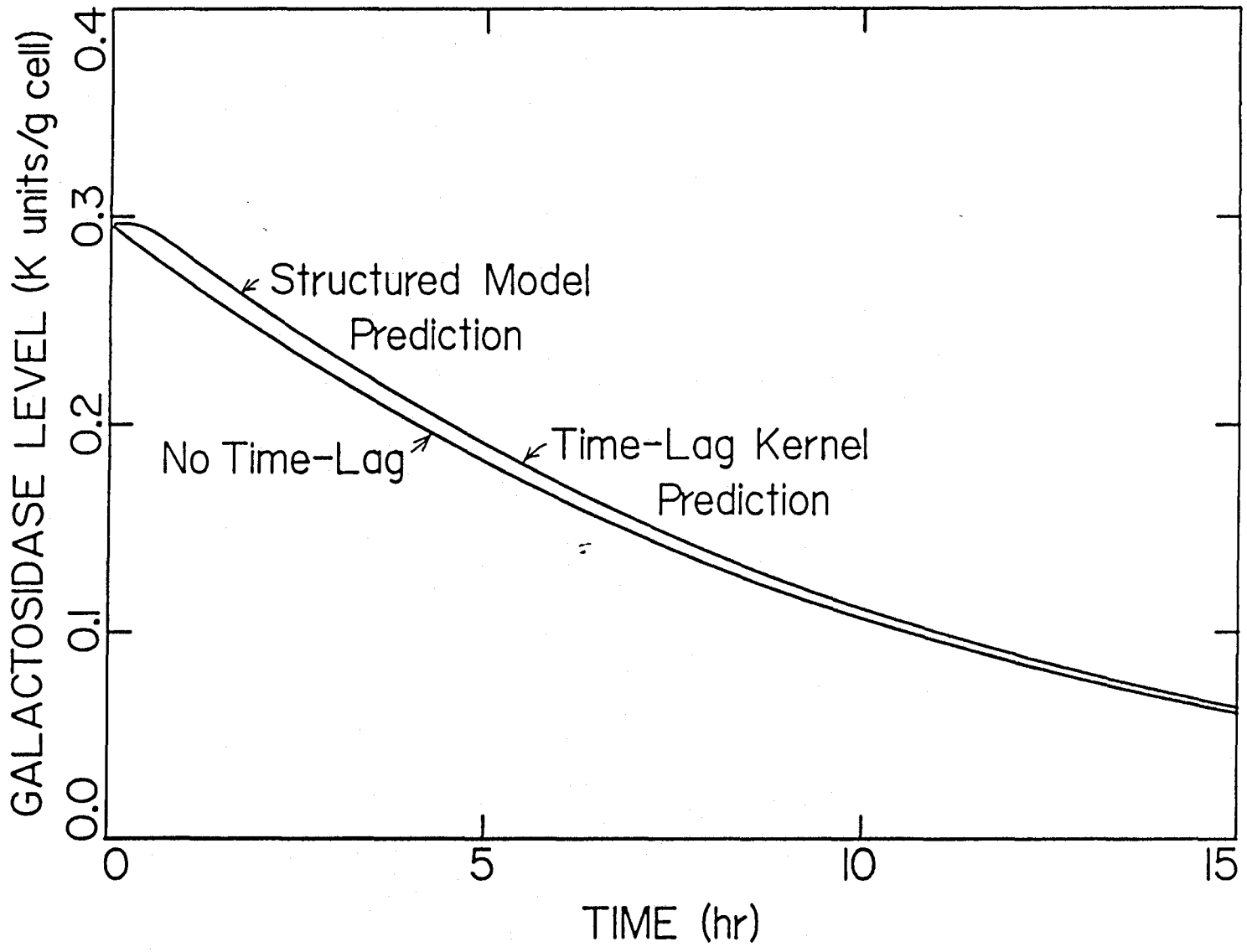


Figure 2.5.5. Comparison of the enzyme profiles calculated with Imanaka's structured model of enzyme production and the time-lag kernel approach after a shift up in the dilution rate from 1.40 hr^{-1} to 1.42 hr^{-1} at t=0 hr.

Undetermined Model Values		
k_8	6.67	K unit/(mg mRNA)-hr

Initial Conditions		
X_0	0.5	g/l
S_{A0}	10.	g/l
S_{B0}	3.	g/l
s_{Bi0}	0.	mg/(g cell)
r_0	0.91	mg/(g cell)
$\overline{rs_{Bi0}}$	0.	mg/(g cell)
m_0	0.	mg/(g cell)
e_0	0.	K units/(g cell)

The simulated results calculated with a full set of eight dynamic equations are plotted in Figures 2.5.7 and 2.5.8. The enzyme concentration for the time-lag approach, shown in Figure 2.5.9, is calculated based on the following dynamic equation:

$$\frac{de(t)}{dt} = k_8 m(t) - \mu e(t), \tag{2.5.50}$$

where the time-lag variable $m(t)$ is activated at the time when S_A becomes smaller than the critical glucose concentration of S_{Ac}. This critical time, denoted as t_c, can be determined from the macroscopic variables without the knowledge of the structure of enzyme production. The slight deviation is due to the fact that the enzyme profile determined by the full set of dynamic equations is based on the internal on/off mechanism of mRNA production, which depends on the value of $(r_c - r)$. This value is not available to an unstructured model that does not consider the internal mechanisms, and the closest on/off switch available to an unstructured model is the crossing of S_{Ac} by S_A. In view of the fact that only macroscopic

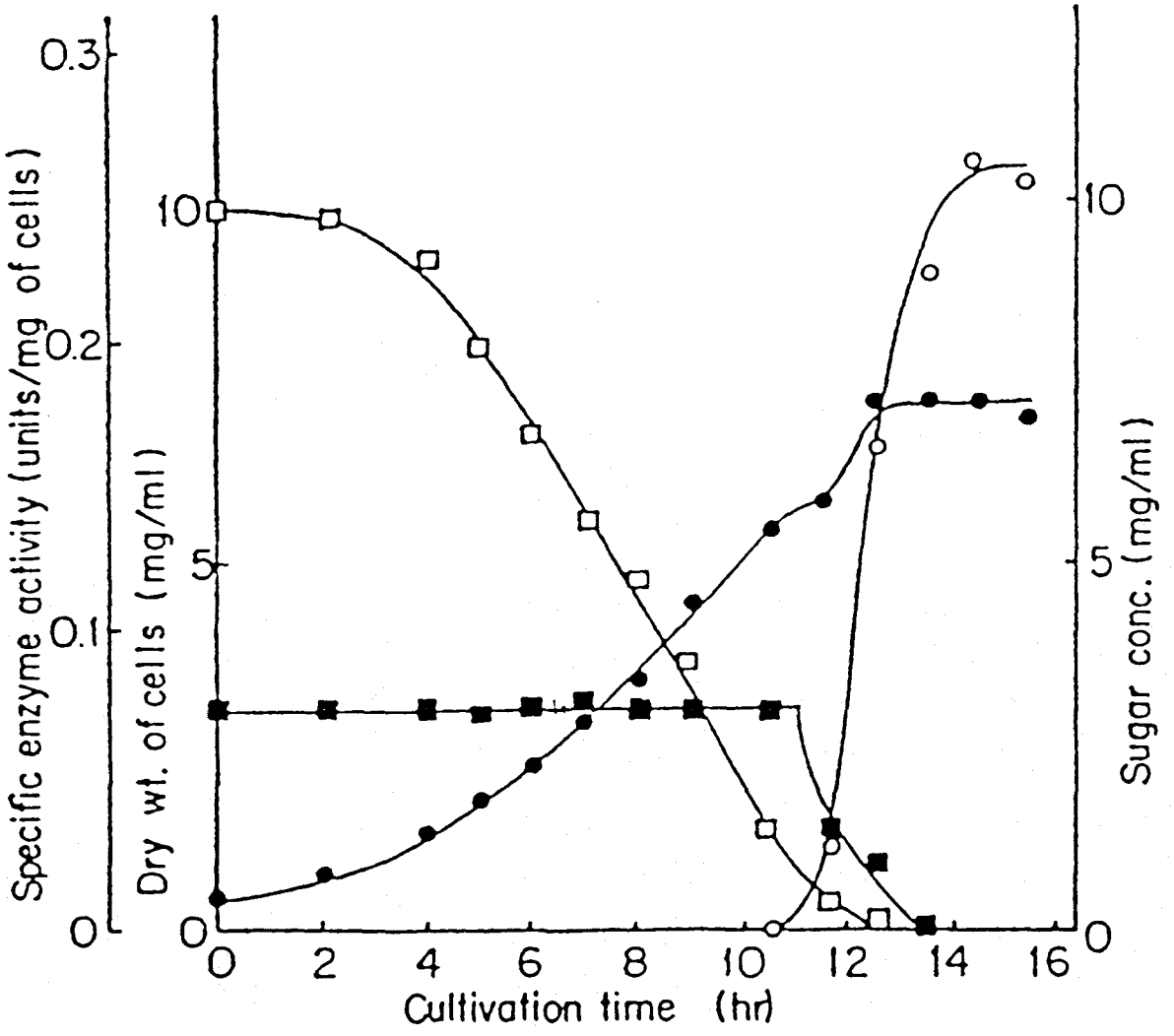


Figure 2.5.6. Time course of cell growth and α-galactosidase production in a mixture of glucose (1%) and galactose (0.3%). Reproduced from Imanaka *et al.* (1973).

variables are used to generate the concentration profiles, the agreement is quite good.

The equivalent time-lag kernel for the batch case can be derived from Equations (2.5.31.4)–(2.5.31.7). As before, instead of manipulating the entire set of equations simultaneously, it is beneficial to make some simplifications. Because U and k_1 are large, one can apply the quasi-steady state assumption to s_{Bi}. Furthermore, because K_{mB} is extremely small, one can assume that $\frac{G_B S_B}{K_{mB}+S_B} = G_B$ for $S_B > K_{mB}$. This leads to:

$$0 = \frac{ds_{Bi}}{dt} = U(G_B - s_{Bi}) - k_1 s_{Bi} - \mu s_{Bi} \tag{2.5.51a}$$

$$s_{Bi} \approx \frac{UG_B}{U + k_1 + \mu} \approx \frac{UG_B}{U + k_1} = 2.5 \quad \text{for} \quad \begin{matrix} S_A < S_{Ac} \\ \& \; S_B > K_{mB} \end{matrix} \tag{2.5.51b}$$

With this simplification, a matrix equation similar to Equation (2.5.43) can be obtained:

$$\underbrace{\begin{bmatrix} \frac{dr(t)}{dt} \\ \frac{d\overline{rs_{Bi}}}{dt} \\ \frac{dm(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}_2(t)}{dt}} = \underbrace{\begin{bmatrix} -k_4 s_{Bi} - k_3 - \mu & k_5 & 0 \\ k_4 s_{Bi} & -k_5 - \mu & 0 \\ -k_6 & 0 & -k_7 - \mu \end{bmatrix}}_{\mathbf{A}_{22}(t)} \underbrace{\begin{bmatrix} r(t) \\ \overline{rs_{Bi}}(t) \\ m(t) \end{bmatrix}}_{\mathbf{x}_2(t)} + \underbrace{\begin{bmatrix} k_2 \\ 0 \\ k_6 r_c \end{bmatrix}}_{\tilde{\mathbf{g}}(t)} \tag{2.5.52}$$

The dynamic matrix, after substituting model parameters, is:

$$\mathbf{A}_{22} = \begin{bmatrix} -1.25 - \mu & 0.0001 & 0 \\ 0.25 & 0.0001 - \mu & 0 \\ -1 & 0 & -8 - \mu \end{bmatrix} \approx \begin{bmatrix} -1.25 - \mu & 0 & 0 \\ 0.25 & -\mu & 0 \\ -1 & 0 & -8 - \mu \end{bmatrix} \tag{2.5.53}$$

$$\begin{aligned} 0 = \det|\mathbf{A}_{22} - \lambda \mathbf{I}| &= \begin{vmatrix} -1.25 - \mu - \lambda & 0 & 0 \\ 0.25 & -\mu - \lambda & 0 \\ -1 & 0 & -8 - \mu - \lambda \end{vmatrix} \\ &= (-1.25 - \mu - \lambda)(-\mu - \lambda)(-8 - \mu - \lambda) \end{aligned} \tag{2.5.54}$$

$$\Longrightarrow \qquad \lambda_1 = -1.25 - \mu \qquad \lambda_2 = -\mu \qquad \lambda_3 = -8 - \mu \tag{2.5.55a}$$

$$\Longrightarrow \quad \mathbf{T} = \begin{bmatrix} -6.75 & 0 & 0 \\ 1.35 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}^{-1} = \begin{bmatrix} -\frac{1}{6.75} & 0 & 0 \\ 0.2 & 1 & 0 \\ \frac{1}{6.75} & 0 & 1 \end{bmatrix} \tag{2.5.55b}$$

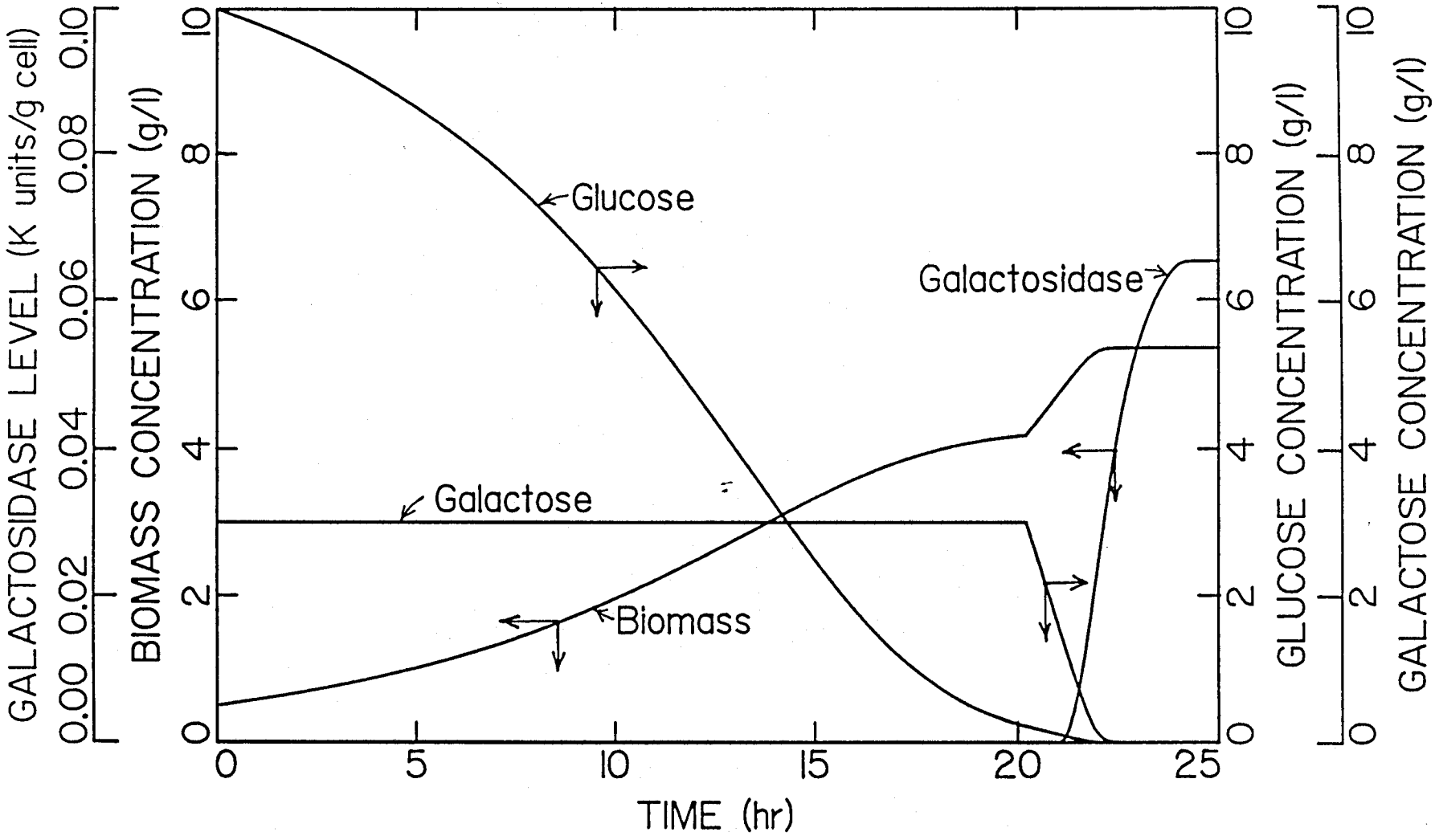


Figure 2.5.7. Time profile of macroscopic state variables (biomass, glucose, galactose, and enzyme) in the batch production of α-galactosidase. See text for the values of model parameters and initial conditions.

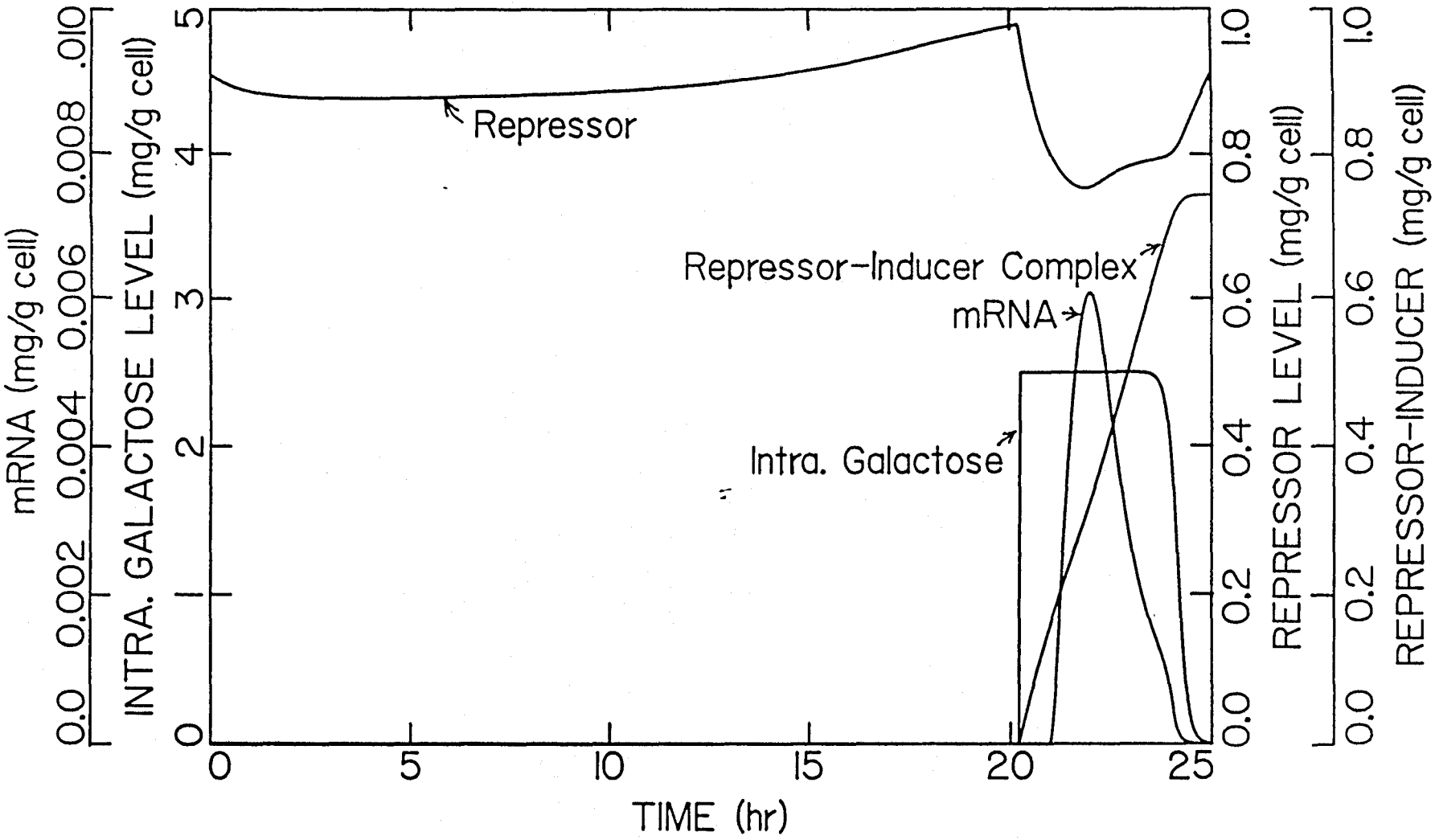


Figure 2.5.8. Time profile of intracellular state variables (intracellular galactose, repressor, repressor-inducer complex, and mRNA) in the batch production of α-galactosidase. See text for the values of model parameters and initial conditions.

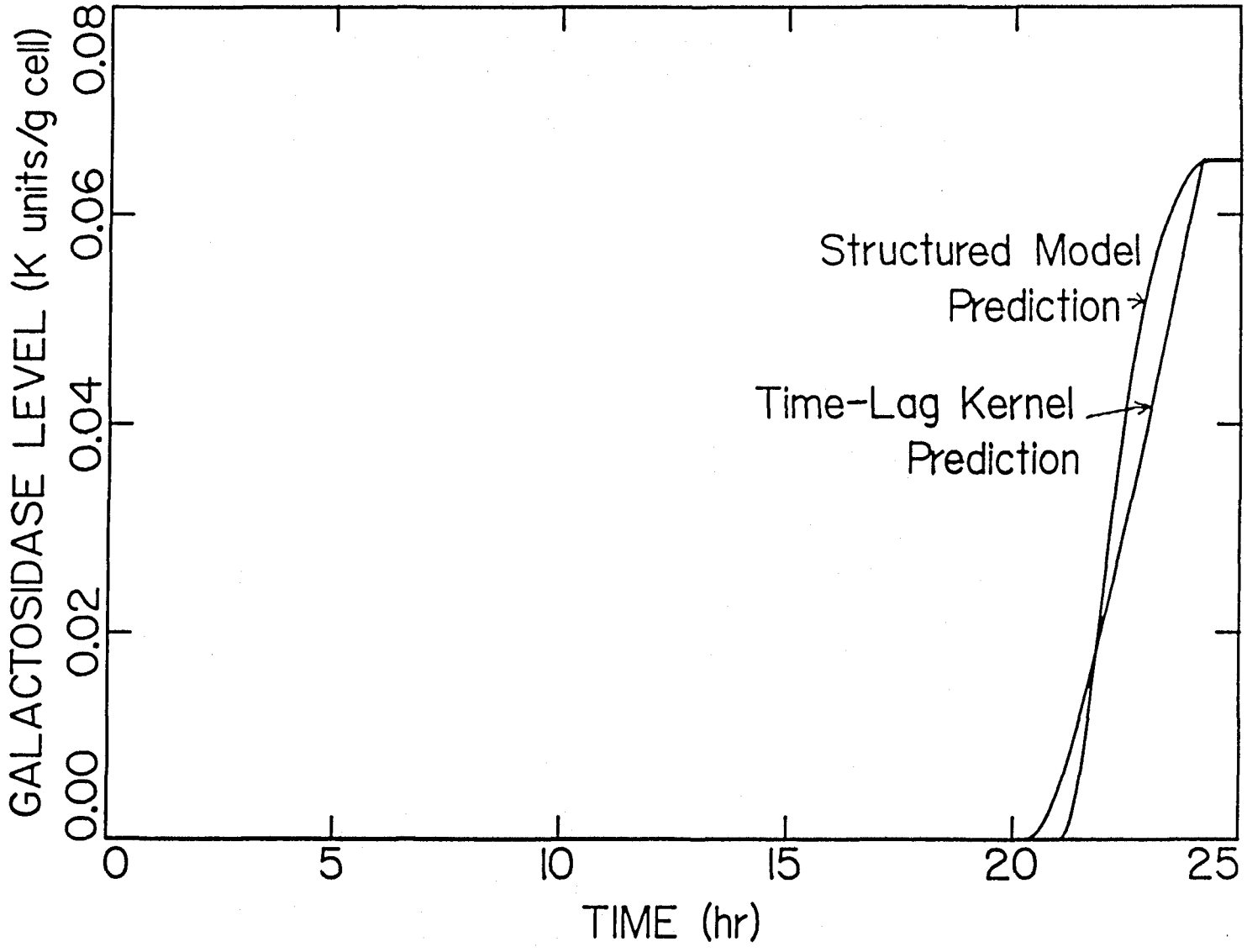


Figure 2.5.9. Comparison of the enzyme profiles calculated with Imanaka's structured model of enzyme production and the time-lag kernel for a batch fermentor.

$$\mathbf{K}(t) = e^{\mathbf{A}_{22}t} = \underbrace{\begin{bmatrix} -6.75 & 0 & 0 \\ 1.35 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} e^{\lambda_1 t} & 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 \\ 0 & 0 & e^{\lambda_3 t} \end{bmatrix}}_{e^{\mathbf{J}t}} \underbrace{\begin{bmatrix} -\frac{1}{6.75} & 0 & 0 \\ 0.2 & 1 & 0 \\ \frac{1}{6.75} & 0 & 1 \end{bmatrix}}_{\mathbf{T}^{-1}} \tag{2.5.55c}$$

$$= \begin{bmatrix} e^{\lambda_1 t} & \approx 0 & 0 \\ 0.2(e^{\lambda_2 t} - e^{\lambda_1 t}) & e^{\lambda_2 t} & 0 \\ \frac{1}{6.75}(e^{\lambda_3 t} - e^{\lambda_1 t}) & \approx 0 & e^{\lambda_3 t} \end{bmatrix}$$

Finally, the time-lag variable for the batch fermentation is:

$$m(t) = \begin{cases} 0 & \text{for } t \le t_c \\ m_{t_c} + \int_{t_c}^{t} k(t-h)X(h)dh & \text{for } t_c < t \end{cases}, \tag{2.5.56}$$

where the kernel $k(t)$ is described by:

$$\begin{aligned} k(t) &= \frac{1}{6.75}\left(e^{\lambda_3 t} - e^{\lambda_1 t}\right) + 0.803e^{\lambda_3 t} \\ &= 0.9511e^{\lambda_3 t} - 0.1481e^{\lambda_1 t} \end{aligned} \tag{2.5.57a}$$

and the pre-integral factor m_{t_c} used to absorb all the information between $t = -\infty$ and $t = t_c$ is:

$$\begin{aligned} m_{t_c} &= \frac{1}{6.75}\left(e^{\lambda_3 t} - e^{\lambda_1 t}\right) r(t_c) + e^{\lambda_3 t}\overset{0}{\cancel{r}}(t_c) \\ &\approx \frac{1}{6.75}\left(e^{\lambda_3 t} - e^{\lambda_1 t}\right) r_c. \end{aligned} \tag{2.5.57b}$$

Tanner's Model: (Tanner, *Biotechnol. Bioeng.*, **12**, 831, 1970.) Another enzymatically based structured model proposed to describe the gluconic acid fermentation will be reduced to an unstructured equivalent with the inclusion of a time-lag kernel. Readers are referred to the original paper for a detailed discussion on the mechanism. The set of ten dynamic equations that comprise this model are listed below:

Biomass: $$\frac{d(X)}{dt} = k_3(RS) = \underbrace{\frac{k_3(RS)}{(R) + (RS)}}_{y} X \tag{2.5.58.1}$$

Substrate: $$\frac{d(S)}{dt} = -k_1(S)(R) + k_2(RS) - k_6(S)(E_1) + k_7(E_1S) \tag{2.5.58.2}$$

Ribosomes: $$\frac{d(R)}{dt} = -k_1(S)(R) + (k_2 + 2k_3 + k_4 + k_5)(RS) \qquad (2.5.58.3)$$

R-S Complex: $$\frac{d(RS)}{dt} = k_1(S)(R) - (k_2 + k_3 + k_4 + k_5)(RS) \qquad (2.5.58.4)$$

Enzyme 1: $$\frac{d(E_1)}{dt} = k_4(RS) - k_6(S)(E_1) + (k_7 + k_8)(E_1S) \qquad (2.5.58.5)$$

Enzyme 2: $$\frac{d(E_2)}{dt} = k_5(RS) - k_9(L)(E_2) + (k_{10} + k_{11})(E_2L) \qquad (2.5.58.6)$$

E1-S Complex: $$\frac{d(E_1S)}{dt} = k_6(S)(E_1) - (k_7 + k_8)(E_1S) \qquad (2.5.58.7)$$

E2-Lactone Complex: $$\frac{d(E_2L)}{dt} = k_9(L)(E_2) - (k_{10} + k_{11})(E_2L) \qquad (2.5.58.8)$$

Gluconolactone: $$\frac{d(L)}{dt} = k_8(E_1S) - k_9(L)(E_2) + k_{10}(E_2L) - k_{12}(L) \qquad (2.5.58.9)$$

Gluconic Acid: $$\frac{d(G)}{dt} = k_{11}(E_2L) + k_{12}(L) \qquad (2.5.58.10)$$

Note that for small values of k_6 and k_7, the first four equations become decoupled from the rest. And it is this simplified special case that is analyzed here. Furthermore, this model is formulated in such a way that the first equation for biomass is redundant; the biomass in this model is proportional to the sum of (R) and (RS):

$$X = K\Big[(R) + (RS)\Big] \qquad (2.5.59)$$

At the beginning of a batch fermentation, $(RS) = 0$. Thus, the observed specific growth rate $y(t) = \frac{k_3(RS)(t)}{(R)(t)+(RS)(t)}$ is zero initially. Because (RS) gradually increases during the course of a batch fermentation, the initial lag phase can be described. At the end of a batch fermentation it settles back to zero, corresponding to the stationary phase. Thus, (RS) is the equivalent of the previously mentioned

"critical enzyme" that directly controls the specific growth rate. This interpretation is consistent with Equation (2.5.58.1)

Although one can rigorously partition the above ten variables into x_1 and x_2 and consider the entire set of dynamic equations in deriving the equivalent kernel function, such full blown treatment is often not necessary. For example, the kernel for a batch fermentation can be adequately approximated by considering only Equation (2.5.58.4), which can be solved to yield:

$$\begin{aligned}(RS)(t) &= \int_{-\infty}^{t} e^{-(k_0+k_3)(t-h)} k_1 S(h) R(h) dh \\ &= \underbrace{(RS)_0}_{0} e^{-(k_0+k_3)t} + \int_{0}^{t} e^{-(k_0+k_3)(t-h)} k_1 S(h) R(h) dh \\ &= \int_{0}^{t} e^{-(k_0+k_3)(t-h)} k_1 S(h) R(h) dh\end{aligned} \tag{2.5.60}$$

where $(RS)_0 = 0$ can be assumed, and, for simplicity, k_2, k_4, and k_5 have been combined into one constant k_0.

$$k_0 = k_2 + k_4 + k_5 \tag{2.5.61}$$

In order to formulate the time-lag kernel in the specific growth rate, one can massage the above kernel integral into the equation for $y(t)$.

$$\begin{aligned}y(t) &= \frac{k_3(RS)(t)}{(R)(t) + (RS)(t)} \\ &\approx \frac{k_3}{(R)_t}(RS)(t) \\ &= \frac{k_3}{(R)_t} \int_{0}^{t} e^{-(k_0+k_3)(t-h)} k_1 S(h) R(h) dh \\ &\approx \int_{0}^{t} (k_0 + k_3) e^{-(k_0+k_3)(t-h)} \mu(h) dh\end{aligned} \tag{2.5.62}$$

The reduction to a Monod cell growth equation used in the last step is discussed extensively in the original paper. Thus, the kernel associated with the specific growth rate is a 0th-order exponential function with a lag time constant of (k_0+k_3):

$$k(t) = (k_0 + k_3) e^{-(k_0+k_3)t} \tag{2.5.63}$$

The following model parameters are used to simulate the batch response of the Tanner model:

Tanner Model Values	
k_0	3.
k_1	1.
k_2	1.
k_3	1.

Initial Conditions	
$(S)_0$	1.50
$(R)_0$	0.05
$(RS)_0$	0.00

The simulated batch response of the model is shown in Figure 2.5.10, and the time-lag kernel calculated directly from the response of Figure 2.5.10 by the Fourier transform method is shown in Figure 2.5.11 along with the derived kernel of Equation (2.5.63). The agreement is rather good, considering the fact that only one dynamic equation is used to approximate the kernel function.

The results of batch simulation using the simplified 0th-order kernel just derived are shown in Figure 2.5.12 for the biomass and substrate concentrations and in Figures 2.5.13 and 2.5.14 for the intrinsic and observed specific growth rates, respectively. It can be seen that the time-lag kernel gives results that are very close to the structured model predictions, which is expected in view of the close agreement of the kernel. An even better fit can be obtained if the kernel function is derived with the entire set of dynamic equations.

The response of a continuous fermentor after a shift up or down in the dilution rate is simulated next. The time-lag kernel for a continuous fermentor can be

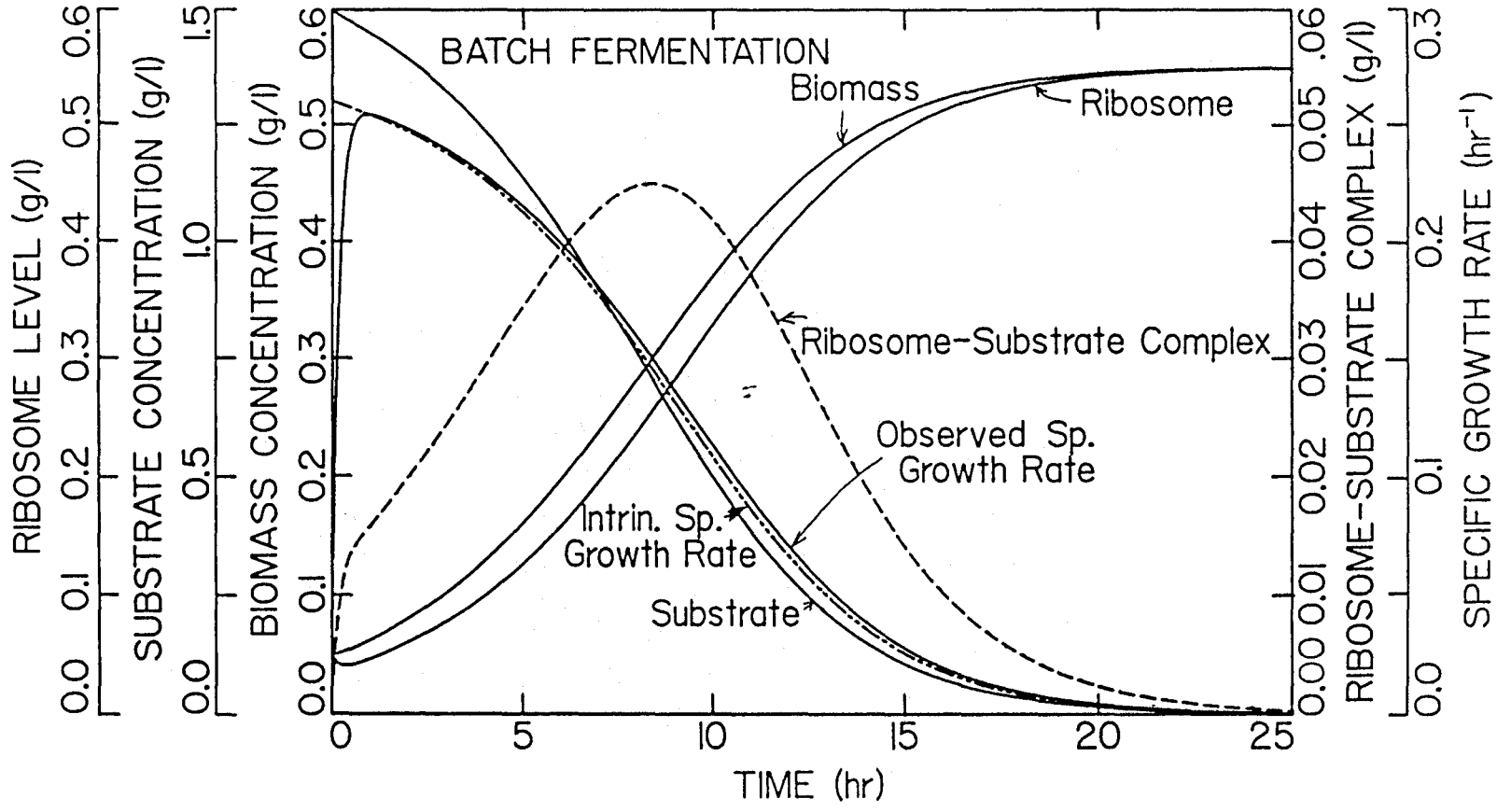


Figure 2.5.10. Dynamic simulation of a batch fermentation with Tanner's model.

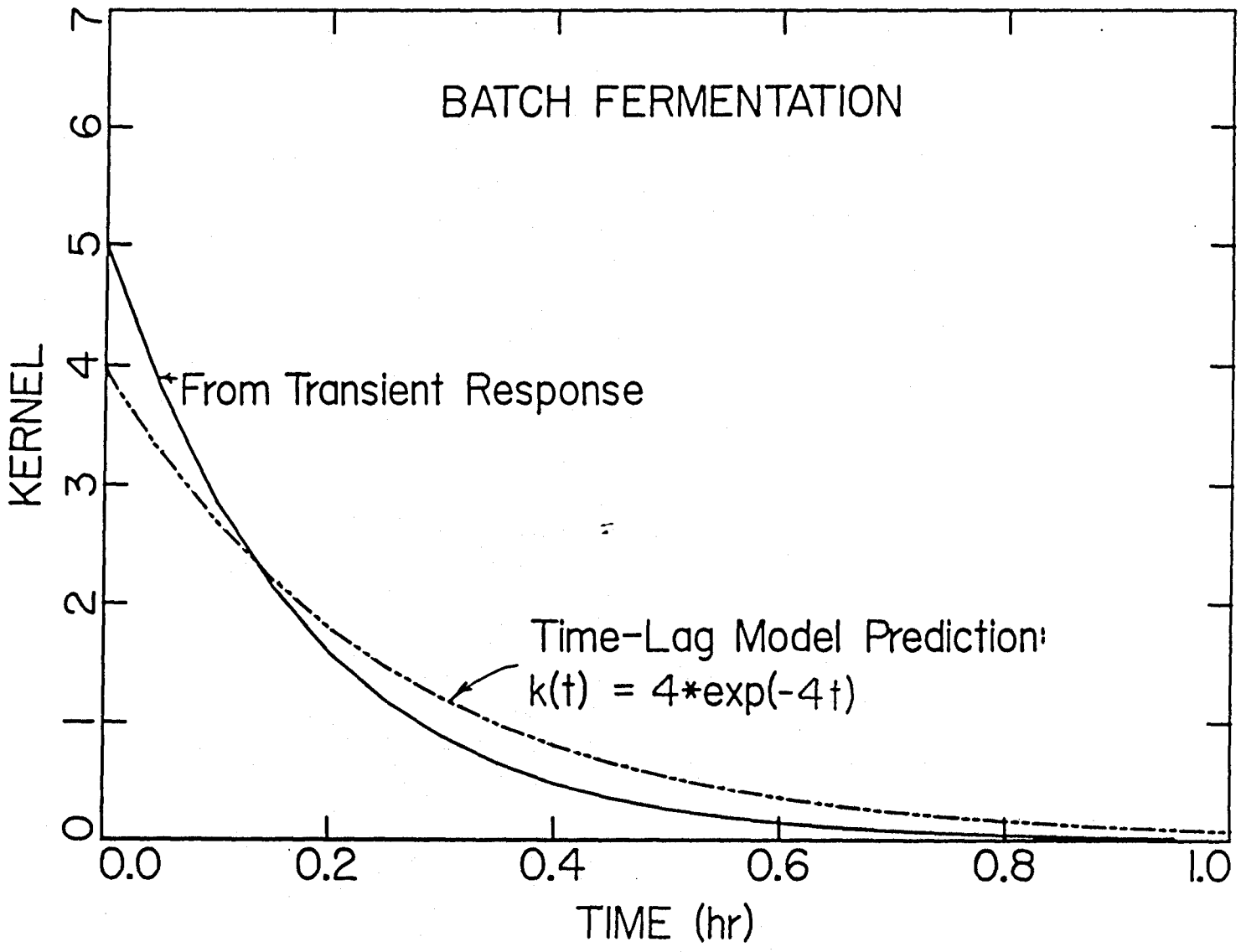


Figure 2.5.11. Comparison of the time-lag kernel calculated directly from the simulated batch response and that approximated from the model equations ($k(t) = (k_0 + k_3)e^{-(k_0+k_3)t}$).

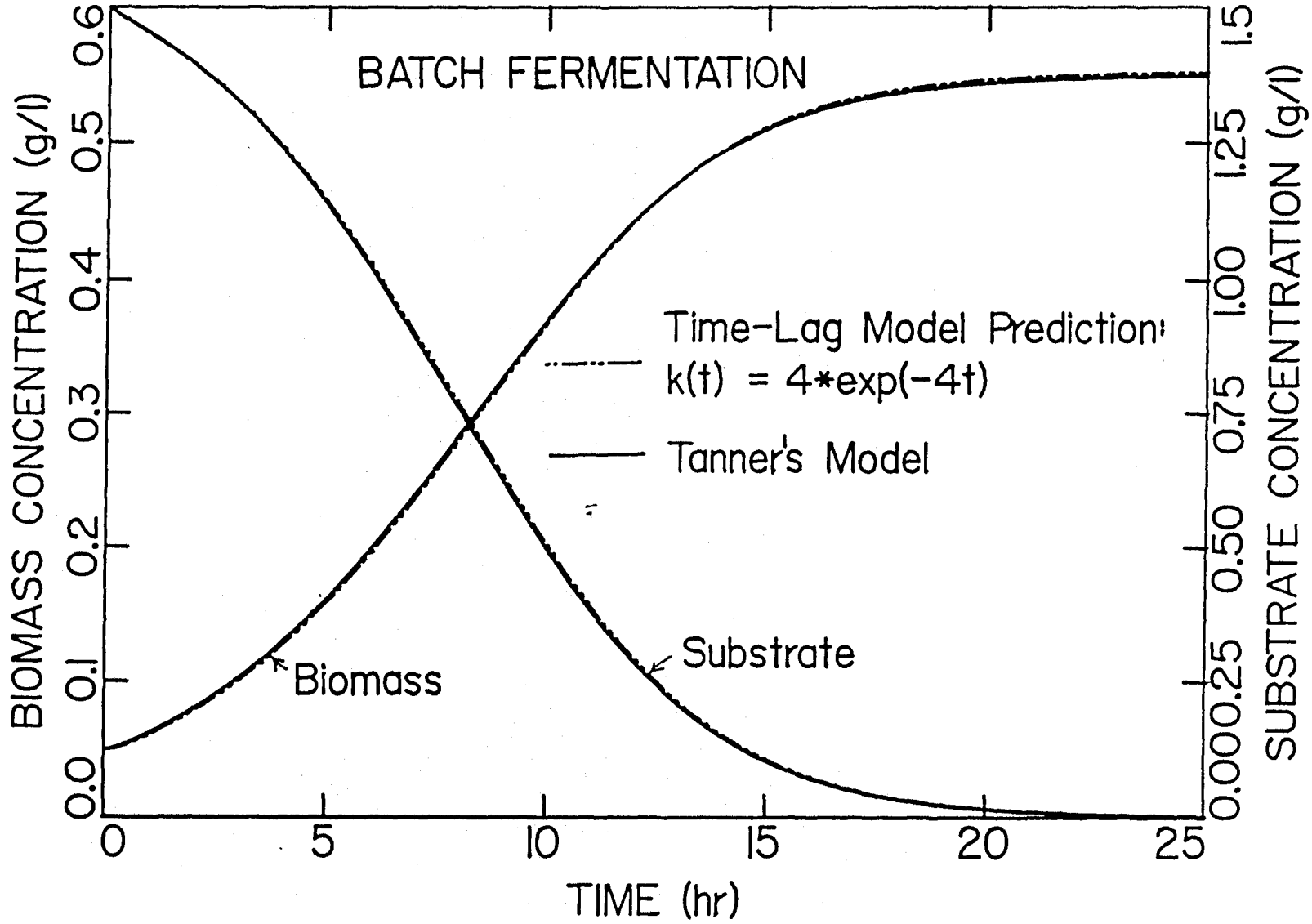


Figure 2.5.12. Comparison of the biomass and substrate concentrations simulated by Tanner's model and those predicted by the time-lag kernel approach.

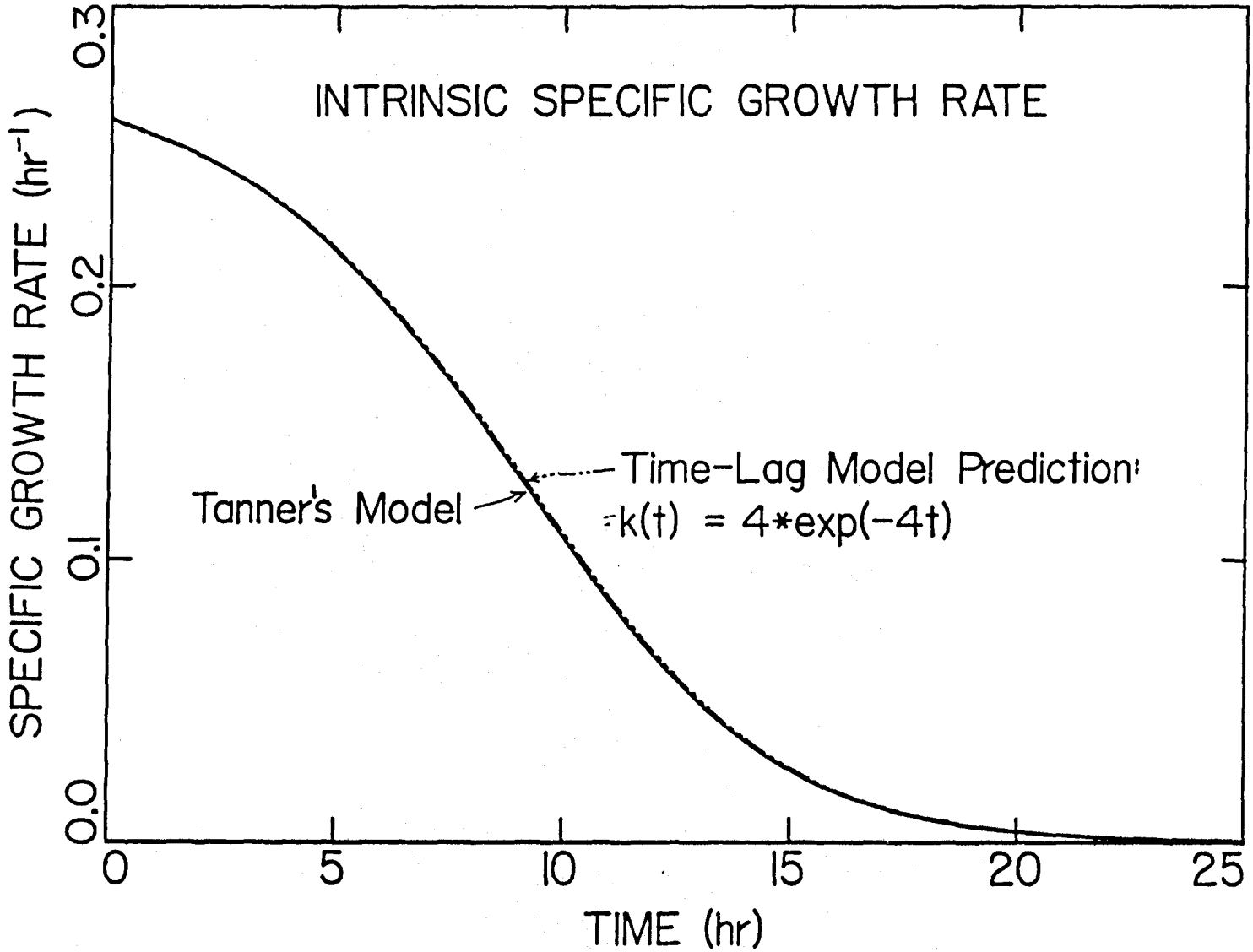


Figure 2.5.13. Comparison of the intrinsic specific growth rate profile calculated by Tanner's model and that predicted by the time-lag kernel approach.

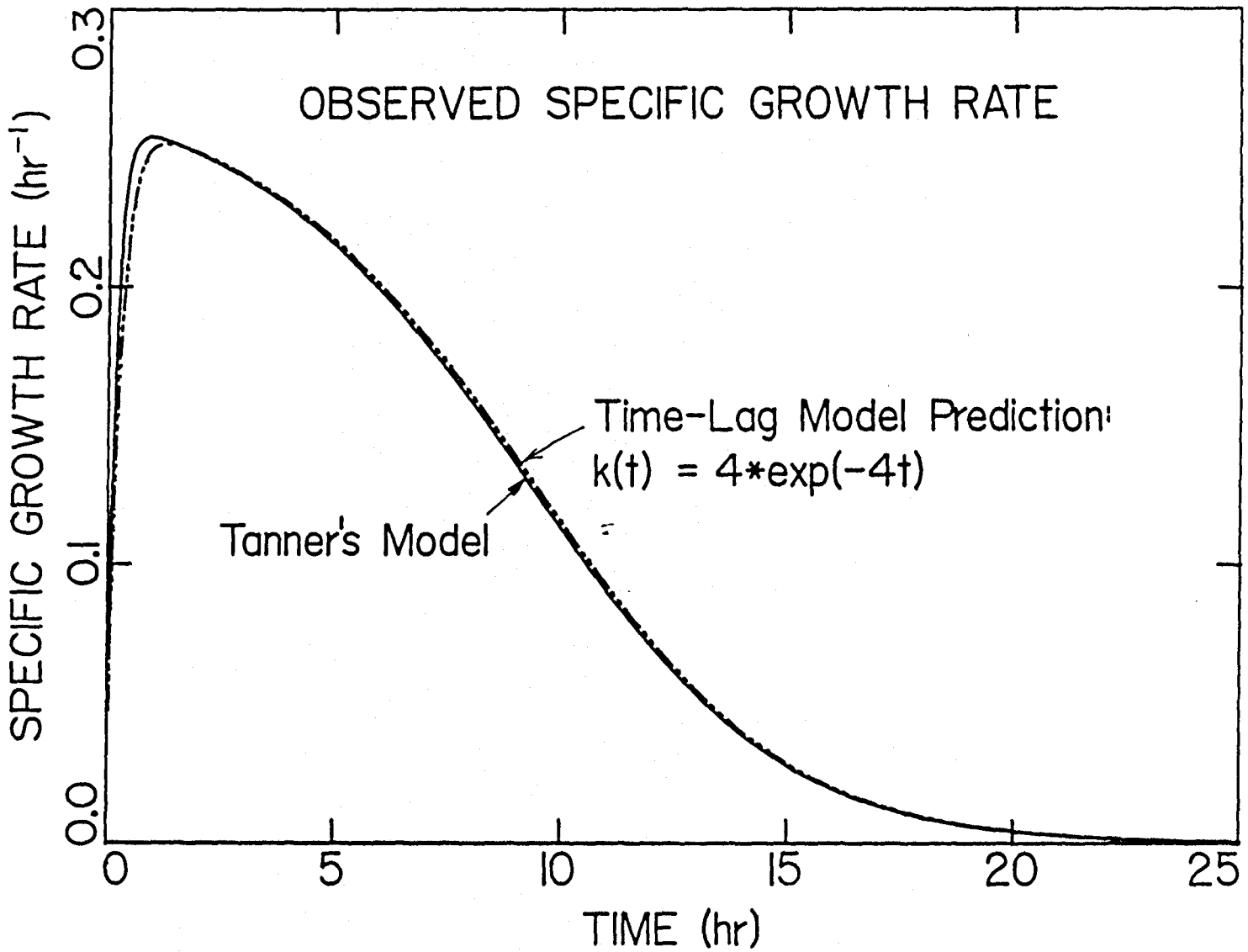


Figure 2.5.14. Comparison of the observed specific growth rate profile calculated by Tanner's model and that predicted by the time-lag kernel approach.

obtained by first partitioning (X) & (S) into $\mathbf{x}_1$ and (R) & (RS) into $\mathbf{x}_2$. The dynamic equations for $(R)(t)$ and $(RS)(t)$ are then linearized around the steady-state values.

$$\underbrace{\begin{bmatrix} \frac{d(R)(t)}{dt} \\ \frac{d(RS)(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}_2(t)}{dt}} = \underbrace{\begin{bmatrix} -k_1(S)_{ss} - D & k_0 + 2k_3 \\ -k_1(S)_{ss} & -(k_0 + k_3) - D \end{bmatrix}}_{\mathbf{A}_{22}(t)} \underbrace{\begin{bmatrix} (R)(t) \\ (RS)(t) \end{bmatrix}}_{\mathbf{x}_2(t)} + \underbrace{\begin{bmatrix} -k_1(R)_{ss}S(t) \\ k_1(R)_{ss}S(t) \end{bmatrix}}_{\tilde{\mathbf{g}}(t)} \tag{2.5.64}$$

It suffices to state that $e^{\mathbf{A}_{22}t}$ can be obtained as before after performing an eigenvalue – eigenvector analysis. They are 0th-order kernels:

Shift Up/Down	Kernel
D=0.2 → 0.3 hr^{-1}:	$k(t) = 5.25e^{-5.25t}$
D=0.3 → 0.2 hr^{-1}:	$k(t) = 6.14e^{-6.14t}$
D=0.2 → 0.3 → 0.2 hr^{-1}:	$k(t) = 5.67e^{-5.67t}$

The time constants are slightly different for each case, reflecting the fact that the original dynamic equations are not perfectly linear and that the quasi-linearization constants differ slightly at different dilution rates.

The fermentor response after a shift-up in the dilution rate is shown in Figure 2.5.15, and the response after a shift-down is shown in Figure 2.5.16. The steady-state values before the disturbance are used as the initial conditions.

Steady-State Values		
	D=0.2 hr^{-1}	D=0.3 hr^{-1}
$(S)_{ss}$	1.05000	1.84286
$(R)_{ss}$	2.23750	1.73030
$(RS)_{ss}$	0.55938	0.74156

The time-lag kernels calculated from the response curves for each of these cases, shown in Figures 2.5.17, 2.5.18, and 2.5.19, are compared with those obtained as a result of model reduction. Because the agreement between the actual kernel and the approximated kernel is once again quite good, the time-lag kernel approach can be used to simulate the transient response of a continuous fermentor.

In summary, one of the attractions of using a complicated structured model is that is can be used to predict the bioreactor behavior under various operating conditions, provided that the kinetic steps are properly identified and all the model parameters are correctly assigned. Through literature examples in this section, it is shown that some of the dynamic steps in a structured model can be eliminated without seriously affecting the predicted output. Furthermore, a structured model can be reduced to an unstructured model with the difference being absorbed by the time-lag kernel. In model translation/reduction, it is often convenient to quasi-linearize a set of nonlinear differential equations. The next logical step is to analyze the eigenvalue and eigenvector of the linearized dynamic matrix $\mathbf{A}(t)$. Such an analysis can yield useful information on the relative time scales of various processes. After grouping variables properly according to the process time constants, one can simplify and reduce the dimension of the system by retaining only the first few most important modes. Eliminating the remaining nonsignificant modes, like pruning a tree, helps to clarify the main feature of the model.

Parametric vs. Non-Parametric Modeling Approaches:

In conjunction with the discussion on the connection between a structured model and an unstructured model, it should be noted that our time-lag kernel modeling approach can also be viewed as a combination of other two opposing modeling approaches. One such example is the classification of models based on

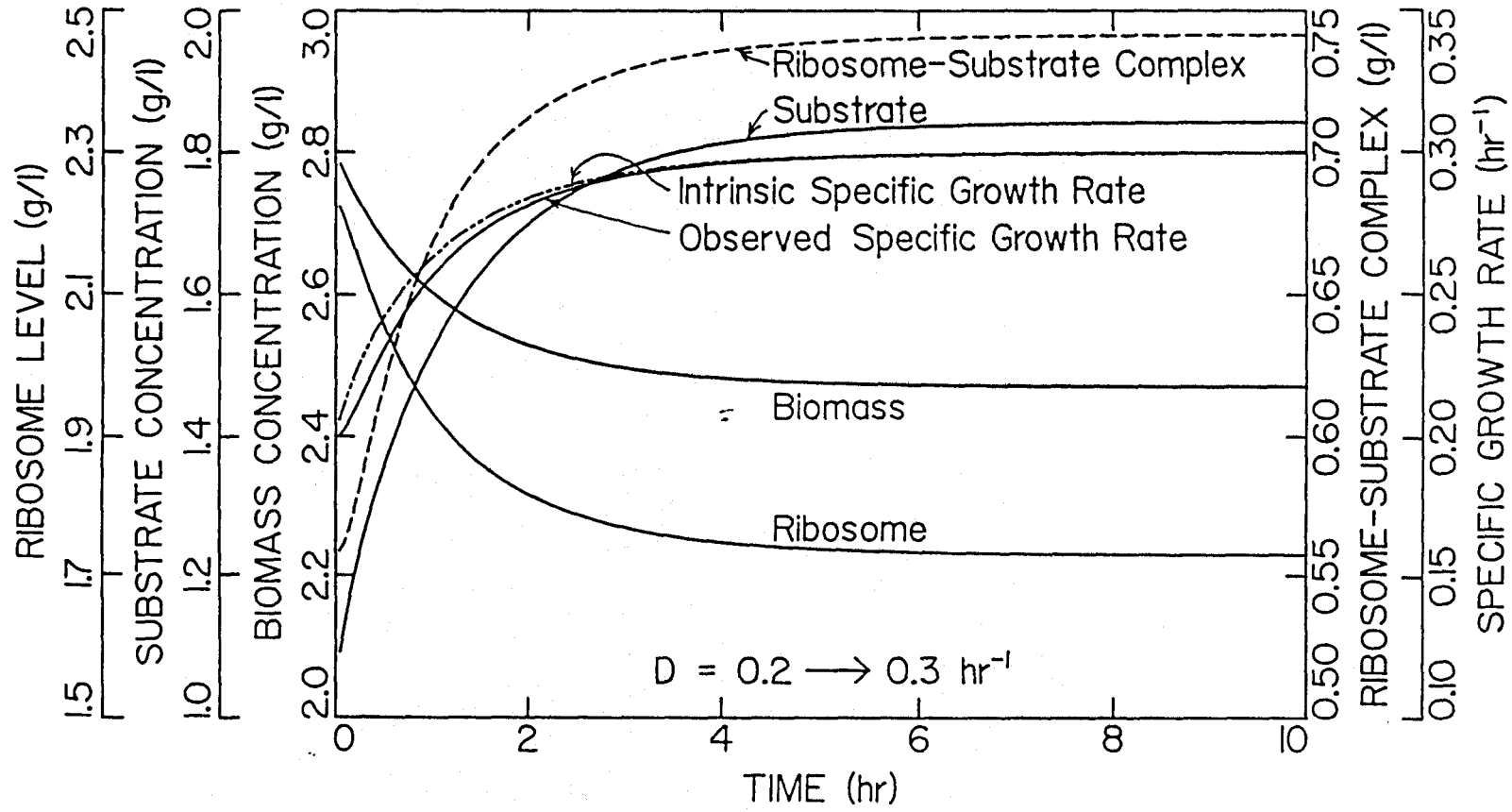


Figure 2.5.15. Simulated transient response of a continuous fermentor after a shift-up in the dilution rate (D=0.2 → 0.3 hr^{-1}) at t=0 hr with Tanner's model.

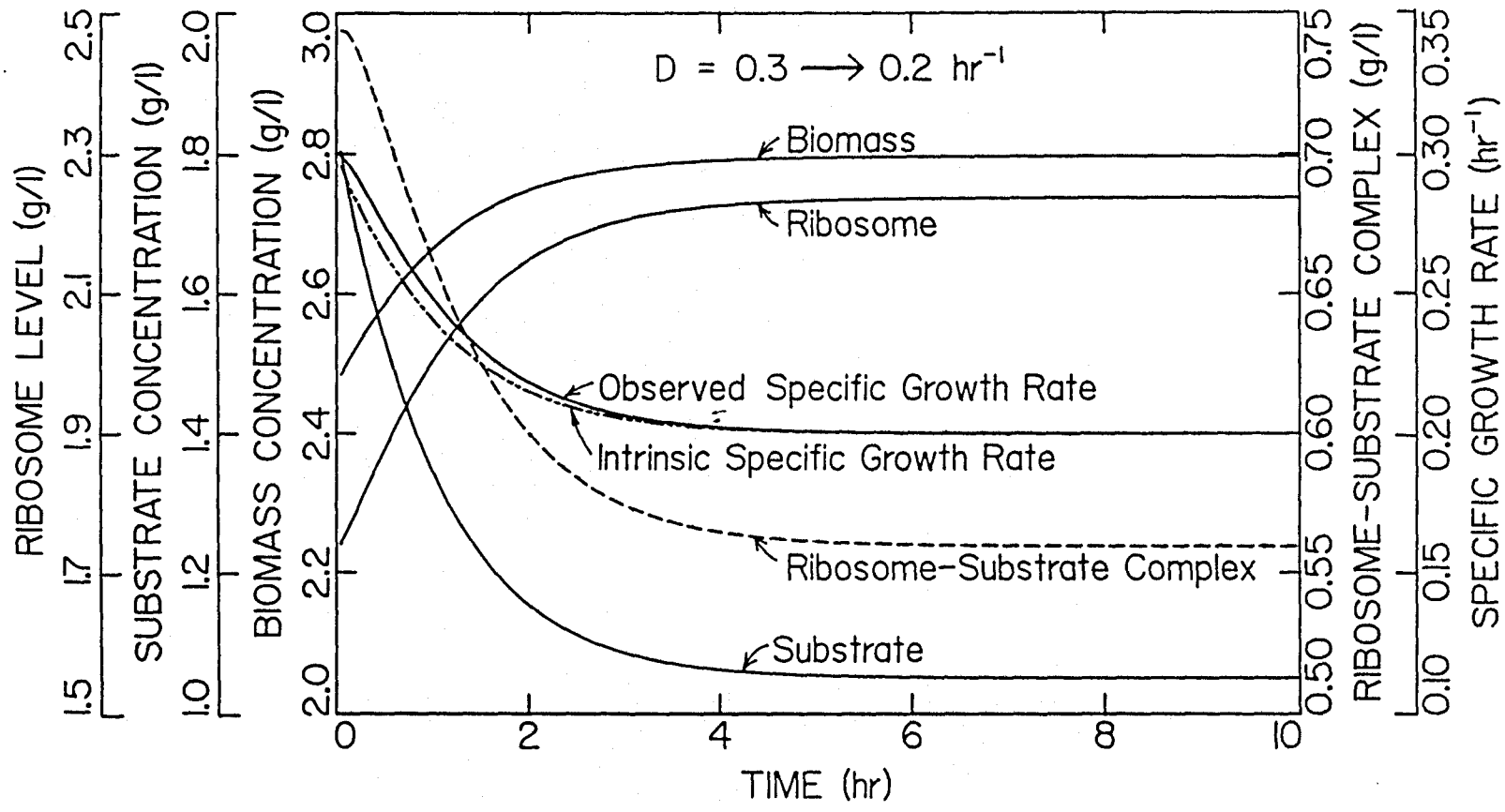


Figure 2.5.16. Simulated transient response of a continuous fermentor after a shift-down in the dilution rate (D=0.3 → 0.2 hr^{-1}) at t=0 hr with Tanner's model.

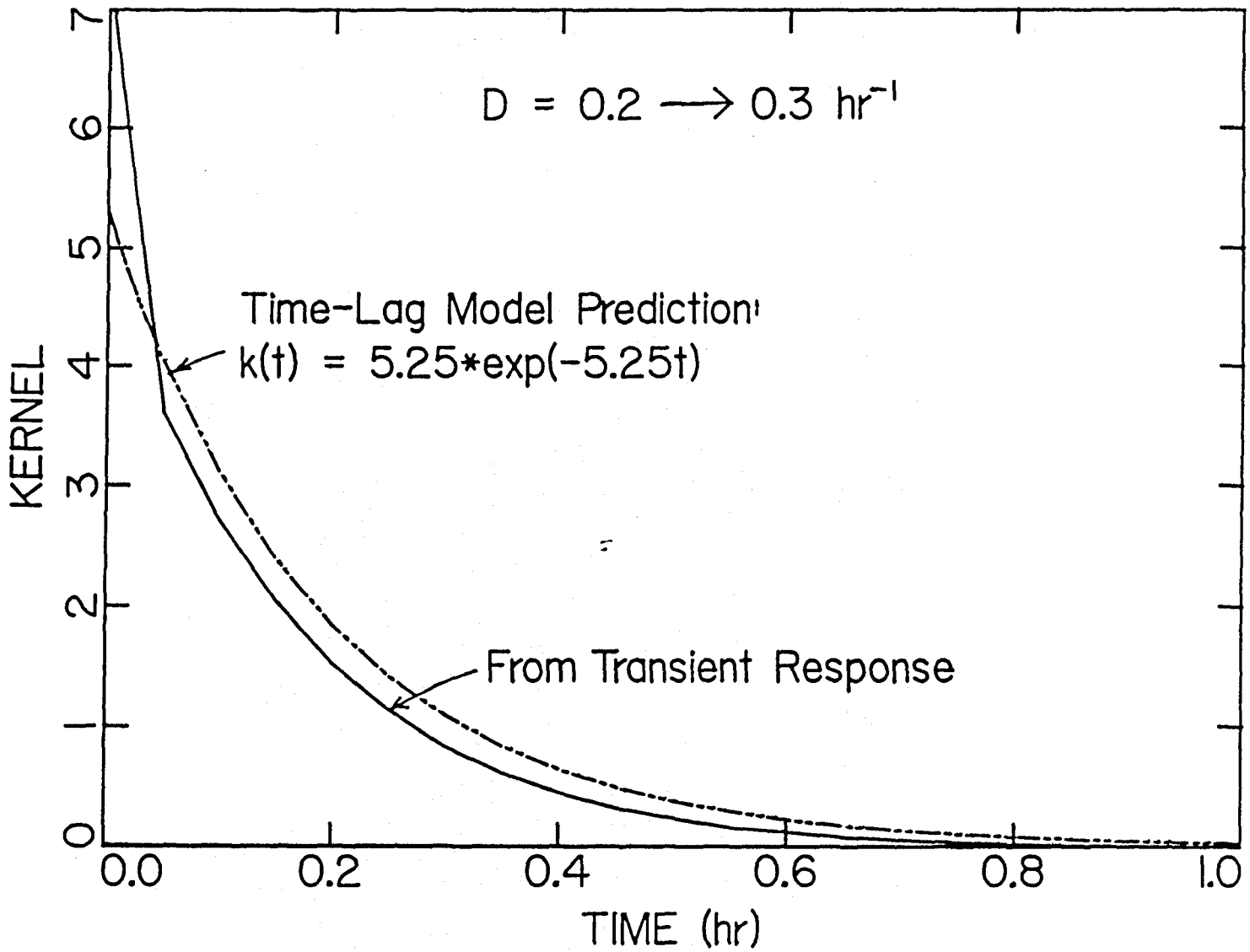


Figure 2.5.17. Comparison of the time-lag kernel calculated directly from the simulated chemostat response with D=0.2 → 0.3 hr^{-1} and that approximated from Tanner's model.

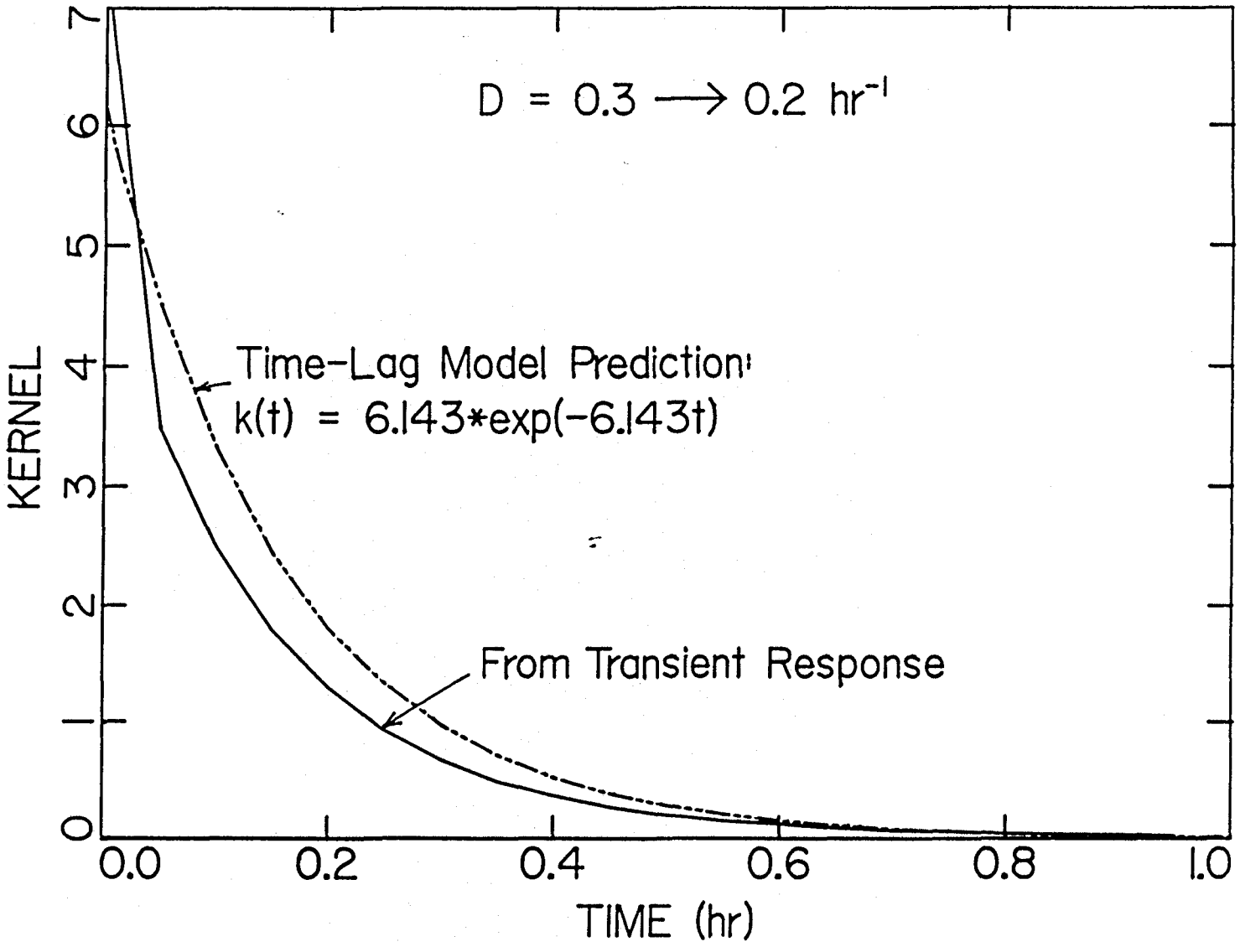


Figure 2.5.18. Comparison of the time-lag kernel calculated directly from the simulated chemostat response with D=0.3 → 0.2 hr^{-1} and that approximated from Tanner's model.

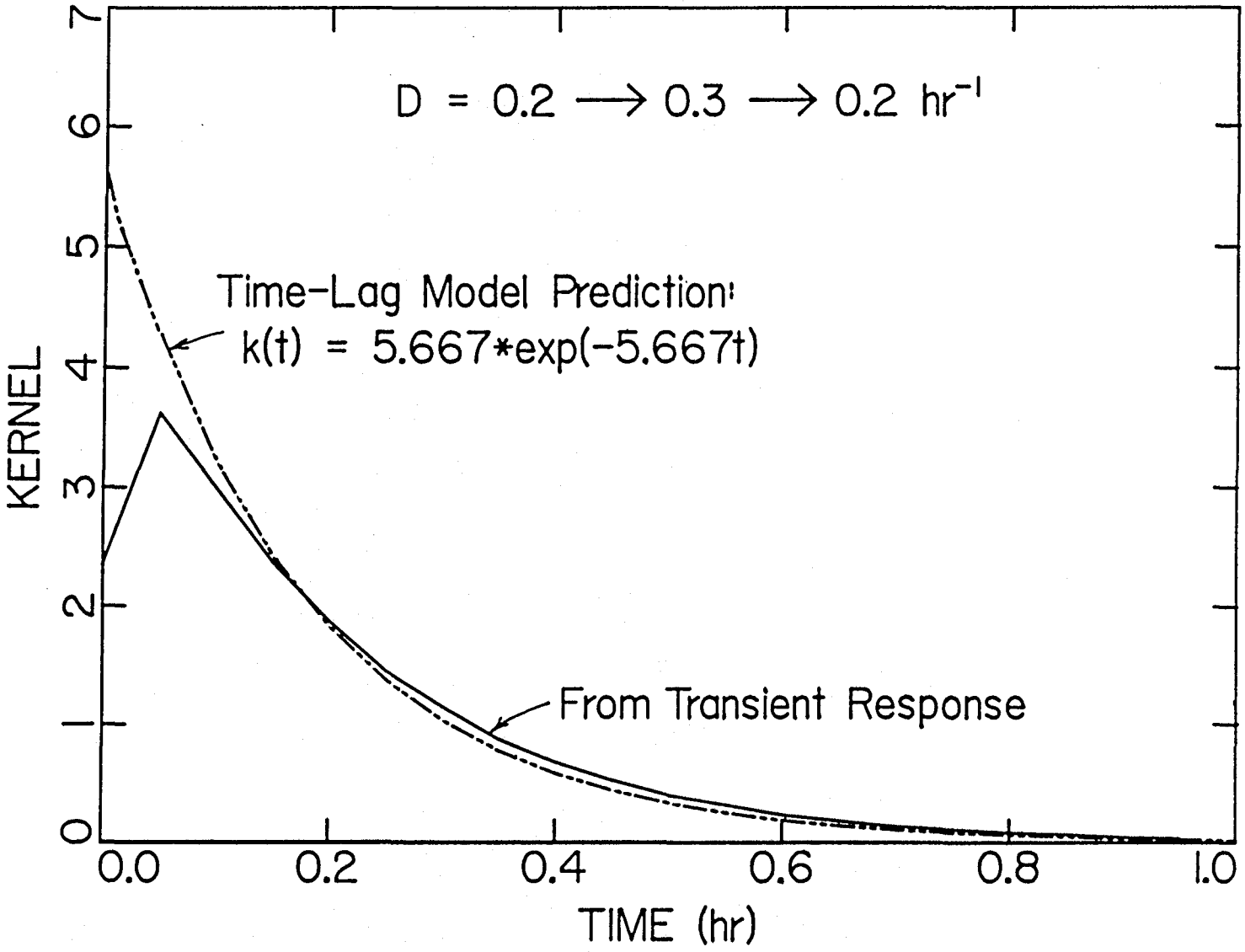


Figure 2.5.19. Comparison of the time-lag kernel calculated directly from the simulated chemostat response with D=0.2 → 0.3 → 0.2 hr^{-1} and that approximated from Tanner's model.

the presence or absence of differential and algebraic equations in the description of the process dynamics.

One of the methods of model classification is based on the representation of one's knowledge. A model can be classified as *parametric* if a *parameter space* is used to describe the process dynamics. In this approach, one's knowledge about the system under consideration is translated into a set of mathematical equations in terms of differential dynamic relationships supplemented, if necessary, by algebraic constitutive relationships. The output of the system is completely determined once one is supplied with the model parameters, initial conditions, and forcing functions. Properly viewed, both the initial conditions and forcing functions may also be considered as additional parameters. The dimensionality of the description in a parametric model is finite.

On the other hand, a *non-parametric model* of a black box type is also frequently used to describe the system dynamics when one's knowledge about the system is poor or when the system is complicated and its description cannot be easily reduced to mathematical equations. The characterization of the system is carried out in a *function space* without resorting to the use of differential equations. For example, one may choose to use Fourier series expansions, spectral densities, autocovariance and cross-covariance matrices, time series, or impulse response and, of course, time-lag kernel functions. One or more of these relationships may be used to transform the forcing input to the system output without assuming the underlying structure of the process. These models are in principle infinitely dimensioned.

The advantage of a parametric model, being finite dimensioned, is that the system can be described concisely with a finite number of parameters. However, there is a price to be paid for this conciseness in terms of large prediction errors if the system orders or model parameters are not correctly chosen. If the system is

complicated or if one cannot make valid assumptions regarding the physical process structure with a certain degree of confidence, then a non-parametric model may be advantageous. Because such a non-parametric model may be infinitely dimensioned, it has the capability of yielding a system output that matches exactly with the observation.

The proposed kernel modeling approach is a hybrid of the parametric and non-parametric approaches. See Figure 2.5.20. For example, state dynamic equations are written explicitly for those variables whose dynamics are well known. There is absolutely no doubt that the state equations for the biomass and substrate concentrations in a chemostat are valid if cell growth is regulated by a limiting substrate. These dynamic equations for macroscopic variables are derived based strictly on material balance concepts; all other effects can be treated as variations in the specific growth rate and/or the yield coefficient. Thus, one can justifiably use a parametric approach to model the macroscopic observations.

On the other hand, one often does not know enough about the dependence of the specific growth rate on other variables. Under these circumstances, it is not practical to derive the dynamic equations for the specific growth rate. In such an attempt, sound judgment must be made as to what to assume and what to ignore, and these assumptions must be verified. In specifying the dynamic equations, the model orders must be known, and functional forms must be supplied. One often uses saturation functions of the Michaelis-Menten type whenever rate expressions are called for, but how often, if ever, are these expressions experimentally justified? Bimolecular elementary reaction rate expressions are often used also, but few studies have actually been conducted with the same degree of rigor that is demanded in the traditional proposal of a chemical reaction mechanism. Model discrimination is seldom performed in biochemical engineering. The moment one *single* such

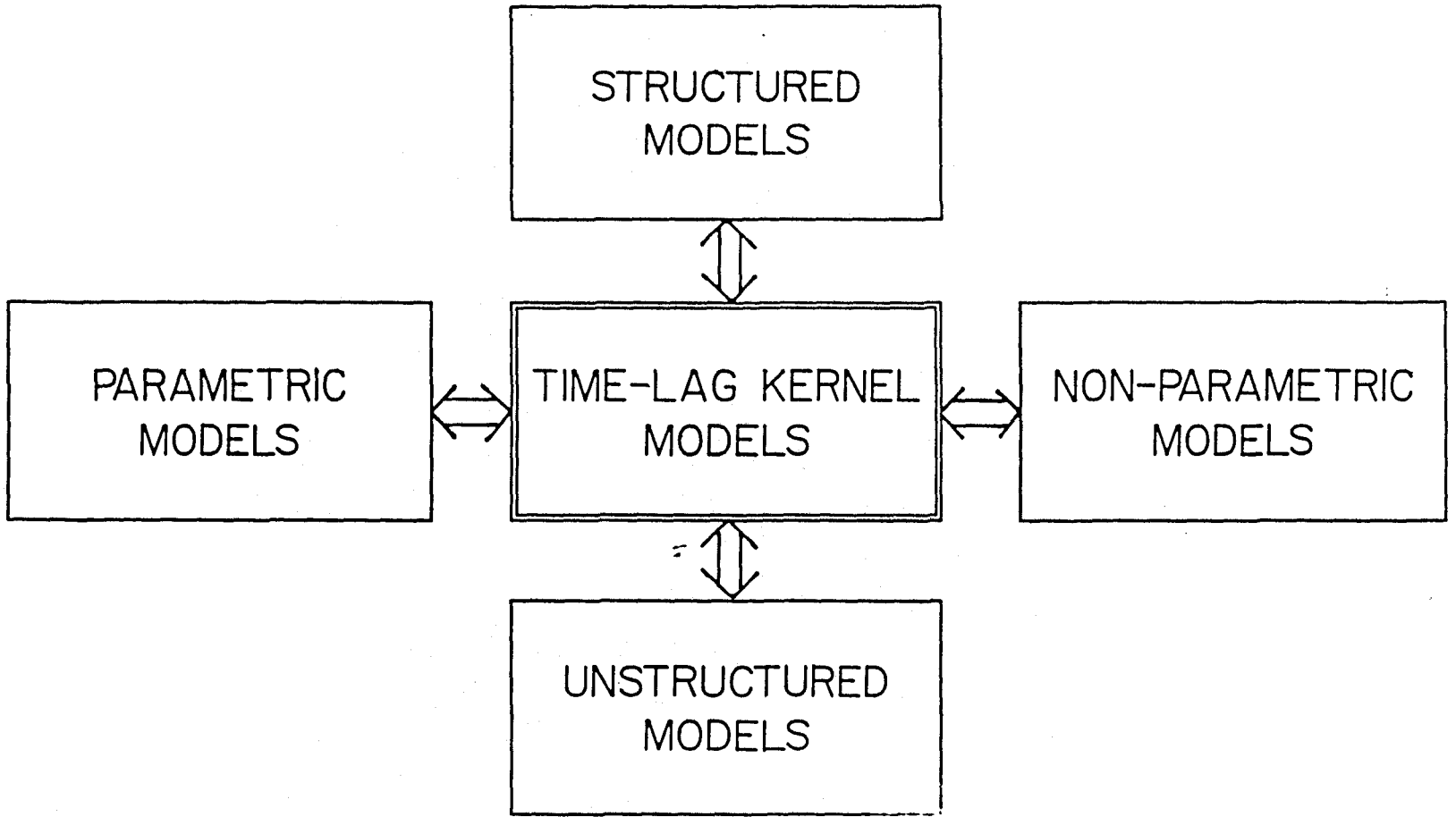


Figure 2.5.20. The kernel modeling approach is a hybrid of the parametric and non-parametric approaches. At the same time, it is also a hybrid of the structured and unstructured approaches.

expression is inserted into a dynamic equation without independent verifications, the model complexity becomes superficial. It contains essentially a black box at this level. More complexity beyond this point is inconsequential because patterns contained inside a black box cannot be seen.

A complex parametric description usually contains numerous model parameters whose values must somehow be evaluated, in addition to the model's sensitivities to variations in the parameters. As pointed out previously, the predicted system behavior can be drastically different if the system order, functional form, or model parameters are not chosen correctly. It has the inherent danger that the entire model can crumble if there exists a weak link in the model such as an incorrect assumption.

Thus, for the part of the process whose dynamics are well-known, one can employ the parametric modeling approach. Whereas, for the part that either is too complicated to be expressed mathematically with confidence or calls for excess investment of resources, one can resort to a non-parametric description. This combined approach is especially suited for a system where the level of understanding is not uniform across the process substructure. And many examples exist in biochemical engineering where the understanding of the process is quite nonuniform. It is not unusual that one can write the differential equations for macroscopic or lumped variables that can be easily quantified (unstructured modeling), but the dynamics for the intracellular components are often vague. One sacrifices knowledge of the process if a purely non-parametric approach is to be taken. On the other hand, one often needs to surpass his means if a purely parametric approach is taken. By using the combined approach, one can optimize effort by fully utilizing the current knowledge, without the danger of overreaching.

One example of such combined modeling approaches is the time-lag chemostat system demonstrated throughout this thesis; it uses differential equations for the biomass and substrate concentrations but gracefully switches to a time-lag kernel relationship to transform the input disturbance in the limiting substrate concentration to the observed specific growth rate. It should be emphasized that the chemostat example is used mainly to illustrate the time-lag approach. By no means does this simple example insist on the use of a time-lag kernel at level of the specific growth rate. If one is quite certain on the dynamics of certain intracellular components, then dynamic equations can be written for these components, and time-lag kernels can be employed for more intricate lower level sub-processes. It is only natural that the transition between a parametric approach and a non-parametric approach should be based on one's judgement as to which level his understanding of the process becomes vague. One should identify the level at which further structural refinement becomes superfluous due to the lack of actual knowledge. There is no need for a time-lag kernel if one's application does not demand accuracy beyond this point. Otherwise, the use of a time-lag kernel is recommended to compensate for the lack of detailed knowledge.

Since a model is to be judged based on its intended purpose, in many applications a time-lag kernel approach is a viable, attractive alternative to either an oversimplified unstructured model (abridged parametric model) that does not perform adequately, an overly complex structured model (fully developed parametric model) whose detailed description is unnecessary, or a purely black box approach (non-parametric model) that has little appeal due to the total lack of process structures.

2.6 COMPUTER SIMULATED SOLUTION OF THE KERNEL

The mathematical form of a convolution integral involving a kernel function is frequently encountered in other fields of studies. As a result, many names are commonly associated with this convolution integral and the related kernel function. For example, in non-Newtonian rheology, especially in the studies of polymeric fluids, the kernel function is known as the *memory function*, usually employed to describe the effect of past history on the apparent viscosity, stress, and strain. It will be interesting to see how the methodology introduced in this study can be applied to solve problems in fluid mechanics. In reactor kinetics and mixing studies in chemical engineering, the kernel function is known as the *residence time distribution function* (RTD). It is functionally equivalent to an *impulse response function* in the studies of systems process control, as previously mentioned during the introduction of the kernel. In a multidimensional form, it is called a *transition matrix* in systems dynamics and optimal control theories, or it is referred to as a *fundamental matrix* in mathematics in dealing with a set of ordinary differential equations. This mathematical equivalence was exploited earlier in reducing a structured model to an unstructured model. Techniques parallel to those used in estimating the residence time distribution will be developed to facilitate the experimental determination of the kernel.

Proper experimental design can greatly simplify the task of estimating the kernel function. Many types of transient experiments can be performed and closely monitored to estimate the kernel function. The ideal perturbation that may be introduced to the system can be either a step change in the input, an impulse change in the input, or a sinusoidally varying input. Although these ideal perturbations can be easily implemented physically in the case of, for example, a tracer injection experiment used to estimate the residence time distribution of a stirred tank or a

tubular reactor, they cannot be easily realized in our demonstration two-dimensional system of biomass and substrate concentrations.

The input variable in our system is the *intrinsic specific growth rate,* which depends on the limiting substrate concentration *in the fermentor*; it is not the substrate concentration in the feed stream, which the operator has a more direct control. An impulse in the input variable requires an impulse in the fermentor substrate concentration, assuming dependence of the intrinsic specific growth rate on the substrate concentration. Although the substrate concentration in the fermentor can be easily increased by injecting a concentrated dose of the substrate, it is difficult to lower the substrate concentration immediately to effect an impulse in the input. Similarly, a step increase in the substrate concentration demands that it be held at a constant value that is a notch above that before the step change. Such trajectories can only be realized by implementing rigorous control actions that can further complicate the dynamics of the system; neither is it easy with the current sensor and control technology to maintain the substrate concentration at a constant, desired level. As a result, an extensive analysis of such purely hypothetical situations imparts little practical benefits and will not be overly emphasized here. Similarly, it is not a trivial matter in achieving a sinusoidal change in the substrate concentration in a fermentor, although a sinusoidal change in either the dilution rate or the feed substrate concentration is attainable with high quality variable speed pumps and gradient mixers.

Since the kernel is equivalent to an impulse response function, it is simply the output of the system after being subjected to an impulse perturbation. Alternatively, it is the first-order derivative of the output of the system after the introduction of a step perturbation in the input. The determination of such a kernel

given the input and the output of the system is termed an *inverse problem*. Theoretically, it is not very difficult. However, because ideal perturbations that greatly simplify the estimation of the kernel cannot be introduced in a fermentor in reality and because random errors and noises cannot be totally eliminated, techniques must be developed to determine the kernel, given the non-ideal input characteristics and noisy output measurements. Of the various methods attempted, the five most promising ones will be discussed here. Namely, they are Fourier transform assisted kernel inversion, estimation based on the time-lag differential equation via polynomial approximation, iterative cyclic approximation, inverse numerical integration, and least-square error via multivariable minimum search.

Fourier Transform Assisted Kernel Inversion

The first method utilizes the well known powerful convolution theorem of the Fourier transform. The convolution integral involving two functions $\mu(t)$ and $k(t)$ in the time domain, $y(t) = \int_{-\infty}^{t} \mu(h)k(t-h)dh$, is a simple product of the two functions in the Fourier domain:

$$Y(\omega) = U(\omega) \cdot K(\omega), \tag{2.6.1}$$

where $Y(\omega)$, $U(\omega)$, and $K(\omega)$ are the Fourier transforms $\mathcal{F}$ of the functions $y(t)$, $\mu(t)$, and $k(t)$, respectively.

$$Y(\omega) = \mathcal{F}\{y(t)\} = \int_{-\infty}^{\infty} e^{i2\pi\omega t} y(t)dt \tag{2.6.2}$$

$$U(\omega) = \mathcal{F}\{\mu(t)\} = \int_{-\infty}^{\infty} e^{i2\pi\omega t} \mu(t)dt \tag{2.6.3}$$

$$K(\omega) = \mathcal{F}\{k(t)\} = \int_{-\infty}^{\infty} e^{i2\pi\omega t} k(t)dt \tag{2.6.4}$$

Collecting all the measured variables in Equation (2.6.1) on one side, one obtains:

$$K(\omega) = U^{-1}(\omega) \cdot Y(\omega). \tag{2.6.5}$$

Thus, theoretically, only simple algebraic manipulations are required to determine the kernel in the Fourier domain. The kernel function in the transformed domain, either Fourier or Laplace, is commonly referred to as the *transfer function* in systems theories. Transfer functions are encountered in all types of systems, chemical, electrical, or mechanical, and they are indispensable in the description of the system dynamics. It is no surprise that such a function is also the backbone of the current time-lag approach. In short, the kernel is determined from the measured input–output pair when the system is subjected to some disturbance.

$$\frac{\mathcal{F}\{\text{output}\}}{\mathcal{F}\{\text{input}\}} = K(\omega) \Longrightarrow \text{Kernel} \tag{2.6.6}$$

Finally, the kernel in the Fourier domain is converted to the time domain through an inverse operation:

$$k(t) = \mathcal{F}^{-1}\{K(\omega)\} = \mathcal{F}^{-1}\left\{U^{-1}(\omega)\cdot Y(\omega)\right\}, \tag{2.6.7}$$

where $\mathcal{F}^{-1}$ signifies the inverse Fourier transform operation as defined by the following equation.

$$k(t) = \mathcal{F}^{-1}\{K(\omega)\} = \int_{-\infty}^{\infty} e^{-i2\pi\omega t}K(\omega)d\omega \tag{2.6.8}$$

Note that a few slightly different versions in common use exist for the definitions of the forward and inverse transforms. Sometimes, the transform pair for an arbitrary function $f(t)$ are defined as:

$$F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} e^{-i\omega t}f(t)dt \tag{2.6.9a}$$

and

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}F(\omega)d\omega. \tag{2.6.9b}$$

Alternatively, the factor of 2π is sometimes divided equally among both expressions:

$$F(\omega) = \mathcal{F}\{f(t)\} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-i\omega t}f(t)dt \tag{2.6.10a}$$

and

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{i\omega t}F(\omega)d\omega. \qquad (2.6.10b)$$

Or, sometimes, they are defined to be the reverse of those used here:

$$F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} e^{-i2\pi\omega t}f(t)dt \qquad (2.6.11a)$$

and

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} = \int_{-\infty}^{\infty} e^{i2\pi\omega t}F(\omega)d\omega. \qquad (2.6.11b)$$

As long as one uses a consistent set of definitions throughout, the exact definition employed does not affect the end results. The advantage of the definition used in this study, over that of Equations (2.6.9a) and (2.6.9b), is that the same subroutine can be used to calculate both the forward transform and the inverse transform without any modification, due to the similarities in both expressions.

The discrete form of the forward Fourier transform is obtained by substituting the integral operation with a summation operation and by substituting dt in the continuous integral notation by the sampling interval Δt. For simplicity, Δt is usually set to unity so that the basic time unit is Δt. Furthermore, the units of frequency (rad/time) becomes $\frac{1}{N}$, and ω in the continuous notation becomes $\frac{j}{N}$ in the digitized version.

$$F_j = \sum_{t=0}^{N-1} f_t e^{i\tilde{\omega}_j t} = \sum_{t=0}^{N-1} f_t e^{i\frac{2\pi j}{N}t} \quad j = 0, 1, 2, \ldots, N-1, \qquad (2.6.12a)$$

where $\tilde{\omega}_j = \frac{2\pi j}{N}$ is the jth Fourier frequency, and N is the number of discrete data represented as an equally spaced time-series. The values for the running index ranges from 0 to $N-1$, and there are N terms in the summation sign. The summation does not include $t = N$ because the function f is assumed to be periodic, *i.e.*, $f_0 = f_N$. The implication of this assumption will be discussed in length in the

later part of this section. In the matrix transformation notation, the above equation is equivalent to:

$$\underbrace{\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix}}_{\mathbf{F}} = \underbrace{\begin{bmatrix} e_{00} & e_{01} & e_{02} & \cdots & e_{0,N-1} \\ e_{10} & e_{11} & e_{12} & \cdots & e_{1,N-1} \\ e_{20} & e_{21} & e_{22} & \cdots & e_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{N-1,0} & e_{N-1,1} & e_{N-1,2} & \cdots & e_{N-1,N-1} \end{bmatrix}}_{\mathbf{E}} \cdot \underbrace{\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}}_{\mathbf{f}} \tag{2.6.12b}$$

$$\mathbf{F} = \mathbf{E} \cdot \mathbf{f}, \tag{2.6.12c}$$

where $e_{jt} = \exp(i\frac{2\pi jt}{N})$ is the jtth element of the *forward transform matrix* $\mathbf{E}$. Note that due to the complementary nature of the frequencies, $F_j = F^*_{N-j}$, where "$*$" denotes the adjoint of a complex matrix, and only F_0 to $F_{\frac{N}{2}}$ need to be calculated. The discrete inverse transform corresponding to Equation (2.6.12a) is:

$$f_t = \frac{1}{N}\sum_{j=0}^{N-1} F_j e^{-i\tilde{\omega}_j t} = \frac{1}{N}\sum_{j=0}^{N-1} F_j e^{-i\frac{2\pi j}{N}t} \quad t = 0, 1, 2, \ldots, N-1, \tag{2.6.13}$$

where the factor $\frac{1}{N}$ results from substituting $d\omega$ with $\frac{1}{N}\Delta j$, which is simply $\frac{1}{N}$ because j takes on integer values. Similarly, in the matrix notation, the above equation is equivalent to:

$$\underbrace{\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}}_{\mathbf{f}} = \frac{1}{N}\underbrace{\begin{bmatrix} e^*_{00} & e^*_{01} & e^*_{02} & \cdots & e^*_{0,N-1} \\ e^*_{10} & e^*_{11} & e^*_{12} & \cdots & e^*_{1,N-1} \\ e^*_{20} & e^*_{21} & e^*_{22} & \cdots & e^*_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e^*_{N-1,0} & e^*_{N-1,1} & e^*_{N-1,2} & \cdots & e^*_{N-1,N-1} \end{bmatrix}}_{\mathbf{E}^*} \cdot \underbrace{\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix}}_{\mathbf{F}} \tag{2.6.13b}$$

$$\mathbf{f} = \frac{1}{N}\mathbf{E}^* \cdot \mathbf{F}, \tag{2.6.13c}$$

where $e^*_{tj} = \exp(-i\frac{2\pi jt}{N})$ is the tjth element of the *reverse transform matrix* $\mathbf{E}^*$. Note that both $\mathbf{E}$ and $\mathbf{E}^*$ are symmetric matrices. In addition, $\mathbf{E}\cdot\mathbf{E}^* = \mathbf{E}^*\cdot\mathbf{E} = \frac{1}{N}\mathbf{I}$.

Recent development in the fast Fourier transform (FFT) algorithm has made the calculation of a Fourier transform much faster, especially for a high-dimensional

problem where N is large, *e.g.*, $N > 500$. The original function is obtained when one sequentially applies the Fourier transform to a function, followed by the inverse Fourier transform. Shifts in both the horizontal axis and the vertical axis do not change the final outcome; neither does the number of points used to describe the function. For completeness, the computer codes needed to perform FFT are included in Appendix D. The algorithm has been tested for different even functions, such as $\frac{1}{x^2+1}$, different odd functions, such as $\frac{1}{x}$, and non-even–non-odd functions, such as e^{-x}. Various tests have further verified that the Fourier transform subroutine presented in Appendix D is fully functional.

Shown in Figures 2.6.1a and 2.6.1b are the computer simulated responses of a biochemical reactor described by Equations (2.2.12) and (2.2.13). When the dilution rate is shifted up from 0.3 hr^{-1} to 0.7 hr^{-1} at t=2 hr, the theoretical response of the system is deterministically calculated based on the assumption that $s_f = 5$ g/l, $Y_s = 0.5$ g/g, and $\mu(s) = \frac{0.5 hr^{-1} s}{0.1g/g+s}$. In this simulation study, a step change in the dilution rate is employed as the disturbance because it can be readily implemented experimentally and because it represents one of the more challenging cases. In addition, with all other variables under close control, a shift-up or shift-down experiment can better reveal the more fundamental nature of the system.

For the purpose of this simulation, the μ versus s curve is assumed to follow a Monod model. However, it need not be so; any constitutive relation can be employed without loss of generality. A valid kernel inversion algorithm should be applicable to any functional form of $\mu(s)$. In simulating the response, the true kernel is assumed to be a first-order function:

$$k(t) = a_0 k_0(t) + a_1 k_1(t) = \left[a_0 \frac{1}{T} + (1 - a_0)\frac{t}{T^2}\right] e^{-\frac{t}{T}} \qquad (2.6.14)$$

with $a_0 = 0.2$ and $T = 1.0$ hr. This function is shown in Figure 2.6.2.

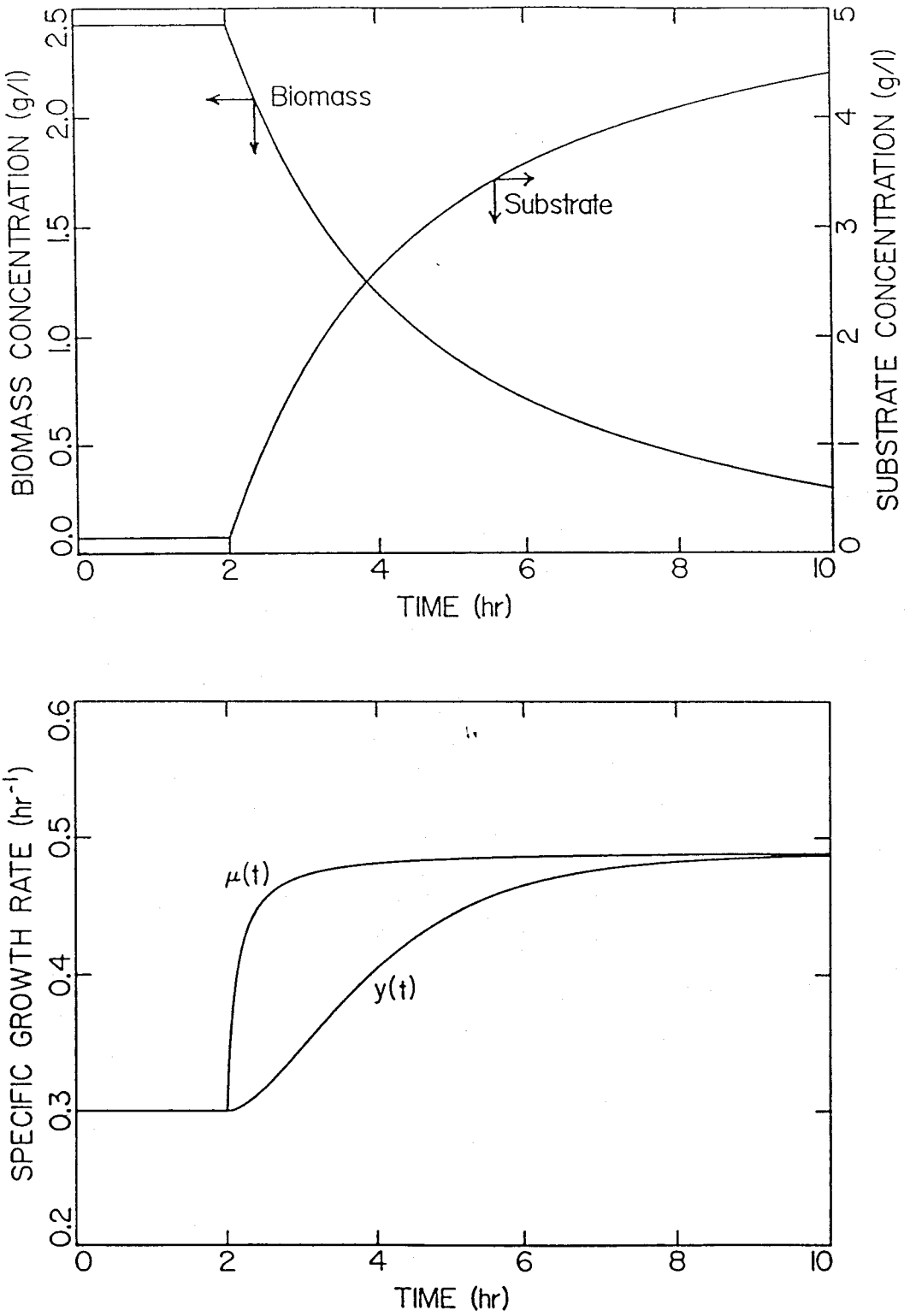


Figure 2.6.1. (a) Biomass and substrate concentrations in a continuously operated bioreactor after a shift-up in the dilution rate from 0.3 hr^{-1} to 0.7 hr^{-1}. (Parameters used: $\mu = \frac{0.5hr^{-1}s}{0.1g/l+s}$; $s_f = 5.0g/l$; $Y_s = 0.5g/g$.) (b) Intrinsic and observed specific growth rates.

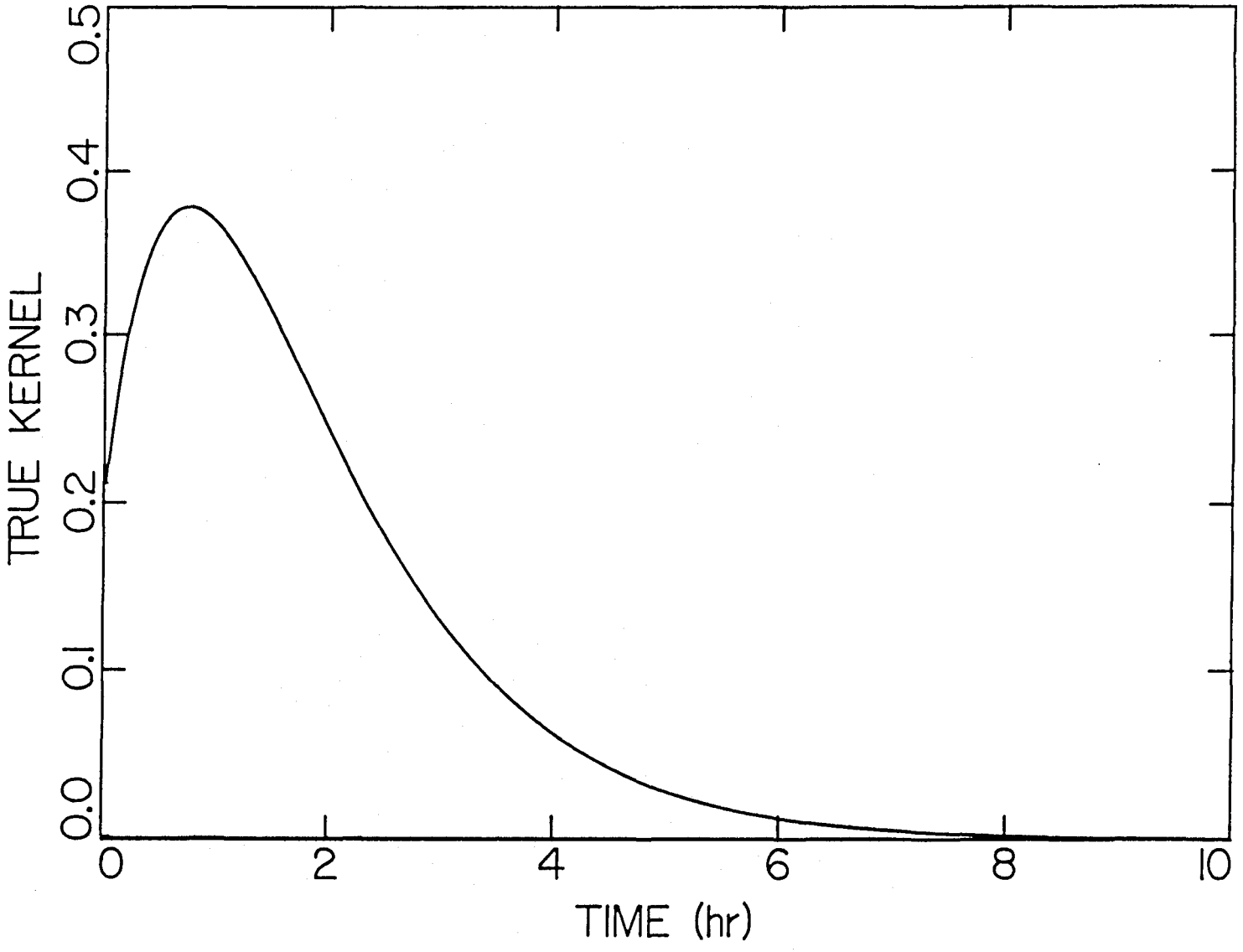


Figure 2.6.2. The assumed first-order kernel function used to simulate the dynamics presented in Figure 2.6.1: $k(t) = a_0 k_0(t) + a_1 k_1(t)$; numerical parameters: $a_0 = 0.2$ and $T = 1.0$ hr.

The objective of this section is to develop the algorithm needed to determine the shape of this kernel function, based on the intrinsic specific growth rate (input), $\mu(t)$, and the observed specific growth rate (output), $y(t)$. However, before an appropriate algorithm can be applied to the actual experimental data to estimate the kernel function for a given system of microbial population, one must evaluate the effectiveness of each algorithm and be made aware of all possible pitfalls. This can best be performed by comparing the two kernels, the first kernel being the estimation based on the given input and the theoretical response of the system and the second one being the original so-called true kernel that is used to generate the output response. More concisely, the objective is to develop a valid solution to the inverse problem, taking into full consideration the practical limitations imposed on the capabilities of the existing experimental apparatus, *e.g.*, non-ideal stimuli and noisy measurements.

Based on the simulated values, Fourier transforms are performed on the deterministic values of $\mu(t)$ and $y(t)$, and the inverse transform of $\frac{Y(\omega)}{U(\omega)}$ is taken naively according to Equation (2.6.7). These basic steps are described by the following graphic statement where each boxed block indicates the mathematical operations performed, *e.g.*, $\boxed{\mathcal{F}}$ being a forward Fourier transform, $\boxed{\mathcal{F}^{-1}}$ being the corresponding inverse Fourier transform, and the outer box being the combined forward-inverse Fourier transforms.

$$\mu \longrightarrow \boxed{\boxed{\mathcal{F}} \xrightarrow{U} \\ \boxed{\mathcal{F}} \xrightarrow{Y}} \boxed{\frac{Y}{U}} \xrightarrow{K} \boxed{\mathcal{F}^{-1}} \longrightarrow k$$
$$y \longrightarrow$$

The results of performing the above sequence of operations in accordance with the description of the convolution theorem of Fourier transforms are presented in Figure 2.6.3. Obviously, the curve of the estimated kernel does not match that predicted by Equation (2.6.7). The main reason for the failure from the point of view

of continuous treatment is that in order for the Fourier transform to be valid as defined in Equations (2.6.2)–(2.6.4), the original function to be transformed must satisfy both the Dirichlet condition, which requires the finiteness of discontinuities in a periodic function, and the condition that the improper integral $\int_{-\infty}^{\infty} |f(t)|dt$ is finite. Thus, our failure can be attributed to the fact that neither $\int_{-\infty}^{\infty} |\mu(t)|dt$ nor $\int_{-\infty}^{\infty} |y(t)|dt$ exists if the increased values of $\mu(t)$ and $y(t)$ after a step disturbance in the dilution rate are not forced down to their respective original values by another equal step change in the dilution rate in the opposite direction. It is emphasized that either the functions should be truly periodic, or, if a certain section of each function is clipped out and repeated according to this template, the resulting periodic functions should still describe the input and output relationships reasonably accurately. However, these conditions are not satisfied by the given response functions used for this simulation when the bioreactor is subjected to a single step shift in the dilution rate. One could theoretically use other types of stimuli, but the impetus for simulation would be lost.

In parallel, the reason for the failure from the perspective of discrete analysis is that in any actual calculation, instead of extending the time to infinity, a time window is ultimately created through which data are viewed because the duration of an experiment is finite. See Figure 2.6.4. Any data extending to the left (past) and right (future) of this imposed time window cannot be seen by the algorithm and are neglected. Thus, an integration from $-\infty$ to $+\infty$ that is needed to obtain the Fourier transform as indicated by Equations (2.6.2)-(2.6.4) is not performed in actuality. Instead, the continuous improper integration from $-\infty$ to $+\infty$ is substituted with a discrete summation from only $t = 0$ to $t = N$, as indicated by Equation (2.6.12).

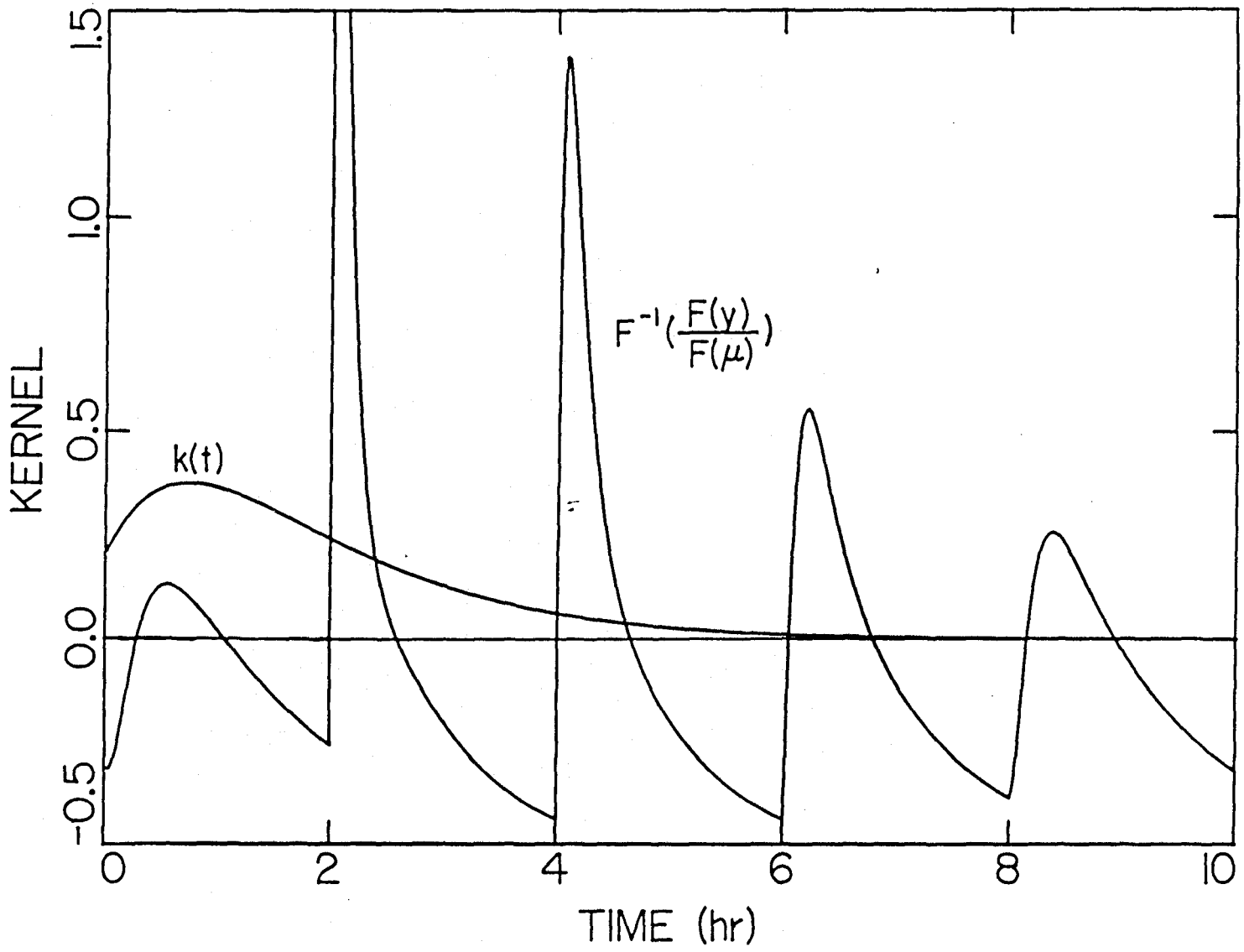


Figure 2.6.3. Direct calculation of the kernel based on $k(t) = \mathcal{F}^{-1}\{K(\omega)\} = \mathcal{F}^{-1}\{U^{-1}(\omega) \cdot Y(\omega)\}$.

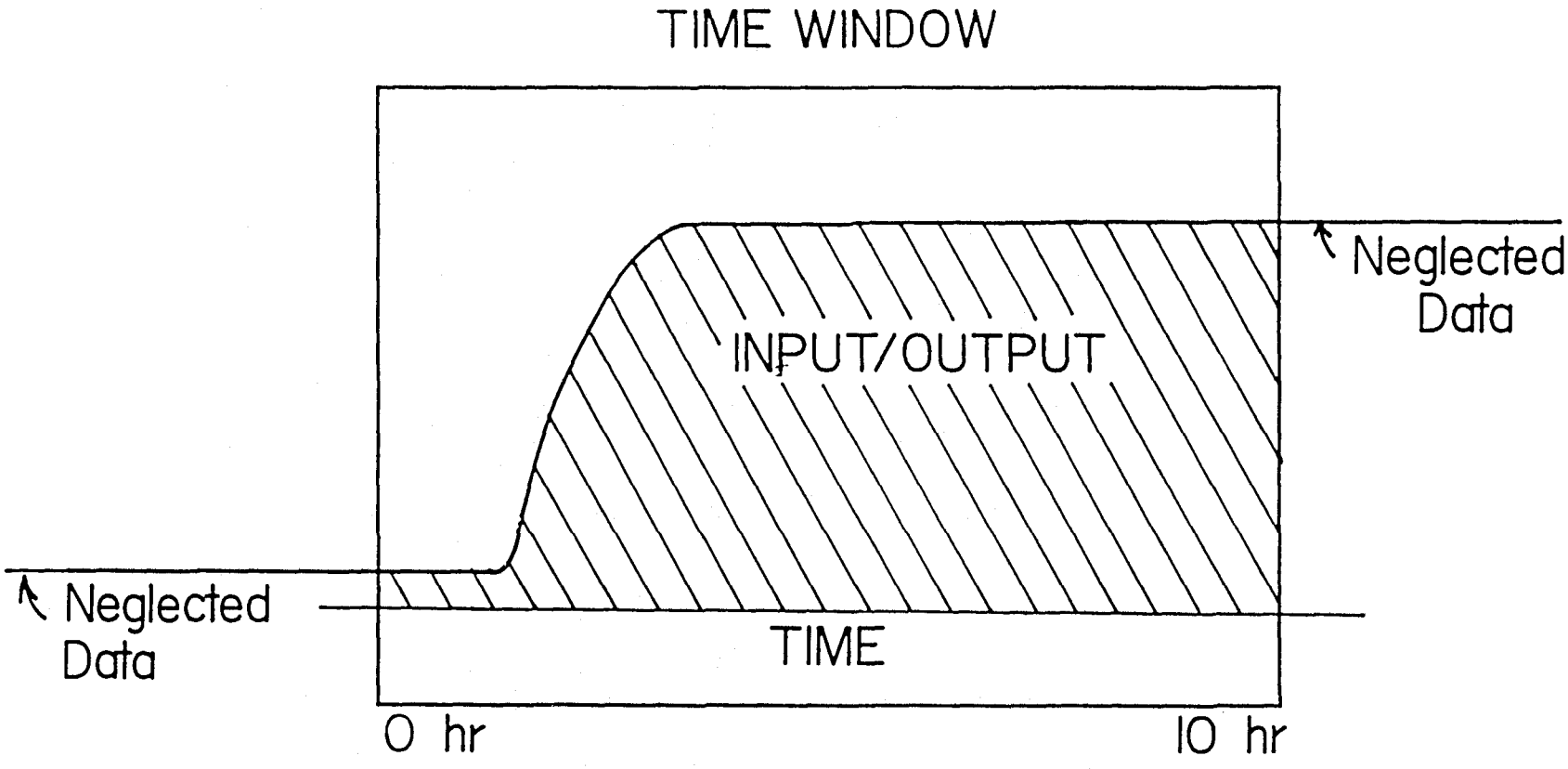


Figure 2.6.4. Data outside the time-window are ignored.

A direct consequence of employing the definition of Equation (2.6.12) is that the function f_t is implicitly assumed to be periodic outside the window. Thus, the relation $f_{nN+t} = f_t$ holds true for $n = \pm 1, \pm 2, \pm 3, \ldots$, which is identical to collapsing all t, both past and future, into the limits defined by the time window. Equivalently, due to the aliasing effect, all frequencies collapse into the interval $(0,\pi)$ in the frequency domain. These properties are expressed graphically in Figure 2.6.5. Moreover, because the function is discontinuous at the end-points where a time window is imposed, there also exist overshoots known as the Gibb's phenomenon, which is caused by the approximation of a discontinuous function by a truncated Fourier series. In short, the method fails because the algorithm is made to believe that both the input and output functions are periodic as indicated in Figure 2.6.5c, both of which generally deviate greatly from the real nature of the given functions. Because of these aliasing properties of digitized Fourier transforms, the Fourier transform method cannot be directly applied to estimate the kernel function when the input to the system is or resembles a step function. Thus, the theories on convolution integrals do not provide much help in this case.

One common trick used on a function that has unequal values at both boundaries of the time window is to subtract a function from it such that the resulting function's end-points are forced to be equal, as indicated in Figure 2.6.6. Mathematically, this is expressed as:

$$f(t) = \mu(t) - g(t). \tag{2.6.15}$$

$$\begin{aligned} y(t) &= \int_{-\infty}^{t} \mu(h)k(t-h)dh \\ &= \int_{-\infty}^{t} [f(h) + g(h)]k(t-h)dh \\ &= \underbrace{\int_{-\infty}^{t} f(h)k(t-h)dh}_{y_f} + \underbrace{\int_{-\infty}^{t} g(h)k(t-h)dh}_{y_g}. \end{aligned} \tag{2.6.16}$$

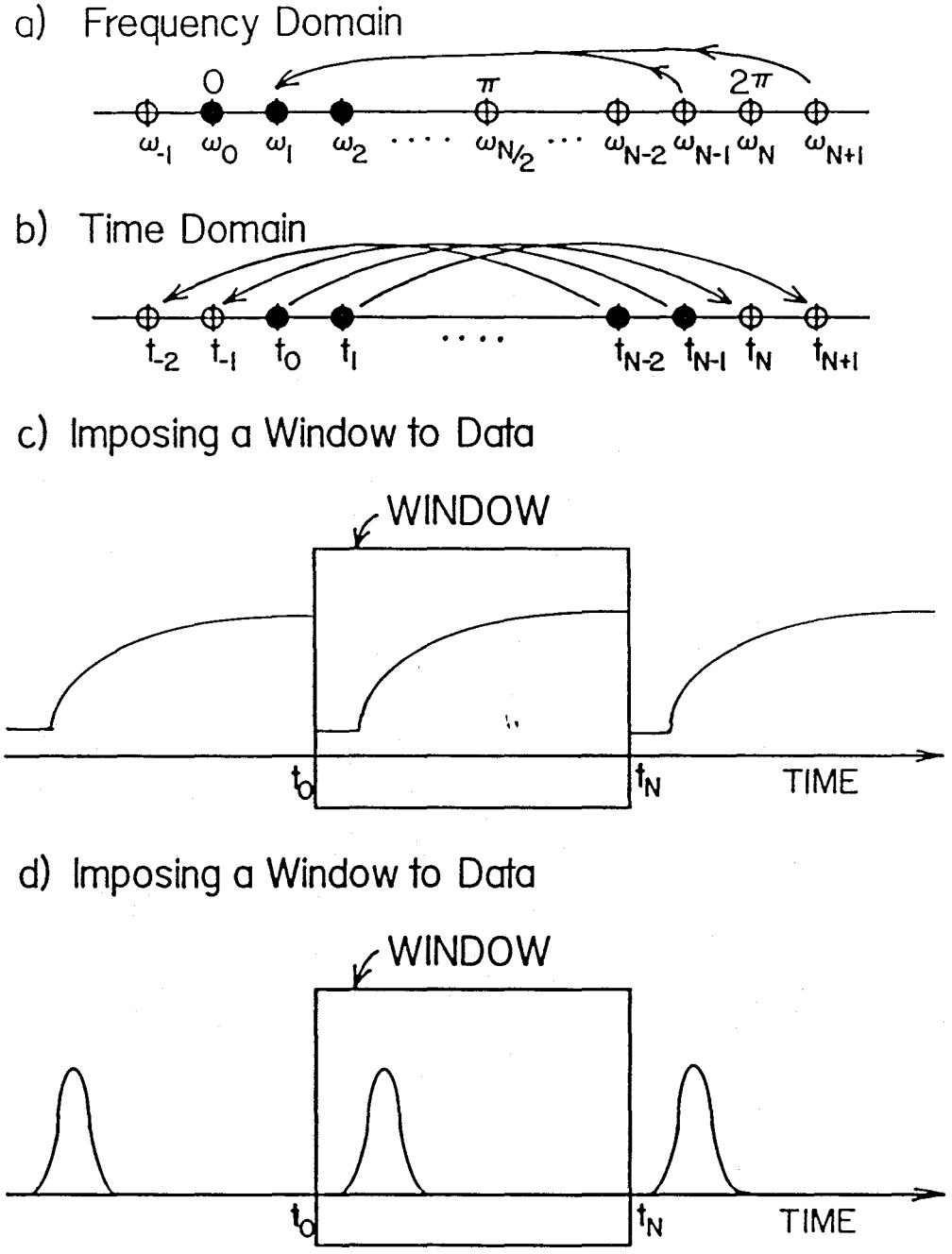


Figure 2.6.5. (a) All frequencies collapse into $(0,\pi)$. (b) In the time domain, the collapsing frequencies bring periodicity to data outside the time-window. (c) Neglected points outside the time-window are forced to be the periodic images of those inside the window. Note the sharp discontinuities at the window boundaries. (d) There is no discontinuity at the window boundaries when the function returns to its starting value.

Thus,

$$\begin{aligned} \mathcal{F}\{y(t)\} &= \mathcal{F}\{y_f(t) + y_g(t)\} \\ &= \mathcal{F}\{y_f(t)\} + \mathcal{F}\{y_g(t)\}, \end{aligned} \tag{2.6.17a}$$

which is equivalent to

$$Y(\omega) = Y_f(\omega) + Y_g(\omega). \tag{2.6.17b}$$

Although $f(t)$ now satisfies the finiteness condition, it is not strictly periodic, as the portion outside the time window is not the exact duplicate of that inside the time window, as shown in Figure 2.6.6. As a result, this technique does not work because each of the following equalities cannot be guaranteed.

$$Y_f(\omega) \stackrel{?}{=} U_f(\omega) \cdot K(\omega) \tag{2.6.18}$$

$$Y_g(\omega) \stackrel{?}{=} U_g(\omega) \cdot K(\omega), \tag{2.6.19}$$

where $U_f(\omega) = \mathcal{F}\{f(t)\}$ and $U_g(\omega) = \mathcal{F}\{g(t)\}$.

Since one can obtain an impulse-like function by differentiating a pulse-like step function, it is hoped that similar techniques of Fourier transforms can be applied to the derivatives of the system input and output. Just such an approach is attempted next. First, the convolution integral is differentiated once with respect to t:

$$y'(t) = \frac{d}{dt}y(t) = \frac{d}{dt}\left[\int_{-\infty}^{t} \mu(h)k(t-h)dh\right] \tag{2.6.20}$$

With a simple change of variables, the above equation is equivalent to:

$$y'(t) = \frac{d}{dt}y(t) = \frac{d}{dt}\left[\int_{0}^{\infty} \mu(t-h)k(h)dh\right]. \tag{2.6.21}$$

Leibnitz's rule of integral differentiation states that for

$$f(t) = \int_{a(t)}^{b(t)} \Phi(\mathbf{h},\mathbf{t})dh, \tag{2.6.22}$$

the derivative is:

$$f(t) = \int_{a(t)}^{b(t)} \frac{\partial \Phi(\mathbf{h},\mathbf{t})}{\partial t}dh + \Phi[b(t),t]\frac{db(t)}{dt} - \Phi[a(t),t]\frac{da(t)}{dt}. \tag{2.6.23}$$

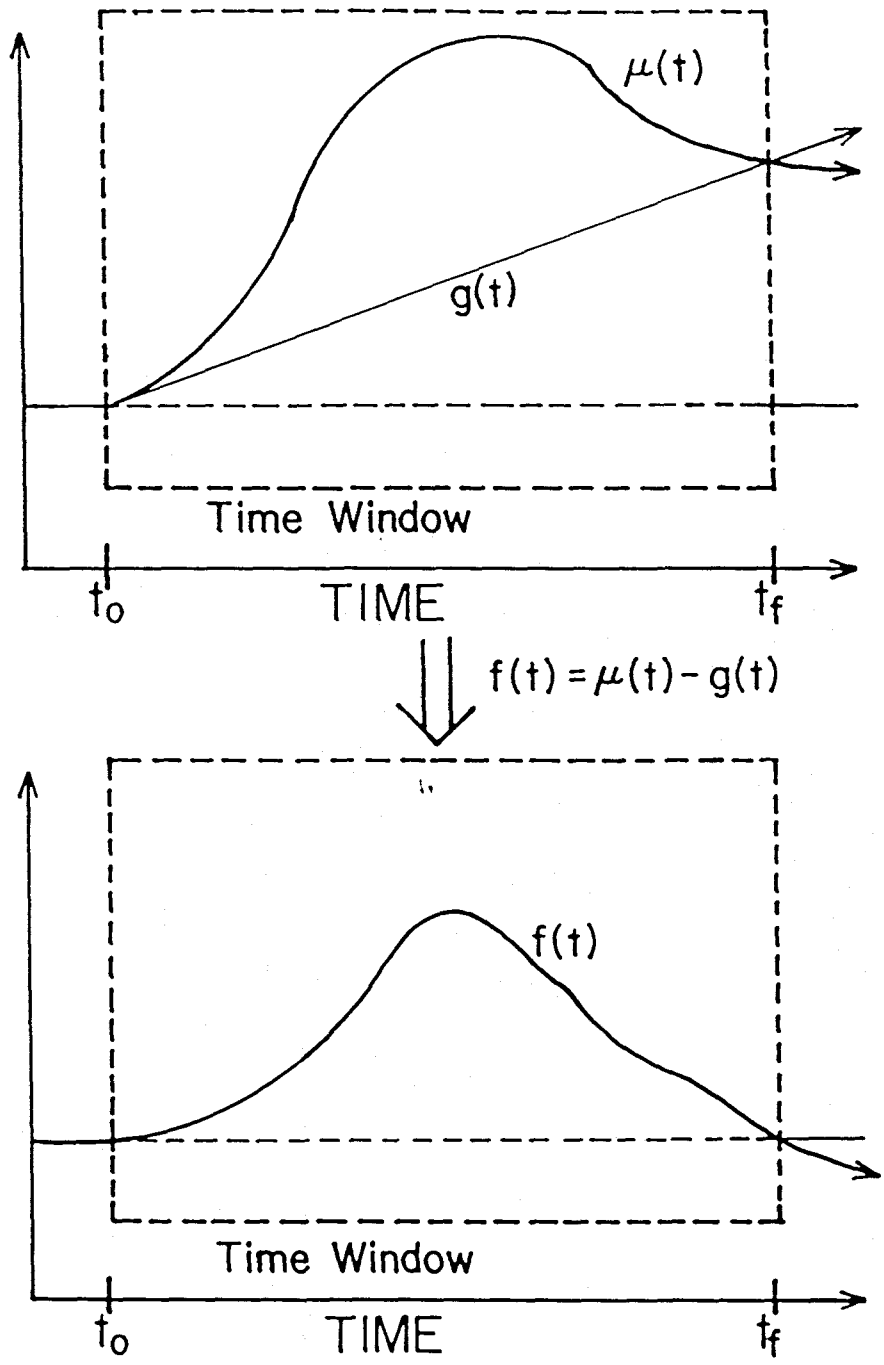


Figure 2.6.6. Forcing the boundary points in a time window to be equal in values by subtracting a triangular function $g(t)$ from the original function $\mu(t)$.

Employing this rule of integral differentiation on Equation (2.6.21), we have:

$$y'(t) = \frac{dy(t)}{dt} = \int_0^\infty \frac{\partial \mu(t-h)}{\partial t} k(h)dh \qquad (2.6.24a)$$

$$= \int_{-\infty}^t \mu(h)\frac{\partial k(t-h)}{\partial t}dh + \mu(t)k(0). \qquad (2.6.24b)$$

Applying Fourier transforms to Equations (2.6.24a) and (2.6.24b) yields:

$$\widetilde{Y}(\omega) = \widetilde{U}(\omega) \cdot K(\omega) \qquad (2.6.25a)$$

and

$$\widetilde{Y}(\omega) = U(\omega) \cdot \widetilde{K}(\omega) + U(\omega)k(0), \qquad (2.6.25b)$$

respectively, where $\widetilde{Y}(\omega) = \mathcal{F}\left\{\frac{dy(t)}{dt}\right\}$, $\widetilde{U}(\omega) = \mathcal{F}\left\{\frac{d\mu(t)}{dt}\right\}$, and $\widetilde{K}(\omega) = \mathcal{F}\left\{\frac{dk(t)}{dt}\right\}$. Of these two equations, the first one is much more useful. As a special case, for a step change in $\mu(t)$, $\frac{d\mu(t)}{dt}$ is an impulse function, and Equation (2.6.24a) simply reduces to:

$$\frac{dy(t)}{dt} = k(t), \qquad (2.6.26)$$

which states that the kernel function is the derivative of the output function $y(t)$. Note that Equation (2.6.25a) is similar in form to Equation (2.6.1).

This algorithm is applied to the same simulation data presented in Figure 2.6.1. The derivative quantities $\frac{dy(t)}{dt}$ and $\frac{d\mu(t)}{dt}$ shown in Figures 2.6.7a and 2.6.7b are approximated by $\frac{y_t - y_{t-1}}{\Delta t}$ and $\frac{\mu_t - \mu_{t-1}}{\Delta t}$, respectively, according to the following digitized scheme.

$$y_t = \sum_{j=0}^{\infty} \mu_{t-j} k_j \Delta t$$

$$= \sum_{j=0}^{t} \mu_{t-j} k_j \Delta t + \sum_{j=t+1}^{\infty} \overbrace{\mu_{t-j}}^{0} k_j \Delta t \qquad (2.6.27a)$$

$$= \sum_{j=0}^{t} \mu_{t-j} k_j \Delta t$$

$$
\begin{aligned}
y_t &= \sum_{j=-\infty}^{t} \mu_j k_{t-j} \Delta t \\
&= \sum_{j=-\infty}^{-1} \overbrace{\mu_j}^{0} k_{t-j} \Delta t + \sum_{j=0}^{t} \mu_j k_{t-j} \Delta t \qquad (2.6.27b) \\
&= \sum_{j=0}^{t} \mu_j k_{t-j} \Delta t
\end{aligned}
$$

Although the above two representations have different running indices in the summation, they are identical when expanded, as they should be.

$$
y_t = \Big[\mu_0 k_t + \mu_1 k_{t-1} + \mu_2 k_{t-2} + \ldots + \mu_{t-2} k_2 + \mu_{t-1} k_1 + \mu_t k_0\Big] \Delta t \qquad (2.6.27c)
$$

The last equalities in both Equations (2.6.27a) and (2.6.27b) are valid only if $\mu_j = 0$ for $j = -1, -2, -3, \ldots$, *i.e.*, if the same type of condition for Laplace transforms holds. This additional, but numerically crucial, condition is often neglected in practice, leading to grossly erroneous conclusions. This fact suggests that one should work in terms of deviation variables or shift the μ axis' reference point so that $\mu_j = 0$ for $j = -1, -2, -3, \ldots$, which can be easily achieved without loss of generality. A similar shift in the axis is also applied to y_t. Furthermore, because all the values of μ_j are *constant* from $j = -\infty$ to $j = -1$, one should perturb the system from a steady-state in order to take advantage of the above relations. Thus, one may encounter difficulties if these conditions are not satisfied.

Taking the difference of Equation (2.6.27), one obtains:

$$
\begin{aligned}
y_t - y_{t-1} &= \left[\sum_{j=0}^{t} \mu_{t-j} k_j \Delta t\right] - \left[\sum_{j=0}^{t-1} \mu_{t-1-j} k_j \Delta t\right] \\
&= \left[\sum_{j=0}^{t-1} \mu_{t-j} k_j \Delta t\right] - \left[\sum_{j=0}^{t-1} \mu_{t-1-j} k_j \Delta t\right] \qquad (2.6.28) \\
&\quad + \mu_0 k_t \Delta t - \overbrace{\mu_{-1}}^{0} k_t \Delta t \\
&= \sum_{j=0}^{t} (\mu_{t-j} - \mu_{t-1-j}) k_j \Delta t
\end{aligned}
$$

The above equation can be written in a difference operator notation:

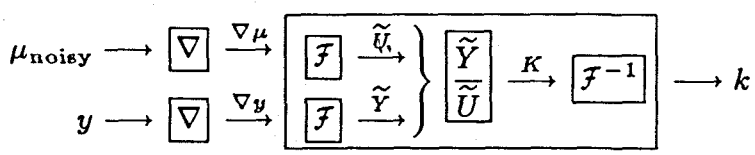
$$\nabla y_t = \left[\sum_{j=0}^{t} (\nabla \mu_t) k_j \Delta t\right] \tag{2.6.29}$$

where

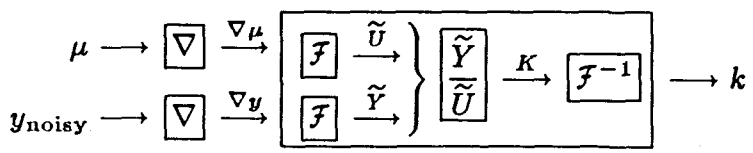
$$\nabla y_t = y_t - y_{t-1} \tag{2.6.30a}$$

$$\nabla \mu_t = \mu_t - \mu_{t-1} \tag{2.6.30b}$$

Note that the differential system input $\frac{\mu_t - \mu_{t-1}}{\Delta t}$ indeed resembles an impulse; consequently, one expects the differential system output $\frac{y_t - y_{t-1}}{\Delta t}$ to be close to the true kernel function. Fourier transforms are applied to these curves and the resulting kernel is displayed in Figure 2.6.7c. The agreement between the originally assumed kernel and the calculated one is quite good. Graphically, the following mathematical operations are performed.



Because there is always noise present in the measurements of both the input and the output in actuality, the simulation is now carried one step further to include the presence of 5% white noise. The noisy functions of $y(t)$ and $\mu(t)$ are shown in Figures 2.6.8a and 2.6.8b. The first set of kernel determination from noisy data assumes that there is noise in $\mu(t)$ but not in $y(t)$. The resulting kernel function, shown in Figure 2.6.8c, indicates that there exists a small, tolerable degree of deviation. However, when there is noise in the system output, this approach breaks down entirely. As shown in Figure 2.6.8d, the kernel resulting from the following series of operations:



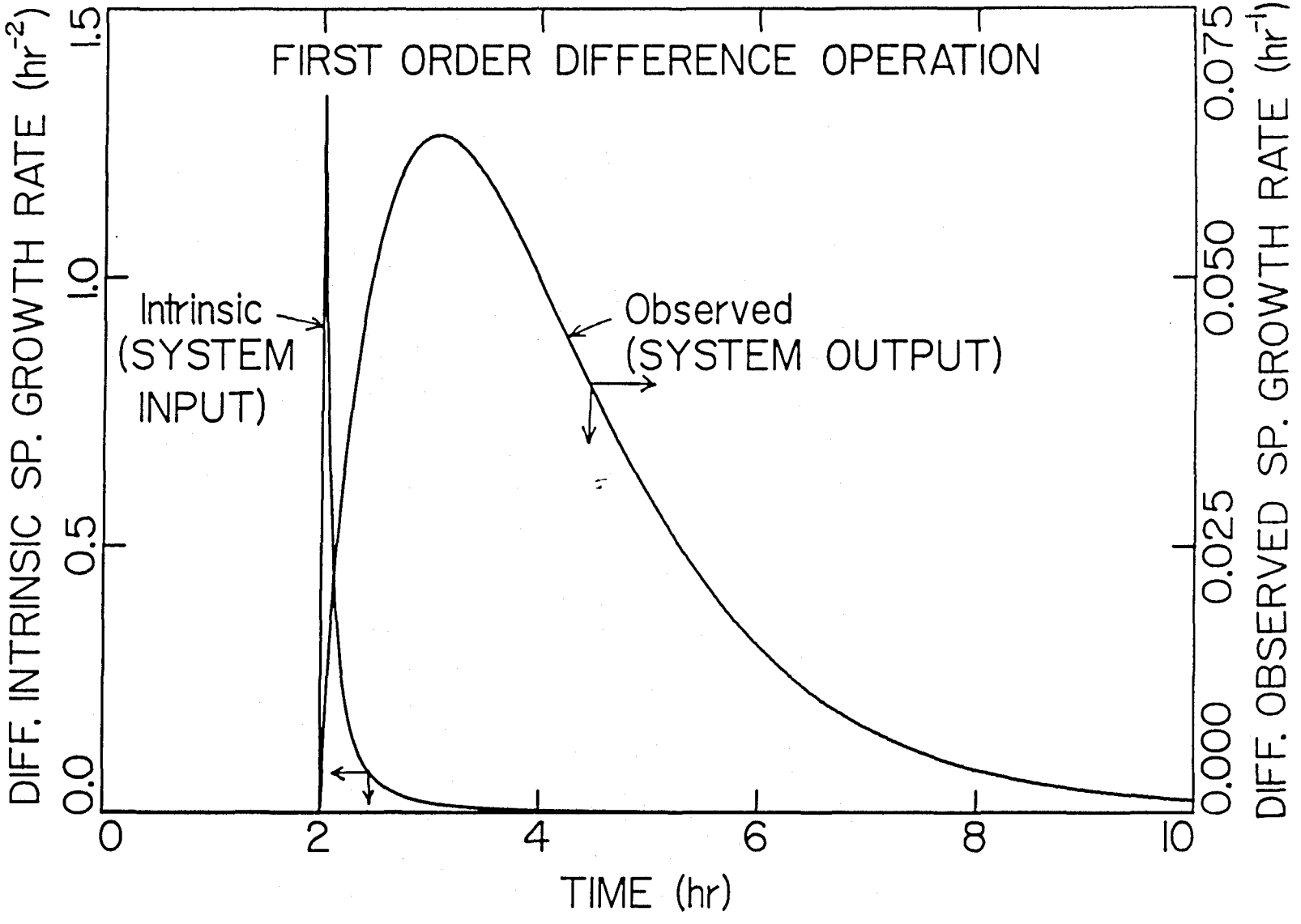


Figure 2.6.7. (a) "Derivative" of intrinsic specific growth rate ($\frac{\mu_t - \mu_{t-1}}{\Delta t}$) and (b) "derivative" of observed specific growth rate ($\frac{y_t - y_{t-1}}{\Delta t}$).

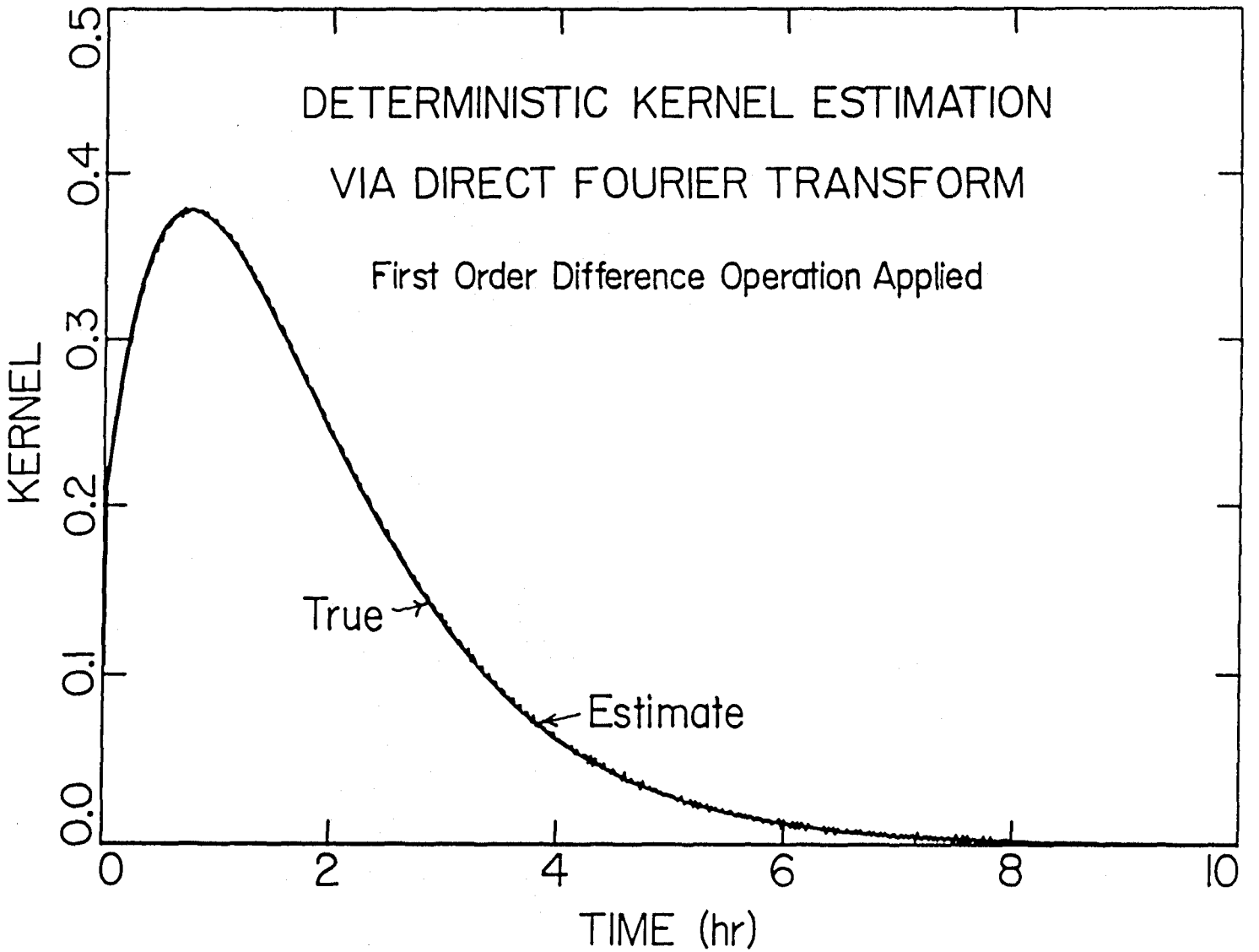
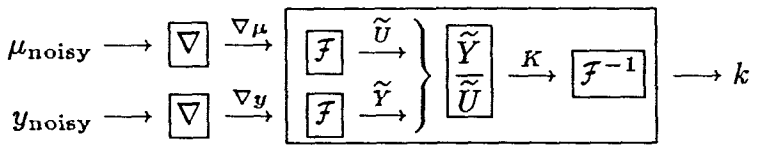


Figure 2.6.7c. (a) Originally assumed first-order kernel function and (b) the calculated kernel based on $k(t) = \mathcal{F}^{-1}\{K(\omega)\} = \mathcal{F}^{-1}\left\{\tilde{U}^{-1}(\omega) \cdot \tilde{Y}(\omega)\right\}$.

becomes totally buried under the noise and appears to be nonsense when a Fourier transform is applied directly to noise-free $\mu(t)$ and noisy $y(t)$. The estimated kernel is equally useless when there are noises in both $\mu(t)$ and $y(t)$.



The reason the direct application of Fourier transforms on noise-free ∇y_t and noisy $\nabla \mu_t$ yields satisfactory results is that $\mu(t)$, being the input to the dynamic system, is part of the integrand, and the integral operation $\int_{-\infty}^{t}$ has a tendency of filtering out the noise in $\mu(t)$. In contrast, the direct application of Fourier transforms fails whenever there is noise in $y(t)$. Taking the difference between y_t and y_{t-1} amplifies the noise to signal ratio to such an extent that ∇y_t is totally swamped in noise. In summary, since a difference operator cannot be applied to noisy $y(t)$ at all, the direct Fourier transform method proves to be rather powerless when the input to the system resembles a step function.

Noise Reduction

Because noise is the source of so much difficulty in the kernel estimation, it is hoped that the aforementioned techniques can perform satisfactorily if the noise level in the system input, and especially in the system output, can be reduced significantly. In this section, noise reduction is attempted via two classical routes: the least-square polynomial fitting in the time domain and the application of a low-pass filter by manipulation of windows in the frequency domain.

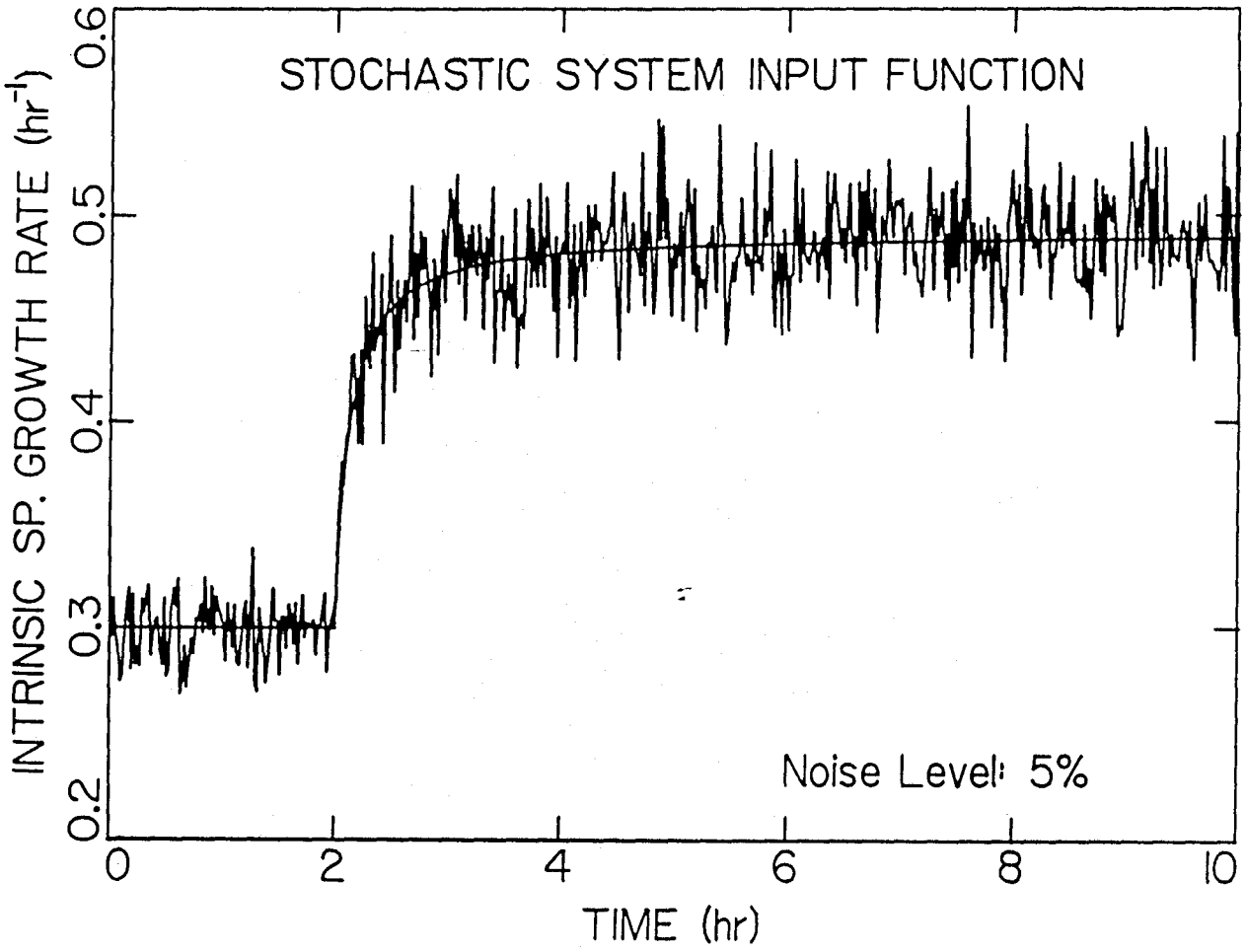


Figure 2.6.8a. Simulated input (*i.e.*, the specific growth rate in the absence of time-lag effects) as a function of time in a continuously operated bioreactor described by the state equations (2.2.10) and (2.2.11) after a shift-up in the dilution rate from 0.3 hr^{-1} to 0.7 hr^{-1}. (Parameters used: $\mu = \frac{0.5hr^{-1}s}{0.1g/l+s}$; $s_f = 5.0g/l$; $Y_s = 0.5g/g$; measurement noise level = 5% .)

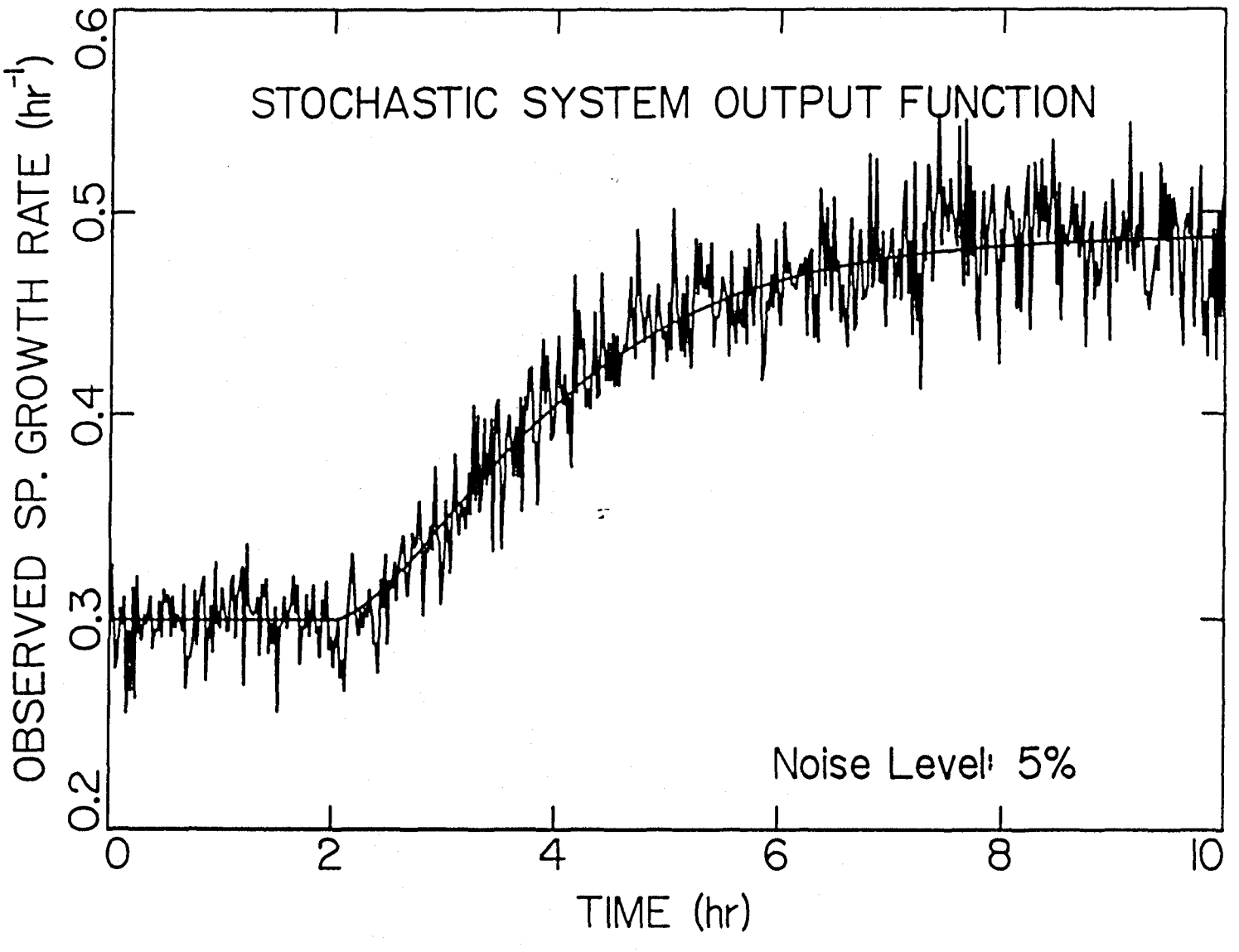


Figure 2.6.8b. Simulated output (*i.e.*, the observed specific growth rate containing time-lag effects) as a function of time. (Measurement noise level = 5%.)

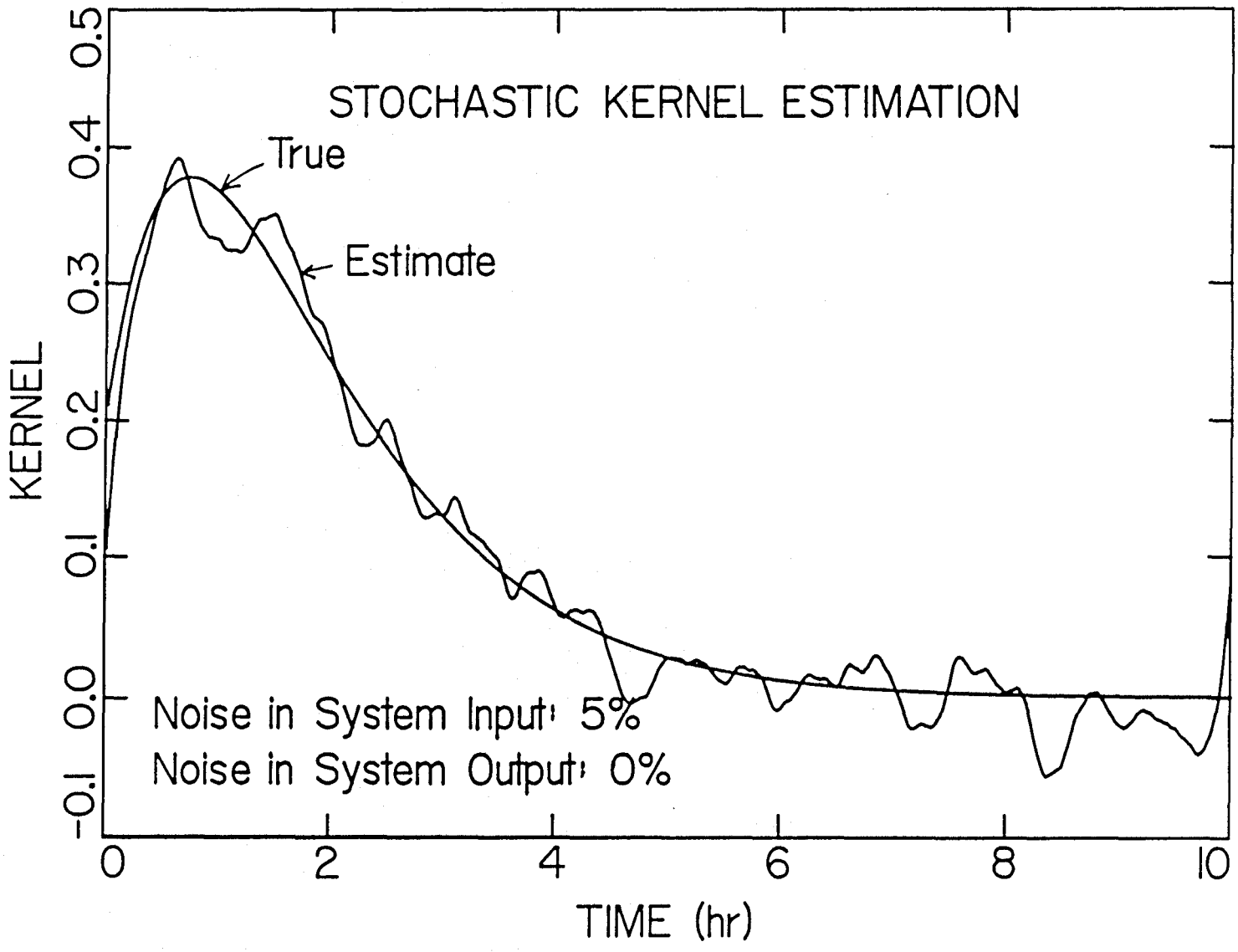


Figure 2.6.8c. True kernel and the estimated kernel based on the direct Fourier transform on the differentials of noisy $\mu(t)$ and noise-free $y(t)$.

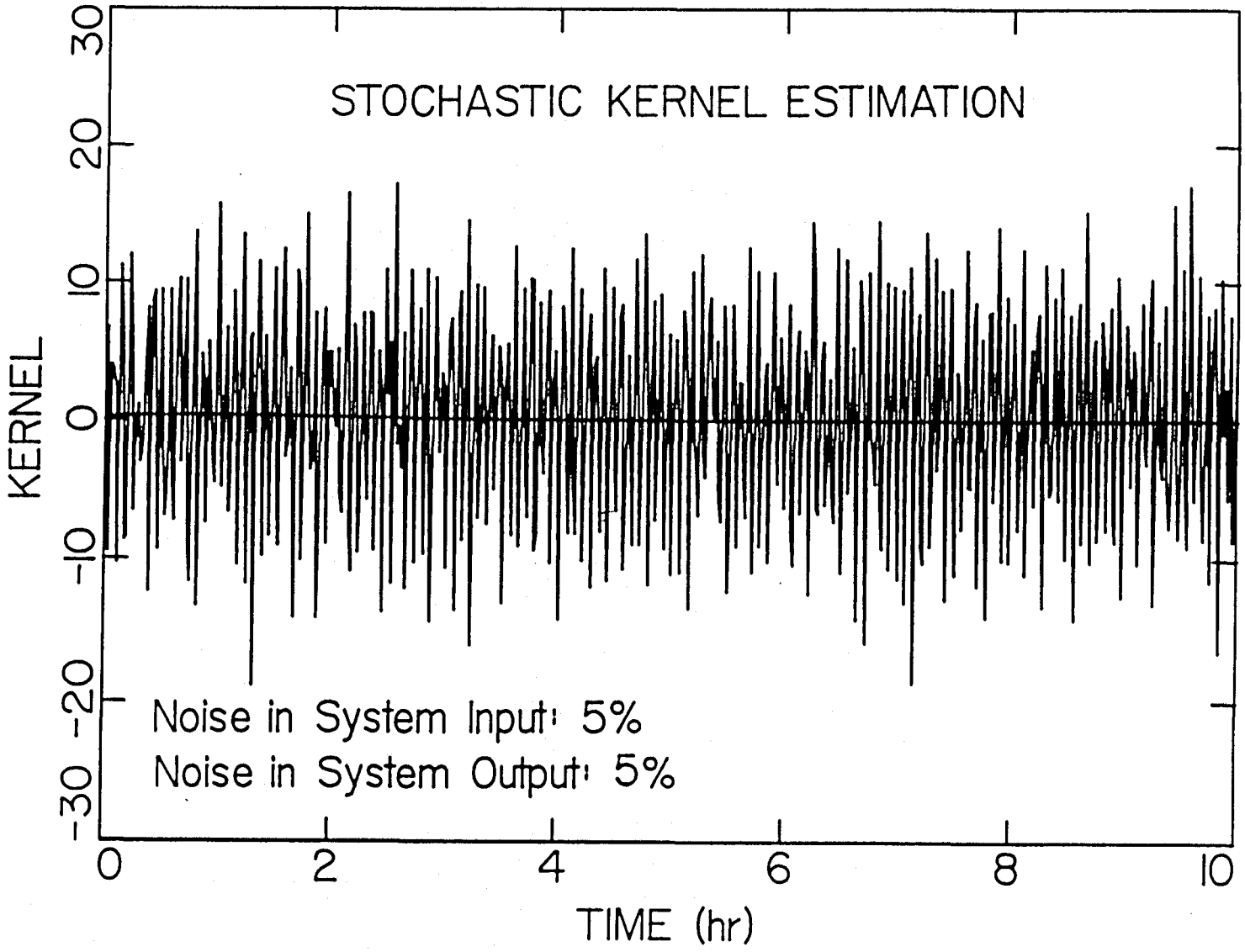


Figure 2.6.8d. True kernel and the estimated kernel based on the direct Fourier transform on the differentials noise-free $\mu(t)$ and noisy $y(t)$.

Multidimensional Linear Regression Analysis

This method is chosen as representative of the various available methods in the time domain. An orthogonal set of polynomials is used to fit the noisy (5%) curves of $\mu(t)$ and $y(t)$, thus removing much of the noise. The following set of coefficients are obtained from linear regression.

Coefficients from Linear Regression on $\mu(t)$								
Degree	1	t	t^2	t^3	t^4	t^5	t^6	t^7
0	0.480							
1	0.462	0.0417						
2	0.442	0.0178	-0.00151					
3	0.424	0.0418	-0.00820	4.97e-4				
4	0.403	0.0882	-0.0314	0.00453	-2.25e-4			
5	0.385	0.149	-0.0787	0.0186	-0.00199	7.90e-5		
6	0.372	0.211	-0.149	0.0497	-0.00850	7.20e-4	-2.40e-5	
7	0.363	0.271	-0.237	0.104	-0.0252	0.00340	-2.41e-4	6.98e-6

Coefficients from Linear Regression on $y(t)$								
Degree	1	t	t^2	t^3	t^4	t^5	t^6	t^7
0	0.445							
1	0.359	0.0190						
2	0.297	0.0607	-0.00465					
3	0.286	0.0758	-0.00885	3.12e-4				
4	0.288	0.0719	-0.00690	-2.44e-5	1.88e-5			
5	0.304	0.0153	0.0371	-0.0131	0.00166	-7.36e-5		
6	0.314	-0.0328	0.0905	-0.0369	0.00663	-5.63e-4	1.83e-5	
7	0.308	0.00427	0.0351	-0.00262	-0.00386	0.00113	1.19e-4	4.40e-5

The above calculation is based on a fitting range of 2–11 hours. Because the polynomial curves have difficulty turning corners quickly, the goodness of fit deteriorates significantly whenever the first two hours of steady-state values are included. In addition, data have been extended to the right of the time window by one hour prior to polynomial fitting so that the much greater deviation from the true values near the end of the interval is effectively avoided. Some of the smoothed polynomials for $\mu(t)$ are shown in Figure 2.6.9a, and those for $y(t)$ are shown in Figure 2.6.9b. From these figures, it can be seen that the agreement between some of the polynomial approximations and the true input and output response curves is indeed quite close. Of these, the 7th-degree polynomial for $\mu(t)$ and the 4th-degree polynomial for $y(t)$ give especially good fits and are used for the subsequent simulation studies.

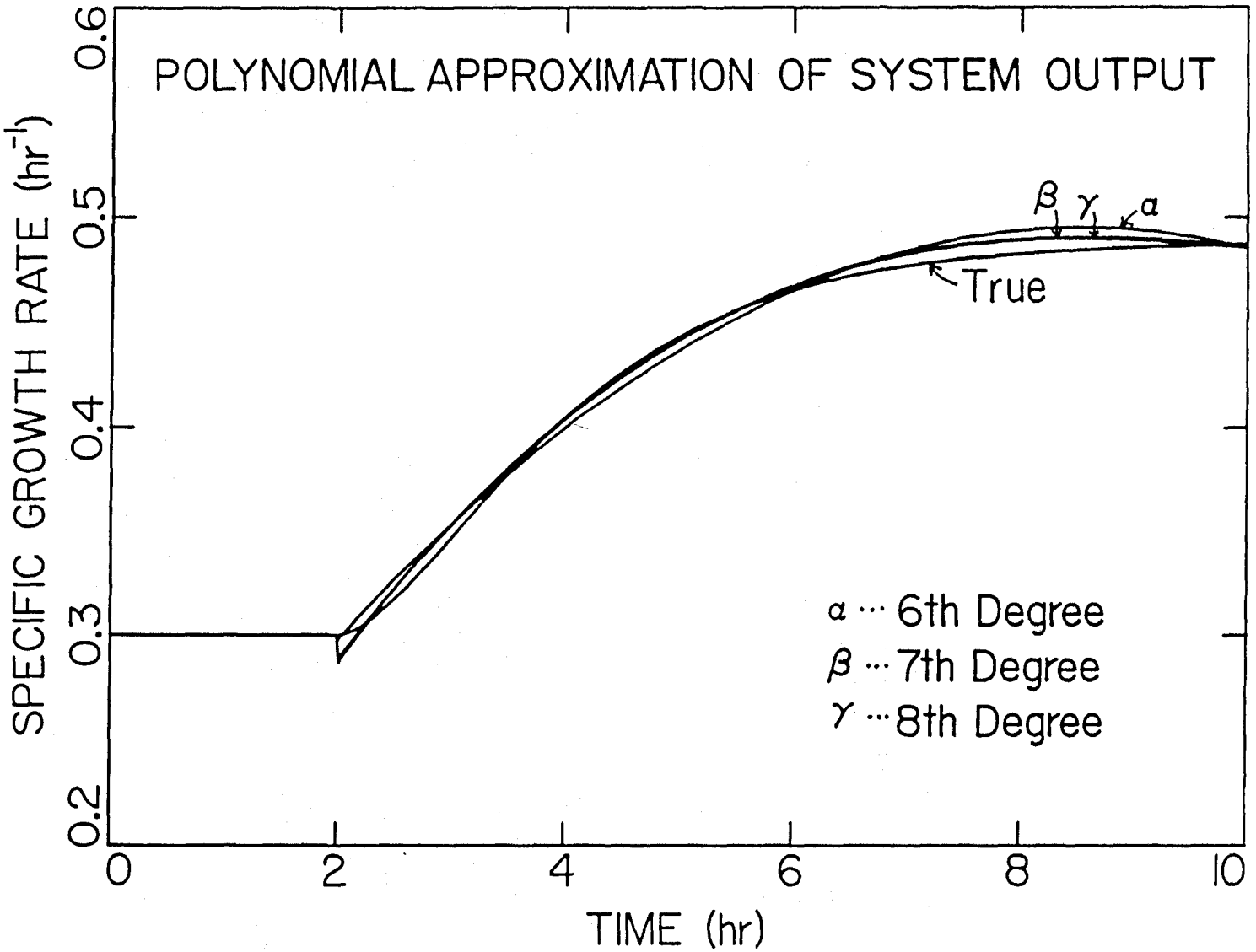


Figure 2.6.9a. Polynomials resulting from linear regression applied to $\mu(t)$ with 5% white noise between 1 and 10 hours in Figure 8a.

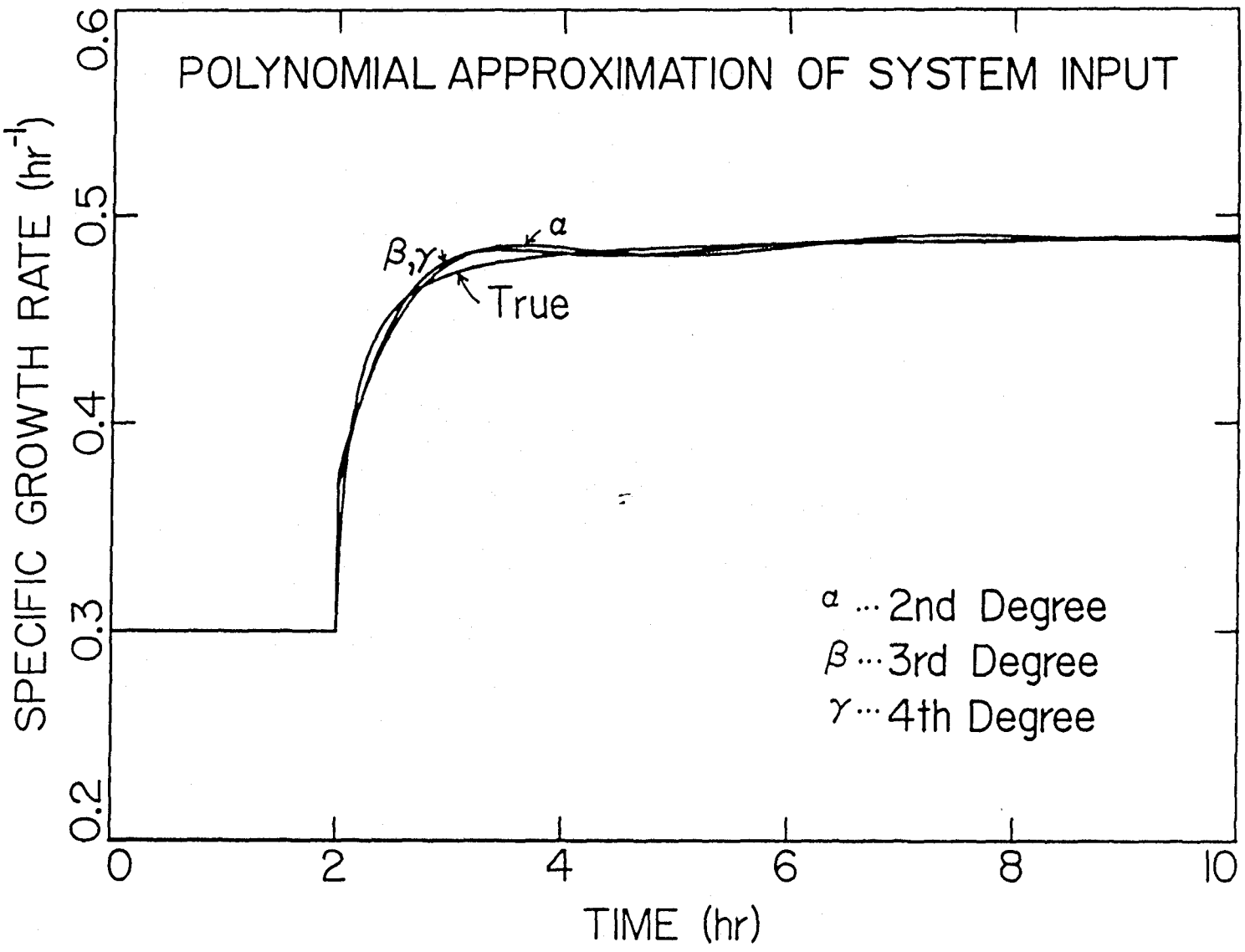
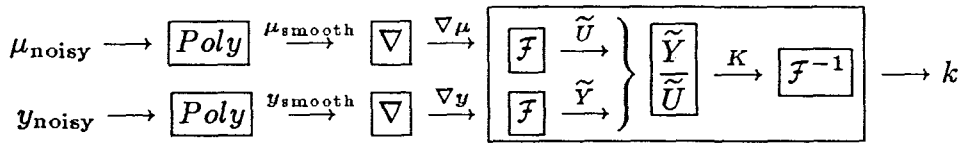
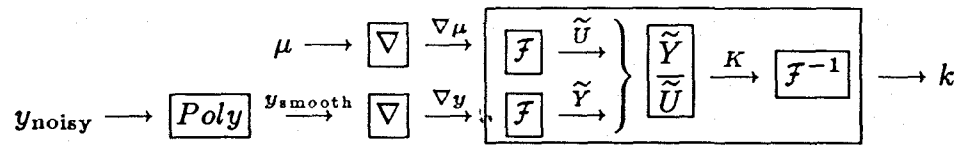


Figure 2.6.9b. Polynomials resulting from linear regression applied to $y(t)$ with 5% white noise between 1 and 10 hours in Figure 8b.

After smoothing with polynomials, the Fourier transform method is then applied to the derivatives of these smoothed functions as described by the following block diagram, where $\boxed{Poly}$ signifies the operation of polynomial smoothing through linear regression.



The resulting kernel function, as can be seen from Curve #4 in Figure 2.6.9c, generally follows the true kernel (Curve #1), granted that it is slightly noisy and there are some noticeable deviations. The effect of noise in $y(t)$ alone is studied by another simulation in which the noise in $\mu(t)$ is totally suppressed, as described by the following scheme.



This calculated kernel, corresponding to Curve #3 in Figure 2.6.9c, closely follows Curve #4. Furthermore, the effect of noise in $\mu(t)$ alone is shown in Curve #2. Thus, the major source of deviation of the estimated kernel from the true one is traced back to the presence of noise in the original $y(t)$, although noise in $\mu(t)$ also contribute to the deviation to a certain degree.

Various other combinations of noise structure in the input and output functions have been attempted. However, the center of the noisy kernel estimate always deviates somewhat from the true kernel, and the agreement is never as good as one wishes. This is due to the fact that although the noise contained in the output function itself is greatly reduced through polynomial smoothing, the slope of the function is not represented well by the polynomial approximation.

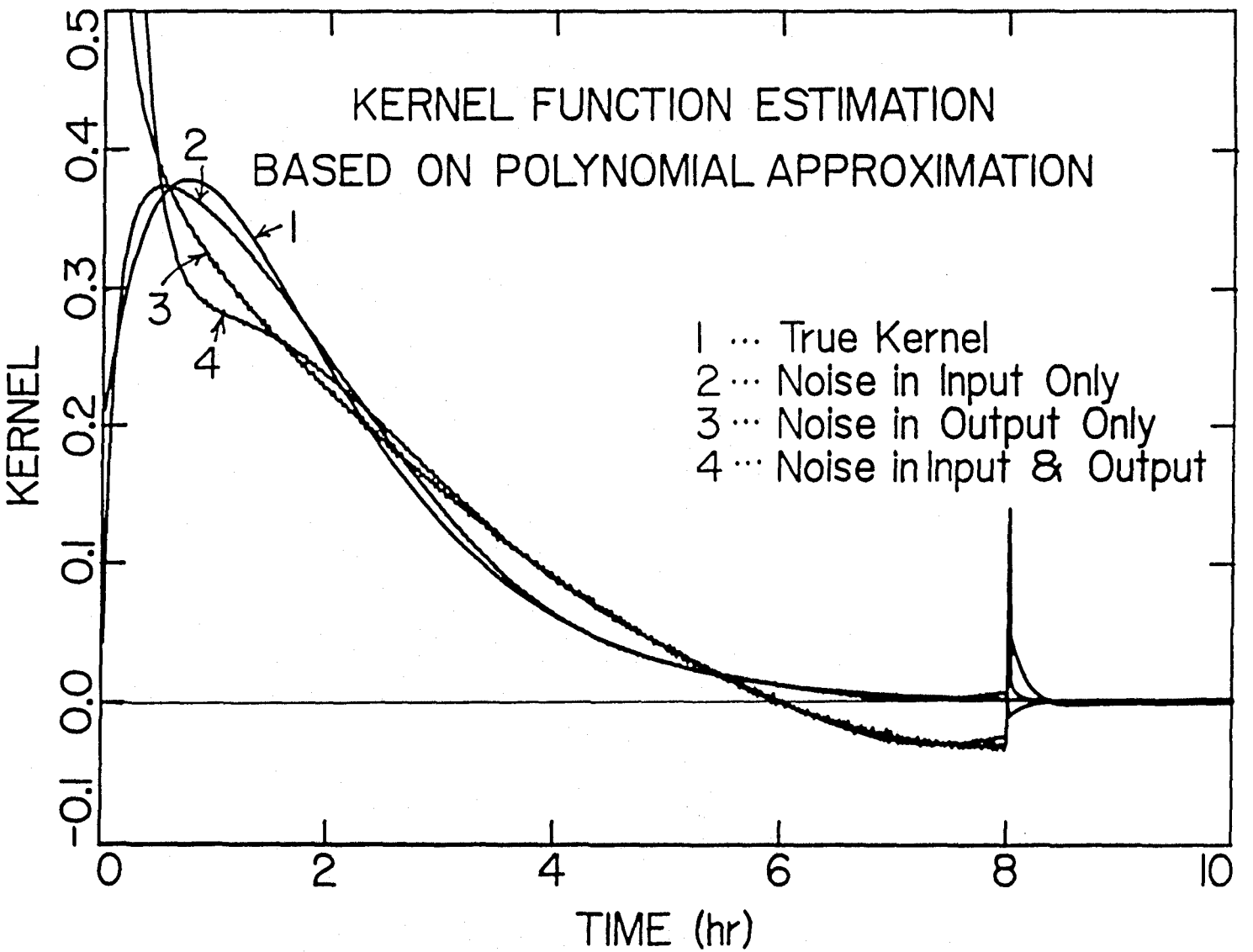


Figure 2.6.9c. True kernel and the estimated kernel based on the Fourier transform on the polynomial approximations to the noisy $\mu(t)$ and/or noisy $y(t)$.

Low-Pass Filter in the Fourier Domain

Another commonly used method of noise reduction is to truncate the high frequencies in the Fourier domain. Theoretically, a Fourier transform is a least-square fit of $f(t)$ to the orthogonal base functions of $\sin(j\tilde{\omega}t)$ and $\cos(j\tilde{\omega}t)$, given the degree of fit. Whereas, the previous regression analysis method calculates the least-square fit of a given function $f(t)$ to a given number of base functions of orthogonal polynomials with degrees of n or less in t. One of the tasks performed by a Fourier transform is to form an inner product between the given function $f(t)$ and the 0th frequency ($\tilde{\omega}_0$) and to extract all the constant components of the original function $f(t)$, which is equivalent to calculating the average value of $f(t)$ over the time window. The first fundamental component of the Fourier transform with a frequency of $\tilde{\omega}_1 = \frac{2\pi}{N}$ extracts from $f(t)$ all the components with that frequency, and so on. As a result, a noisy and highly oscillating function will have large amplitudes at high frequencies, where the amplitude is defined as $|F_j| \equiv \sqrt{\mathrm{Re}\{F_j\} + \mathrm{Im}\{F_j\}}$. On the other hand, the amplitudes of the discrete Fourier transform for a smooth function are generally small except at low frequencies. Thus, through spectrum analysis, information on the distribution of frequencies can be obtained. (A similarly structured harmonic analysis enables one to extract the dominant frequencies and phases of a time series.)

$$f_t = \underbrace{\frac{1}{N}\sum_{j=0}^{l-1} F_j e^{-i\tilde{\omega}_j t}}_{\text{smooth part}} + \underbrace{\sum_{j=l}^{m-1} F_j e^{-i\tilde{\omega}_j t}}_{\text{intermediate part}} + \underbrace{\sum_{j=m}^{N-1} F_j e^{-i\tilde{\omega}_j t}}_{\text{noisy part}} \qquad (2.6.31)$$

Therefore, smoothing can be effectively accomplished by imposing a window after the original function is transformed into the frequency domain. Many different types of filters or windows, each with a unique characteristic filter transfer function, have been proposed in the past. Shown in Figure 2.6.10a is the original boxcar window in the frequency domain. An ideal low-pass filter, shown in Figure 2.6.10b,

will allow all the frequencies less than the cutoff frequency, ω_c, to pass unaltered, and none of the component with frequencies higher than ω_c will pass through. To reduce Gibb's phenomenon caused by a suddenly truncated Fourier series and to accelerate the convergence, this ideal filter is often modified. For example, a linear filter with a transition band of δ is shown in Figure 2.6.10c. Finally, a cosine window, another commonly used filter, is shown in Figure 2.6.10d.

The filter used in this simulation study is a cosine one defined by the following equation:

$$H(\omega) = \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_c - \delta \\ \frac{1}{2}\left[1 + \cos\left(\frac{\omega - \omega_c + \frac{\delta}{2}}{\delta}\pi\right)\right] & \text{for } \omega_c - \delta \leq \omega \leq \omega_c + \delta \\ 0 & \text{for } \omega_c + \delta \leq \omega \leq \pi \ , \end{cases} \tag{2.6.32}$$

where δ is the transition band which, for simplicity, is set in this study to be $2\omega_c$. The function is graphically represented in Figure 2.6.11. The degree of smoothing obtainable depends on the cutoff frequency ω_c, and this is demonstrated in Figures 2.6.12 and 2.6.13 for $\mu(t)$ and $y(t)$, respectively, where it is shown that noise levels can be greatly reduced when ω_c is small. However, deviations from the true values at two ends of the smoothed function tend to become large for small ω_c. This is due to the fact that the end points of the original curves do not match and the Fourier transform attempts to force them to meet at the half way point. Furthermore, after a window is applied to a function, the function becomes more distorted at sharp corners, *e.g.*, at the onset of disturbance, when ω_c is small. Faithful reconstruction of the original function becomes difficult when only a few low frequencies can be used.

Similar to the approach taken in polynomial approximation, functions of $\mu(t)$ and $y(t)$ smoothed with frequency windows are used to generate a kernel based on the following scheme where $\boxed{Window}$ indicates smoothing through the application

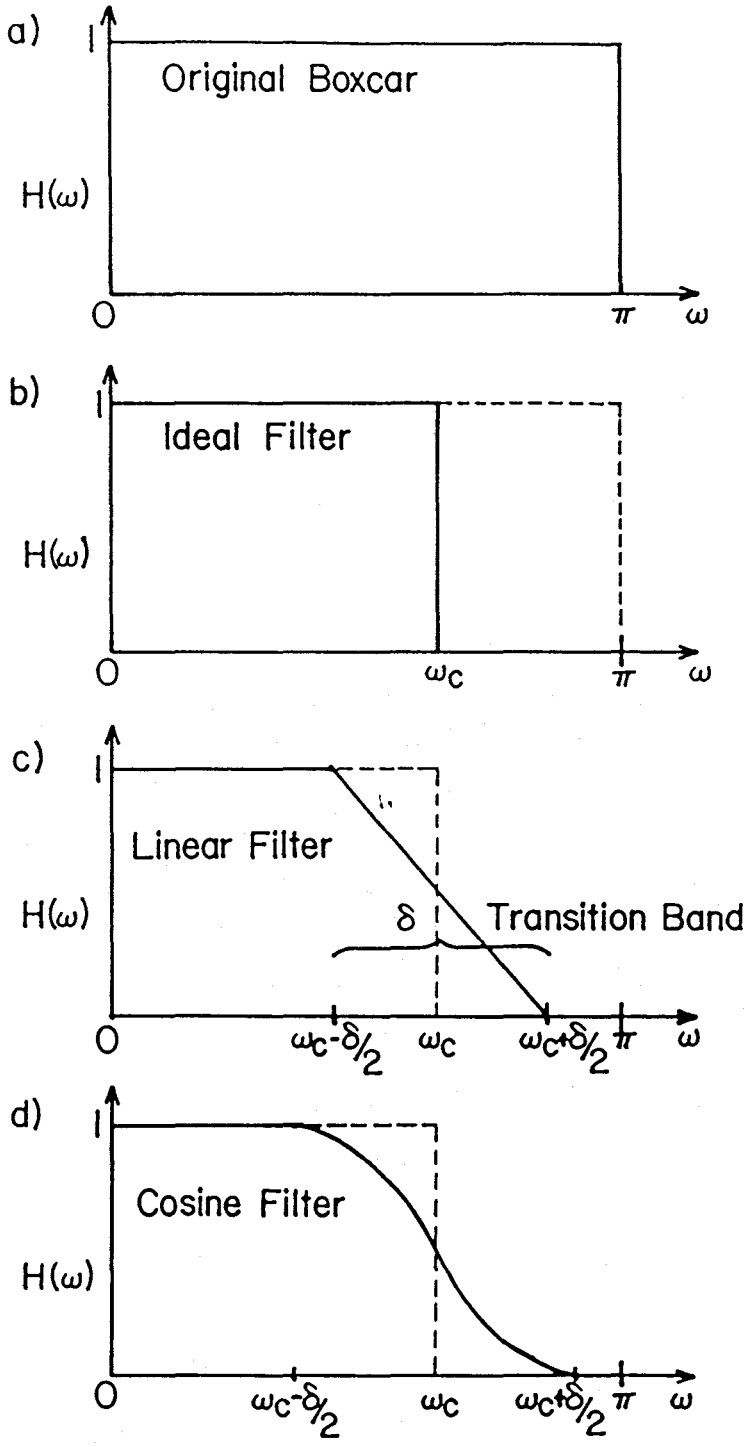


Figure 2.6.10. a) The original boxcar window. b) An ideal filter transfer function with cutoff frequency ω_c. c) A linear filter with a transition band of δ. d) A cosine filter with a transition band of δ.

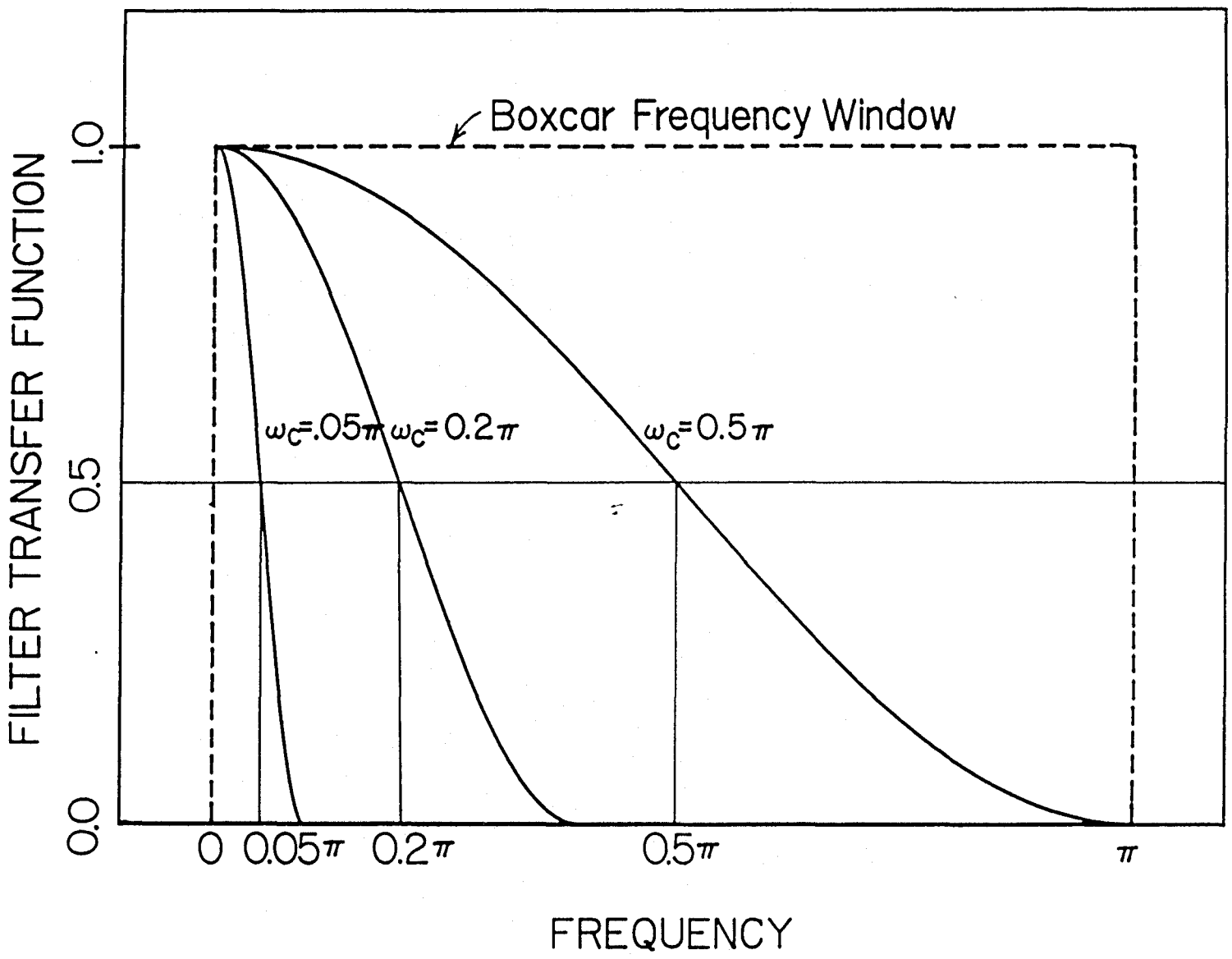


Figure 2.6.11. Cosine low-pass filter transfer function. The displayed functions are used to obtain the smoothed curves shown in Figures 2.6.12 and 2.6.13.

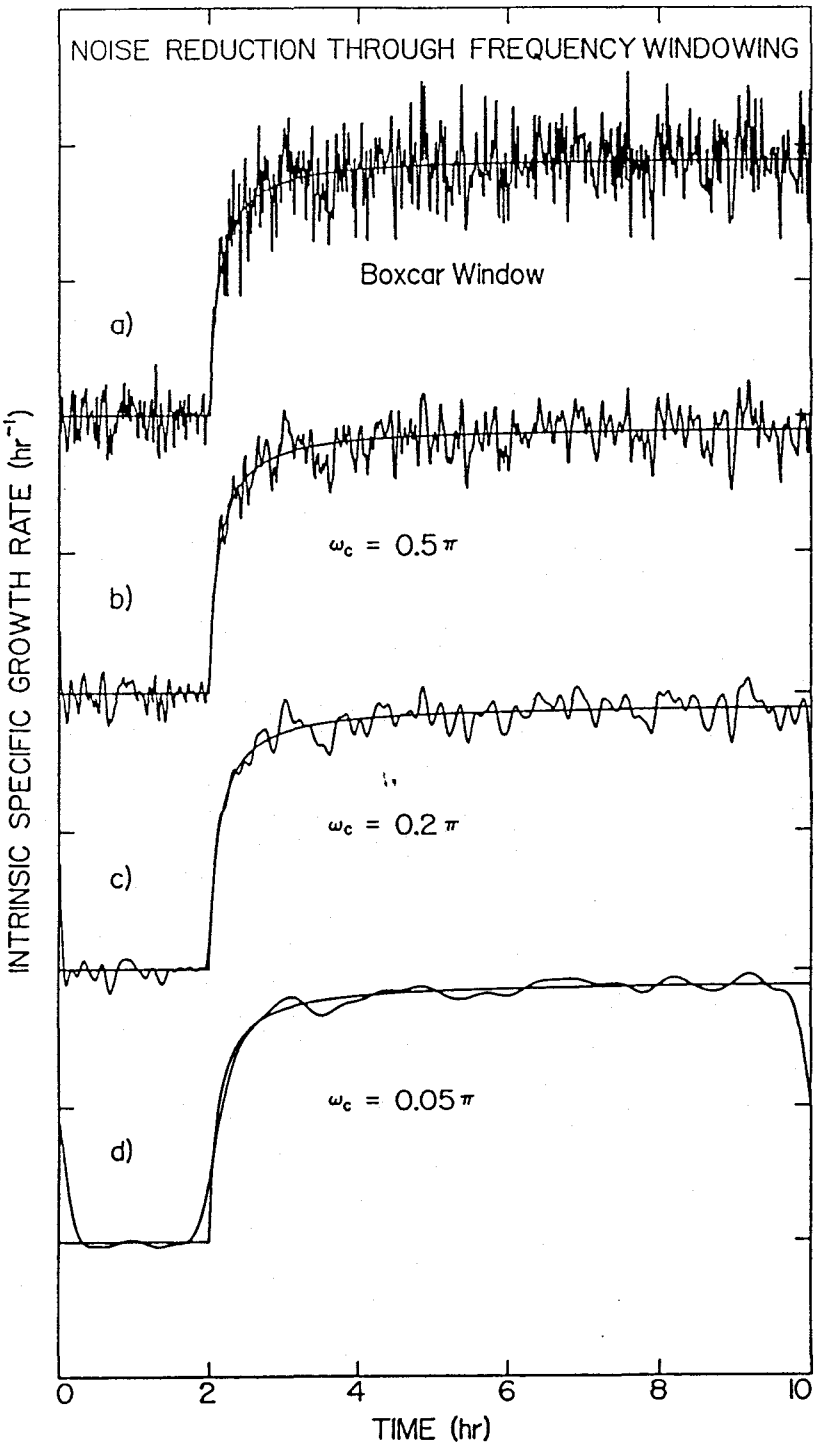


Figure 2.6.12. a) The original function of $\mu(t)$ with 5% noise. b) Noise reduction via a cosine low-pass filter with $\omega_c = 0.5\pi$. c) $\omega_c = 0.2\pi$. d) $\omega_c = 0.05\pi$

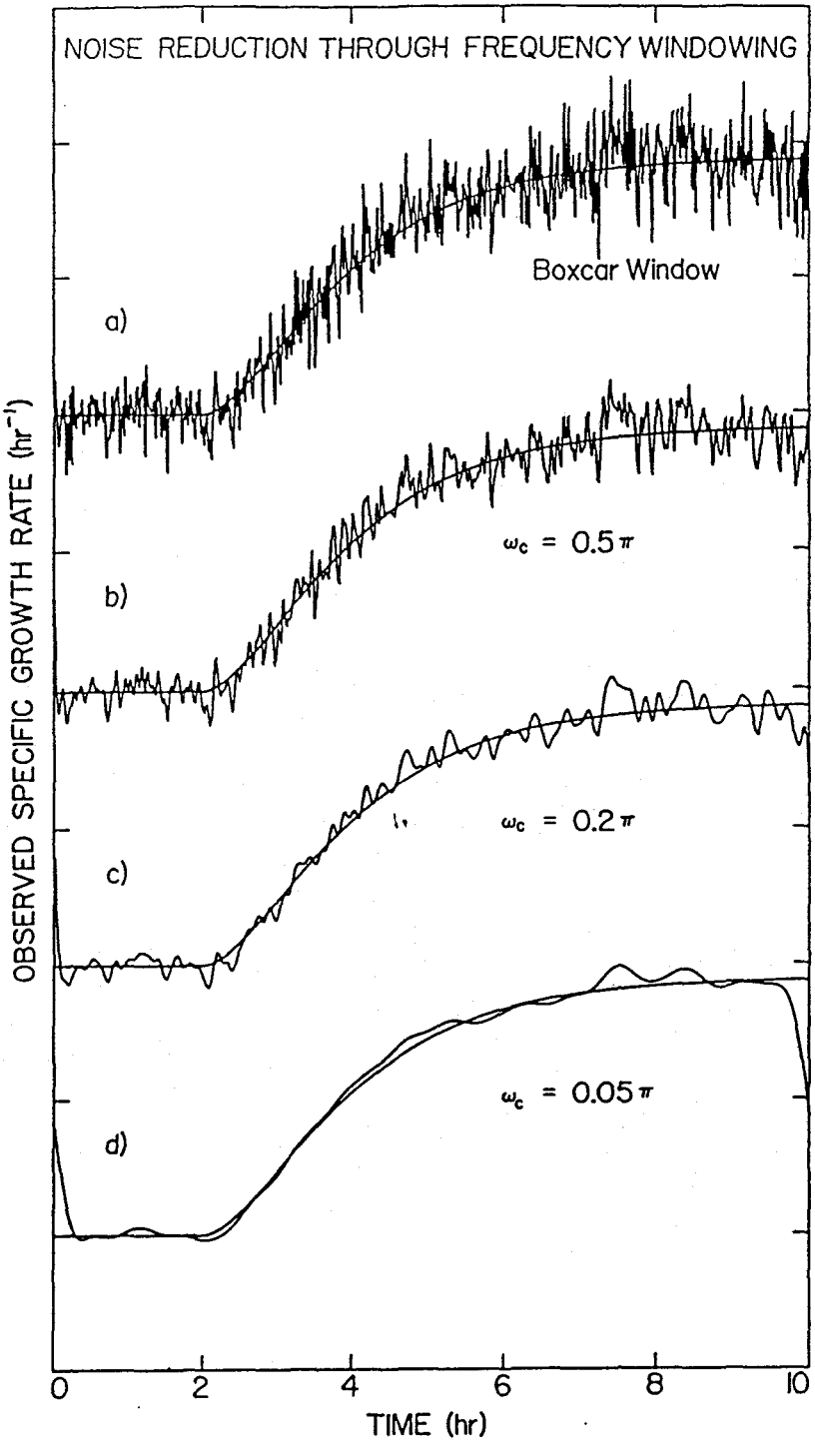
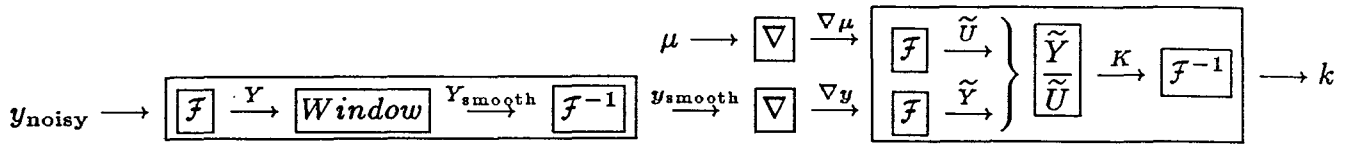
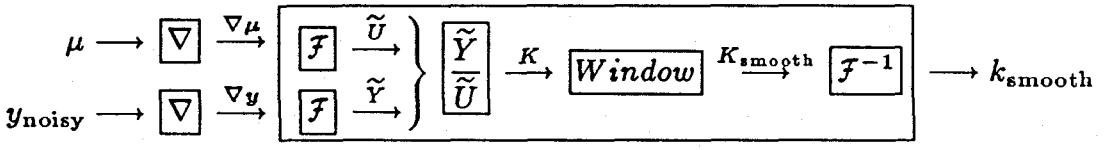


Figure 2.6.13. a) The original function of $y(t)$ with 5% noise. b) Noise reduction via a cosine low-pass filter with $\omega_c = 0.5\pi$. c) $\omega_c = 0.2\pi$. d) $\omega_c = 0.05\pi$

of a frequency window, and the results of the calculation are shown in Figure 2.6.14.



These generally disastrous results, except for perhaps $\omega_c = 0.05\pi$, are not at all surprising, considering the amount of noise still remaining in the smoothed functions. Note that the estimated kernels are not only noisy and unreliable, but the degrees of variation also span a few orders of magnitudes. A slight variation of the above scheme is presented below, and the results of applying frequency windows immediately before taking the final Fourier inversion to obtain the kernel are plotted in Figure 2.6.15.



The above scheme is preferred over applying two separate frequency windows on the noisy $\mu(t)$ and $y(t)$ individually. This is because the subsequent division operation $\frac{\widetilde{Y}}{\widetilde{U}}$ totally cancels out the effect of prior frequency windows; it is also numerically disastrous (*i.e.*, division by 0) beyond the transition band.

The main reason for the failure will be briefly pursued here. For example, Figure 2.6.16 is the power spectral density of the noisy (5%) $y(t)$, where power is defined as:

$$P(\omega) = F(\omega) \cdot F^*(\omega) \tag{2.6.33}$$

In the above equation, "$\star$" denotes the complex conjugate operation. Except for the very low frequency components that yield the deterministic part of the curve, this power spectral density function is basically the Fourier transform of the auto-correlation function of the noise. The rather evenly spread spectrum of the system input verifies that the noise used in this simulation is indeed white, just as white light is composed of photons of various frequencies.

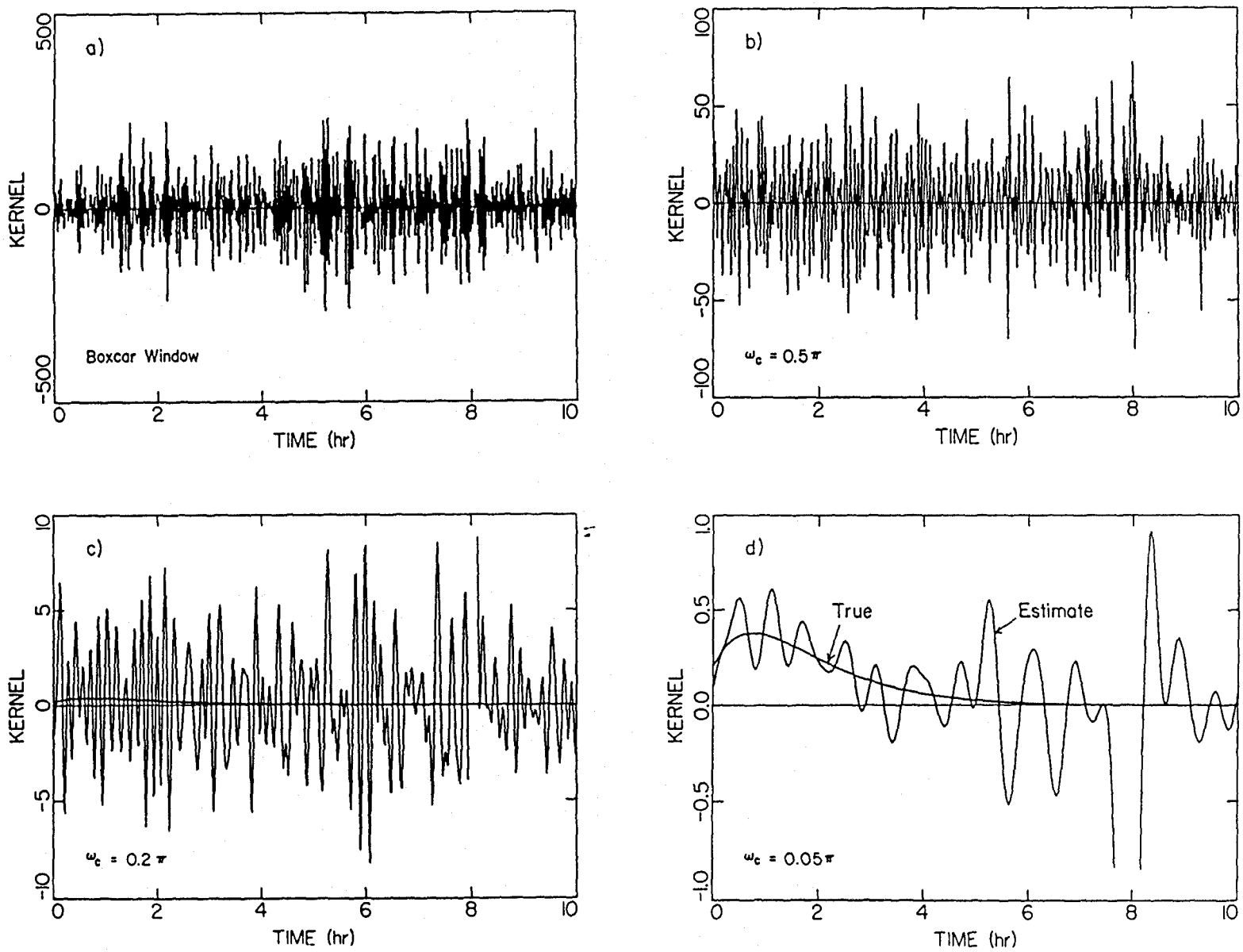


Figure 2.6.14. Totally chaotic kernel obtained after the application of frequency windowing to $\widetilde{Y(\omega)}$. with a) boxcar b) $\omega_c = 0.5\pi$, c) $\omega_c = 0.2\pi$, and d) $\omega_c = 0.05\pi$.

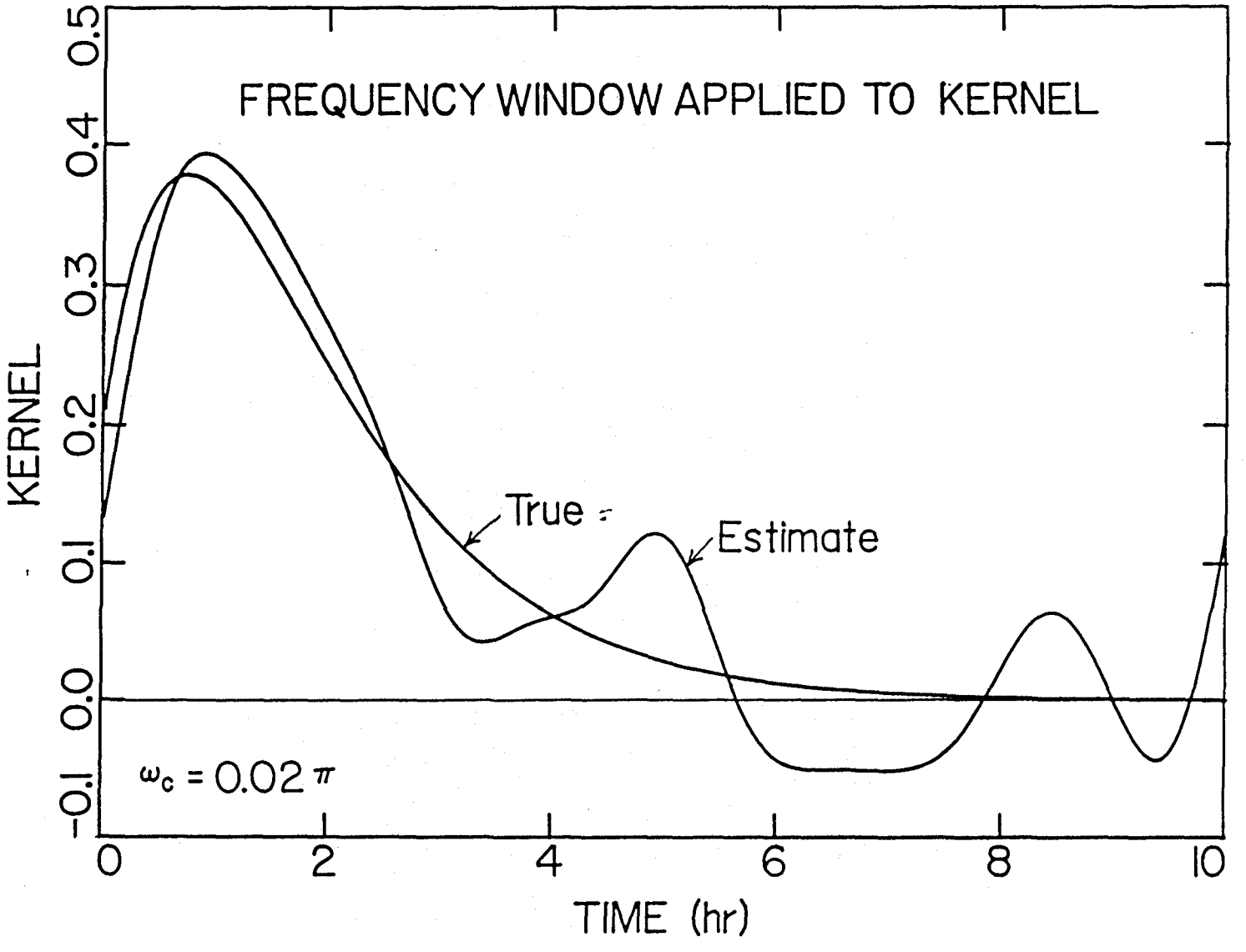


Figure 2.6.15. Kernel obtained after the application of frequency windowing to $\frac{\widetilde{Y(\omega)}}{\widetilde{U(\omega)}}$ with $\omega_c = 0.02\pi$

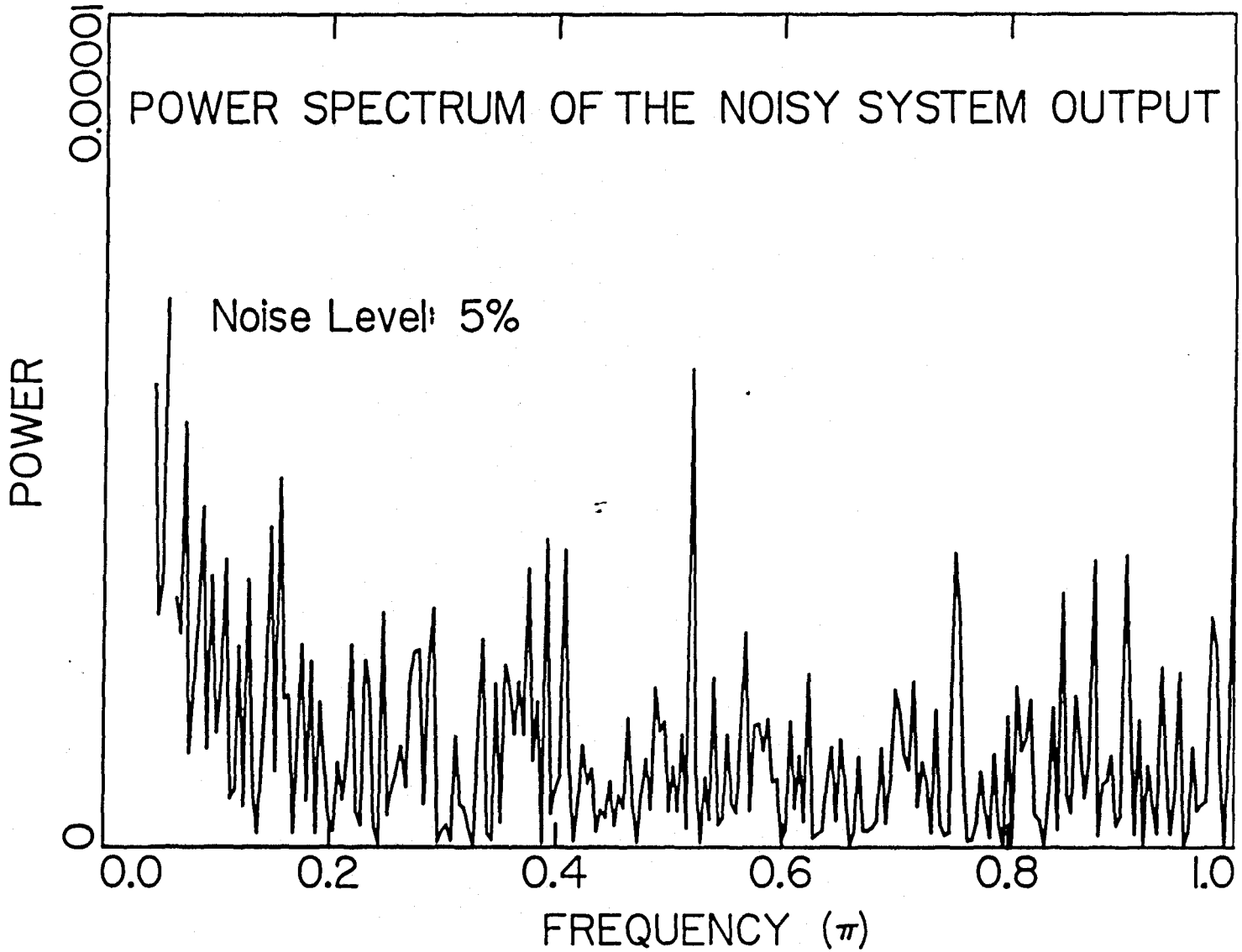


Figure 2.6.16. Power spectrum of the noisy (5%) $y(t)$.

Most deterministic functions have less than half a dozen or so frequencies where the power spectrum is significantly nonzero. For example, the power spectrum of deterministic $\nabla y(t)$ is shown in Figure 2.6.17. Those of other deterministic quantities such as $\mu(t)$, $y(t)$, and $k(t)$ all decay to zero just as quickly, usually within the first five frequencies. All the difficulties with estimating the kernel function via the Fourier transforms, based on a near-step stimulus, is that a first-order difference operation must be carried out beforehand. Because ∇ is a high-pass filter, this operation is highly undesirable from the perspective of model parameter estimation.

The periodogram of the first $k(t)$ presented in Figure 2.6.14, which is estimated from noise free $\mu(t)$ and noisy $y(t)$ with no frequency window, is contrasted against that of the true $k(t)$ in Figure 2.6.18. The power spectrum of the true $k(t)$ approaches zero so quickly, as frequency is increased, that it cannot be shown distinctly in the same figure. The periodograms of other variables involved in this simulation, such as $\nabla\mu(t)$ and $\nabla y(t)$, also display a similar behavior, namely that the low frequency components are highly suppressed while the high frequency components are magnified by the difference operation.

Note that the power for the disastrous $k(t)$ increases almost linearly as frequency is increased; whereas, the power for the deterministic $k(t)$ is all contained within the first few frequencies. The power spectrum of $k(t)$ estimated with $\omega_c = 0.5\pi$, shown in Figure 2.6.19, basically displays the same behavior as the previous one that is without any frequency window, except that the power at high frequencies is reduced by the applied cosine filter with a reflection point at 0.5π. However, it still falls short of cleaning up the undesirable frequency components, and the final kernel estimate is not satisfactory. Other cutoff frequencies did not significantly improve the kernel estimation, nor did the application of another frequency window to $K(\omega)$ immediately before the last inverse Fourier transform.

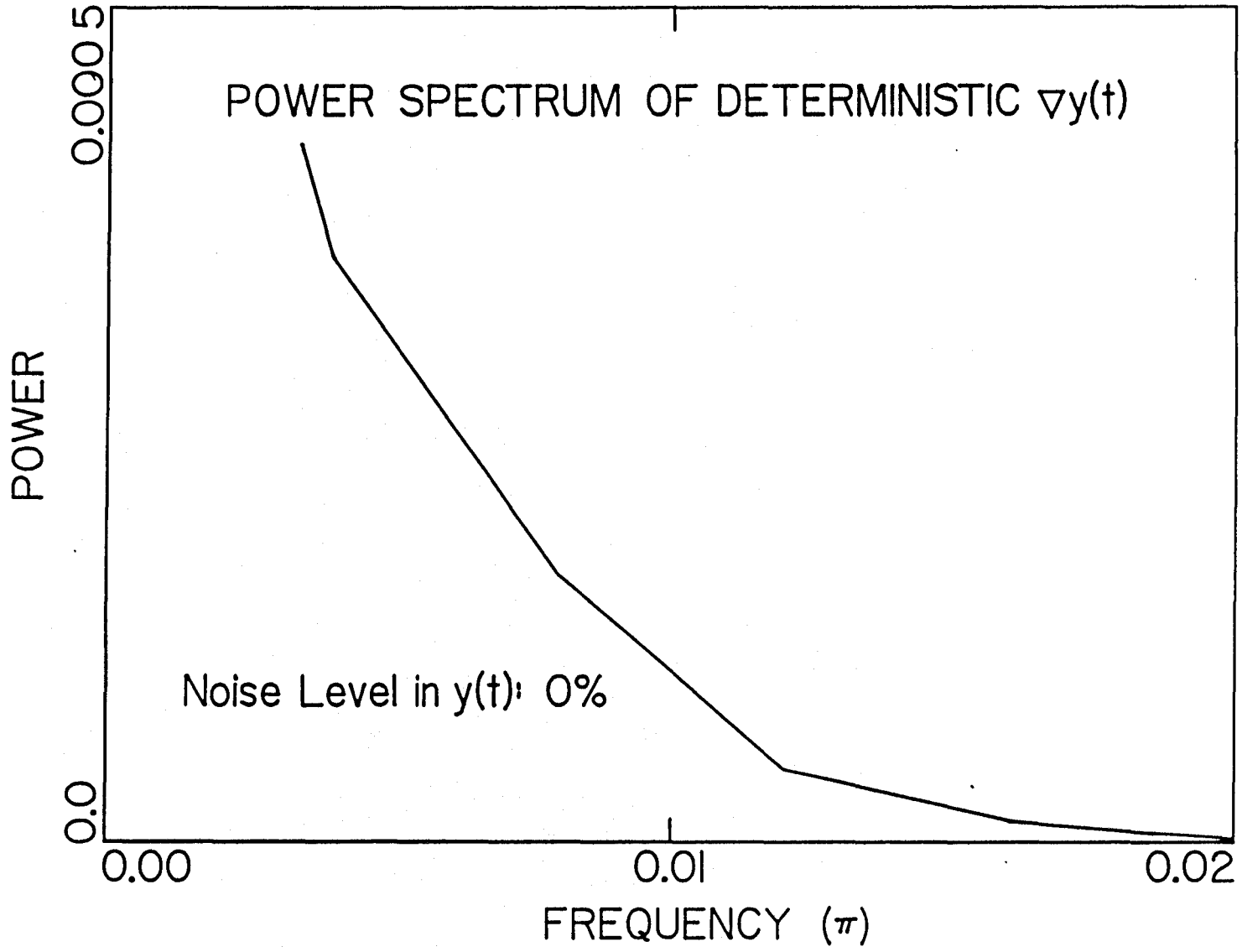


Figure 2.6.17. Power spectrum of deterministic $\nabla y(t)$.

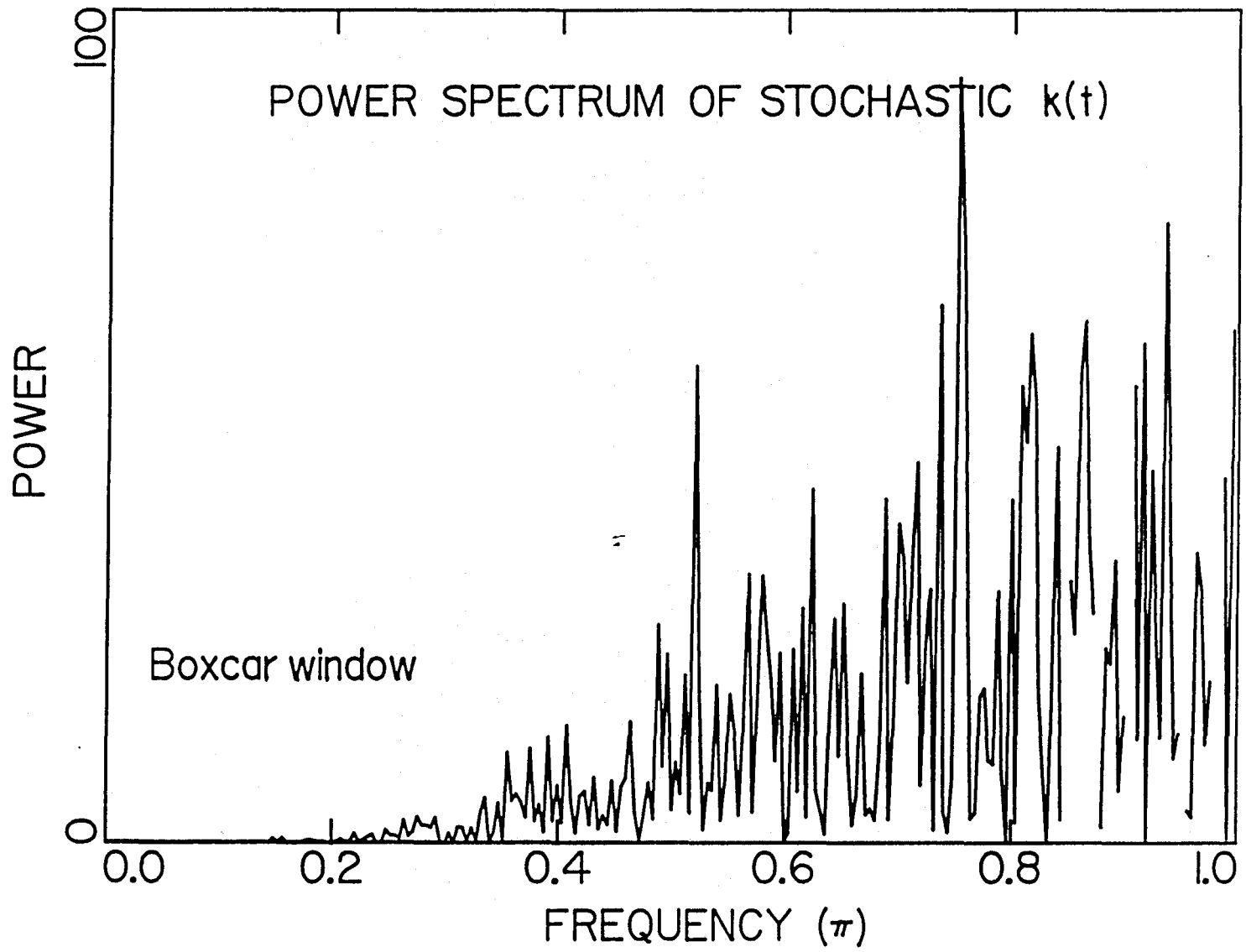


Figure 2.6.18. Periodogram of the estimated $k(t)$ of Figure 2.6.14a. The periodogram of the true $k(t)$ cannot be seen in this scale because all the power is contained in the first few frequencies.

Again, the failure is due to the fact that the small waves in the smoothed functions make the slopes of the smoothed functions *totally* unreliable, although the functions themselves are perfectly usable in certain other applications.

Time Series Analysis

Finally, time series analysis has also been applied to kernel estimation based on the auto-covariance and cross-covariance matrices of the system input and output. Results similar to those of Fourier transforms have been obtained. Namely, the kernel is estimated correctly in the absence of noises. However, the time series method fails badly whenever there are noises in the system output for a step input, despite the fact that this method has been specifically developed to solve just this type of problem in which dynamic models are to be identified from noisy data.

Tapering in the Time Domain

When a time window is applied arbitrarily, the end points may not necessarily match nicely, though the function may be truly periodic. This gives rise to a phenomenon known as *leakage*, which is the direct result of viewing a finite section of an infinitely long data series. Leakage phenomenon is manifested in an oscillatory Fourier transform. To reduce the leakage, tapering techniques analogous to the windowing techniques in the frequency domain can be employed in the time domain to soften the edges of a boxcar window so that less weight is applied to the points near the edge of the window. While many taper functions have been used in the past, the split cosine bell taper shown in Figure 2.6.20 is used in this simulation and in the analysis of sinusoidal dilution rate experiments in the subsequent chapters. However, the use of tapering in the time domain is generally not advised if the curve near two ends of the time window initially is not close to zero, for doing so will highly distort the shapes of the system input and output response curves,

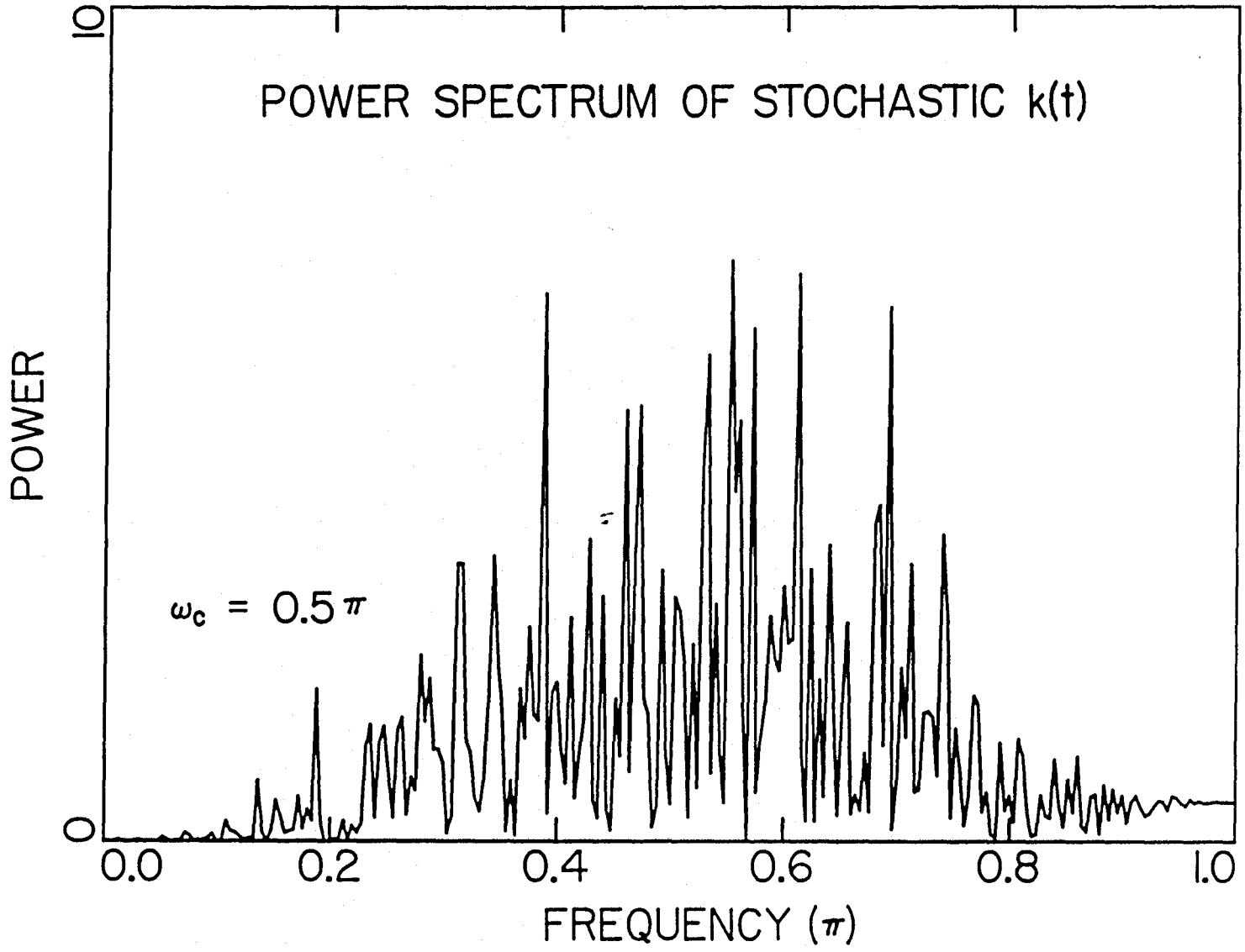


Figure 2.6.19. Periodogram of the estimated $k(t)$ of Figure 2.6.14b. The cosine filter applied cuts off the high frequency components.

resulting in questionable estimates for the kernel.

$$\text{Taper} = \begin{cases} \frac{1}{2}\left[1 - \cos(\frac{t}{t_{\text{taper}}}\pi)\right] & \text{for } 0 \le t \le t_{\text{taper}} \\ 1 & \text{for } t_{\text{taper}} \le t \le t_f - t_{\text{taper}} \\ \frac{1}{2}\left[1 - \cos(\frac{N-t}{t_{\text{taper}}}\pi)\right] & \text{for } t_f - t_{\text{taper}} \le t \le t_f \ , \end{cases} \tag{2.6.34}$$

where t_{taper} is the last point where taper will be applied.

Direct Polynomial Approximation

A variation of smoothing via the polynomial approximation is simply to find the coefficients of least-square fit of the noisy function to the time-lag differential equation, given the order of the polynomial. All the higher derivatives of this function are then directly calculated from these coefficients. In previous sections, it has been derived that the following relationship holds for a linear combination of $k(t) = \sum_{i=0}^{n} a_i k_i(t)$, where $k_i(t)$ is the ith exponential distribution function $\frac{1}{T}\frac{t}{T}^i e^{-\frac{t}{T}}$.

$$\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{n} \left[\sum_{j=i}^{n} \binom{j}{i} a_{n-j} \right] T^i \frac{d^i \mu(t)}{dt^i} \tag{2.6.35}$$

Specifically, for the first three kernels, namely the 0th-order $k(t) = \frac{1}{T}e^{-\frac{t}{T}}$, the first-order $k(t) = \left(a_0\frac{1}{T} + a_1\frac{t}{T^2}\right)e^{-\frac{t}{T}}$ with $(a_0 + a_1 = 1)$, and the second-order $k(t) = \left(a_0\frac{1}{T} + a_1\frac{t}{T^2} + a_2\frac{t^2}{T^3}\right)e^{-\frac{t}{T}}$ with $(a_0 + a_1 + a_2 = 1)$, the following respective linear differential equations are satisfied.

$$T\frac{dy(t)}{dt} + y(t) = \mu(t) \tag{2.6.36}$$

$$T^2\frac{d^2y(t)}{dt^2} + 2T\frac{dy(t)}{dt} + y(t) = \mu(t) + a_0 T\frac{d\mu(t)}{dt} \tag{2.6.37}$$

$$T^3\frac{d^3y(t)}{dt^3} + 3T^2\frac{d^2y(t)}{dt^2} + 3T\frac{dy(t)}{dt} + y(t) = \mu(t) + (2a_0 + a_1)T\frac{d\mu(t)}{dt} + a_0T^2\frac{d^2\mu(t)}{dt^2} \tag{2.6.38}$$

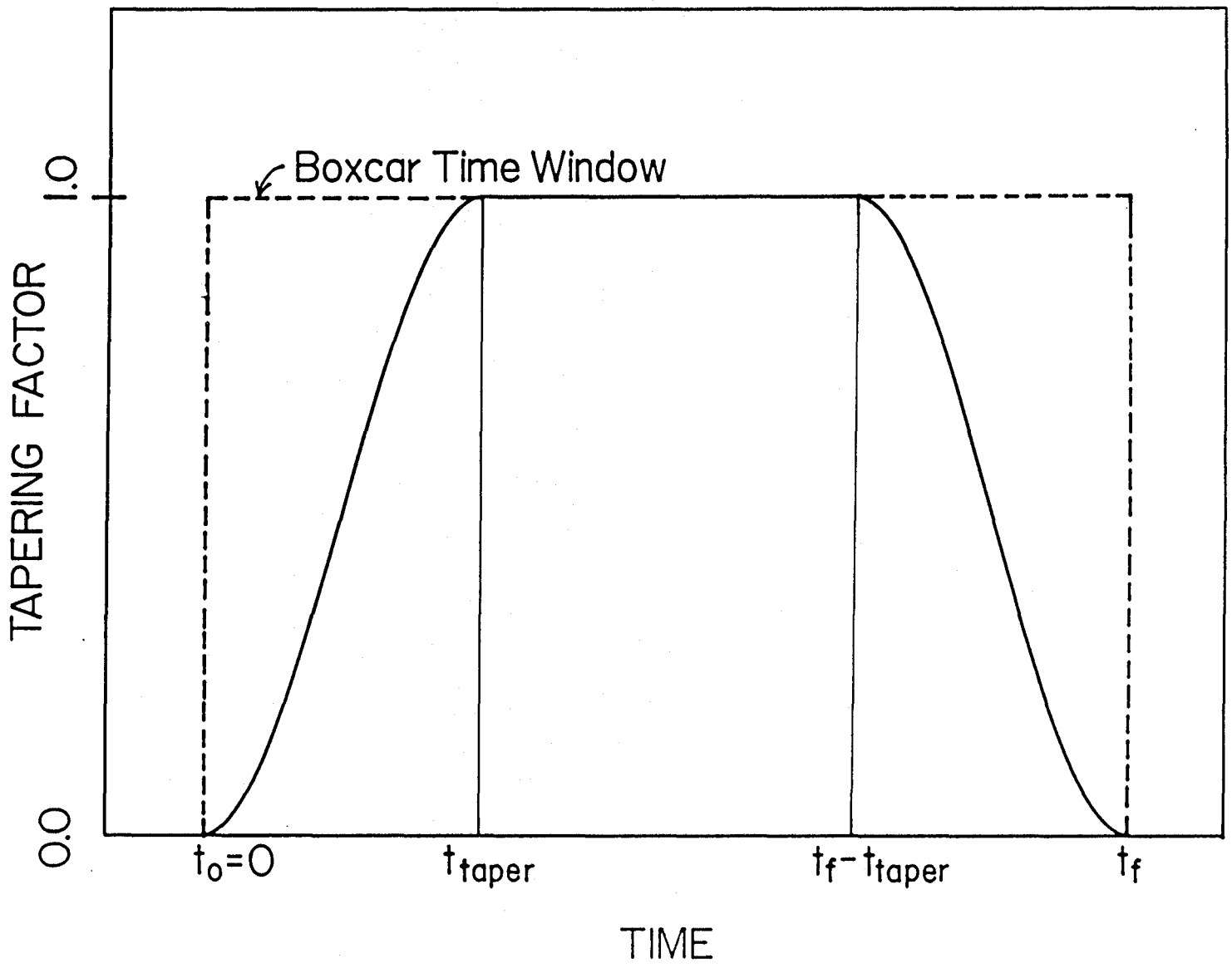


Figure 2.6.20. A taper function is applied to the boxcar time window to soften the sharp edges.

Equations (2.6.36)-(2.6.38) can be written in a general form:

$$g(t) \equiv \sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{n} \alpha_i \frac{d^i \mu(t)}{dt^i}, \tag{2.6.39}$$

where $\alpha_i = T^i \sum_{j=i}^{n} \binom{j}{i} a_{n-j}$. In this estimation scheme, a suitable order n is first assumed for the kernel to be estimated, and a value for T is subsequently assumed to start the process. Then the left hand side of the above equation, *i.e.*, $g(t) \equiv \sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i}$ is evaluated and treated in the linear regression scheme as the dependent variable. Subsequently, the coefficients α_is are obtained by approximating the function $g(t)$ with the remaining $n+1$ base functions of $\frac{d^i \mu(t)}{dt^i}$, which can be directly evaluated from the polynomial approximation to the measured $\mu(t)$.

In the stochastic digital domain, the last equation can be expressed as:

$$\bar{g}_t = \alpha_0 \mu_t + \alpha_1 \left[\frac{d\mu}{dt}\right]_t + \alpha_2 \left[\frac{d^2\mu}{dt^2}\right]_t + \ldots + \alpha_n \left[\frac{d^n\mu}{dt^n}\right]_t + \epsilon_t \qquad t = 0, 1, 2, \ldots, \tag{2.6.40}$$

where $\bar{g}_t$ denotes the estimated values of $g(t)$ based on the smoothed $y(t)$, and ϵ_t the errors due to the fact that the above correlation does not always hold true in the presence of noises or the kernel is not exactly a linear combination of the base exponential distribution functions. Expressed in a matrix format, ready to be subjected to numerical calculation, the above equation is:

$$\underbrace{\begin{bmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \bar{g}_2 \\ \vdots \\ \bar{g}_t \end{bmatrix}}_{\bar{\mathbf{g}}} = \underbrace{\begin{bmatrix} \mu_0 & \left[\frac{d\mu}{dt}\right]_0 & \left[\frac{d^2\mu}{dt^2}\right]_0 & \cdots & \left[\frac{d^n\mu}{dt^n}\right]_0 \\ \mu_1 & \left[\frac{d\mu}{dt}\right]_1 & \left[\frac{d^2\mu}{dt^2}\right]_1 & \cdots & \left[\frac{d^n\mu}{dt^n}\right]_1 \\ \mu_2 & \left[\frac{d\mu}{dt}\right]_2 & \left[\frac{d^2\mu}{dt^2}\right]_2 & \cdots & \left[\frac{d^n\mu}{dt^n}\right]_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_t & \left[\frac{d\mu}{dt}\right]_t & \left[\frac{d^2\mu}{dt^2}\right]_t & \cdots & \left[\frac{d^n\mu}{dt^n}\right]_t \end{bmatrix}}_{\boldsymbol{\mu}} \cdot \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}}_{\boldsymbol{\alpha}} + \underbrace{\begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_t \end{bmatrix}}_{\boldsymbol{\epsilon}}, \tag{2.6.41}$$

Or in a more compact notation:

$$\bar{\mathbf{g}} = \boldsymbol{\mu} \cdot \boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{2.6.42}$$

where the columns of μ are assumed to be mutually linearly independent. This rank condition is automatically satisfied if the polynomial used to estimate $\mu(t)$ is of at least degree n. The sum of squared errors is:

$$J = \sum_{j=0}^{t} \epsilon_j^2 = \epsilon^T \epsilon = (\bar{\mathrm{g}} - \mu \cdot \alpha)^T (\bar{\mathrm{g}} - \mu \cdot \alpha). \tag{2.6.43}$$

From linear regression theories, the estimate of α, denoted henceforth as $\hat{\alpha}$, when J is minimized is:

$$\hat{\alpha} = \mu^T \mu^{-1} \mu^T \mathrm{g}. \tag{2.6.44}$$

Finally the value of J at the minimum is:

$$J_{\min} = \bar{\mathrm{g}}^T \bar{\mathrm{g}} - \hat{\alpha} \mu^T \mu \hat{\alpha}. \tag{2.6.45}$$

For all orders of $k(t)$, the first coefficient corresponding to the term $\mu(t)$, *i.e.*, α_0 should be unity. If not, another value of T is assumed and the least-square fitting process is repeated. Finally, when the coefficient corresponding to $\mu(t)$ converges to unity, the remaining coefficients α_is are converted to a_is, with which the kernel function is now completely defined. This process is repeated with a new assumption on the order of the kernel.

As a demonstration, this technique is applied to the same example of a step increase in the dilution rate. The following set of coefficients is obtained assuming the degree of the least-square polynomial for $\mu(t)$ to be 7 and that for $y(t)$ to be 4; these, as shown previously, were the degrees that proved to represent adequately

the corresponding noisy curves.

Coefficients from Linear Regression								
	1	t	t^2	t^3	t^4	t^5	t^6	t^7
μ	3.63e-1	2.71e-1	-2.37e-1	1.04e-1	-2.52e-2	3.40e-3	-2.41e-4	6.98e-6
μ'	2.71e-1	-4.73e-1	3.12e-1	-1.01e-1	1.70e-2	-1.45e-3	4.89e-5	
μ''	-4.73e-1	6.24e-1	-3.02e-1	6.81e-2	-7.24e-3	2.93e-4		
μ'''	6.24e-1	-6.04e-1	2.04e-1	-2.90e-2	1.47e-3			
$\mu^{(4)}$	-6.04e-1	4.08e-1	-8.69e-2	5.86e-3				
$\mu^{(5)}$	4.08e-1	-1.74e-1	1.76e-2					
$\mu^{(6)}$	-1.74e-1	3.52e-2						
$\mu^{(7)}$	3.52e-2							

Coefficients from Linear Regression					
	1	t	t^2	t^3	t^4
y	2.88e-1	7.19e-2	-6.90e-3	-2.44e-5	1.88e-5
y'	7.19e-2	-1.38e-2	-7.31e-5	7.52e-5	
y''	-1.38e-2	-1.46e-4	2.26e-4		
y'''	-1.46e-4	4.51e-4			
$y^{(4)}$	4.51e-4				

From these curves the vector $\bar{\mathbf{g}}$ is constructed from y, y', ..., and the matrix μ

from μ, μ', ..., for the assumed order of the kernel according to the following table.

Order	Time-Lag Differential Equation
True	$y'' + 2y' + y = \mu + 0.2\mu'$
0	$Ty' + y = \alpha_0\mu$
1	$T^2y'' + 2Ty' + y = \alpha_0\mu + \alpha_1\mu'$
2	$T^3y''' + 3T^2y'' + 3Ty' + y = \alpha_0\mu + \alpha_1\mu' + \alpha_2\mu''$
3	$T^4y'''' + 4T^3y''' + 6T^2y'' + 4Ty' + y = \alpha_0\mu + \alpha_1\mu' + \alpha_2\mu'' + \alpha_3\mu'''$

For example, when n is assumed to be 1, as the assumed kernel time constant T changes, so do the the coefficients α_0 and α_1 calculated through least-square regression. Shown in Figure 2.6.21 is the variation in the least-square regression estimate of the first coefficient (α_0) on the right hand side of the equation in the above table. The kernel time constant T corresponding to where α_0 is unity is 1.153 hour. From the variations of α_0 and α_1 as a function of T, the fraction of the 0th-order component in the composite kernel, namely a_0, is calculated and is shown in Figure 2.6.22. The least-square estimates of the kernel time constant and coefficients subject to the condition that $\alpha_0 = 1$ are displayed below for other assumed kernel orders.

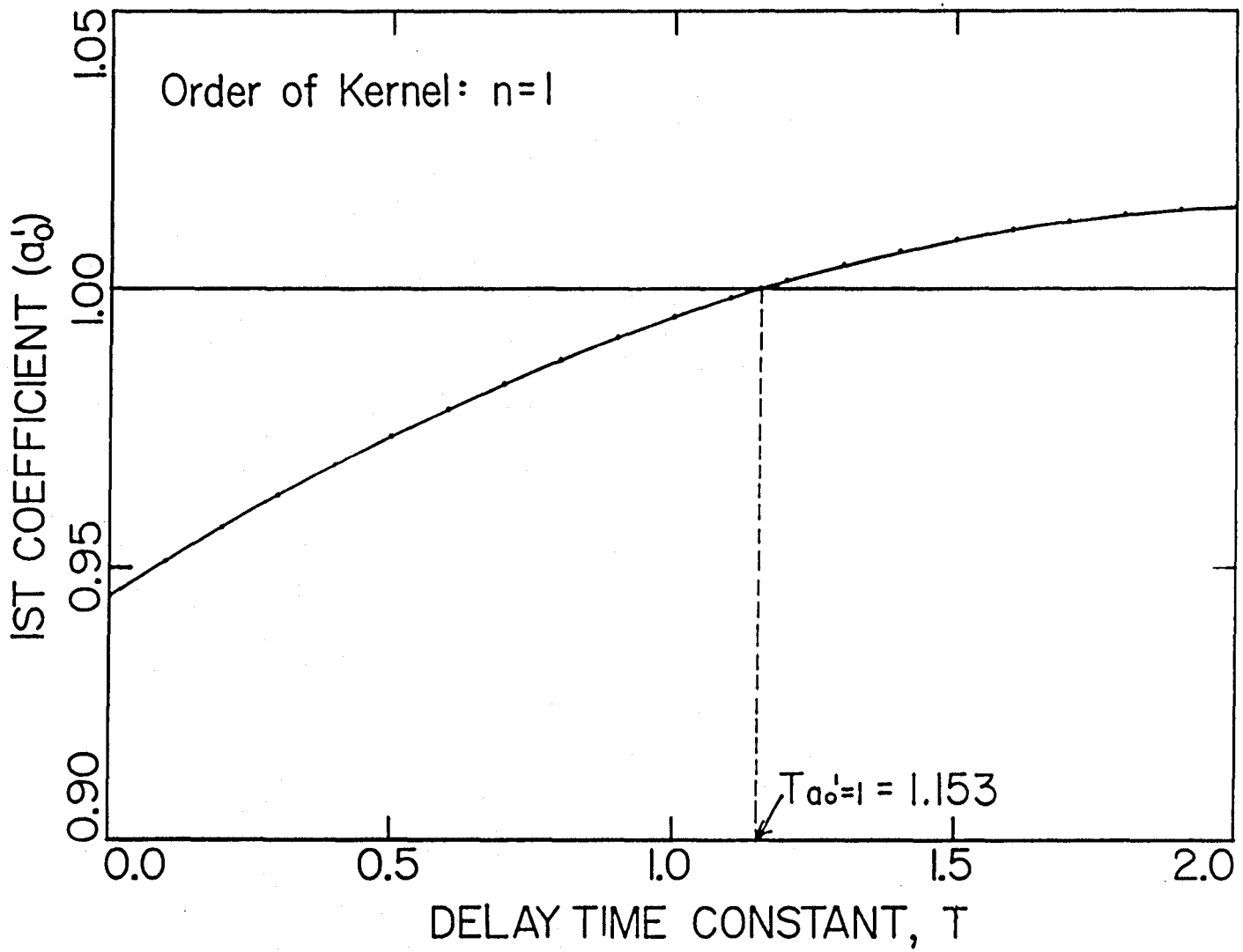


Figure 2.6.21. Variation of the least-square estimated first coefficient associated with $\mu(t)$ in the differential equation $\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{n} \alpha_i \frac{d^i \mu(t)}{dt^i}$ as a function of the assumed kernel time constant T.

$$k(t) = a_0 k_0(t) + a_1 k_1(t)$$

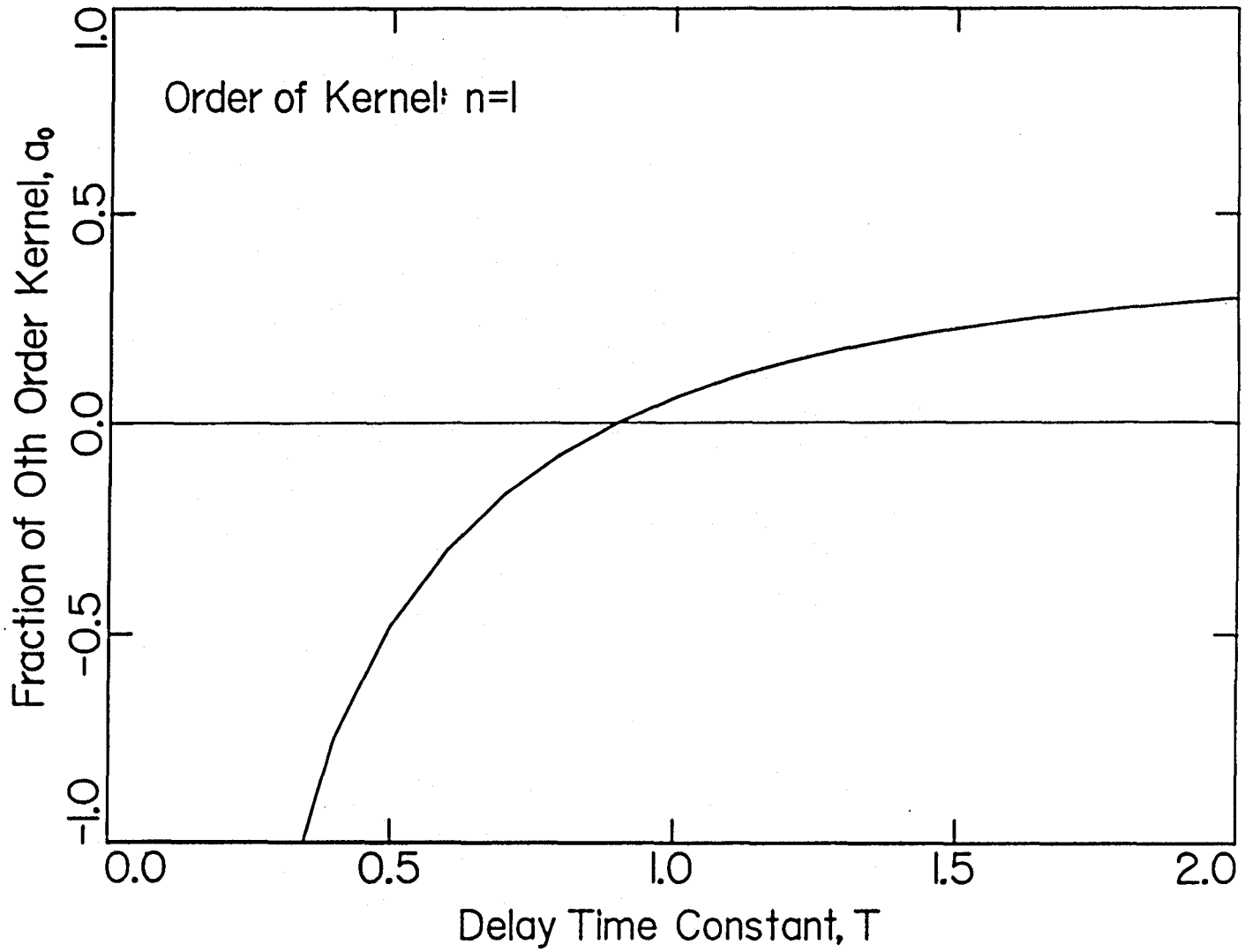


Figure 2.6.22. Variation of the 0th-order kernel component (a_0) as a function of the assumed kernel time constant T. ($k(t) = a_0 k_0(t) + a_1 k_1(t)$.)

Summary of Least-Square Values							
n	J	T	α_0	α_1	α_2	α_3	α_4
0	0.0891	1.686	1.0000				
1	0.0655	1.153	1.0000	0.1480			
2	0.0589	0.922	1.0000	0.1322	-0.0568		
3	30.0917	3.574	1.0000	5.2856	5.5902	1.627	
4	1.5139	2.043	1.0000	-3.6392	-2.1596	1.534	1.582
1	True	1.	1.0000	0.2			

Note that the values of the error function J at $\alpha_0 = 1$ are about the same for $n = 0$, $n = 1$, and $n = 2$. However, they increase significantly for $n \geq 3$. From these values of J, one can be quite confident that the order of the kernel is either 0, 1, or 2, but probably not 3 or more.

Based on these estimates of the coefficients in the time-lag differential equation, the kernel parameters are completely specified by Equation (2.6.39). The coefficients on the right hand side of that equation represent the contribution from each exponential distribution base function, namely $T^i \sum_{j=i}^{n} \binom{j}{i} a_{n-j}$. They are listed below

for the first ten kernel orders. It is basically a table of binomial coefficients.

Coefficients for $\frac{d^i\mu(t)}{dt^i}$ on the RHS of Equation (2.6.37)										
Order of					i					
Kernel (n)	0	1	2	3	4	5	6	7	8	9
0	1									
1	1	1								
2	1	2	1							
3	1	3	3	1						
4	1	4	6	4	1					
5	1	5	10	10	5	1				
6	1	6	15	20	15	6	1			
7	1	7	21	35	35	21	7	1		
8	1	8	28	56	70	56	28	8	1	
9	1	9	36	84	126	126	84	36	9	1

For example, for $n = 3$, all rows up to and including $n = 3$ in the above table are extracted and shown below.

Example: Coefficients for n=3				
	$i\ \left(\times T^i \frac{d^i\mu(t)}{dt^i}\right)$			
	0	1	2	3
a_3	1			
a_2	1	1		
a_1	1	2	1	
a_0	1	3	3	1

Directly from this table, the following set of equations is obtained.

$$\begin{aligned}
\text{1st column:}\quad \alpha_0 &= (1\cdot a_3 + 1\cdot a_2 + 1\cdot a_1 + 1\cdot a_0\ ) \times T^0\mu \\
\text{2nd column:}\quad \alpha_1 &= (\quad 1\cdot a_2 + 2\cdot a_1 + 3\cdot a_0\ ) \times T^1\frac{d\mu(t)}{dt} \\
\text{3rd column:}\quad \alpha_2 &= (\quad\quad 1\cdot a_1 + 3\cdot a_0\ ) \times T^2\frac{d^2\mu(t)}{dt^2} \\
\text{4th column:}\quad \alpha_3 &= (\quad\quad\quad 1\cdot a_0\ ) \times T^3\frac{d^3\mu(t)}{dt^3}
\end{aligned}$$

These equations give rise to:

$$\underbrace{\begin{bmatrix}\alpha_0\\ \alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix}}_{\boldsymbol{\alpha}} = \underbrace{\begin{bmatrix}1 & 1 & 1 & 1\\ 3 & 2 & 1 & 0\\ 3 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}}_{\mathbf{B}} \cdot \underbrace{\begin{bmatrix}a_0\\ a_1\\ a_2\\ a_3\end{bmatrix}}_{\mathbf{a}} . \tag{2.6.46}$$

Because the above transformation matrix **B** has a triangular form, the coefficients can be easily solved in a reverse order to derive the kernel parameters. Finally, the results are presented in the table below, and the corresponding estimated kernels are plotted in Figure 2.6.23.

Summary of Estimated Kernel Parameters

n	T	a_0	a_1	a_2	a_3	a_4
0	1.686	1.0000				
1	1.153	0.1284	0.887			
2	0.922	-0.0668	0.277	0.790		
3	3.574	0.0356	0.331	0.710	-0.0769	
4	2.043	0.0908	-0.183	-0.512	-0.5700	2.175
1	1.	0.2	0.8	True		

As predicted by the error function J, reasonable estimates are obtained with $n = 0$, $n = 1$, and $n = 2$. The reason that the estimated kernels, although close, do not exactly match the true one is that although the polynomial approximations of the system input and output functions quite accurately describe these curves,

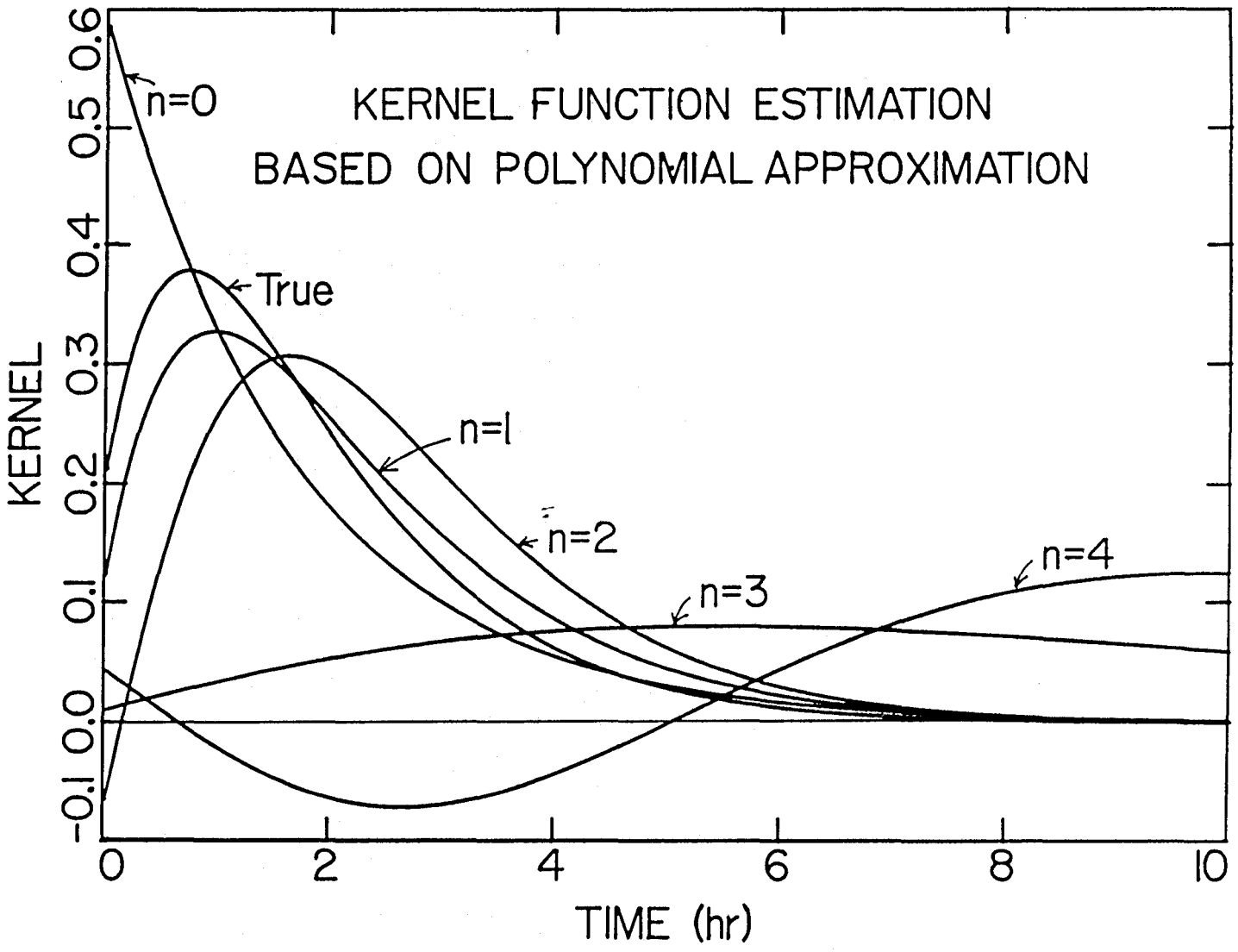


Figure 2.6.23. Estimated kernels based on the 7th-degree polynomial approximation of noisy $\mu(t)$ and 4th-degree polynomial approximation of noisy $y(t)$.

the derivatives of these approximations do not represent the *slopes* of $\mu(t)$ and $y(t)$ as well as the functions themselves. This fact is shown in Figure 2.6.24 where the first-order derivative of the polynomial approximation of $y(t)$ is contrasted with that of the true one. The agreement deteriorates even more quickly for higher-order derivatives.

It is important to note that the *minimum* value of $J = \sum_t \epsilon_t^2$ over the entire range of T generally decreases as the order of the kernel is increased. This is to be expected since there are now more parameters that can be manipulated to ensure a better fit. The error functions for the first few orders are plotted against T in Figure 2.6.25. Furthermore, the minima of these error functions are shown in Figure 2.6.26.

However, the value of T that gives rise to the minimum error function does not always yield the parameters needed to satisfy the condition that $\alpha_0 = 1$. The following table gives the two values of T, the first one corresponding to $\alpha_0 = 1$ and

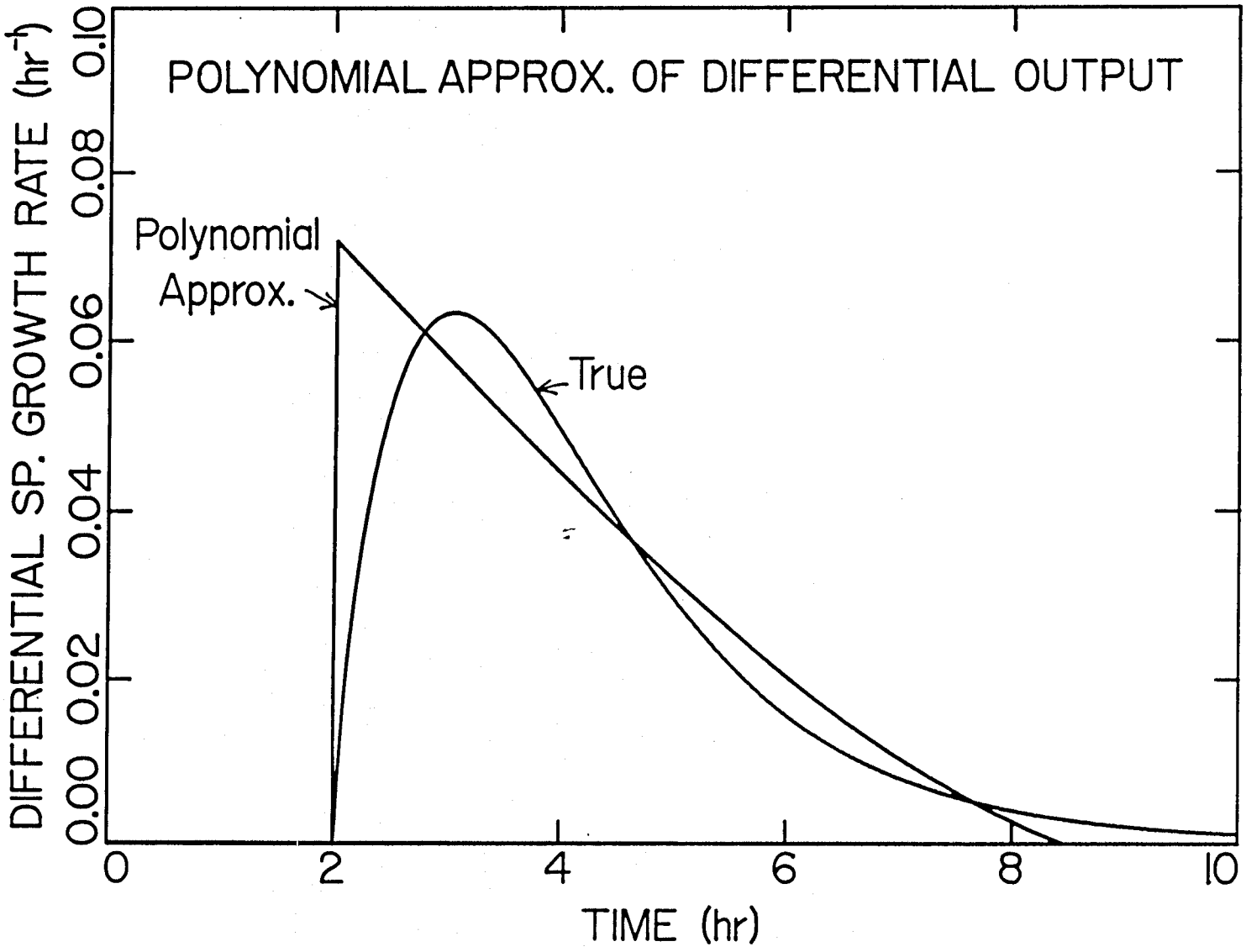


Figure 2.6.24. Comparison of the first-order derivative of the polynomial approximation of the system output with that of the true value.

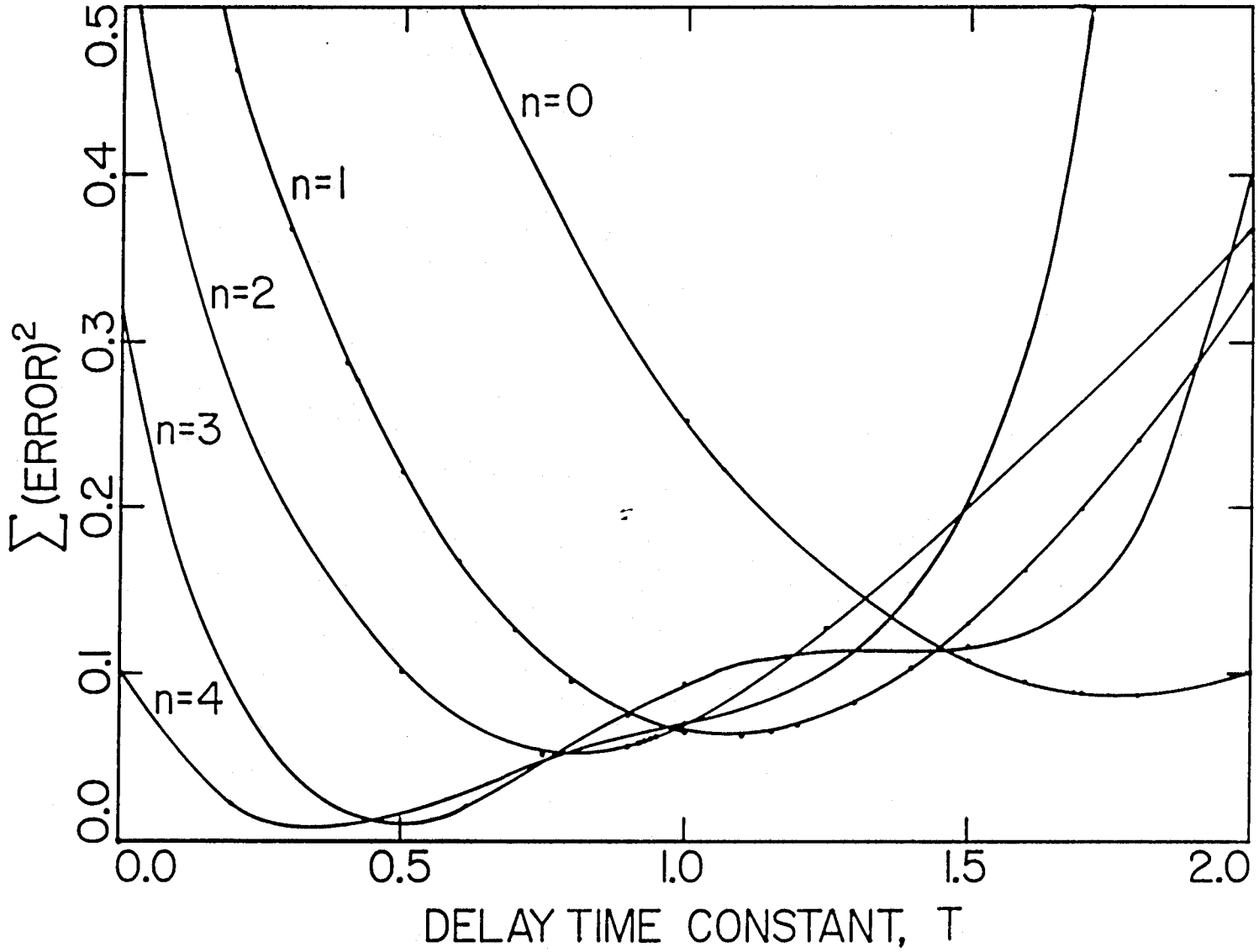


Figure 2.6.25. Least-square error of the kernel estimate as a function of the delay time constant T for different assumed kernel orders.

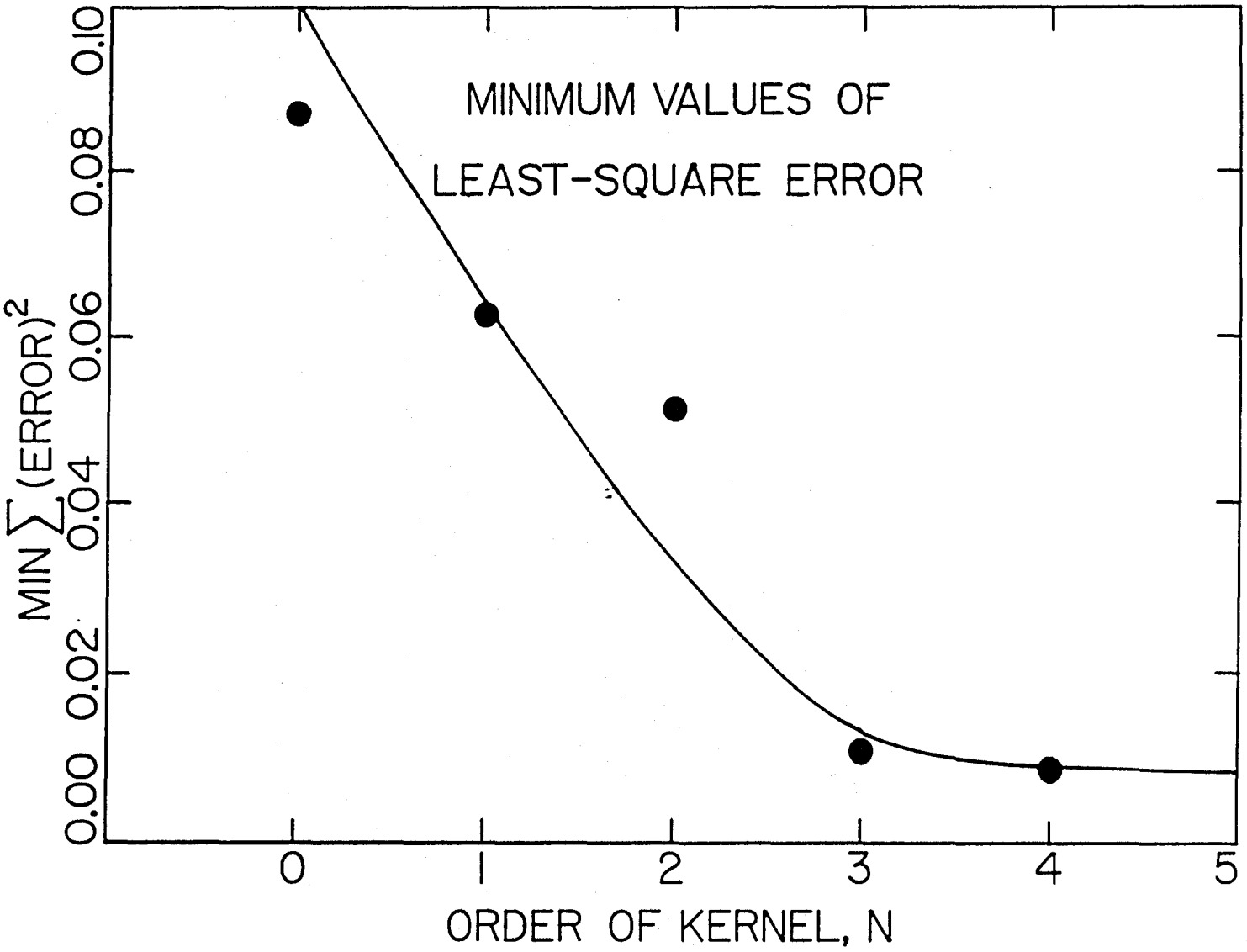


Figure 2.6.26. Minimum values of the error function versus assumed kernel orders.

the second one corresponding to minimum J.

Summary Error Functions and Kernel Parameters				
n		T	α_0	J
0	(a_0=1)	1.686	1.0000	0.0891
0	(min J)	1.766	1.0000	0.0870
1	(a_0=1)	1.153	1.0000	0.0655
1	(min J)	1.071	0.9974	0.0627
2	(a_0=1)	0.922	1.0000	0.0589
2	(min J)	0.810	0.9965	0.0511
3	(a_0=1)	3.574	1.0000	30.0917
3	(min J)	0.480	0.9980	0.0108
4	(a_0=1)	2.043	1.0000	1.5139
4	(min J)	0.345	0.9956	0.0087

The discrepancy between the two values of T can be described by a factor defined as the ratio of $\frac{T_{\alpha_0}}{T_{\min}}$. If a kernel can indeed be expressed as a linear combination of n independent exponential distribution base functions, then in the absence of any noise, $\alpha_0 = 1$ should occur exactly when there is a perfect match, *i.e.* when $J = 0$. When the system is corrupted by noises, the T where α_0 is unity should be close to the T where the error function is minimized. The equivalence of these two delay time constants results in a unit discrepancy factor. Thus, one expects this discrepancy factor to be unity if kernel order is correctly guessed. This factor is plotted in Figure 2.6.27 as a function of the assumed order of the kernel. Once again, from this plot it can be concluded that the order of the kernel is either 0, 1,

or 2, with $n = 1$ being the best guess because it yields a discrepancy factor nearest to one. The original true kernel used for this simulation indeed has $n = 1$.

Finally, it should be noted that the above table indicates that α_0 is not at all sensitive to variations in other kernel parameters such as T. Similarly, the error function J near its minimum value is relatively flat and is insensitive to variations in parameters such as T. See Figure 2.6.25.

Iterative Cyclic Approximation

In the Fourier transform method considered previously, it was demonstrated that Fourier transforms cannot be applied naively and directly on the system input and output pair, for the convolution theorem does not hold for non-periodic functions. Rather, the transform must be applied on the differenced quantities when the system input closely resembles a step function where the function values at the beginning of the experiment are significantly different from those at the end. The need for difference operations is based on the implied periodical assumption on the functions to be transformed. The failure of the Fourier transform method in the presence of noise in the system output is attributed to difference operations, which amplified the noise. The next method considered for kernel inversion attempts to modify the system input and output so that Fourier transforms can be applied on them *directly*, instead of on the differenced quantities.

First, the problem of how the system input and/or output functions should be modified will be addressed. To begin, there is no question that the integral relationship between the system input $\mu(t)$, and the system output $y(t)$ in the continuous time domain described below is true:

$$y(t) = \int_{-\infty}^{t} \mu(h)k(t-h)dh. \qquad (2.6.47)$$

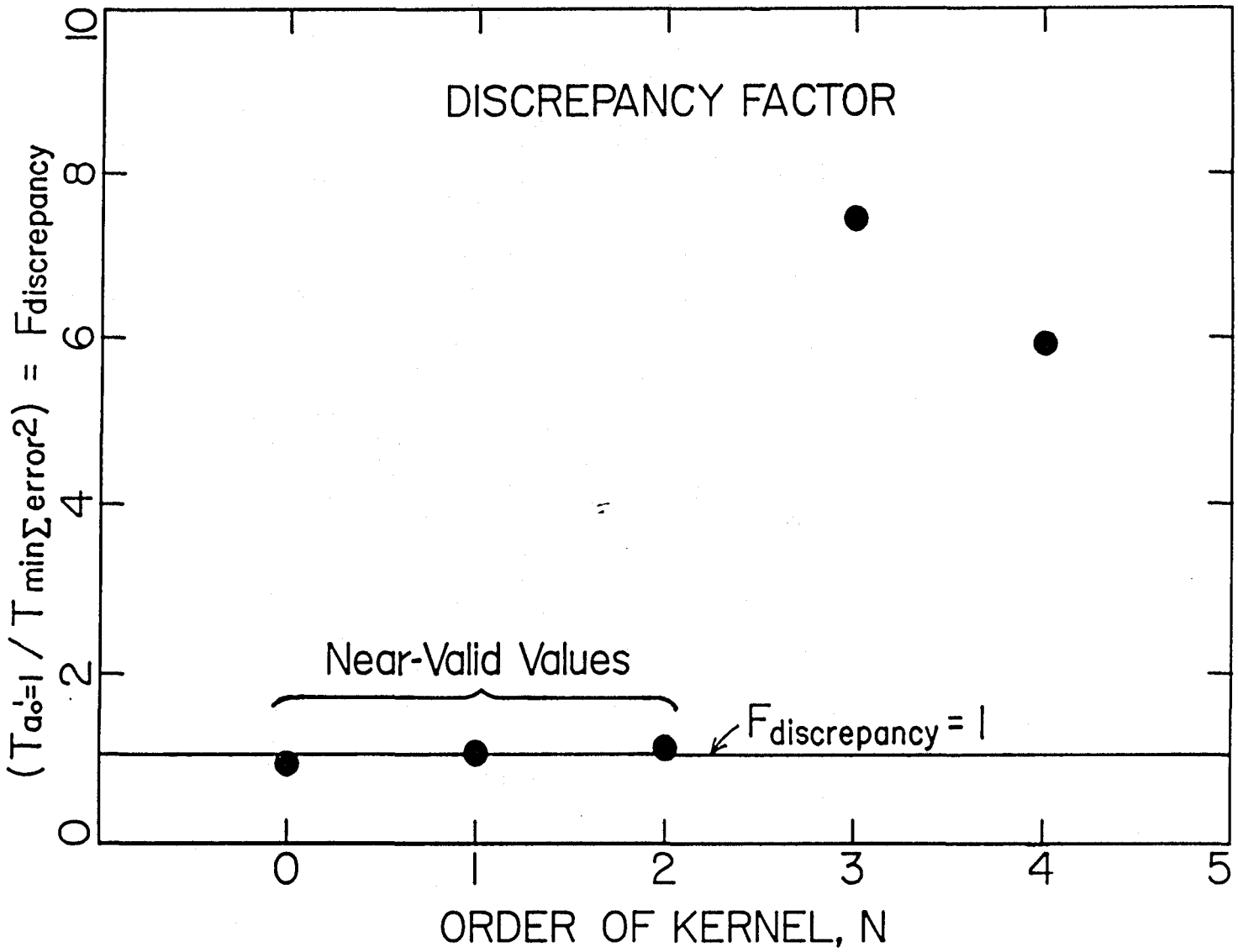
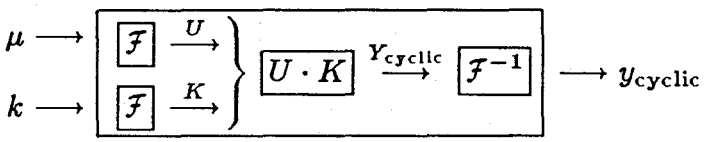


Figure 2.6.27. Discrepancies between the Ts where $\alpha_0 = 1$ and where the error function is at its minimum.

Furthermore, the corresponding digital form is also valid within the time window:

$$y_t = \sum_{j=-\infty}^{t} \mu_{t-j} k_j \Delta t. \tag{2.6.48}$$

If the system is perturbed from an established steady-state, then $\mu_j = 0$ for $j = -1, -2, -3, \ldots$, where μ_t and y_t are formulated as perturbation variables. Figure 2.6.28 shows the system output calculated this way and the function calculated according to the following operation in accordance with the convolution theorem of the Fourier transform, which states that the transform of the system output is the product of the transform of the system input and the transform of the kernel.



Thus, if the function $y_{\text{cyclic}}(t)$ can be approximated, one can directly apply a Fourier transform to obtain the kernel.

A system with an ideal, perfect step system input at $t = t^*$ is presented in Figure 2.6.29. A Fourier transform assumes that the function will repeat the same pattern outside the time window, *i.e.*, infinitely periodical. From the Fourier transform's perspective, the system effectively has two disturbances, a step up at $t = t^*$ and a step down at $t = 0$ (or at $t = t_f$), respectively designated as $\mu_a(t)$ and $\mu_b(t)$ in the figure, although the step down input is artificially created by the time window in the transform and does not exist in reality.

$$\mu_{\text{cyclic}}(t) = \mu_a(t) + \mu_b(t). \tag{2.6.49}$$

In essence, the second disturbance is a virtual one. Since the convolution integrals and, hence, the Fourier transforms are additive, one can treat the system input as two separate step functions and calculate the system response to each of the stimuli independently. These individual system output functions are noted in the

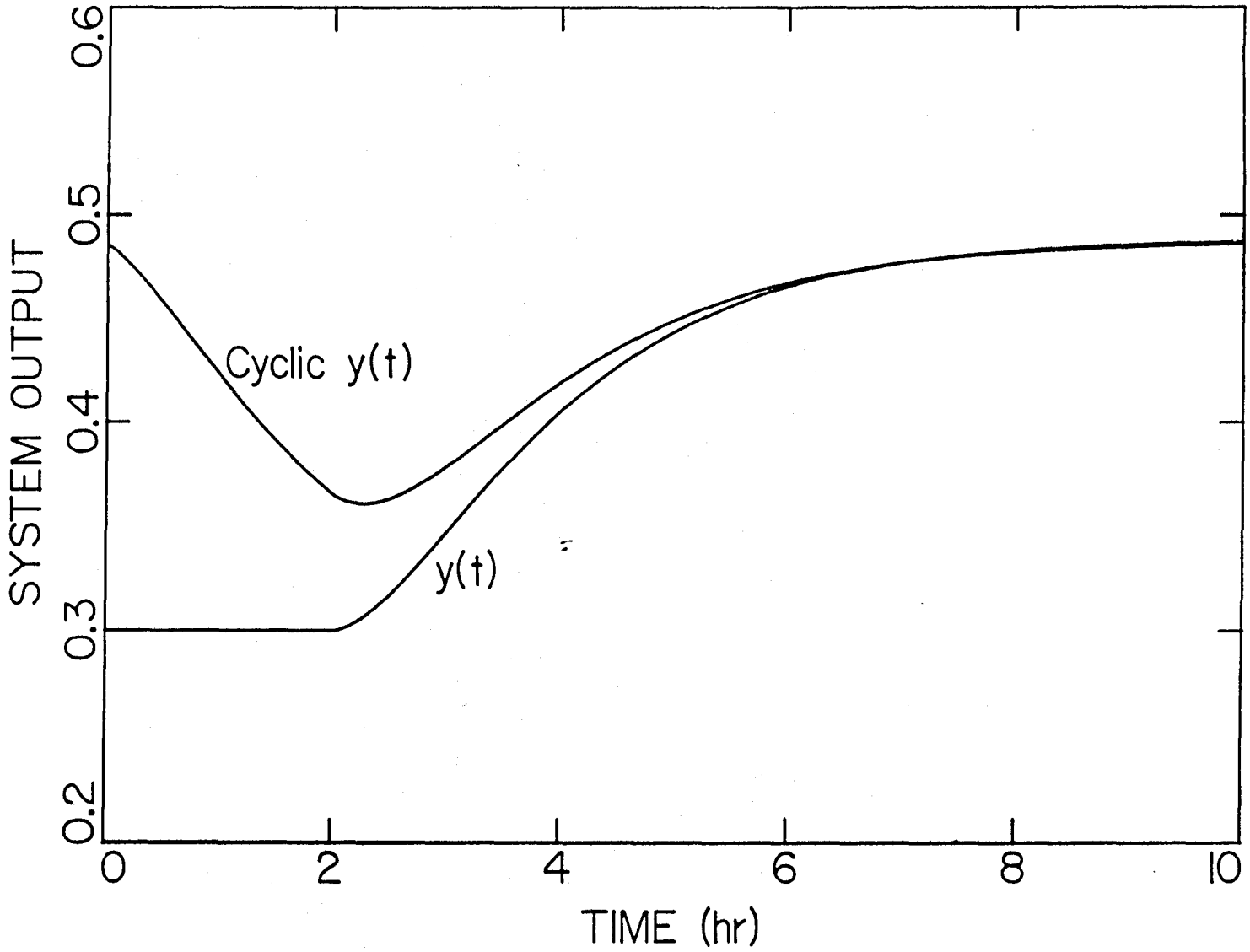


Figure 2.6.28. Comparison of the true system output $y(t)$ and the cyclic $y(t)$ calculated according to the convolution theorem of Fourier transform. Specifically, $y_{\text{cyclic}}(t) = \mathcal{F}^{-1}\{\mathcal{F}\{k(t)\} \cdot \mathcal{F}\{\mu(t)\}\}$.

figure as $y_a(t)$ and $y_b(t)$. Note that $y_b(t)$ is a vertically reflected and horizontally shifted image of the physically measured $y_a(t)$ because the step inputs have the same magnitudes but are oppositely directed. Therefore, the cyclic system output $y_{\text{cyclic}}(t)$ can be easily obtained.

$$y_{\text{cyclic}}(t) = y_a(t) + y_b(t), \tag{2.6.50}$$

where

$$y_a(t) = \int_{-\infty}^{t} \mu_a(h)k(t-h)dh \qquad \Longrightarrow \quad y_t \tag{2.6.51a}$$

$$y_b(t) = \int_{-\infty}^{t} \mu_b(h)k(t-h)dh \qquad \Longrightarrow \quad y_{\text{high}-ss} - y_{t^*+t} \tag{2.6.51b}$$

To be sure, there are also an infinite number of these step ups and downs in the past in addition to the two just considered, and each of these will contribute to the final observed response. If the time window is wide enough, the effect of these past virtual disturbances will have decayed to insignificant levels corresponding to the tail section of the kernel function. Besides being perfectly rectangular, long $t_f - t^*$ makes it an ideal system.

However, the system input encountered in real applications frequently is not so ideal. One such non-ideality is illustrated in Figure 2.6.30. Non-ideality comes from two independent sources: a time interval of $t_f - t^*$ that is not long enough to establish fully a new steady-state and a near-step input imposed on the system. As before, this non-ideal input can be decomposed into $\mu_a(t)$ and $\mu_b(t)$, with the known observed effect of $y_a(t)$ for $\mu_a(t)$ and the presently unknown effect of $y_b(t)$ for $\mu_b(t)$ that remains be calculated once the kernel is known.

$$\mu(t) = \mu_a(t) + \mu_b(t). \tag{2.6.52}$$

$$y_{\text{cyclic}}(t) = y_a(t) + y_b(t). \tag{2.6.53}$$

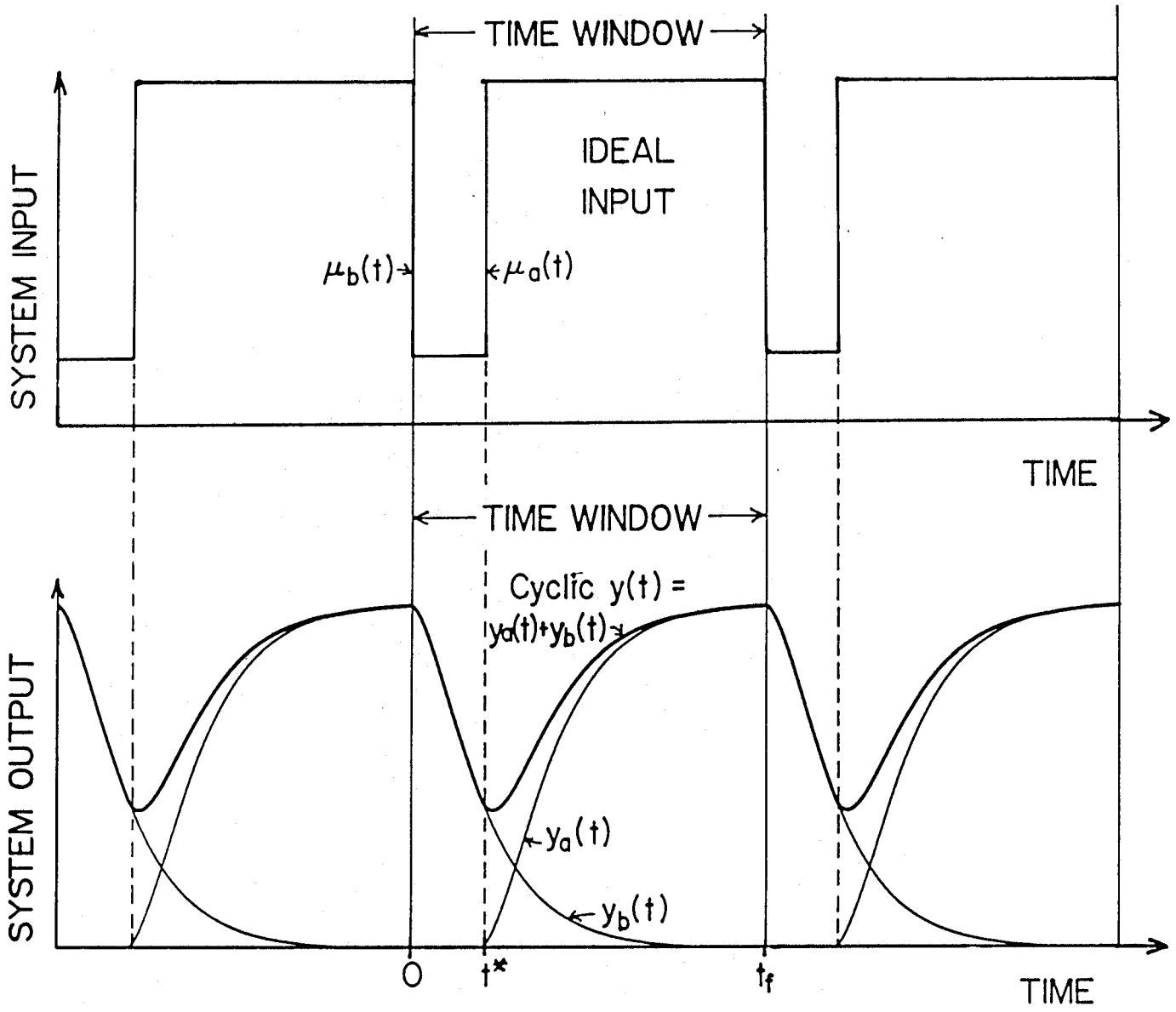


Figure 2.6.29. An ideal rectangular input at $t = t^*$. The virtual step down at $t = 0$ is due to the imposition of a time window. The system output $y_a(t)$ and $y_b(t)$ are the shifted and reflected image of each other.

$$y_a(t) = \int_{-\infty}^{t} \mu_a(h)k(t-h)dh \qquad \Longrightarrow \quad y_t \tag{2.6.54a}$$

$$y_b(t) = \int_{-\infty}^{t} \mu_b(h)k(t-h)dh \tag{2.6.54b}$$

But because the kernel itself is not yet known, an iterative process of kernel estimation is suggested. Note that except for the near-step up at $t = -(t_f - t^*)$, $\mu_b(t)$ is an almost perfect step down. Part of what makes an input function ideal is the length of $t_f - t^*$. Thus, a near-step up at the distant past can be safely neglected if $t_f - t^*$ is long enough. As opposed to the ideal step input, the imperfection in the shape of the step input makes $\mu_a(t)$ and $\mu_b(t)$ nonsymmetrical, which, in turn, makes $y_a(t)$ and $y_b(t)$ nonsymmetrical.

The once decomposed $\mu_b(t)$ can be further decomposed into $\mu_{b1}(t)$, $\mu_{b2}(t)$, and $\mu_{b3}(t)$:

$$\mu_b(t) = \mu_{b1}(t) - \mu_{b2}(t) - \mu_{b3}(t) \tag{2.6.55}$$

$$y_b = y_{b1}(t) - y_{b2}(t) - y_{b3}(t). \tag{2.6.56}$$

$$y_{b1}(t) = \int_{-\infty}^{t} \mu_{b1}(h)k(t-h)dh \qquad \Longrightarrow \quad y_{\text{high-ss}} - y_{t^*+t} \tag{2.6.57a}$$

$$y_{b2}(t) = \int_{-\infty}^{t} \mu_{b2}(h)k(t-h)dh \qquad \Longrightarrow \quad \sum_{j=0}^{t} (\mu_{\text{high-ss}} - \mu_{t^*+t-j})k_j \Delta t \tag{2.6.57b}$$

$$y_{b3}(t) = \int_{-\infty}^{t} \mu_{b3}(h)k(t-h)dh \qquad \Longrightarrow \quad \sim 0 \tag{2.6.57c}$$

This is graphically illustrated in Figure 2.6.32. In summary, the cyclic $y(t)$ is to be calculated as:

$$y_{\text{cyclic}}(t) = y_a(t) + y_{b1}(t) - y_{b2}(t) - y_{b3}(t), \tag{2.6.58}$$

The decomposed $\mu_{b1}(t)$ contains most of the parental function $\mu_b(t)$ and has the same but inverted shape as the original physically measured system input. The response to $\mu_{b1}(t)$ is designated as $y_{b1}(t)$, and it constitutes the first-order major correction to $y(t)$ in achieving the true $y_{\text{cyclic}}(t)$. The response of the system to the

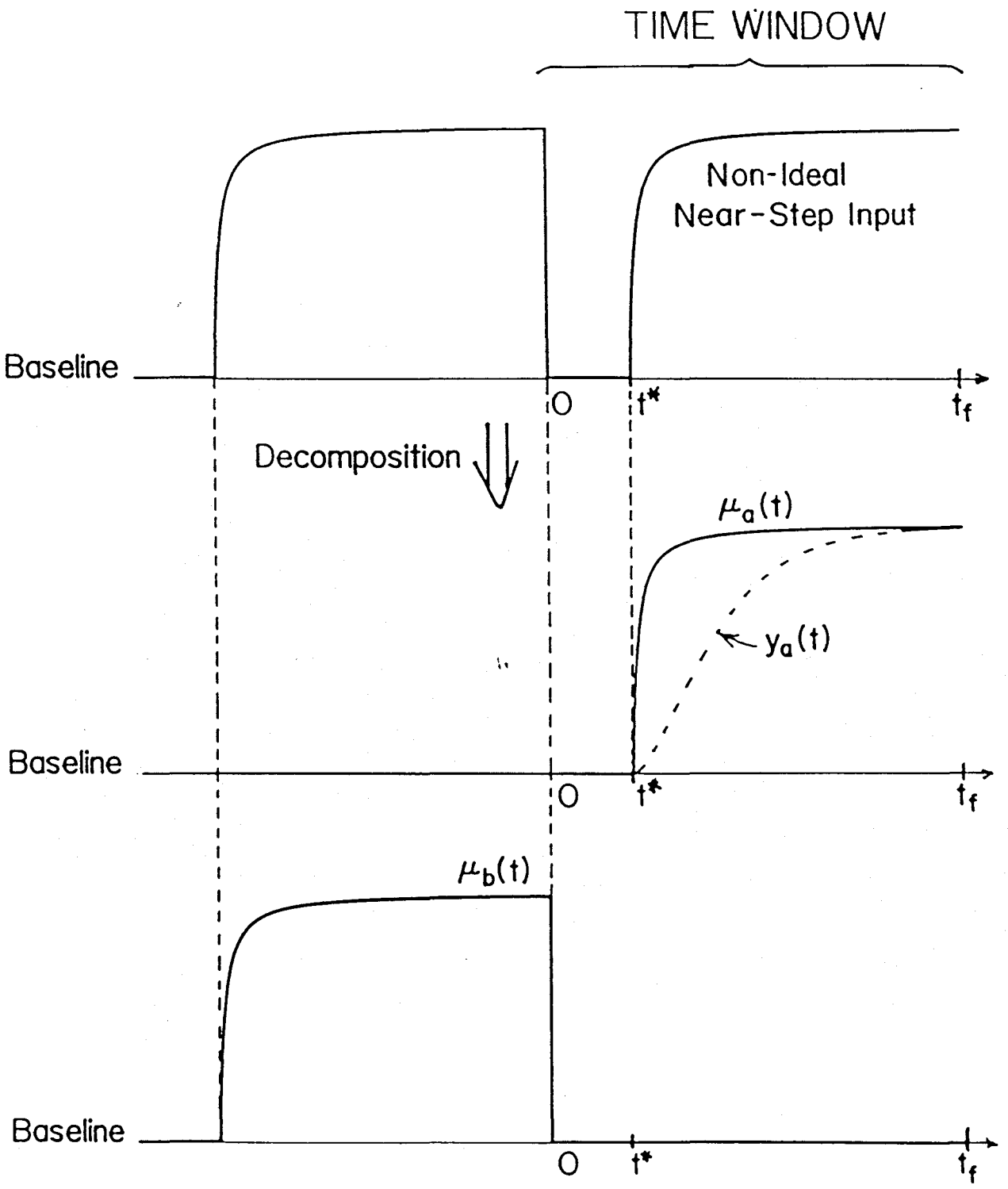


Figure 2.6.30. Decomposition of a non-ideal system input into $\mu_a(t)$ whose response is known and $\mu_b(t)$ whose response is to be estimated.

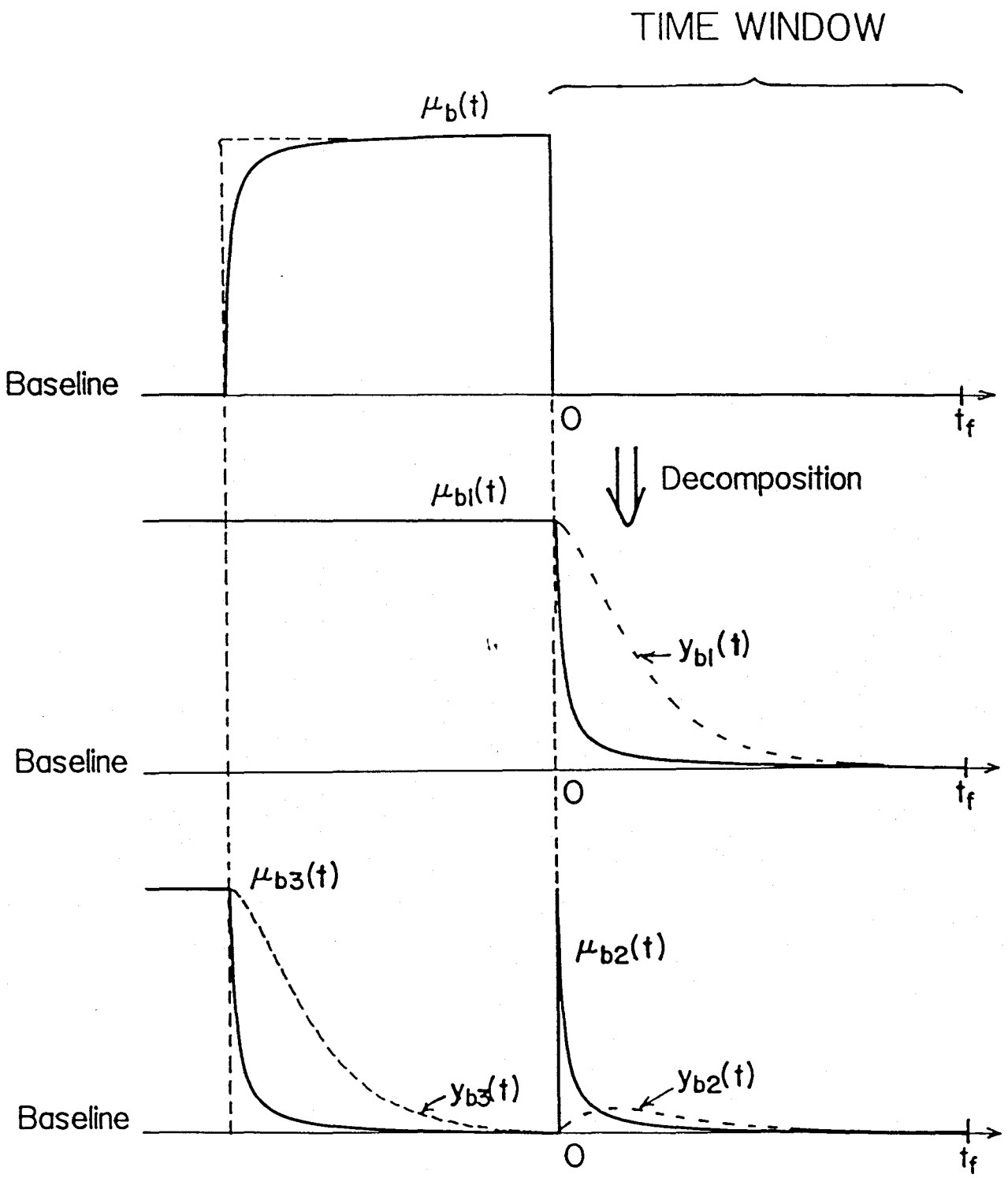
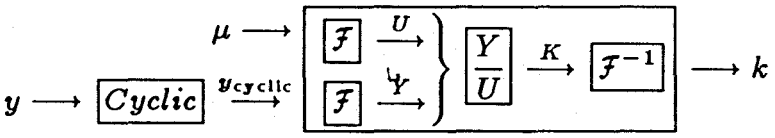


Figure 2.6.31. Further decomposition of $\mu_b(t)$ into $\mu_{b1}(t)$, $\mu_{b2}(t)$, and $\mu_{b3}(t)$. The responses to $\mu_{b1}(t)$ and $\mu_{b3}(t)$ are known. Only the response to $\mu_{b2}(t)$ need be estimated.

input $\mu_{b1}(t)$ is simply the inverted image of the actual measured and known system output. So is the response to $\mu_{b3}(t)$, except that only an insignificant amount of residual will be felt in the time window if the kernel decays to 0 within the time interval $t_f - t^*$. This response will be termed third-order or tertiary correction. Thus, the only part of the original $\mu(t)$ that has not yet been considered is $\mu_{b2}(t)$, which is shaped like an impulse, suggesting that the second-order correction $y_{b2}(t)$ should be similar to a reduced version of the kernel.

Because the secondary correction cannot be obtained without some a priori knowledge on the kernel, it is neglected in the first approximation of the cyclic $y(t)$. Figure 2.6.32 shows the relative magnitude and the relationship of these correction terms in our simulation. The kernel estimated through a straight forward Fourier transform of $\mu(t)$ and $y_{\text{cyclic}}(t)$ is shown in Figure 2.6.33.



The results of first iteration is reasonable but slightly oscillatory due to the neglected second-order correction. In the second iteration, the second-order correction is calculated based on the kernel after the first iteration. Figure 2.6.34 shows that after this correction the cyclic $y(t)$ used in the second iteration is already indistinguishable from the true cyclic $y(t)$. The kernel fails to converge to the true curve even in the absence of noises in both $\mu(t)$ and $y(t)$, although the iterative algorithm is convergent and there is little difference between the true $y_{\text{cyclic}}(t)$ and the $y_{\text{cyclic}}(t)$ calculated from the estimated kernel. This fact demonstrates that more than one kernel can give rise to the identical observed behavior; the relationship is not one-to-one, and the kernel is not unique. (It is unique, though, if the kernel is constrained to be a linear combination of the base exponential distribution functions.)

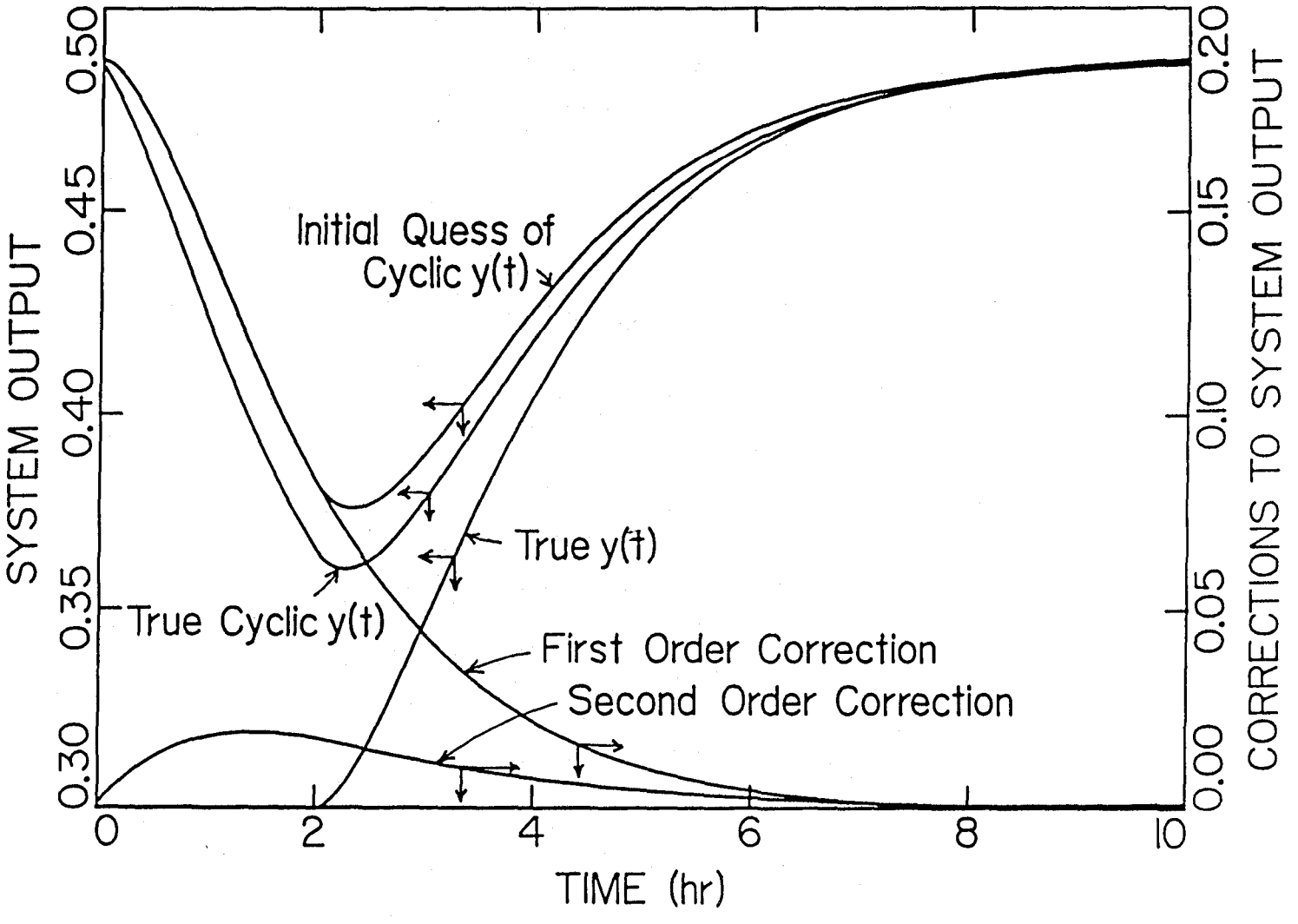


Figure 2.6.32. Various correction terms that transform the original $y(t)$ into $y_{\text{cyclic}}(t)$. The first-order correction, which is the reflected image of the original $y(t)$, represents the majority of the overall needed correction. The initial guess of $y_{\text{cyclic}}(t)$ is the summation of the true $y(t)$ and this first-order correction.

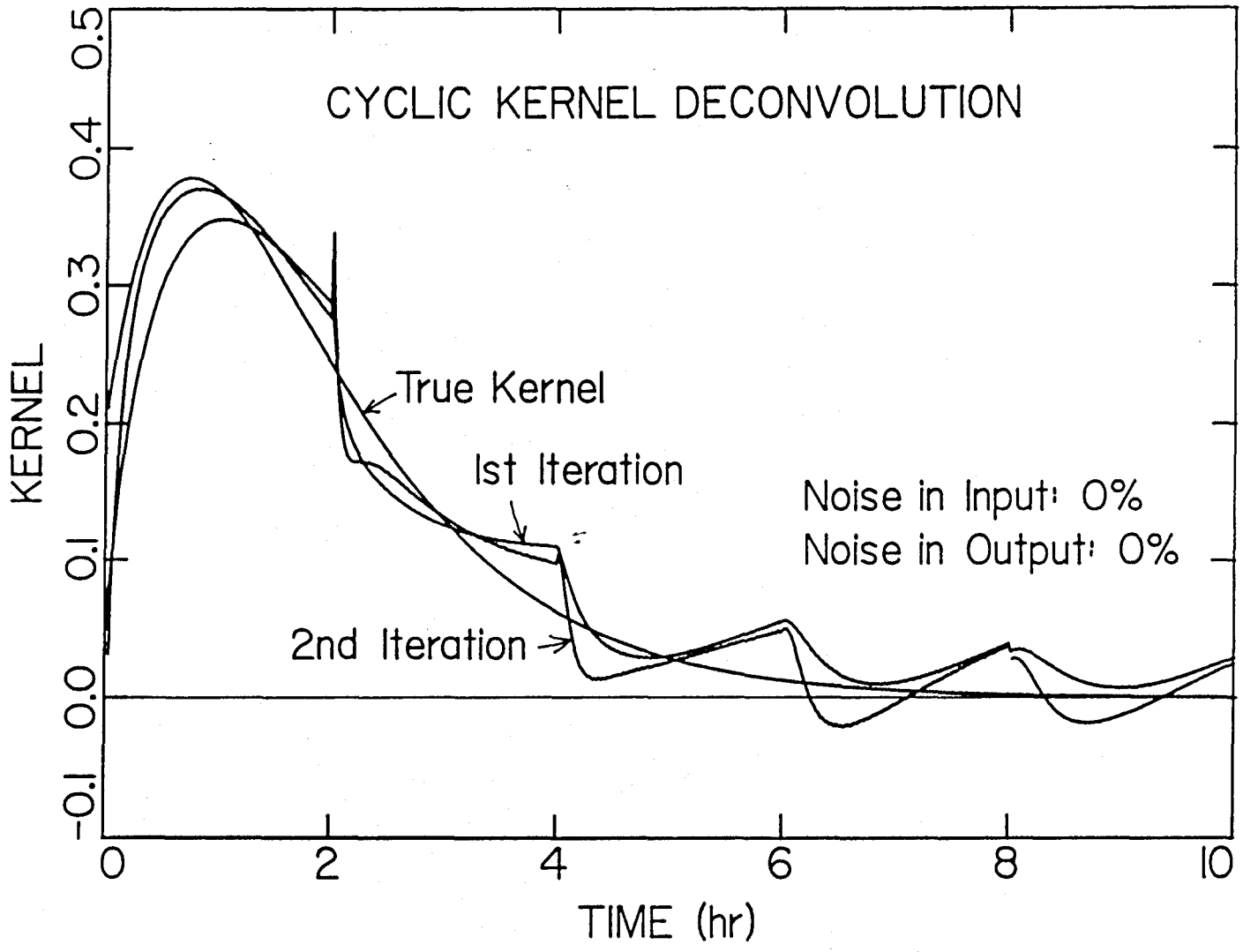


Figure 2.6.33. Kernel estimated by applying cyclic kernel deconvolution to deterministic $y(t)$ and $\mu(t)$. There is no further visually detectable change in the kernel shape after the second iteration.

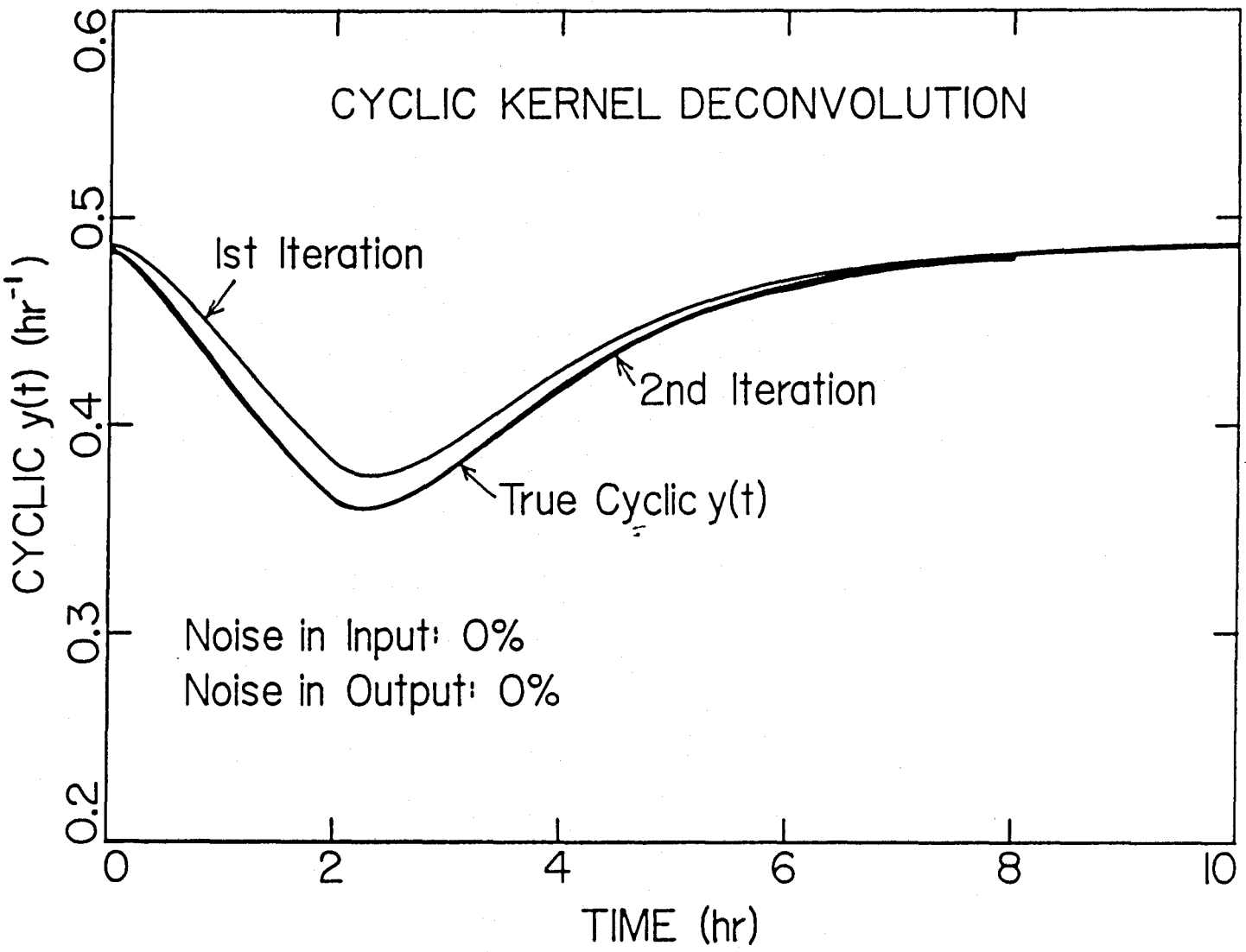
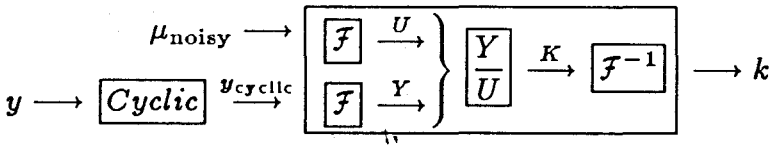


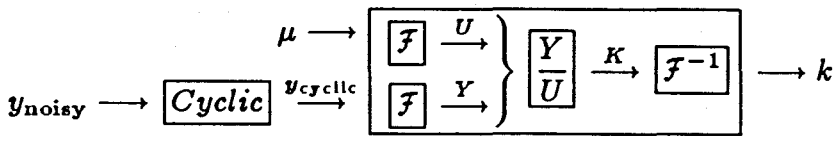
Figure 2.6.34. Cyclic $y(t)$ estimated by applying cyclic kernel deconvolution to deterministic $y(t)$ and $\mu(t)$. There are no significant changes after the second iteration.

Due to the initial failure to include the secondary correction to $y_{\text{cyclic}}(t)$, the kernel's oscillation around the true curve persists once the kernel is allowed to do so. Note that the kernel resulting from the Fourier transform operations, as any other function, is also implicitly periodic. For example, the two end-points in the given time window are always equal, and the true kernel can never be forced out of this Fourier transform-based scheme if it does not have equal end-points. Figure 2.6.35 demonstrates the effect of lengthening the pre-perturbation time from 2 hours to 4 hours. This technique can be used to push the onset of oscillations in the estimated kernel to a later time.

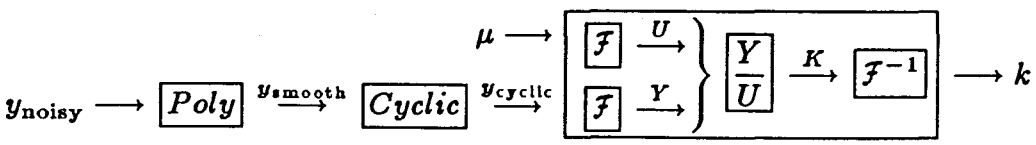
The converged estimates of the kernel function in the presence of noise in the system input $\mu(t)$ based on the following scheme are shown in Figure 2.6.36.



Although the estimated kernel oscillates around the true curve, much closer agreement is possible when it is further fitted to a set of base exponential distribution functions in a least-square sense. On the other hand, the presence of noise in the system output $y(t)$ is extremely detrimental to kernel estimation via a Fourier transform on the cyclic $y(t)$, as illustrated in Figure 2.6.37.



Finally, noise reduction through either polynomial approximation or frequency windowing is attempted at various points along the same route as before. The objective here is to force $k(t)$ to converge while suppressing the noise.



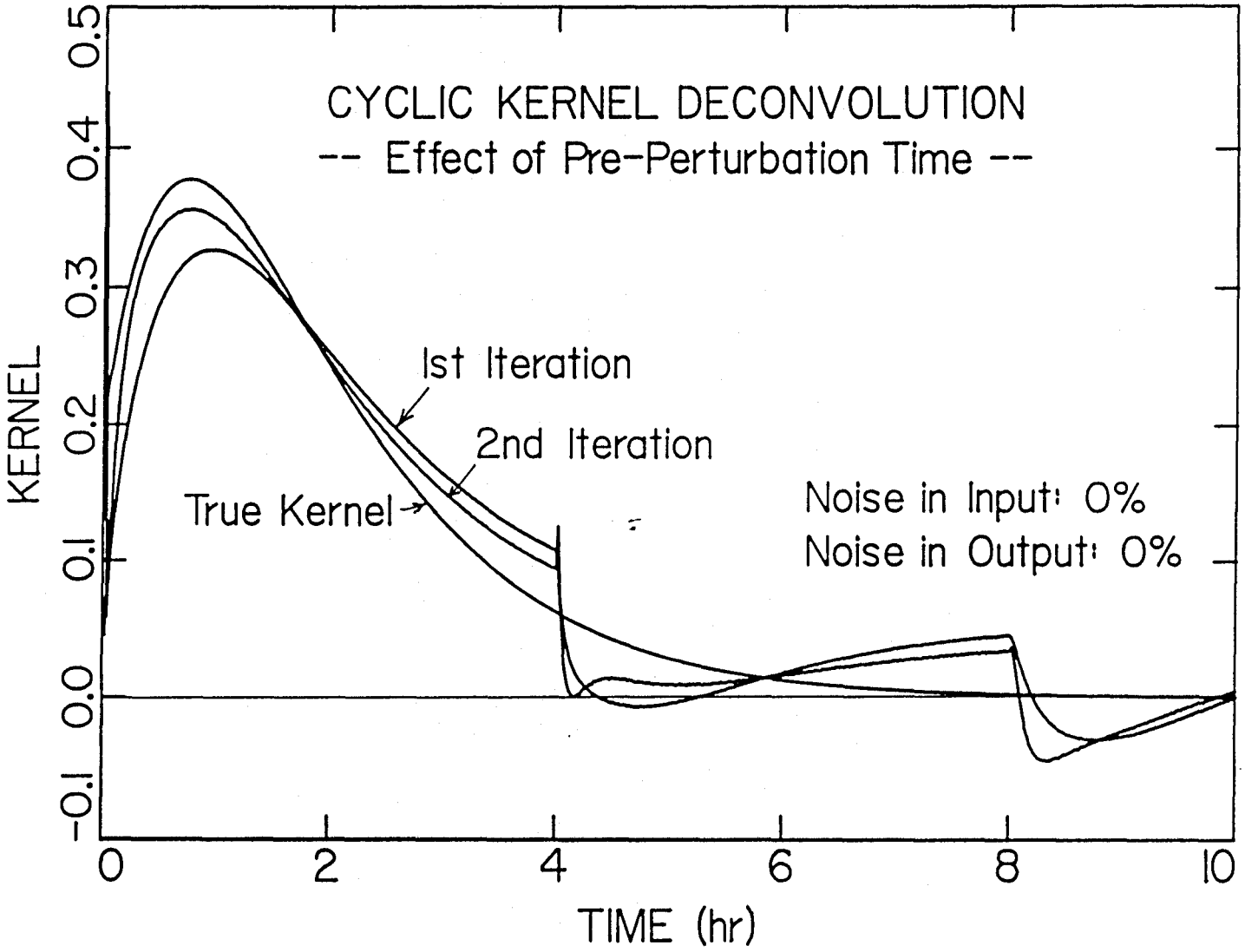


Figure 2.6.35. Kernel estimated by applying cyclic kernel deconvolution to deterministic $y(t)$ and $\mu(t)$. The pre-perturbation time t^* is increased from 2 hours to 4 hours.

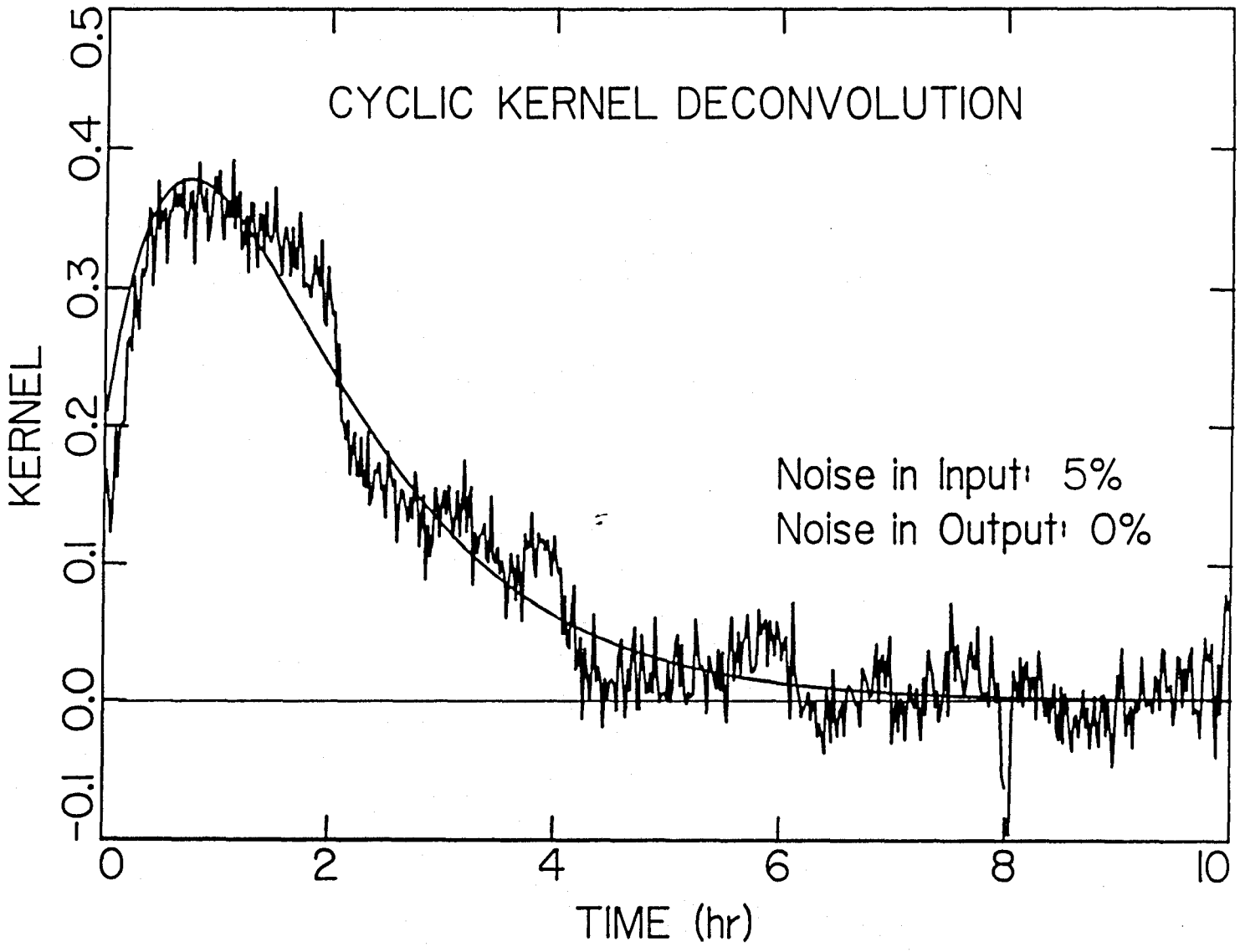


Figure 2.6.36. Kernel estimation through cyclic deconvolution on noisy (5%) system input $\mu(t)$ and deterministic system output $y(t)$.

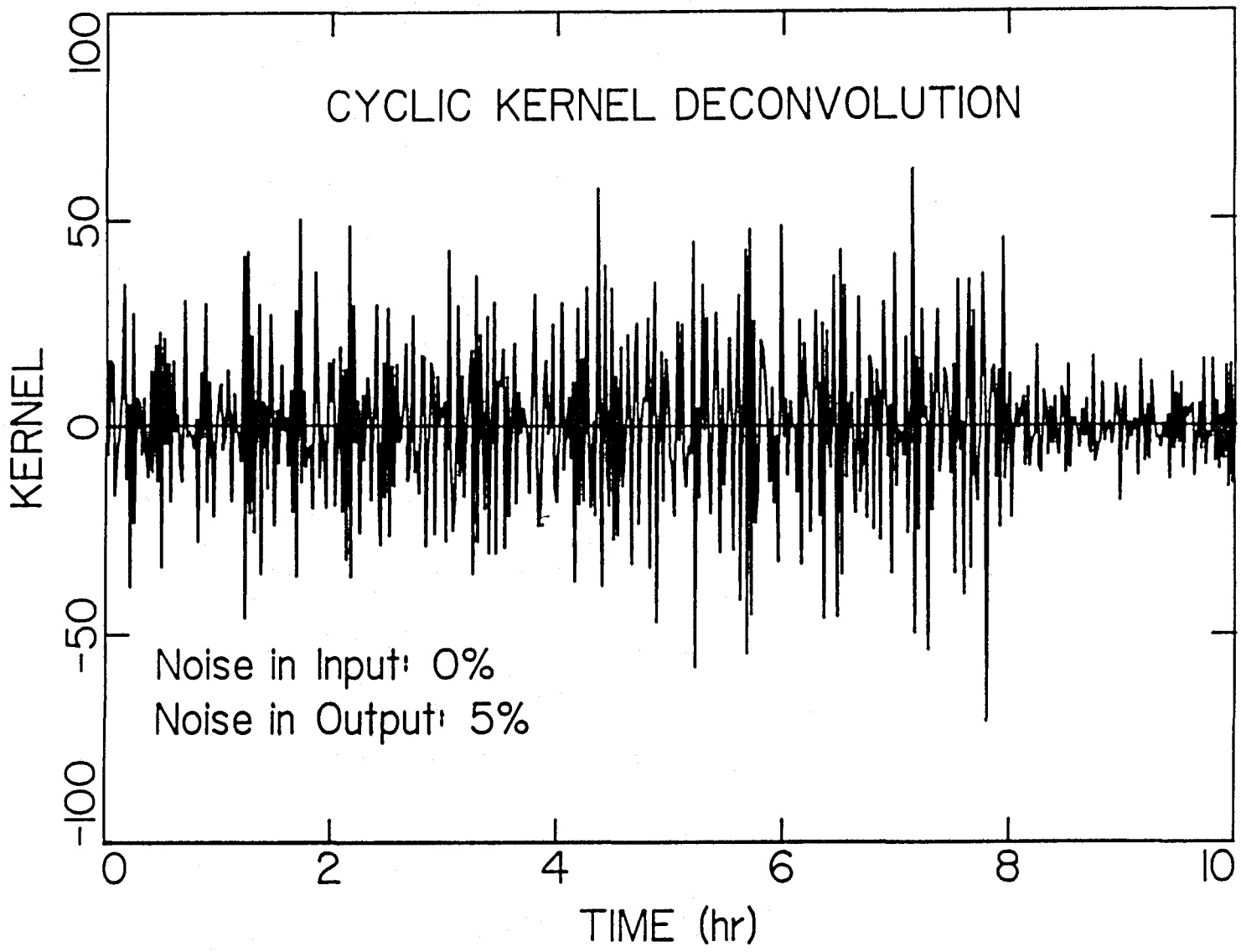
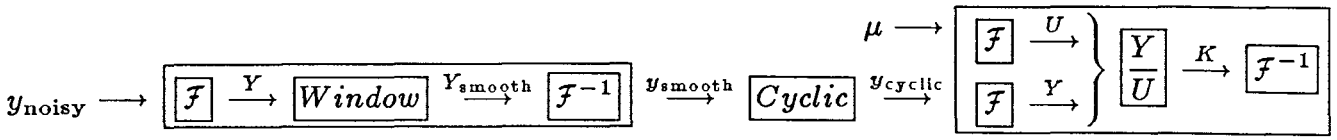
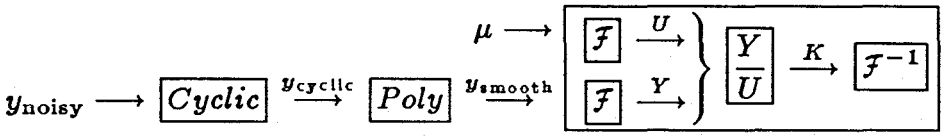
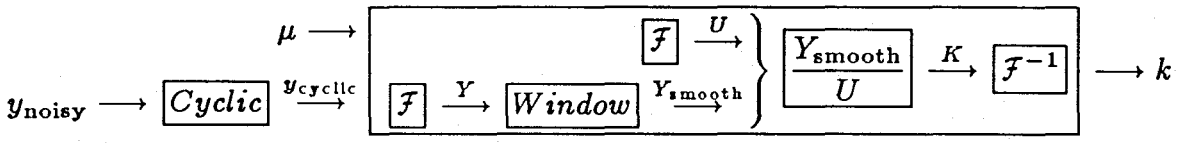


Figure 2.6.37. Kernel estimation through cyclic deconvolution on deterministic system input $\mu(t)$ and noisy (5%) system output $y(t)$.

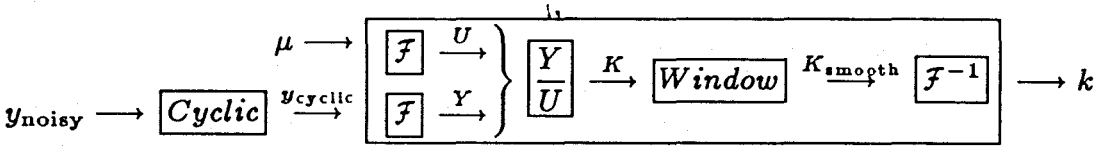


For example, frequency windows can be applied to smooth the original noisy $y(t)$ before forming $y_{cyclic}(t)$. The results from the above set of operations, though not shown, are not encouraging due to the fact that the noise in the kernel in each iteration is not filtered, which, in turn, causes the next estimate of $y_{cyclic}(t)$ to be noisy. This problem can be eliminated if smoothing by polynomial approximation or by a frequency window is applied at the point after $y_{cyclic}(t)$ has been formed.





which is equivalent to the following block diagram.



The results of these operations are shown in Figure 2.6.38 and 2.6.39 for $k(t)$ and $y_{cyclic}(t)$. It is clear that there is a vast improvement over the results shown in Figure 2.6.37, which do not have smoothing applied. As before, the agreement of the estimated $y_{cyclic}(t)$ with the true $y_{cyclic}(t)$ is quite good, partly due to the fact that the end-points are equal, but the agreement of the estimated $k(t)$ with the true $k(t)$ is not as good. Nonetheless, considering the fact that the system input structure used throughout simulations in this section is most demanding, the method survived the tough test quite well. In summary, the method of cyclic kernel deconvolution studied in this section is similar to the original Fourier transform method, but it avoids taking the difference operation that is generally considered undesirable in a noisy function.

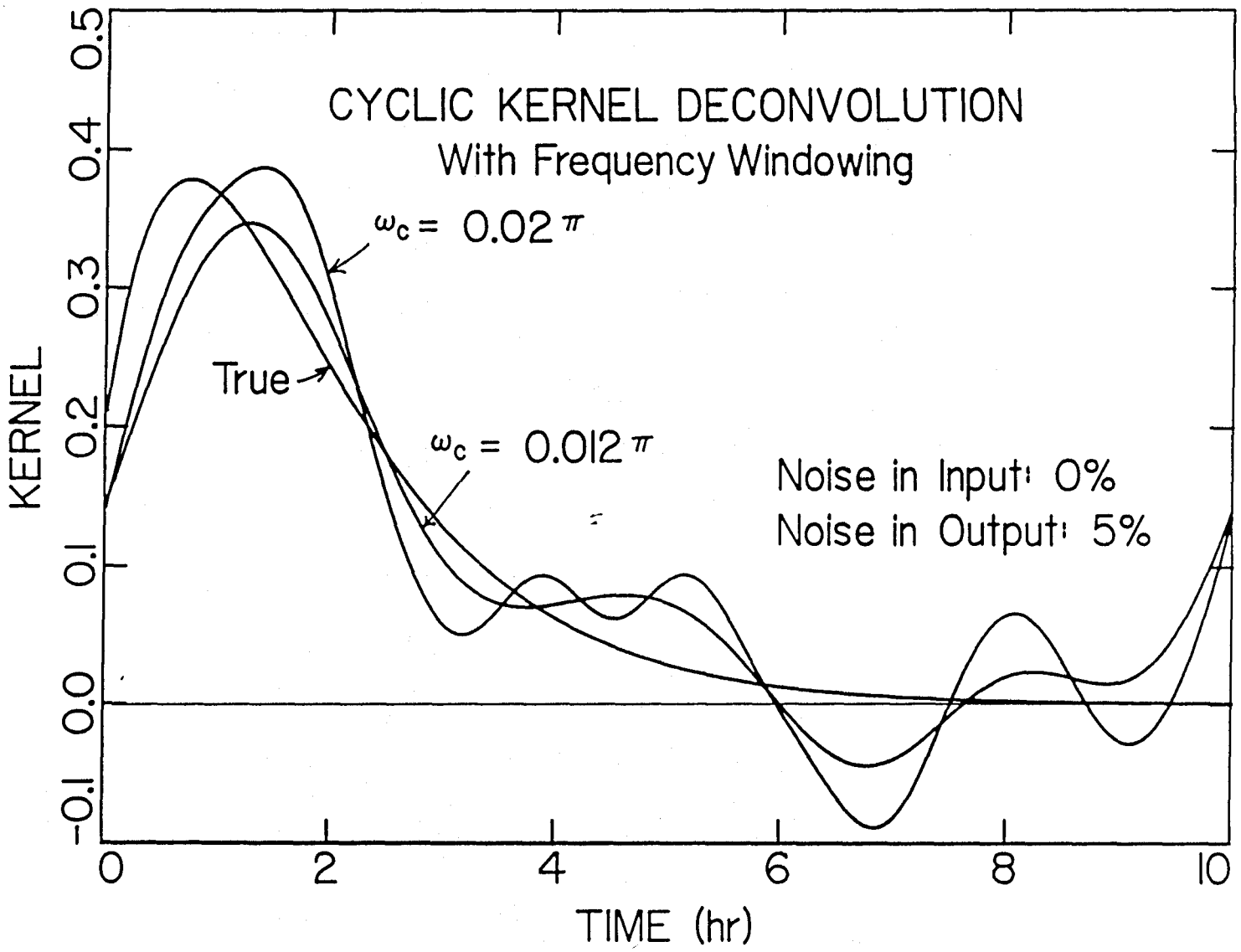


Figure 2.6.38. Kernel estimation through cyclic deconvolution on deterministic system input $\mu(t)$ and noisy (5%) system output $y(t)$, with frequency windows.

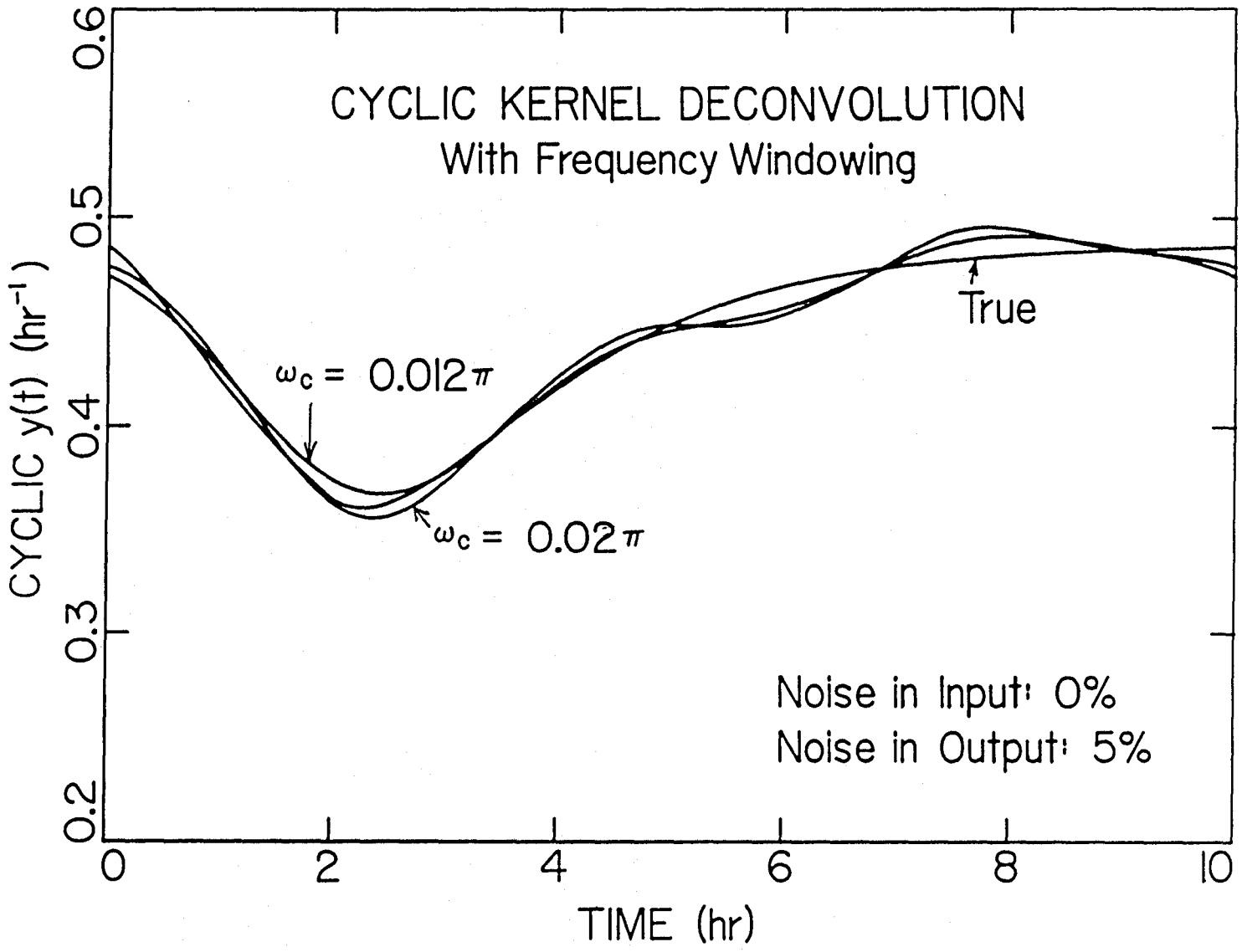


Figure 2.6.39. Cyclic $y(t)$ estimated through cyclic deconvolution on deterministic system input $\mu(t)$ and noisy (5%) system output $y(t)$, with frequency windows.

Inverse Numerical Integration

An alternative approach is to solve the set of digitized integral equations directly. Expanding Equation (2.6.27a) $y_t = \sum_{j=0}^{t} \mu_{t-j} k_j \Delta t$ yields:

$$\begin{aligned}
y_0 &= \left\{\mu_0 k_0 \Delta t\right\} \\
y_1 &= \left\{\mu_1 k_0 + \mu_0 k_1 \Delta t\right\} \\
y_2 &= \left\{\mu_2 k_0 + \mu_1 k_1 + \mu_0 k_2 \Delta t\right\} \\
&\vdots \\
y_j &= \left\{\mu_j k_0 + \mu_{j-1} k_1 + \ldots + \mu_1 k_{j-1} + \mu_0 k_j \Delta t\right\} \\
&\vdots \\
y_{t-1} &= \left\{\mu_{t-1} k_0 + \mu_{t-2} k_1 + \ldots + \mu_1 k_{t-2} + \mu_0 k_{t-1} \Delta t\right\} \\
y_t &= \left\{\mu_t k_0 + \mu_{t-1} k_1 + \ldots + \mu_1 k_{t-1} + \mu_0 k_t \Delta t\right\},
\end{aligned} \tag{2.6.59}$$

which can be represented very compactly in a matrix notation:

$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} \mu_0 & 0 & 0 & \ldots & 0 \\ \mu_1 & \mu_0 & 0 & \ldots & 0 \\ \mu_2 & \mu_1 & \mu_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_t & \mu_{t-1} & \mu_{t-2} & \ldots & 0 \end{bmatrix}}_{\boldsymbol{\mu}} \cdot \underbrace{\begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ \vdots \\ k_t \end{bmatrix}}_{\mathbf{k}} \cdot \Delta t. \tag{2.6.60}$$

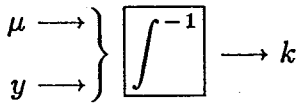
Note that in the above equation it is assumed that $\mu_j = 0$ for $j = -1, -2, -3, \ldots$, which can be physically achieved if one introduces perturbation from a steady state and works in deviation variables. Due to this assumption of $\mu_j = 0$, the matrix $\boldsymbol{\mu}$ is lower triangular, and the solution for $\mathbf{k}$ can be easily obtained recursively from

k_0 to k_t without resorting to the high-dimensional matrix inverse.

$$
\begin{aligned}
k_0 &= \frac{1}{\mu_0}\left\{\frac{y_0}{\Delta t}\right\} \\
k_1 &= \frac{1}{\mu_0}\left\{\frac{y_1}{\Delta t} - \mu_1 k_0\right\} \\
k_2 &= \frac{1}{\mu_0}\left\{\frac{y_2}{\Delta t} - \mu_2 k_0 - \mu_1 k_1\right\} \\
&\vdots \\
k_j &= \frac{1}{\mu_0}\left\{\frac{y_j}{\Delta t} - \mu_j k_0 - \mu_{j-1} k_1 - \ldots - \mu_1 k_{j-1}\right\} \\
&= \frac{1}{\mu_0}\left\{\frac{y_j}{\Delta t} - \sum_{k=0}^{t-1} \mu_{j-k} k_k\right\} \\
&\vdots \\
k_{t-1} &= \frac{1}{\mu_0}\left\{\frac{y_{t-1}}{\Delta t} - \mu_{t-1} k_0 - \mu_{t-2} k_1 - \ldots - \mu_1 k_{t-2}\right\} \\
&= \frac{1}{\mu_0}\left\{\frac{y_{t-1}}{\Delta t} - \sum_{k=0}^{t-2} \mu_{j-k} k_k\right\} \\
k_t &= \frac{1}{\mu_0}\left\{\frac{y_t}{\Delta t} - \mu_t k_0 - \mu_{t-1} k_1 - \ldots - \mu_1 k_{t-1}\right\} \\
&= \frac{1}{\mu_0}\left\{\frac{y_t}{\Delta t} - \sum_{k=0}^{t-1} \mu_{t-k} k_k\right\}
\end{aligned}
\tag{2.6.61}
$$

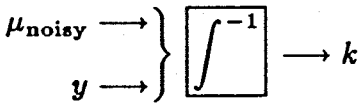
Because the above equations suggest division by μ_0, any steady-state values prior to the introduction of the near-step perturbation have $\mu = 0$ and are not used in the calculation. In addition, because a factor of $\frac{1}{\mu_0}$ is involved in the calculation of every k_j, this scheme may not always yield a curve $k(t)$ with a 0th moment of exact unity, as it should be, if there are noises in μ_0. Any error in the kernel estimation due to incorrect μ_0 will propagate to the next step, for the determination of the next k_j depends on all the k_js calculated up to that point. This algorithm's sensitivity to uncertainties in μ_0 is shown in Figure 2.6.40, where deterministic system input and output are used per the following schematic in which $\boxed{\int^{-1}}$ signifies the inverse

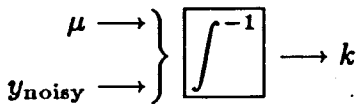
numerical integration.



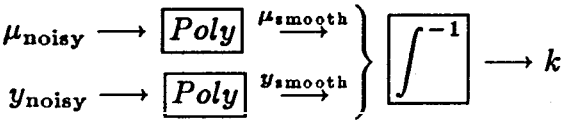
Too small of an estimation for μ_0 tends to cause oscillations in $k(t)$, as demonstrated in Figure 2.6.40b. However, as long as the initial guess of μ_0 is within the same order of magnitude of the true value, the algorithm seems to work well, as evidenced in Figure 2.6.40c. As the guesses of μ_0 deviate further away from the true value, the estimated kernels as well as the areas under these curves start to diverge from the true curve. See Figure 2.6.40d-f. One need not be concerned about this divergent behavior because it can be detected by the deviation in the curve's first moment and because it is unlikely that a specific growth rate of over 1 hr^{-1} is to be guessed. It is further suggested that the area be normalized at the end of the calculation.

The results of numerical simulation using this approach show the same tendency as other methods, namely that although errors in $\mu(t)$ are automatically reduced via the filtering effect of the summation process and can be tolerated, any errors in $y(t)$ directly affect the calculation of k_j. Errors in k_j, in turn, propagate through the summation and affect the calculation of the next k_{j+1}. The cumulative effect again leads to the general failure of the method when there are noises in $y(t)$. See Figures 2.6.41 and 2.6.42 resulting form the following respective operations.





Finally, noisy curves of $\mu(t)$ and $y(t)$ are first smoothed by either polynomial approximations or frequency windowing as before, and the results are shown in Figures 2.6.43 and 2.6.44, respectively.



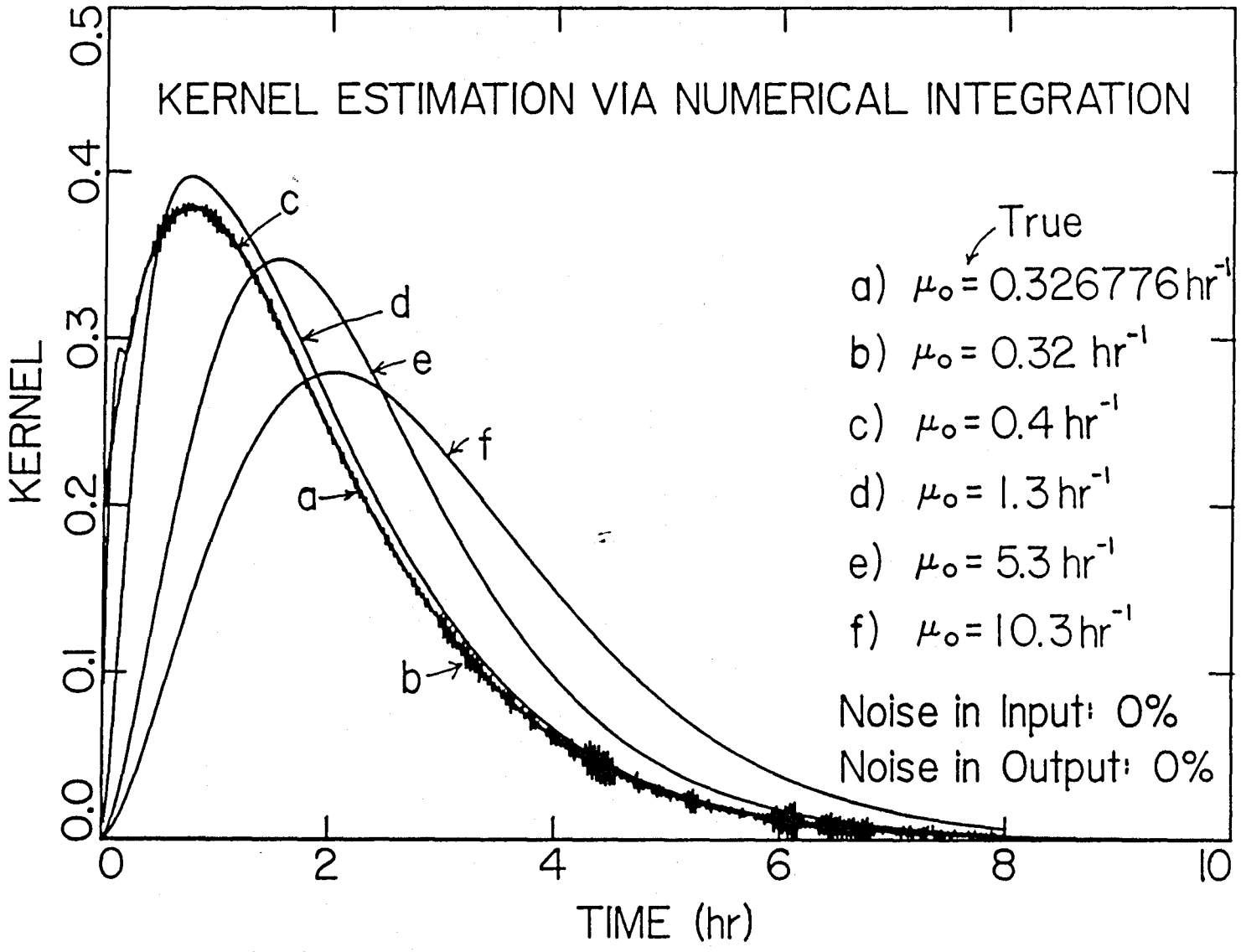
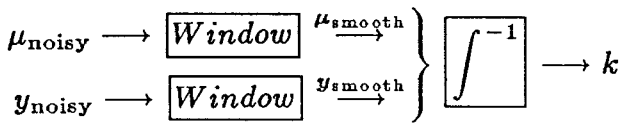


Figure 2.6.40. Inverse numerical integration is used to estimate the kernel. Shown are the effects of wrong guesses in the initial $\mu(0)$ on kernel estimation based on deterministic $\mu(t)$ and $y(t)$.



Least-Square Error via Multivariable Minimum Search

The last method to be considered seeks the kernel model parameters that minimize the sum of square-errors between the measured $y(t)$ and the predicted $y(t)$ based on the assumed kernel. The algorithm is described in the flow diagram of Figure 2.6.45. First, the order of the kernel is assumed so that the number of parameters can be identified. One then provides the initial guesses for the distribution of weights for each base function, a_is, and the kernel time constant, T. A kernel function is constructed from these given parameters, and the system output is predicted from this kernel, given the system input. The objective function J is formulated according to the following equation, and the sum of square-errors is evaluated.

$$J = \int \epsilon^2 = \int (y_{\text{measured}} - y_{\text{predicted}})^2 \quad \Longrightarrow \quad \sum_{t=0}^{N} (y_{\text{measured}} - y_{\text{predicted}})_t^2 \tag{2.6.62}$$

Any of the well established numerical methods on multivariable function minimization may be employed to seek $J = \min_{a_i,T} \left\{ \sum_t \epsilon_t^2 \right\}$. This process is represented graphically in Figure 2.6.46 for a first-order kernel with two variables, namely a_0 and T.

For all practical purposes, the noise in the system input and output may be assumed to be normal and independent, *i.e.*, $\frac{(error)_i}{\sigma_i} \sim N(0,1)$. The sum of the normalized square-errors, $\sum_{i=1}^{N} \frac{(error)_i^2}{\sigma_i^2}$, can thus be expected to follow a χ^2 probability distribution function with $\nu = N - 1$ degrees of freedom. Usually, N is large, and probability tables do not list the *chi*^{2} distribution for a large N. Fortunately, it can be approximated by a $N(\nu, 2\nu)$ as a result of the central limit theorem.

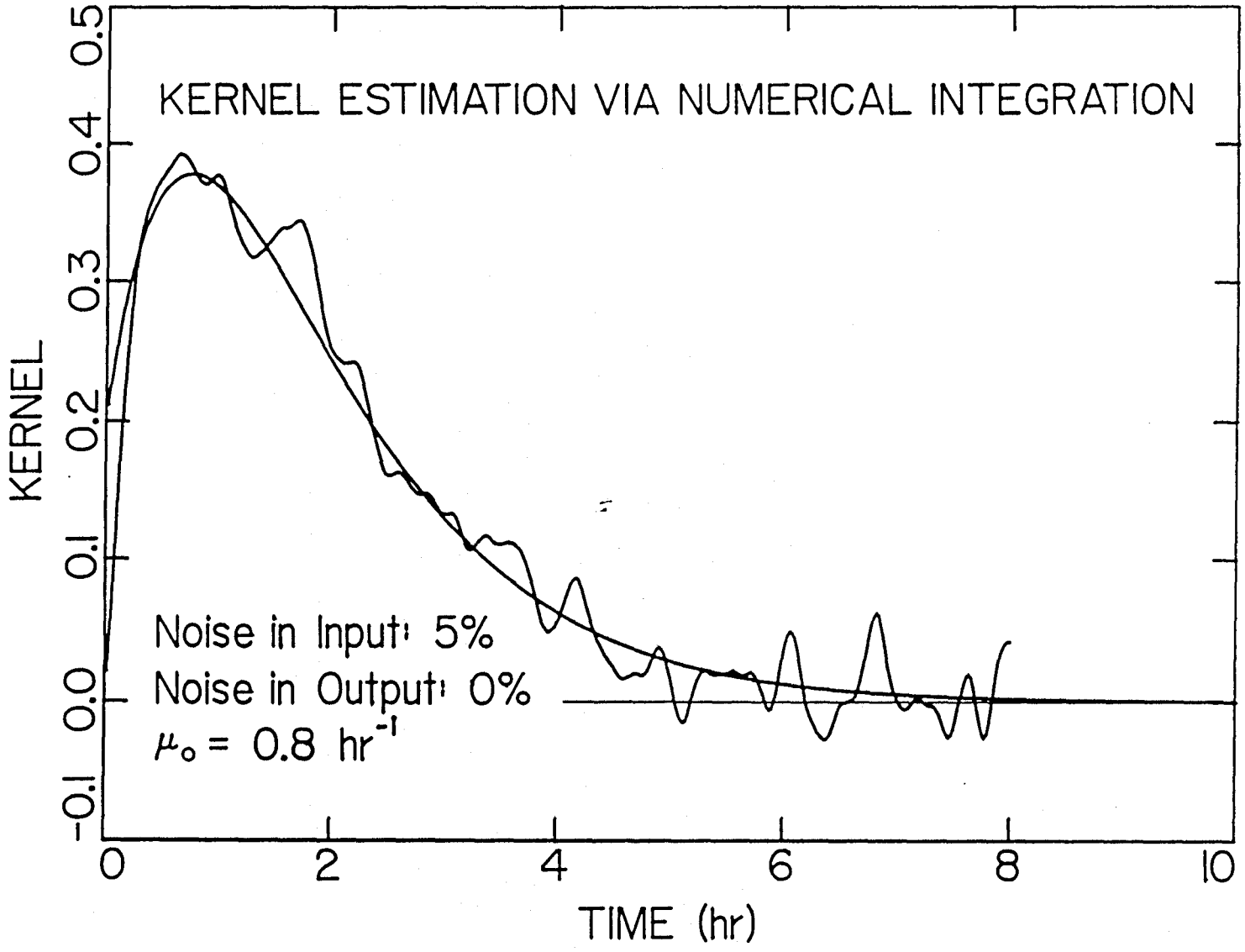


Figure 2.6.41. Kernel estimation by inverse numerical integration with 5% noise in $\mu(t)$ only.

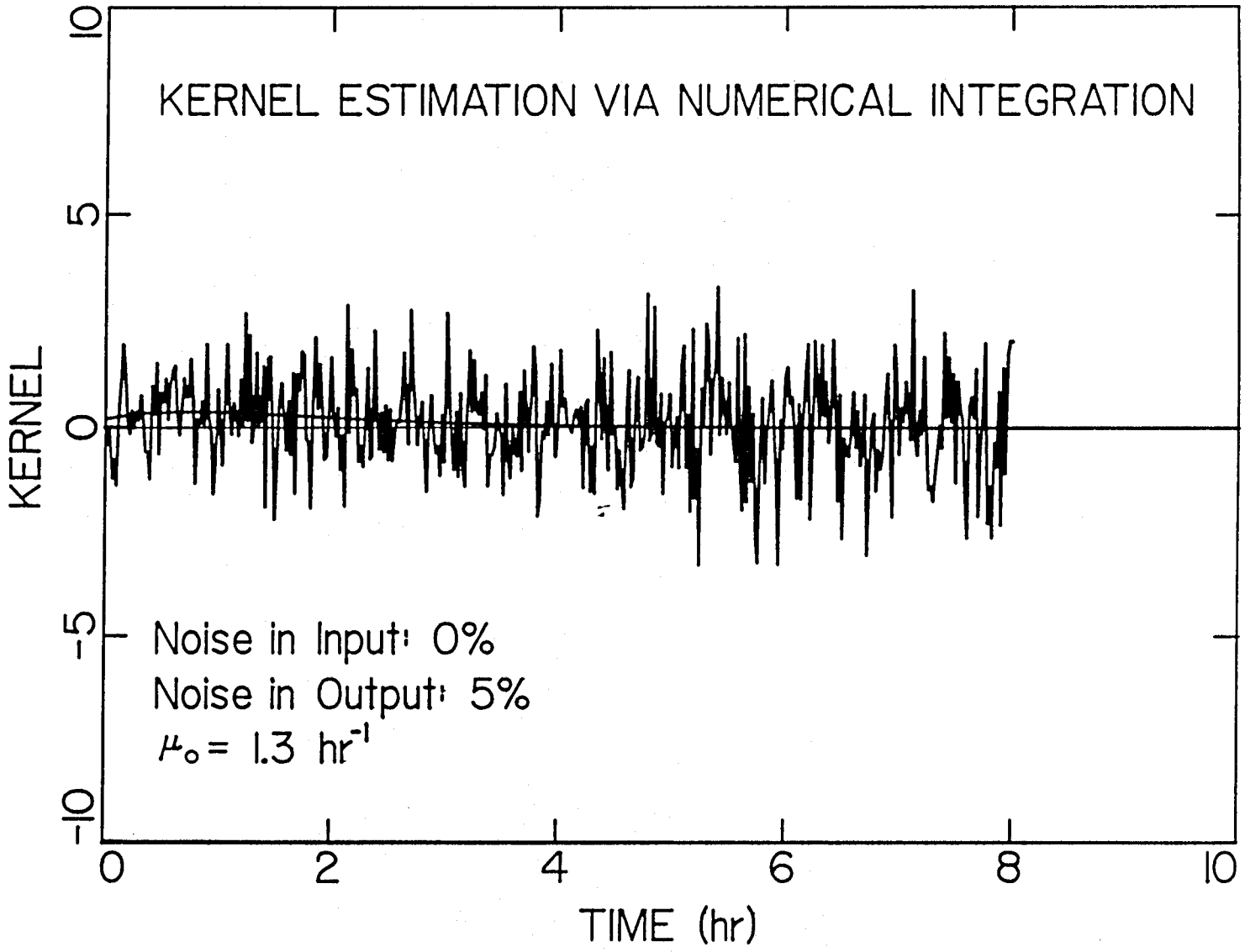


Figure 2.6.42. Kernel estimation by inverse numerical integration with 5% noise in $y(t)$ only.

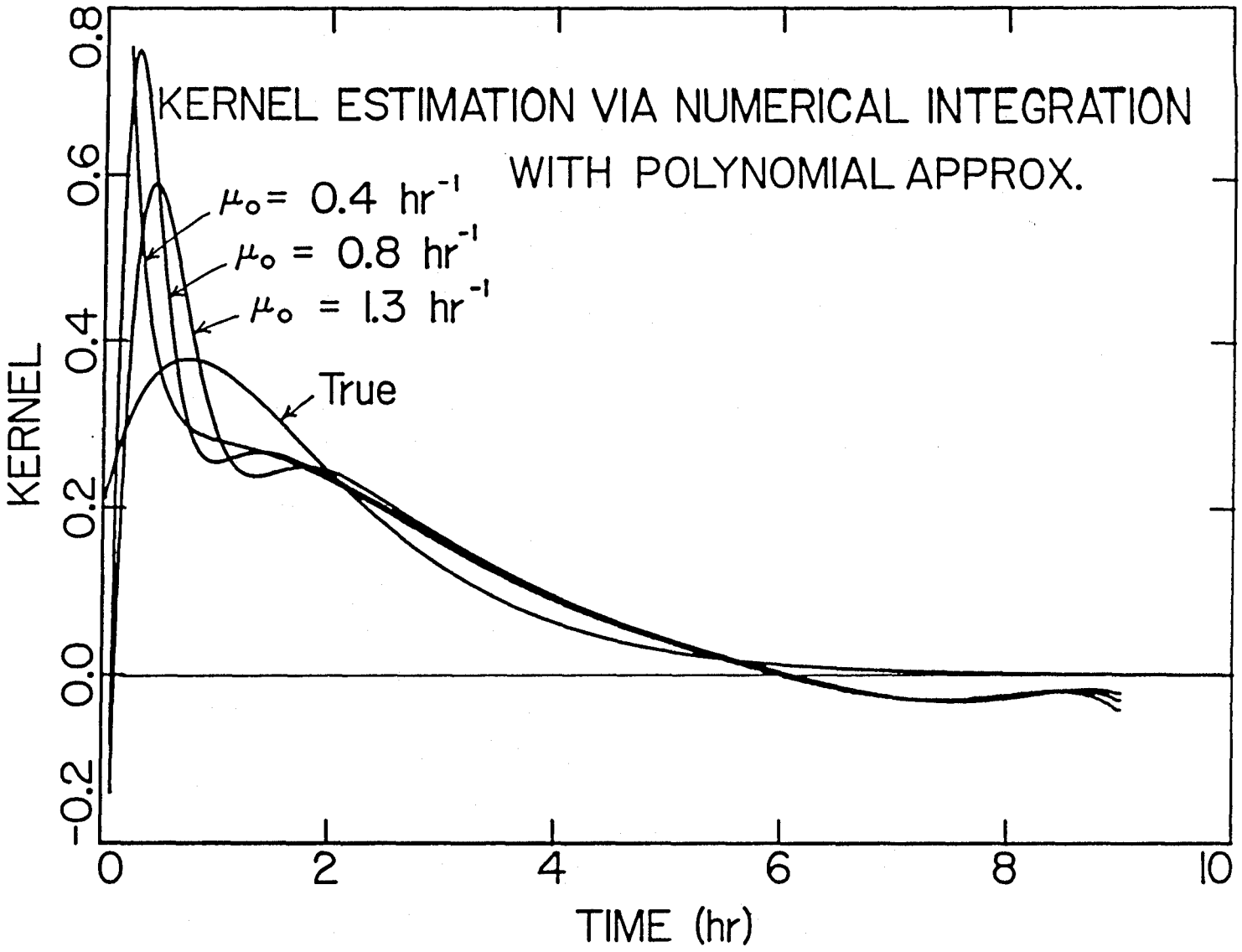


Figure 2.6.43. Kernel estimation by inverse numerical integration after the noisy (5%) $\mu(t)$ and $y(t)$ are smoothed by polynomial approximations.

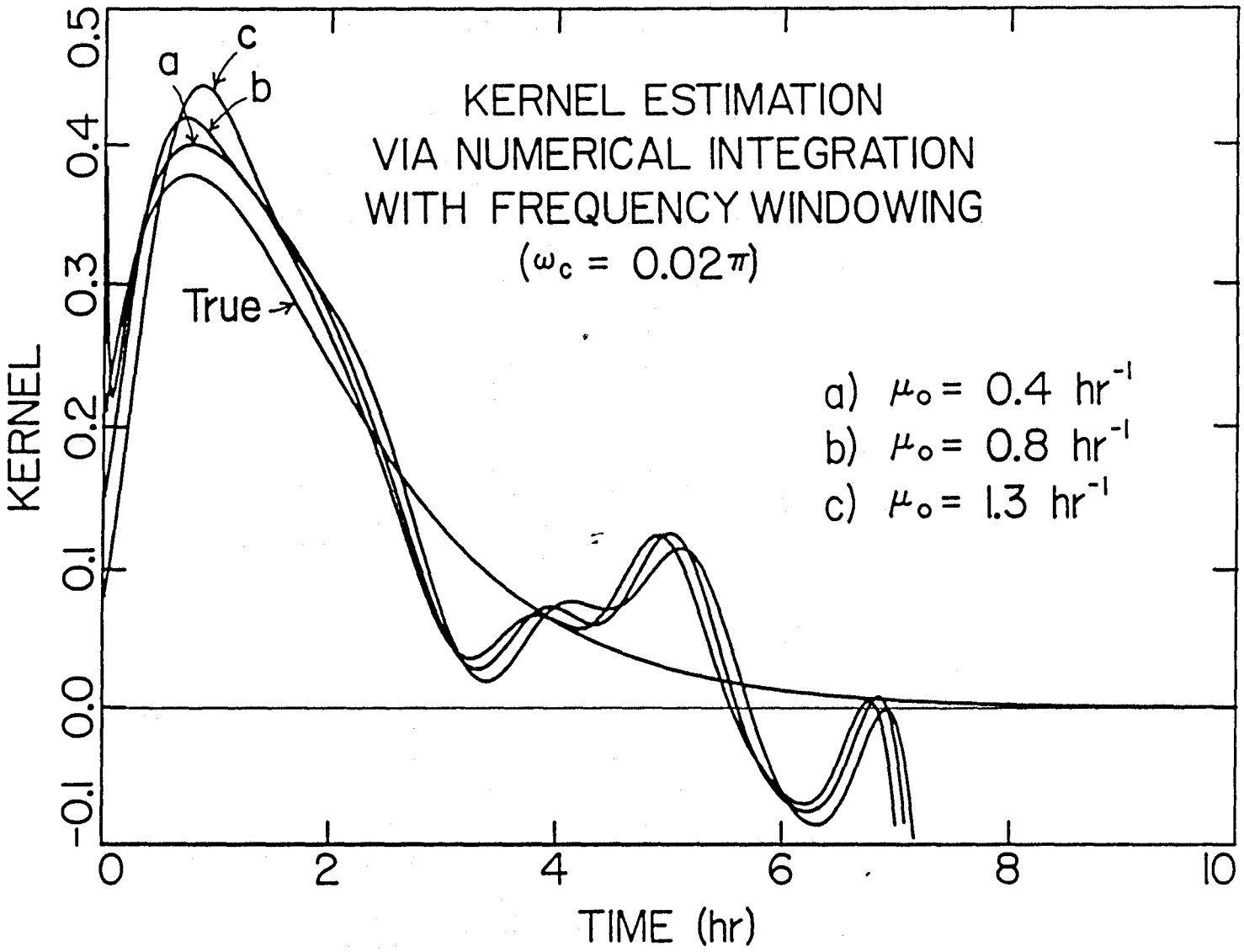


Figure 2.6.44. Kernel estimation by inverse numerical integration after the noisy (5%) $\mu(t)$ and $y(t)$ are smoothed by applying frequency windows with $\omega_c = 0.02\pi$.

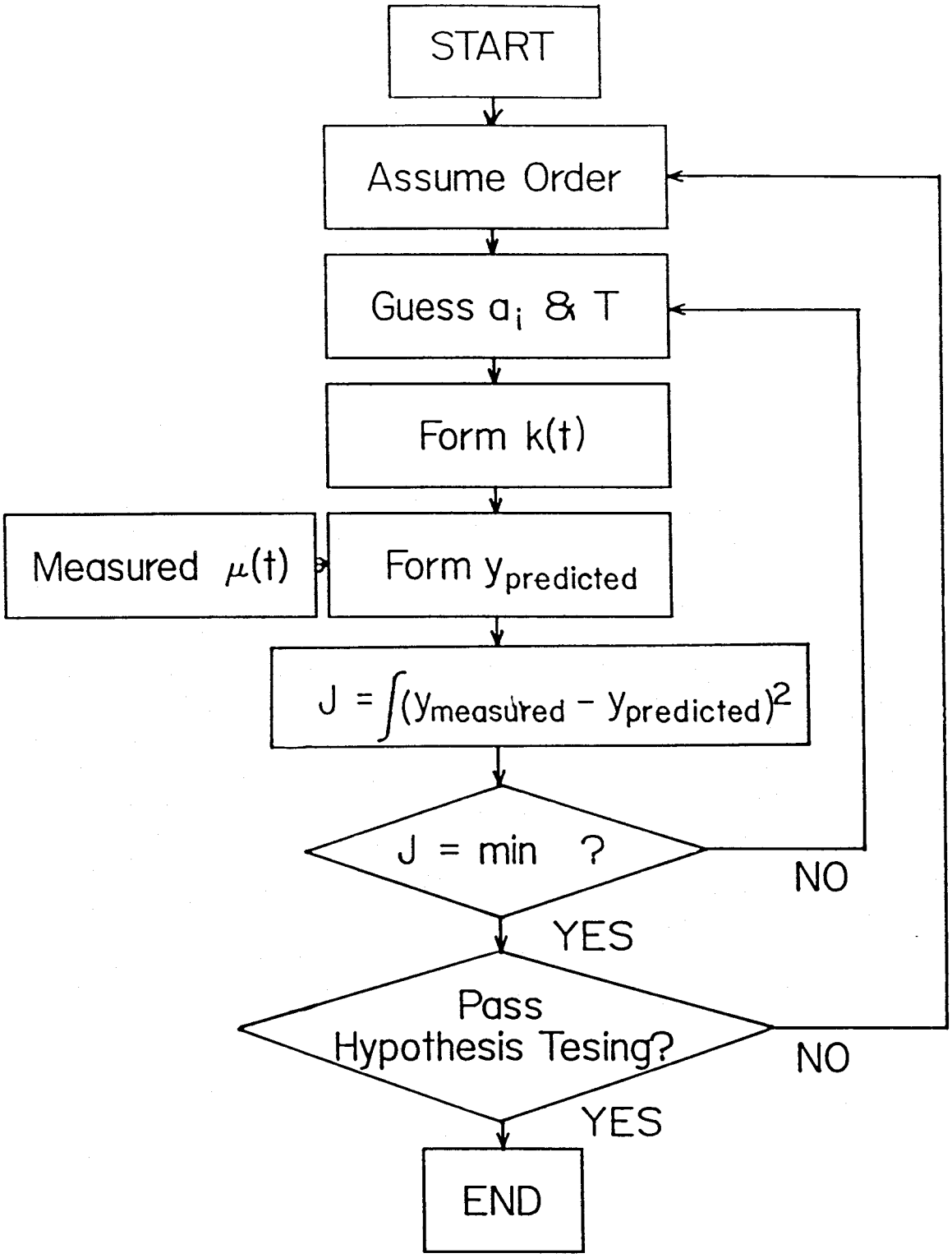


Figure 2.6.45. Double-looped flow diagram of kernel model parameter determination by searching for the minimum of the sum of square-errors between the measured $y(t)$ and the predicted $y(t)$.

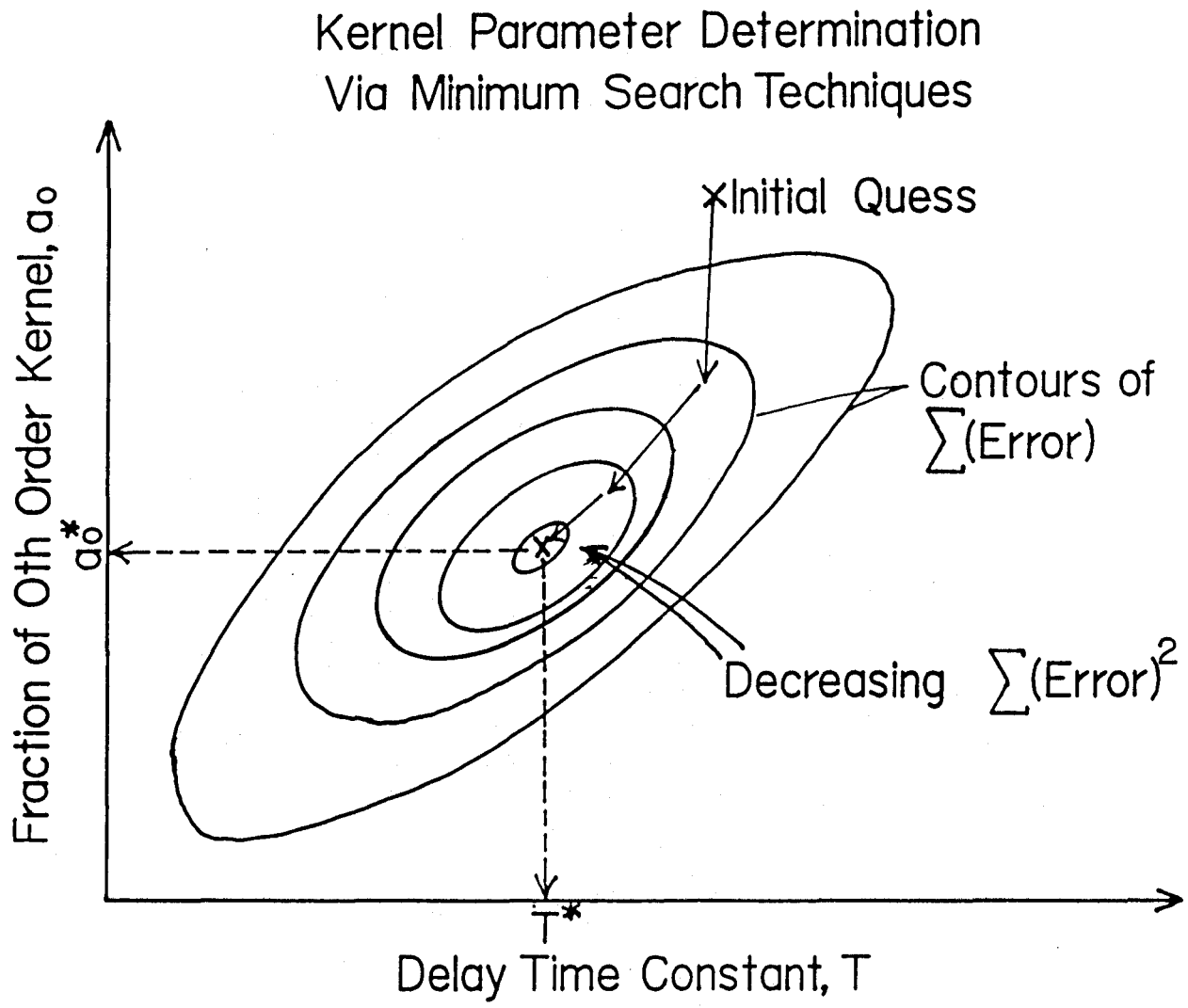


Figure 2.6.46. Example of two-dimensional minimization of square-errors for a first-order kernel.

If the assumed order of the kernel is too small, there are not enough parameters to fit adequately the measured system output, and a significant amount of errors can be expected. Conversely, if the assumed order of the kernel is correct, the amount of errors between the predicted $y(t)$ and the measured $y(t)$ will match the χ^2 probability distribution function. Increasing the kernel order beyond the true one will not reduce the objective function by any significant amount.

For example Figure 2.6.47 demonstrates the use of statistical hypothesis testing to derive the correct kernel order. Given the desired level of significance, *e.g.*, 95%, the point at which the cumulative probability distribution function is 0.95 can be located either from the graph or from a table. Following the dotted arrow in the figure, the corresponding value of the objective function is noted. The kernel order is increased until the minimum of the sum of square-errors comes down to the noted value. In Figure 2.6.47, a second-order kernel is needed to satisfy the specified confidence level.

In our simulation, the same system input and the true kernel functions considered throughout this section are used. Noises corresponding to 5 % of the true values are then added to the intrinsic specific growth rate $\mu(t)$ and the observed specific growth rate $y(t)$ of the culture to represent the uncertainty present in the actual measurements. The reconstruction of the kernel is initiated by assuming its order to be 0 and by picking an arbitrary value for the lag time-constant T. (A reasonable value for T should be of the same order of magnitude as the bioreactor time constant, *i.e.*, the reciprocal of the specific growth rate.) From the noisy transient data of $\mu(t)$ and $k(t)$, the curve for the theoretically predicted observed specific growth rate is generated by integrating the transformed differential equation for $y(t)$. The mean square deviation of the $y(t)$ predicted by the kernel away from the

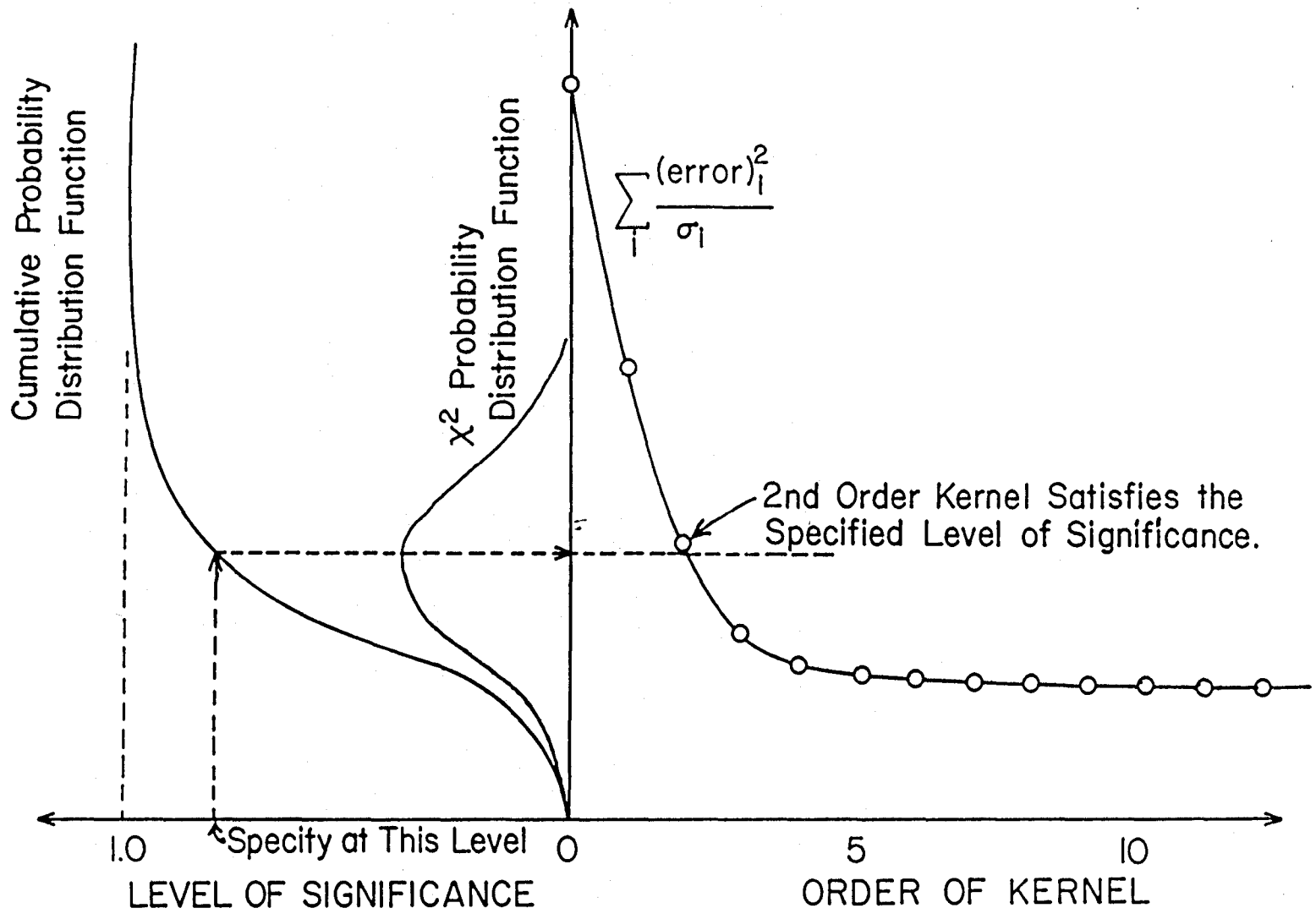


Figure 2.6.47. Kernel order determination through statistical hypothesis testing with a χ^2 probability distribution function.

observed $y(t)$ is then minimized over the choice of T.

$$\min_{T}(y_{measured} - y_{predicted})^2 \qquad (2.6.63)$$

After the above objective function is minimized with respect to T, the sum of normalized square deviations is obtained. A χ^2-test is used to test the original hypothesis that the kernel is 0th-order. A confidence interval of 95 % is a reasonably good value to apply to the hypothesis testing. If the test result on the minimized objective function indicates that the deviation is statistically significant, the order of the kernel $k(t)$ is increased by 1, and the entire minimization process is repeated. Otherwise, the hypothesis is accepted, and the reconstruction process is terminated. In this example, as the order of the kernel is increased from 2 to 3, the decrease in the objective function is negligible, the algorithm decides that the the order of the kernel is 1, which is indeed the true value. From the noisy transient data of $\mu(t)$ and $y(t)$, the kernel was reconstructed by minimizing the residue expressed in Equation (2.6.62). The simplified minimization algorithm is presented in the first part of Appendix D. The resulting kernel is shown in Figure 2.6.48c.

Simulation Studies of Other Types of Perturbation

Since the frequency response function or the pulse response function can be considered merely as another representation of the impulse response function (*i.e.*, the kernel), various other types of signals such as sinusoidal, periodic rectangular, periodic triangular, or pulse can all be utilized to determine experimentally the shape of the kernel. The previous example of a step increase in the dilution rate represents the most difficult case of estimating the impulse response function from a pulse experiment. Because the impulse response function is the slope of the output from a pulse input, the problem is basically that of estimating the slope of the output from a noisy curve, which is not trivial no matter which method is used. In

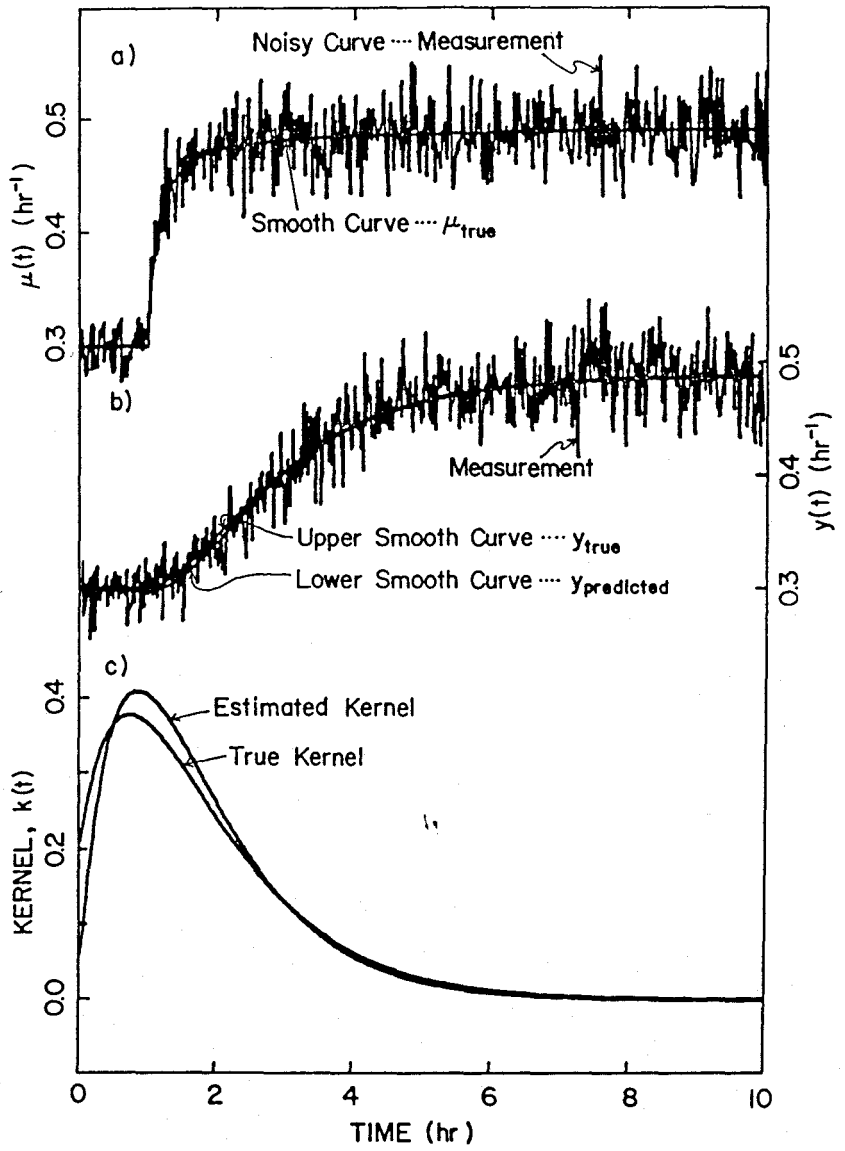
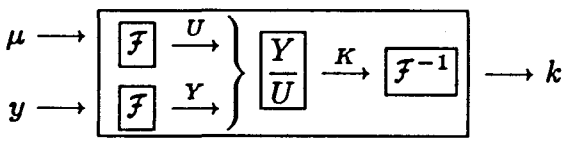


Figure 2.6.48. (a) Simulated input (*i.e.*, the specific growth rate in the absence of time-lag effects) as a function of time in a continuously operated bioreactor described by the state equations (2.2.10) and (2.2.11) after a shift-up in the dilution rate from 0.3 hr^{-1} to 0.7 hr^{-1}. (Parameters used: $\mu = \frac{0.5hr^{-1}s}{0.1g/l+s}$; $s_f = 5.0g/l$; $Y_s = 0.5g/g$; noise level in measurement = 5% .) (b) Simulated output (*i.e.*, the observed specific growth rate containing time-lag effects) as a function of time. (Upper smooth curve: the true value of $y(t)$; lower smooth curve: the calculated value of $y(t)$ based on the estimated kernel function of (c). (c) True and estimated shapes of the kernel.

general, a method can be expected to work well for other types of stimuli if it works well for a step input. Subjecting a numerical method to the toughest test helps to force out its deficiencies so that one is made aware of the reason and circumstances where an algorithm succeeds or fails. Furthermore, if possible, one would like to find a way of estimating the kernel from a strictly shift-up or shift-down experiment starting from a well defined steady-state so that the response from each case can be contrasted, especially if there exists any difference.

Although none of the previously described algorithms manages exceptionally well in noisy conditions, they work quite well when the direct input, not the derivative, is an impulse function or remotely resembles an impulse function. For example, the Fourier transform method is applied to a situation where the levels of both the input and output return to their respective starting values within the time-window, so that there are no discontinuities at the time-window boundaries when each function is repeated on both side of the time-window. This periodic version of the function is conceptually indicated in Figure 2.6.5d. As a specific example, Figures 2.6.49a and 2.6.49b display the results of simulating the response of the time-lag system where the dilution rate is suddenly increased from the original steady-state value of 0.3 hr^{-1} to 0.7 hr^{-1} at $t = 1$ hr and subsequently restored back to 0.3 hr^{-1} at $t = 2$ hr. The parameters used in this simulation are the following: $s_f = 5$ g/l, $Y_s = 0.5$ g/g, and $\mu(s) = \frac{0.5 hr^{-1} s}{1g/l+s}$. As before, Fourier transforms are performed on $\mu(t)$ and $y(t)$, which are transformed to $U(\omega)$ and $Y(\omega)$, respectively, and the results of the inverse transform of $\frac{Y(\omega)}{U(\omega)}$ are shown in Figure 2.6.50.



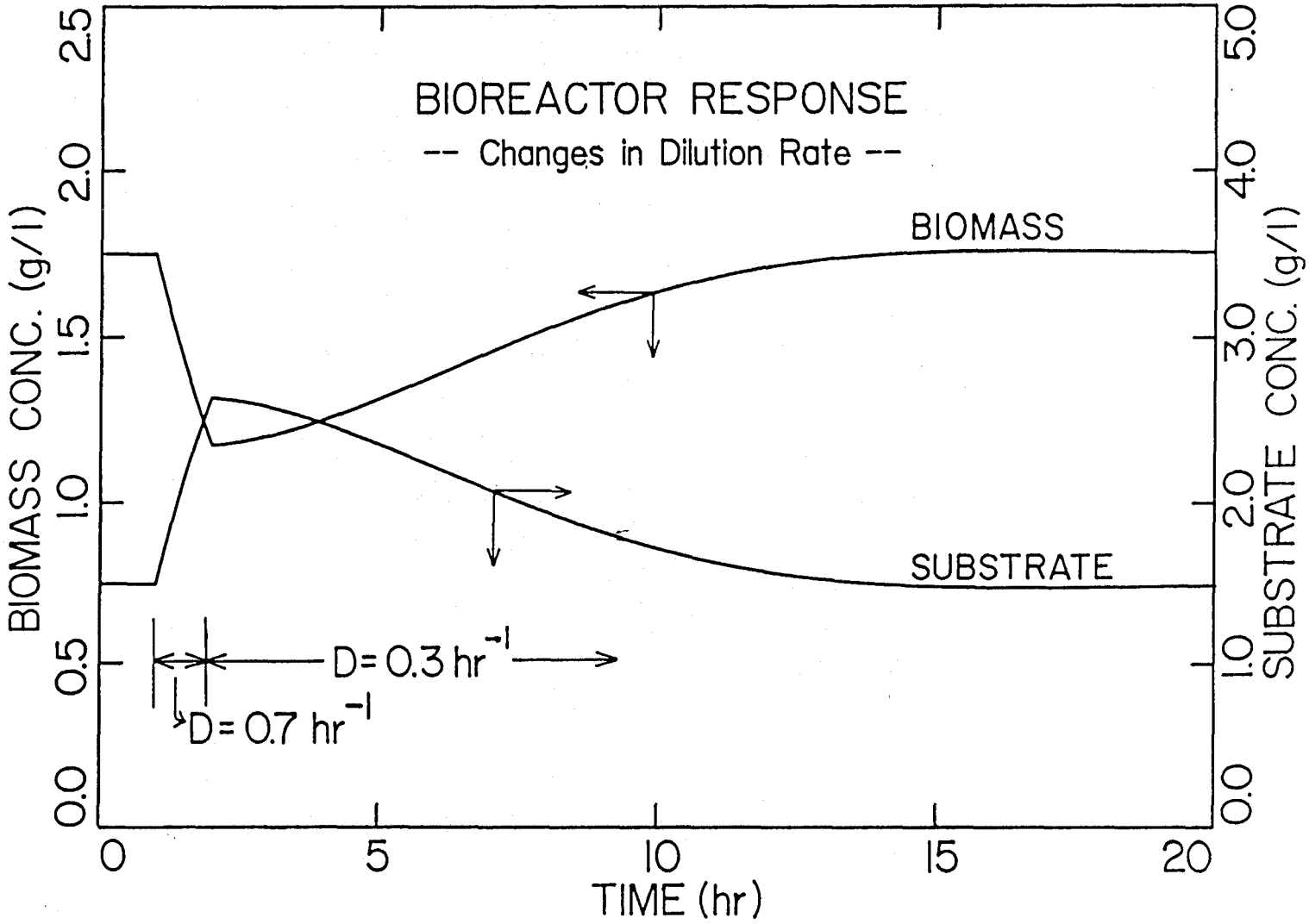


Figure 2.6.49a. Biomass and substrate concentrations in a continuously operated bioreactor after a shift-up in the dilution rate from 0.3 hr^{-1} to 0.7 hr^{-1} at 1 hr a symmetrical shift-down at 2 hr. (Parameters used: $\mu = \frac{0.5 hr^{-1} s}{0.1 g/l + s}$; $s_f = 5.0 g/l$; $Y_s = 0.5 g/g$).

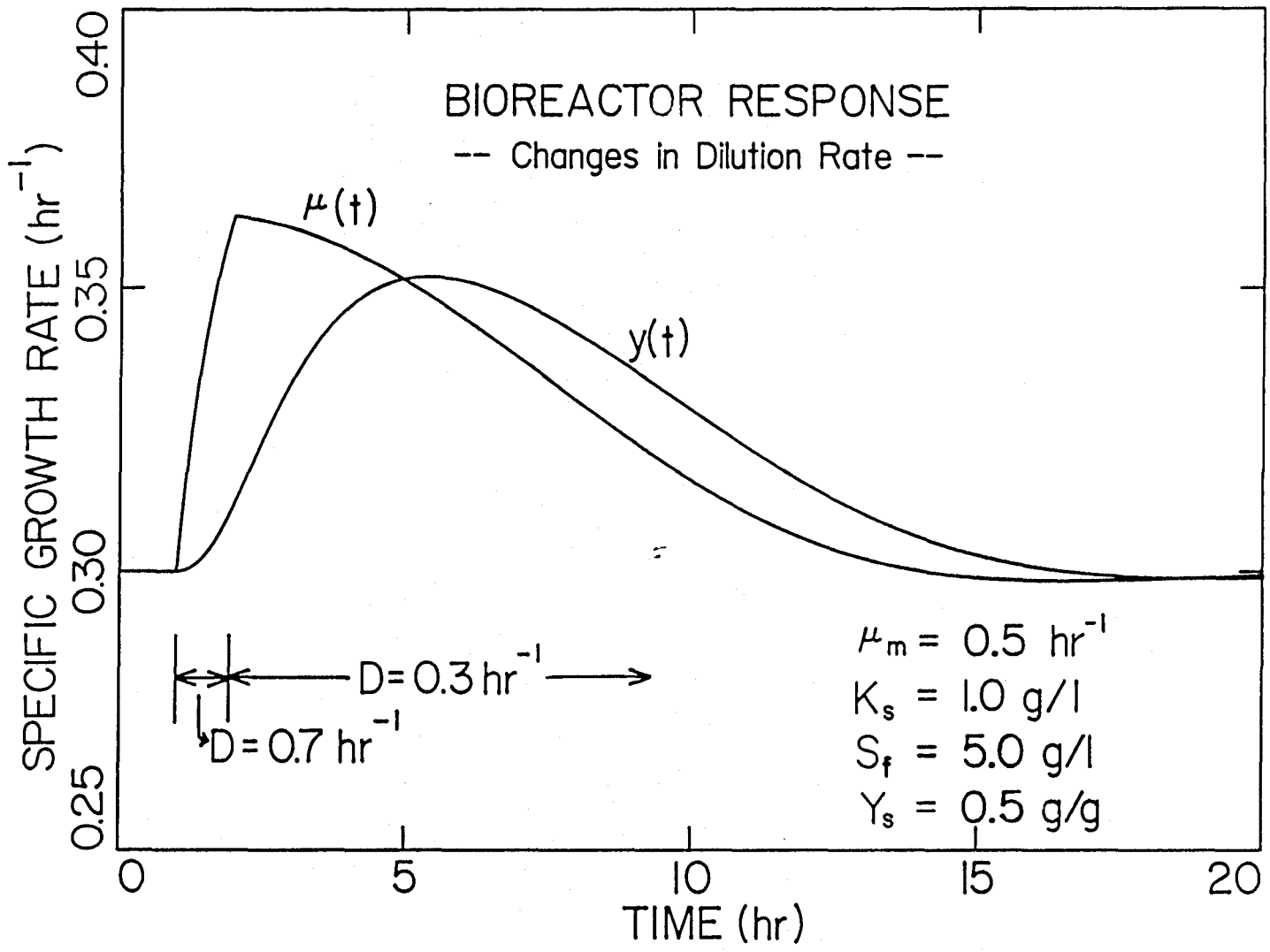


Figure 2.6.49b. Intrinsic and observed specific growth rates in a continuously operated bioreactor after a shift-up in the dilution rate from 0.3 hr^{-1} to 0.7 hr^{-1} at 1 hr a symmetrical shift-down at 2 hr. (Parameters used: $\mu = \frac{0.5 hr^{-1} s}{0.1 g/l + s}$; $s_f = 5.0 g/l$; $Y_s = 0.5 g/g$).

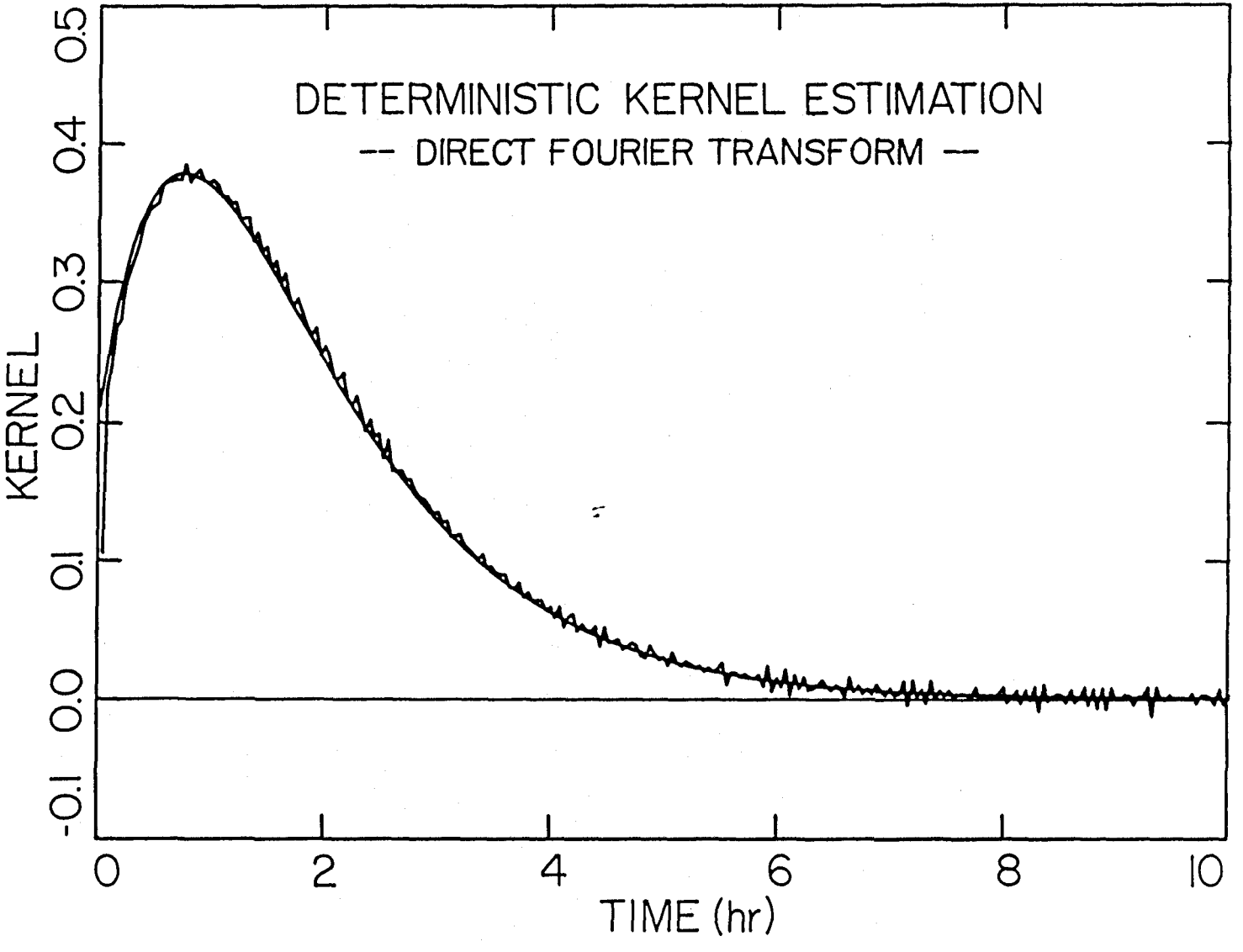
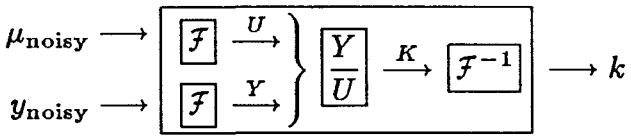


Figure 2.6.50. Direct calculation of the kernel based on

$$k(t) = \mathcal{F}^{-1}\{K(\omega)\} = \mathcal{F}^{-1}\{U^{-1}(\omega) \cdot Y(\omega)\}$$

Indeed, the Fourier transform method can be effectively utilized to estimate the kernel function in this deterministic case. The kernel estimated from the noisy input and output signals is shown in Figure 2.6.51, which indicates that the Fourier transform method can be effectively used.



Further extensive simulation shows that other methods can be used to give good results for this case.

Much better agreement between the true kernel and the estimated one can be achieved if an impulse can be applied to the system. This last simulation demonstrates the difficulty in achieving an impulse. Although only one hour of perturbation in the dilution rate is used, the system input $\mu(t)$ is not at all close to an impulse. Furthermore, the estimated kernel is in considerably better agreement with the true one if the noise level is decreased. (Of course, in almost all methods, the reconstructed kernel coincides with the true one in the absence of noise.)

Summary

Throughout this section, the system input, $\mu(t)$, is assumed to be the true specific growth rate in the absence of time-lag effects. Given $s(t)$, this true specific growth rate $\mu(t)$ is obtained from a μ versus s curve constructed from a series of steady state experiments, in which the time-lag effects are eliminated. The system output, $y(t)$, is the observed specific growth rate, which is derived from the slope of the culture growth curve. Much better results can be expected when a Kalman filter is applied to reduce the noise level in the measurement of the specific growth rate. This is especially true when the problem is more demanding, as in the case of a near step input to the system. In this case, Fourier transforms, as well as all other methods considered in this section, do not yield the exact answer, and deviations

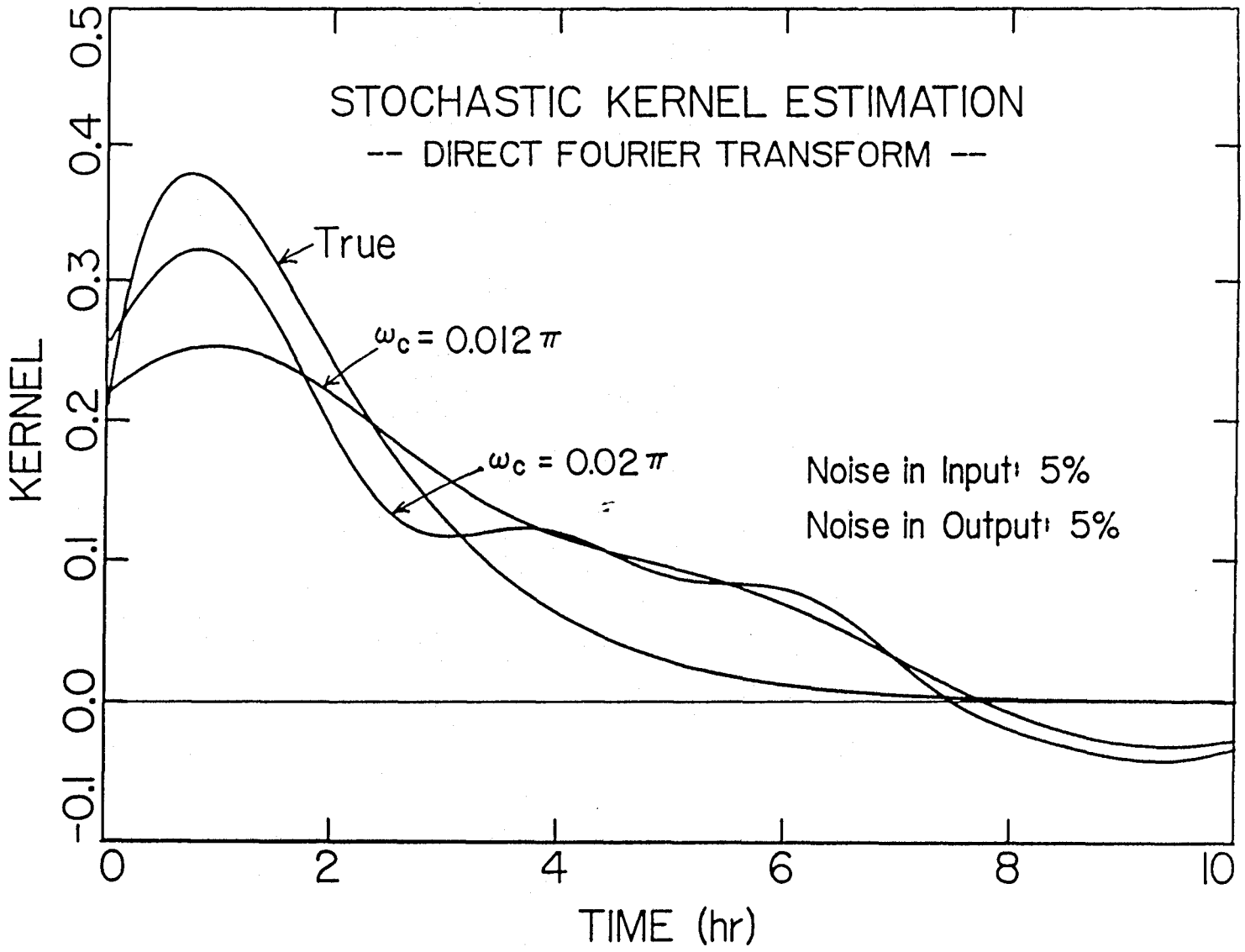


Figure 2.6.51. Stochastic kernel estimation via direct Fourier transform on noisy (5%) $\mu(t)$ and noisy (5%) $y(t)$.

between the true kernel and the estimated curve is rather significant when there is noise in the system output. This is because a kernel is sensitive to the system output and cannot be estimated exactly; conversely, the system output is insensitive to the kernel, and a slight variation in the kernel model parameters or in the functional shape does not significantly affect the predicted output.

Kernel deconvolution strongly depends on the nature of the disturbance. If the input is impulse-like, *i.e.*, if the value of the input starts and ends at the same level, the techniques of Fourier transforms can be directly applied without taking the difference operation to estimate the kernel function in the convolution integral. The Fourier transform method or any of the variations thereof also give good results if the input is periodic. In these cases, the estimation of the kernel can be accomplished without resorting to the use of exponential distribution functions, although it may eventually be approximated by an expansion of such functions to facilitate the mathematical analysis of the system.

In summary, the following is a list of the methods discussed in this section.

Method	Use of Base Functions
1 Fourier Transform	No
a) Noise Reduction via Linear Regression	No
b) Noise Reduction via Frequency Windows	No
2 Time Series Analysis	No
3 Polynomial Approximations	Yes
4 Iterative Cyclic Approximation	No
5 Inverse Numerical Integration	No
6 Quadratic Minimum Search	Yes

Of these methods, only polynomial approximation and quadratic minimum search are confined to the use of linear combination of base functions. All other methods are not confined by any specific set of base functions; thus, they are generally applicable to any arbitrary functions and at times may be preferred. Nonetheless, they may be slightly modified to accommodate exponential distribution base functions. One way is to approximate the resulting kernel function by a set of these base functions at the end of calculation. Typical computer programs used to calculate the impulse response function by utilizing Fourier transforms and other principles are listed in Appendix D.

2.7 LOCAL STABILITY ANALYSIS

Stability analysis of an open loop system is of utmost importance in any process control consideration. For a successful venture, one must have a model that can display the correct dynamic behaviors, especially the instabilities, oscillations, and hysteresis. An inappropriate choice of models can lead to an unfruitful exercise. No advanced control theories can fully rectify the mistake of choosing a model that cannot describe the oscillatory instabilities if they are indeed observed in practice. Any existing inherent unstable tendencies must be fully incorporated in any sound control strategies. This section attempts to analyze theoretically the stability of a time-lag model.

One of the advantages of transforming an integral differential equation into a purely differential one is that one can directly utilize the well-developed stability and bifurcation theories to analyze mathematically the behavior of the time-lag system in the same way as one would analyze a system with no time-lag. The type of dynamic equations most frequently encountered in biochemical engineering, including the examples used here, are nonlinear in nature. Although there are a number of methods generally employed for the analysis of a set of nonlinear

dynamic equations, each method is limited in applicability, and there is currently no general technique that can be universally applied. In this section, the dynamics of a chemostat culture in the presence of time-lag kernels is analyzed, employing techniques from the classical linearized stability analysis.

Formulation of State Equations

As a fundamental example, the non-dimensionalized state equations for a simple chemostat, without product formation, is presented below.

$$\frac{dx(t)}{dt} = [-1 + \mu[s(t)]]\, x(t) \tag{2.7.1a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s}\mu[s(t)]x(t) \tag{2.7.1b}$$

The states in the above equations are clearly the biomass concentration, $x(t)$, and the limiting substrate concentration, $s(t)$. The model parameters are the specific growth rate, μ, and the substrate to cell yield coefficient, Y_s. The stability of this classical system has been previously analyzed (Koga and Humphrey, 1967).

It is conceivable that time-lag effect may be present in either or both of these growth parameters. First, the presence of a time-lag only in the specific growth rate will be considered.

$$\frac{dx(t)}{dt} = \left[-1 + \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right] x(t) \tag{2.7.2a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s}\left[\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right] x(t) \tag{2.7.2b}$$

These dynamic equations can be analyzed more readily, if the observed specific growth rate is treated as an additional state, so that the integrals in the above equations are eliminated.

$$\frac{dx(t)}{dt} = [-1 + y(t)]\, x(t) \tag{2.7.3a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s}y(t)x(t) \tag{2.7.3b}$$

$$y(t) = \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh = \int_{0}^{\infty} \mu[s(t-h)]k(h)dh \qquad (2.7.4)$$

As discussed in previous sections, the differential form of the last equation depends on the order of the kernel. For a 0th-order kernel $k_0(t)$, the last equation is equivalent to one additional differential equation:

$$\frac{dy(t)}{dt} = \frac{1}{T}[-y(t) + \mu(t)] \qquad \text{0th Order} \qquad (2.7.5)$$

For a purely 1st-order kernel $k_1(t)$, the two additional differential equations are:

$$\left.\begin{aligned} \frac{dy(t)}{dt} &= z(t) \\ \frac{dz(t)}{dt} &= \frac{1}{T^2}[-2Tz(t) - y(t) + \mu(t)] \end{aligned}\right\} \qquad \text{1st Order} \qquad (2.7.6)$$

For a mixed 0th- and 1st-order kernel $k(t) = a_0k_0(t) + a_1k_1(t)$, the following two equations are added.

$$\left.\begin{aligned} \frac{dy(t)}{dt} &= z(t) \\ \frac{dz(t)}{dt} &= \frac{1}{T^2}\left[-2Tz(t) - y(t) + \mu(t) + a_0T\frac{d\mu(t)}{dt}\right] \end{aligned}\right\} \qquad \text{Combined 1st Order} \qquad (2.7.7)$$

Steady-States

The steady-state values of x_0 and s_0 are obtained by setting the above differential equations to zero. Thus, the manipulation is purely an algebraic one.

$$0 = [-1 + y_0]\,x_0 \qquad (2.7.8a)$$

$$\Rightarrow \quad y_0 = 1 \;\text{ or }\; x_0 = 0 \quad \text{(Washout)}$$

$$0 = 1 - s_0 - \frac{1}{Y_s}y_0x_0 \qquad (2.7.8b)$$

$$\Rightarrow \quad x_0 = Y_s(1 - s_0)$$

$$y_0 = \lim_{t\to\infty}\int_0^{\infty} \mu[s(t-h)]k(h)dh$$

$$= \mu[s_0]\underbrace{\int_0^{\infty} k(h)dh}_{1} = \mu[s_0] \qquad (2.7.9)$$

$$\Rightarrow \quad \mu(s_0) = \mu_0 = 1 \text{ or } s_0 = \mu^{-1}(1)$$

Alternatively, the steady-state value for μ_0 can be obtained by setting each of the differential equations of $\frac{dy(t)}{dt}$ and/or $\frac{dz(t)}{dt}$ to 0. Thus, the steady-state values of x_0, s_0, and μ_0 are not altered by the presence of time-lag effects. This is certainly expected, because there should be no time-lag effects at steady-states.

0th-Order Kernel

Linearization: First, how the stability will be affected by the inclusion of a 0th-order time-lag in the specific growth rate will be investigated via the linearized state equations around the steady-state values. The transformed state equations are:

$$\frac{dX(t)}{dt} = \overbrace{(-1 + y_0)}^{0} X(t) + x_0 Y(t) \tag{2.7.10a}$$

$$\frac{dS(t)}{dt} = -\frac{1}{Y_s} \overbrace{y_0}^{1} X(t) - S(t) - \frac{1}{Y_s} x_0 Y(t) \tag{2.7.10b}$$

$$\frac{dY(t)}{dt} = \frac{1}{T}\mu_0' S(t) - \frac{1}{T} Y(t) \tag{2.7.10c}$$

The transformed variables $X(t)$, $S(t)$, and $Y(t)$ are defined as:

$$X(t) = x(t) - x_0 \tag{2.7.11a}$$

$$S(t) = s(t) - s_0 \tag{2.7.11b}$$

$$Y(t) = y(t) - y_0 \tag{2.7.11c}$$

And μ_0' is the slope of the μ versus s curve at the steady-state value of s.

$$\mu_0' = \left.\frac{d\mu(s)}{ds}\right|_{s_0} \tag{2.7.12}$$

Jacobian Matrix and Characteristic Equation: The Jacobian matrix for the linearized set of differential equations is:

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & x_0 \\ -\frac{1}{Y_s} & -1 & -\frac{1}{Y_s} x_0 \\ 0 & \frac{1}{T}\mu_0' & -\frac{1}{T} \end{bmatrix} \tag{2.7.13}$$

The characteristic equation for the differential equations (2.7.10a–c) is:

$$0 = \det|\lambda \mathbf{I} - \mathbf{J}| = \begin{vmatrix} \lambda & 0 & -x_0 \\ \frac{1}{Y_s} & \lambda + 1 & \frac{1}{Y_s}x_0 \\ 0 & -\frac{1}{T}\mu_0' & \lambda + \frac{1}{T} \end{vmatrix}, \qquad (2.7.14)$$

$$0 = \lambda^3 + (1 + \frac{1}{T})\lambda^2 + \frac{1}{T}(1 + C)\lambda + \frac{1}{T}C$$

where C is the stability variable that persistently appears in stability analysis.

$$C = \frac{1}{Y_s}\mu_0' x_0 = \mu_0'(1 - s_0) \qquad (2.7.15)$$

Necessary and Sufficient Conditions: The first-order necessary and sufficient condition for the system of differential equations to be asymptotically stable is that all the roots of the characteristic equation must lie to the left of the imaginary axis on complex plane, *i.e.*, the real part of each of the roots shall be negative.

There are many ways to analyze this characteristic equation. Solving the roots directly is not always the best approach, especially in view of the fact that currently there exists no formula for degrees of 5 or above. Since our objective is to find the *signs* of the roots, the application of such classical tests as the Routh test may be more appropriate. The Routh test states that all the coefficients a_is in the following nth-degree characteristic equation must be positive.

$$a_0\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \ldots + a_n = 0 \qquad (2.7.16)$$

Secondly, all the elements in the first column of the Routh array being positive constitutes the necessary and sufficient condition for a stable system, where the

Routh array is formulated for a nth-degree polynomial as follows.

Row	Elements			
1	a_0	a_2	a_4	...
2	a_1	a_3	a_5	...
3	b_1	b_2		
4	c_1	c_2		
5	d_1			
6	e_1			

The first two rows are taken from the coefficients in the characteristic equation. The subsequent rows are calculated as:

$$\begin{array}{llll} b_1 = \frac{a_1 a_2 - a_0 a_3}{a_1} & b_2 = \frac{a_1 a_4 - a_0 a_5}{a_1} & \cdots \\ c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1} & c_2 = \frac{b_1 a_5 - a_1 b_3}{b_1} & \cdots \\ \cdots & \cdots & \cdots \end{array} \tag{2.7.17}$$

Thus, the Routh array for a 0th-order kernel is:

0th-Order Kernel

Row	Routh Array Elements	
1	1	$\frac{1}{T}(1+C)$
2	$1+\frac{1}{T}$	$\frac{1}{T}C$
3	$\frac{1}{T}[(1+C)(1+\frac{1}{T})-C]$	
4	$\frac{1}{T}C$	

The consequence of the first condition on the signs of the coefficients of the characteristic equation is that $C > 0$ must hold for a stable non-washout steady-state. Subsequent analysis of the Routh array also confirms that $C > 0$ is the most restrictive condition in the sense that all other conditions are automatically satisfied if $C > 0$. Since $1 - s_0$ is always positive from material constraints, the necessary

condition for a stable steady-state reduces to $\mu_0' > 0$ for a system with a time-lag in the specific growth rate characterized by a 0th-order kernel. This is the same as the original two-dimensional state equations without time-lag.

The stability for a general constitutive form of $\mu(s)$ is illustrated in Figure 2.7.1, where the stability can be easily inferred from the slope of the $\mu(s)$ curve as it crosses the operating curve corresponding to the dilution rate. The operating curve is a constant of 1 due to the non-dimensionalization by the dilution rate. Having a negative slope, the second steady-state for a substrate inhibition kinetics is always unstable.

The inclusion of a 0th-order kernel in the time-lag makes it possible for the roots of the characteristic equation to have imaginary parts; whereas, the roots for the state equations without time-lag are always real, and the approach to the steady-state is always exponential. The direct consequence is that the system approach to the non-washout steady-state can be oscillatory if the time delay constant T is large enough. The boundary between the exponential and the damped oscillatory return to steady-state can be calculated from the following algebraic equation, which corresponds to repeated roots for the characteristic equation.

$$\left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2 = 0, \tag{2.7.18a}$$

where

$$p = \frac{1}{3T^2}\left(-T^2 + T + 3TC - 1\right) \tag{2.7.18b}$$

$$q = \frac{1}{27T^3}\left(2T^3 + 24T^2 - 9T^2C - 3T - 9TC + 2\right). \tag{2.7.18c}$$

The graphical interpretation of the stability variable C is illustrated in Figure 2.7.2. It is the slope of the curve $\mu(s)$ at s_0 multiplied by the baseline $1 - s_0$, or equivalently it is the length of the right hand side of the triangle extending from

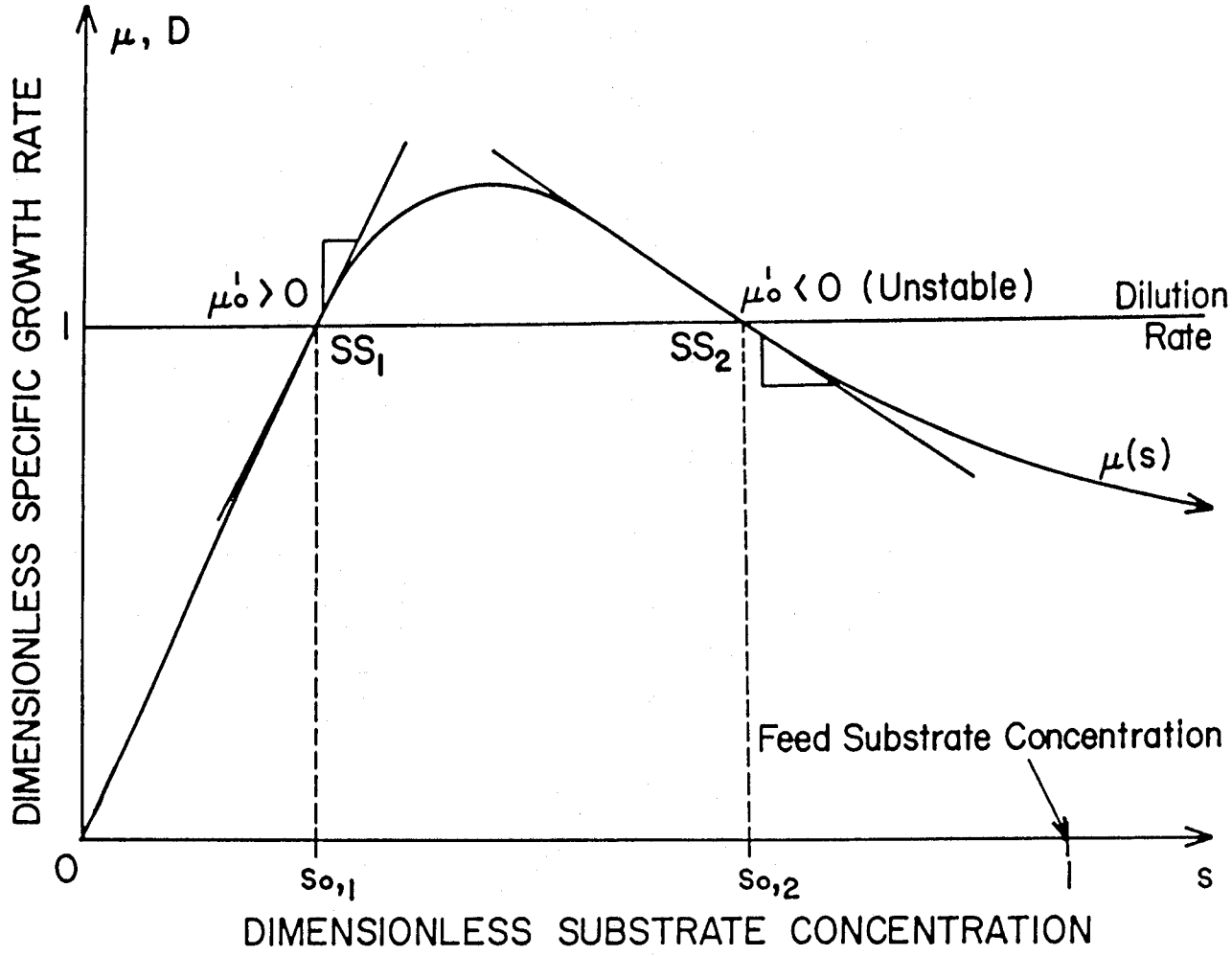


Figure 2.7.1. Stability of a set of two-dimensional chemostat state equations depends on the sign of $\frac{d\mu(s_0)}{dt}$. The steady-state is always unstable if $\frac{d\mu(s_0)}{dt<0}$.

$s = 1$. The effect of changes in the dilution rate on C is illustrated in that figure. The region of damped oscillations is mapped in Figure 2.7.3.

Mixed 0th- & 1st-Order Kernel

Although the 0th-order kernel is not strong enough to cause the non-washout steady-state to be outright unstable, it can induce damped oscillations. Thus, one may expect that the system may be forced to become unstable if the order of time-lag is increased. As shown later, this is indeed so when a 1st-order kernel is employed.

When the stability variable C or the time-lag constant T is small, the roots of the above fourth-order characteristic equation lie to the left of Point A in Figure 2.7.4. They are all real and negative, and the system approaches the non-washout steady-state in an exponential manner. As C or T is increased, the roots pass through Point A where the roots are repeated. This corresponds to the oscillation boundary in that figure. As C or T is increased even further, the repeated roots split into the imaginary plane in mirror images, and the system approaches the steady-state with damped oscillations. Finally, Point B in Figure 2.7.4 is reached, where the roots cross the imaginary axis. The non-washout steady-state is no longer stable. It can be proved that at this point the system enters into a state of sustained oscillations. Briefly, it can be shown that:

a) there is no stable steady-state to attract the trajectories in the x and s plane;

b) the trajectories must be confined within the boundaries imposed by material balance, *e.g.*, $0 \leq s(t) \leq 1$, $0 \leq x(t) \leq \frac{1}{Y_s}$, $0 \leq y(t) \leq \max \mu(s)$, and $0 \leq z(t) \leq \max \frac{d\mu(s)}{ds}$;

c) the trajectories cannot cross each other at any one point within these boundaries, for doing so would violate the fundamental existence and uniqueness

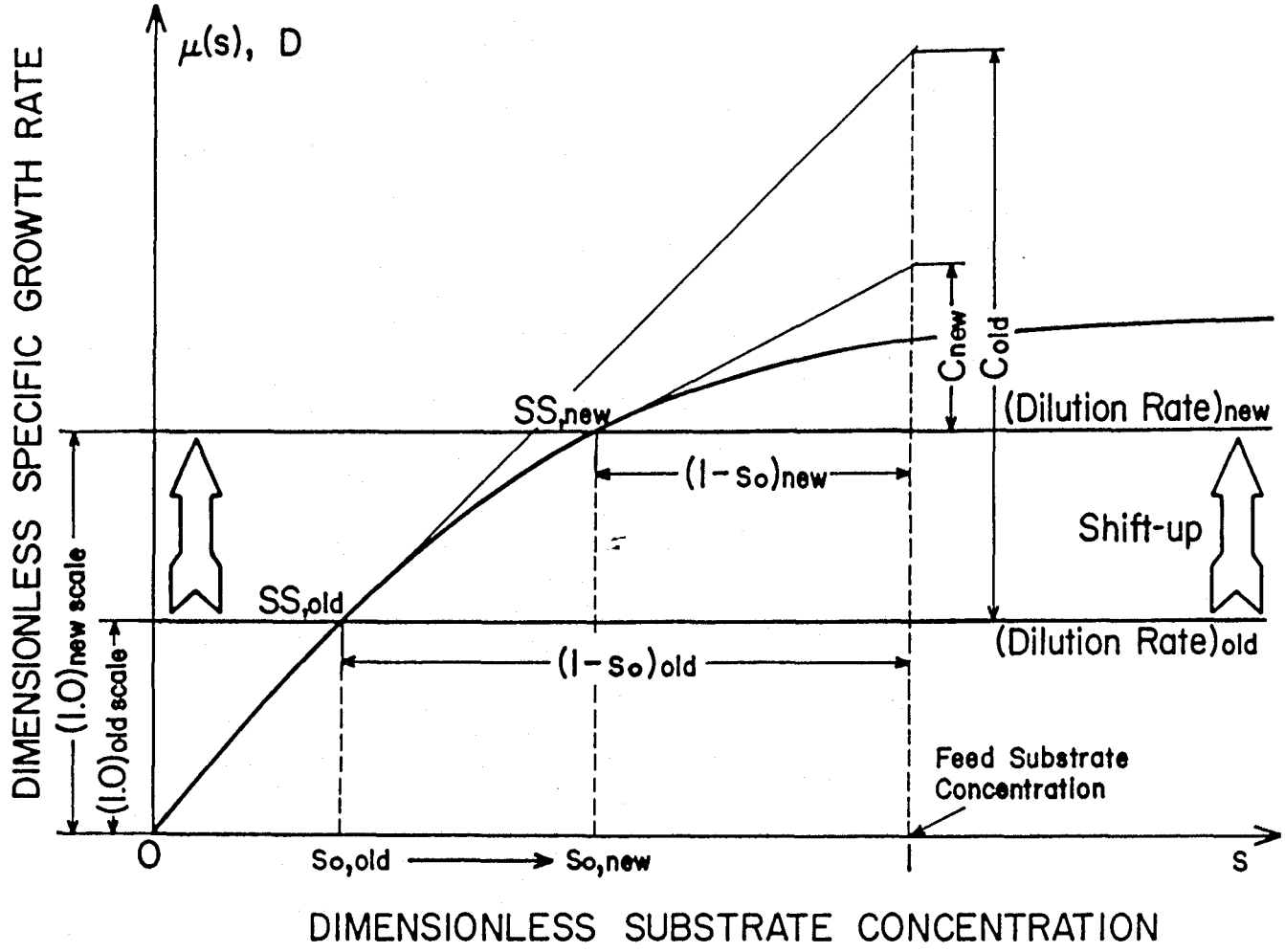


Figure 2.7.2. Graphical interpretation of the stability variable C. The effect of shifts in the dilution rate on C is shown graphically.

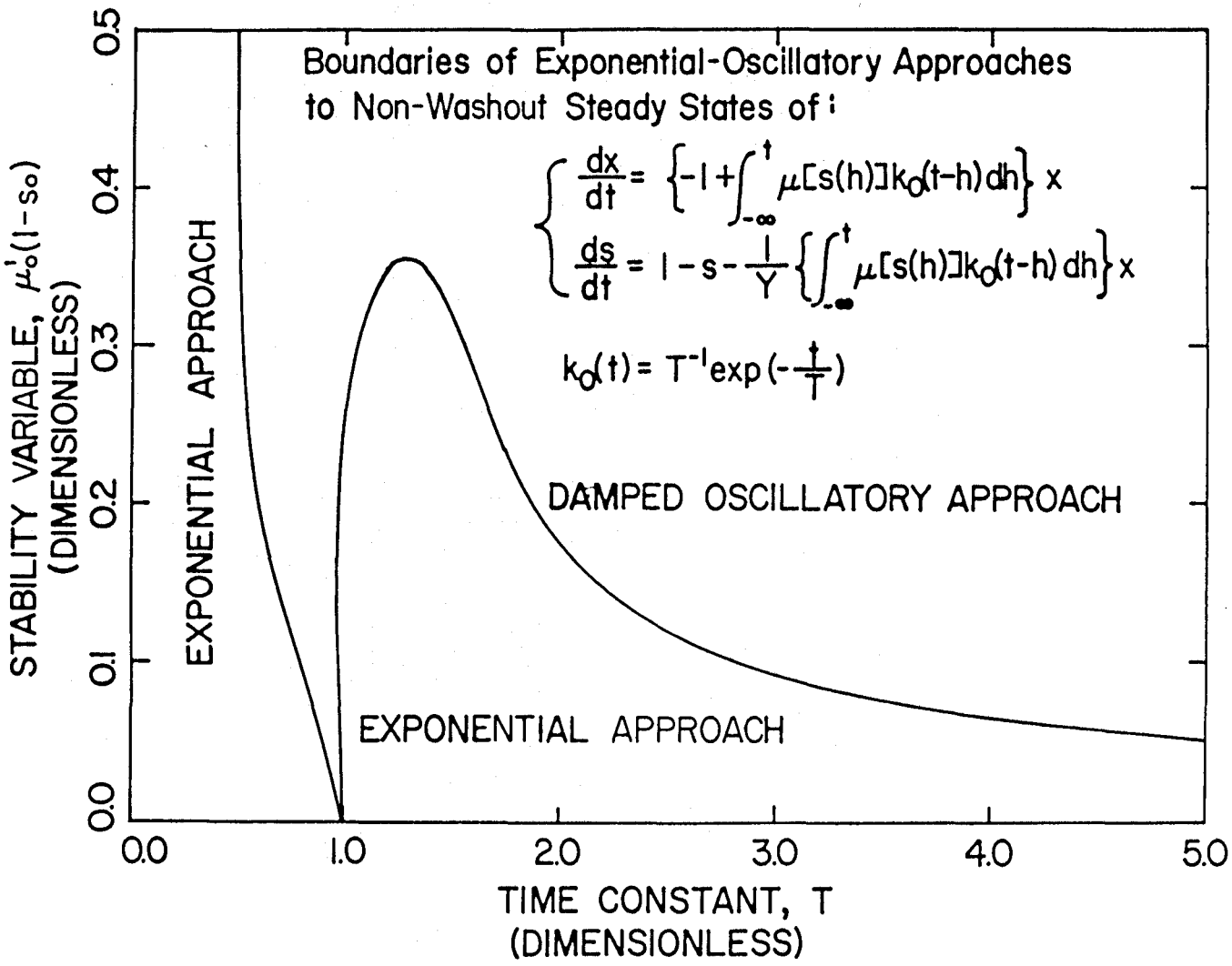


Figure 2.7.3. Oscillation boundary for a two-dimensional chemostat with a 0th-order time-lag kernel.

theorem of ordinary differential equations.

Under these constraints, the trajectories must be oscillatory. In a two-dimensional system, the trajectories must enter a stable limit cycle; however, for a higher-dimensional system, they need not be. For example, in a three-dimensional system the trajectories may have the option of spiraling around the surface of a torus.

To find these oscillation and stability boundaries corresponding to Points A and B in Figure 2.7.4 for a mixed 0th- and 1st-order kernel $k(t) = a_0 k_0(t) + a_1 k_1(t)$, a similar approach as outlined for the 0th-order kernel is performed. First, the linearized set of equations are:

$$\frac{dX(t)}{dt} = \overbrace{(-1 + y_0)}^{0} X(t) + x_0 Y(t) \tag{2.7.19a}$$

$$\frac{dS(t)}{dt} = -\frac{1}{Y_s} \overbrace{y_0}^{1} X(t) - S(t) - \frac{1}{Y_s} x_0 Y(t) \tag{2.7.19b}$$

$$\frac{dY(t)}{dt} = Z(t) \tag{2.7.19c}$$

$$\begin{aligned} \frac{dZ(t)}{dt} = &-a_0 \frac{1}{T} \frac{1}{Y_s} \mu_0' \overbrace{y_0}^{1} X(t) + \frac{1}{T^2}(1 - T a_0) \mu_0' S(t) \\ &- \frac{1}{T^2}(1 + T a_0 C) Y(t) - \frac{2}{T} Z(t) \end{aligned} \tag{2.7.19d}$$

The corresponding Jacobian matrix and characteristic equations are:

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & x_0 & 0 \\ -\frac{1}{Y_s} & -1 & -\frac{1}{Y_s} x_0 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 \frac{1}{T} \frac{1}{Y_s} \mu_0' & \frac{1}{T^2}(1 - T a_0) \mu_0' & -\frac{1}{T^2}(1 + T a_0 C) & -\frac{2}{T} \end{bmatrix} \tag{2.7.20}$$

$$0 = \det|\lambda \mathbf{I} - \mathbf{J}| = \begin{vmatrix} \lambda & 0 & -x_0 & 0 \\ \frac{1}{Y_s} & \lambda + 1 & \frac{1}{Y_s} x_0 & 0 \\ 0 & 0 & \lambda & -1 \\ a_0 \frac{1}{T} \frac{1}{Y_s} \mu_0' & -\frac{1}{T^2}(1 - T a_0) \mu_0' & \frac{1}{T^2}(1 + T a_0 C) & \lambda + \frac{2}{T} \end{vmatrix}$$

$$0 = \lambda^4 + \frac{1}{T}(T + 2)\lambda^3 + \frac{1}{T^2}(2T + 1 + T a_0 C)\lambda^2 + \frac{1}{T^2}(1 + C + T a_0 C) + \frac{1}{T^2} C \tag{2.7.21}$$

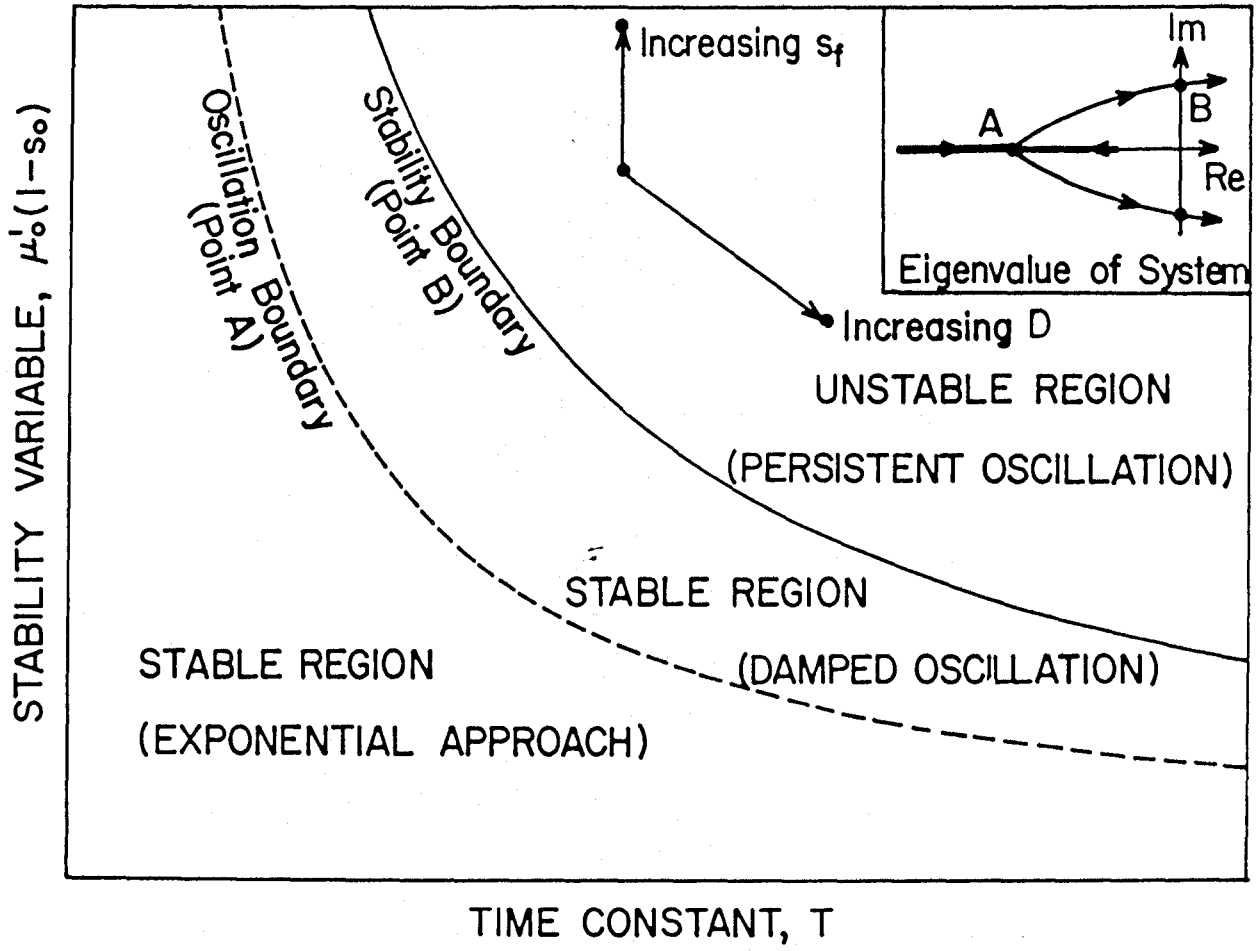


Figure 2.7.4. Oscillation and stability boundaries depend on the stability variable $C = \mu_0'(1 - s_0)$ and the lag time constant T. They are calculated from the critical points A and B in the roots of the characteristic equation.

And the following Routh array is obtained to perform the Routh test.

Mixed 0th- & 1st-Order Kernel

Row	Routh Array Elements		
1	1	$\frac{1}{T^2}(2T+1+Ta_0C)$	$\frac{1}{T^2}C$
2	$\frac{1}{T}(T+2)$	$\frac{1}{T^2}(1+C+Ta_0C)$	0
3	$\frac{1}{T^3}(T+2)(2T+1+Ta_0C)-\frac{1}{T^2}(1+C+Ta_0C)$	$\frac{1}{T^3}(T+2)C$	
4	c_1		
5	$\frac{1}{T^3}(T+2)C$		

The term c_1 in the above Routh array is the most restrictive one.

$$0 < c_1 b_1 = \left[\frac{1}{T^3}(T+2)(2T+1+Ta_0C)-\frac{1}{T^2}(1+C+Ta_0C)\right]\left[\frac{1}{T^2}(1+C+Ta_0C)\right] - \frac{1}{T^4}(T+2)^2C \tag{2.7.22}$$

After some algebraic manipulation, the above inequality is reduced to a polynomial form.

$$T(1+Ta_0)C^2+\left[(T^3+2T^2+T-2\frac{(1+Ta_0)}{(1-2a_0)}\right]C-\frac{(2T^2+4T+2)}{(1-2a_0)}<0 \tag{2.7.23}$$

After applying the quadratic formula, the two solutions of the above equation are found in the following simple forms, of which only the boxed second solution is of any practical concern.

$$-\frac{(T^2+2T+1)}{(1+Ta_0)} < \boxed{0<C<\frac{2}{(1-2a_0)T}} \tag{2.7.24}$$

The stability boundaries for some representative values of a_0 are plotted in Figure 2.7.5. Note that there exist stability boundaries for $a_0 < 0.5$, which indicates that the system can be unstable if C or T is large. For $0.5 \leq a_0$ the 0th-order kernel contributes more significantly to the combined kernel, and the roots of the

characteristic equation do not cross the imaginary axis for $0 < C$ as in the purely 0th-order kernel. Note that $a_0 = 0$ corresponds to a purely 1st-order kernel. Although not shown, the boundaries for the onset of damped oscillations exist for all a_0 to the left of the stability boundaries.

Higher-Order Kernels and Higher-Dimensional Equations

The stability boundaries for a higher-order time-lag kernel are calculated in a similar manner, and they are plotted in Figure 2.7.6. As the order of the kernel is increased, the C and T parameter space corresponding to stable steady-states are cornered further.

In summary, a full spectrum of dynamic behavior, including damped oscillations (when a 0th-order kernel is included) and sustained oscillations (when a 1st-order kernel is included) can be predicted. The inclusion of the time-lag kernel greatly extends the utility of the simple bioreactor model considered in this section to a wide range of oscillatory behaviors, which it cannot predict in its original unmodified form.

The inclusion of kernels in other variables such as Y_s and x can also be analyzed in a manner similar to what was done with time-lags in the specific growth rate. The same approach can also be extended to other state equations that contain more model parameters, such as a maintenance term in the biomass state equation (Herbert's model), a maintenance term in the substrate state equation (Pirt's model), substrate inhibition, variable yield coefficients, cell recycle, or spatial inhomogeneities. It would be interesting to investigate the effect of time-lag on modes of operations other than a CSTR, *e.g.*, plug flow, batch, or fed-batch. In addition, the approach described herein can be carried out for a higher-dimensional system of product formation, mixed substrates, or mixed cultures. Considering the variety of transient behaviors that are possible with the simplest set of chemostat state

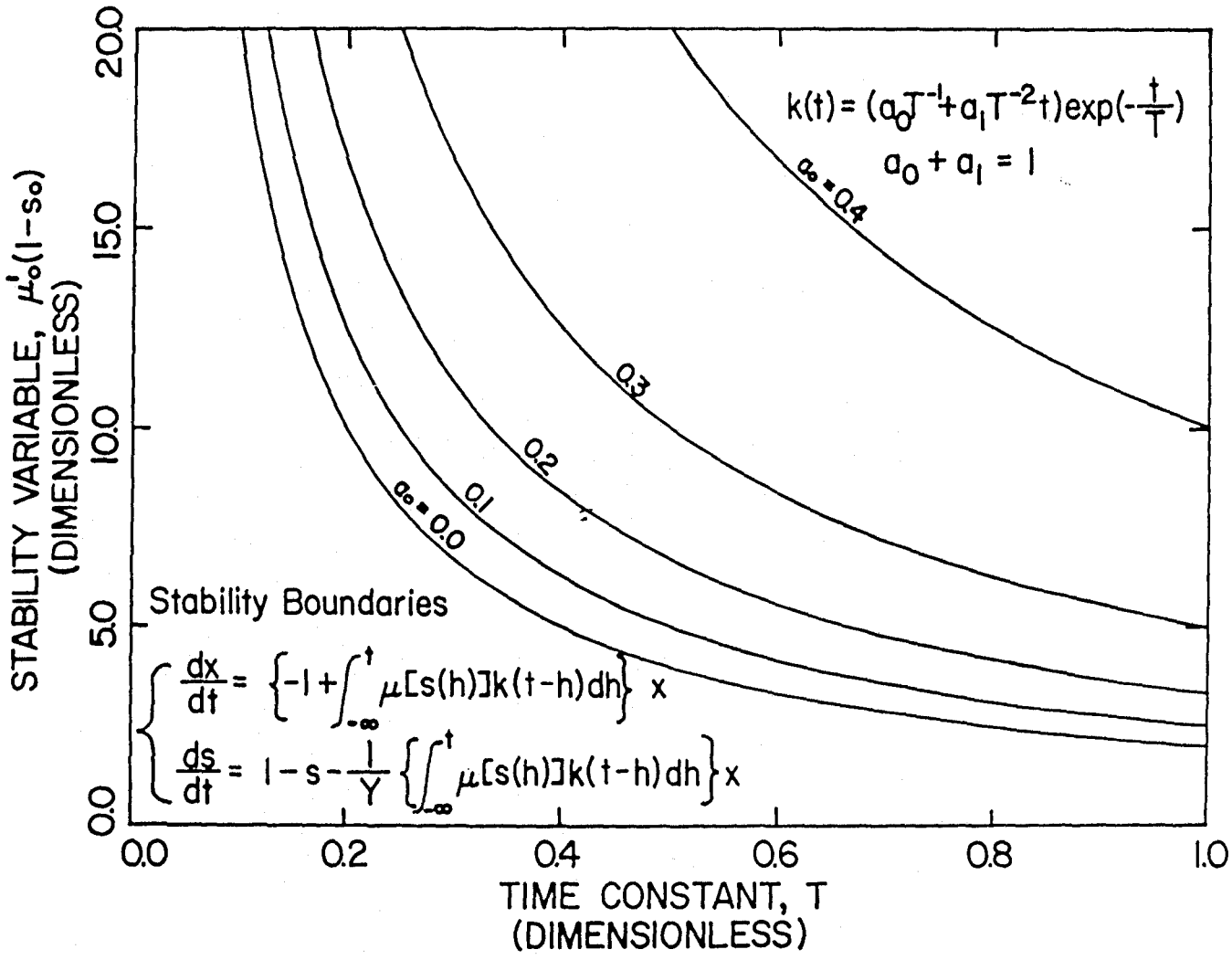


Figure 2.7.5. Stability boundaries for a two-dimensional chemostat with a mixed 0th- and 1st-order kernel. The boundaries exist for $0 \leq a_0 < 0.5$.

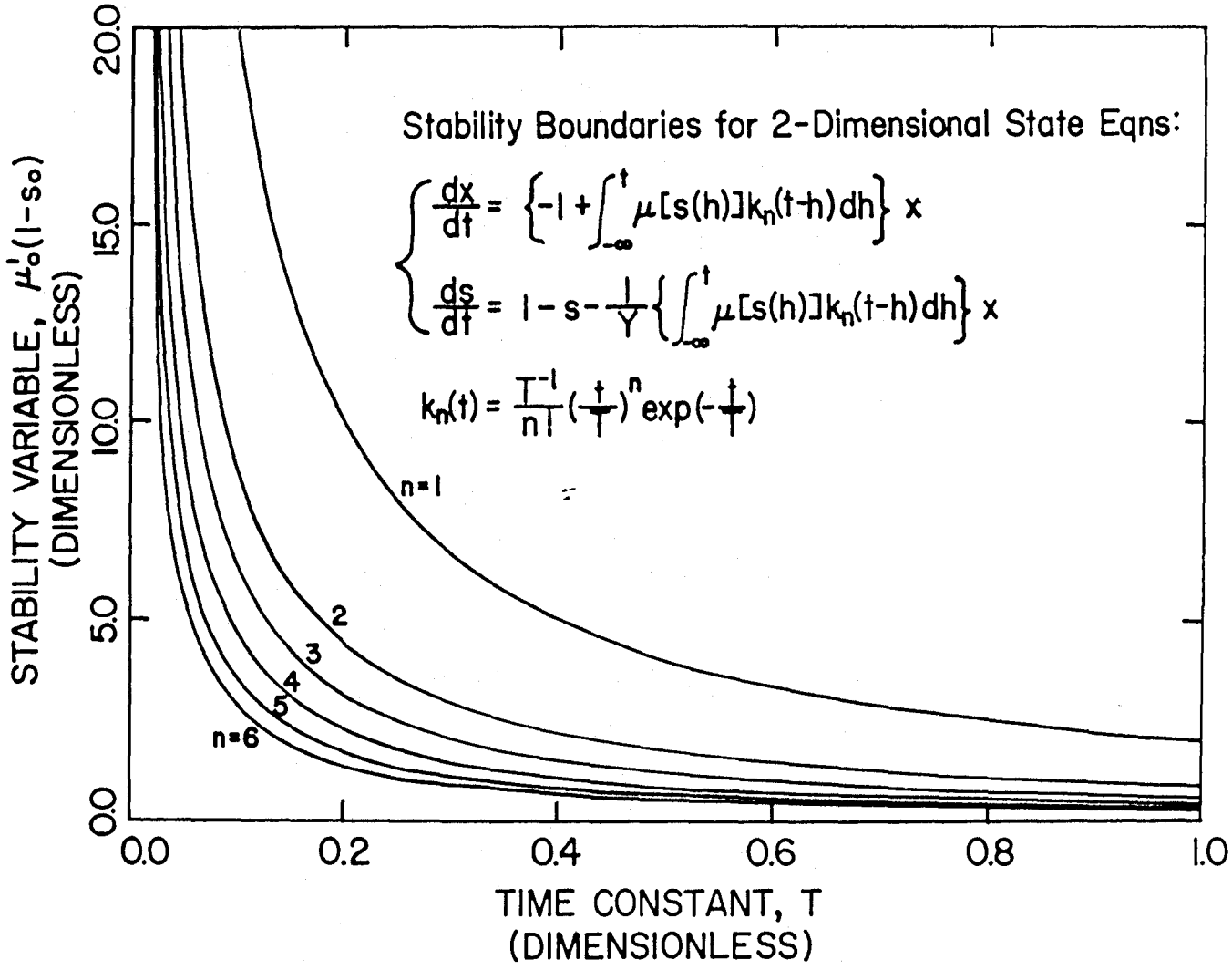


Figure 2.7.6. Stability boundaries for a two-dimensional chemostat with higher-order kernels.

equations, an extremely rich collection of dynamic responses is expected to emerge from applying such an analysis to these slightly more-complicated systems. Furthermore, it is suggested that the effect of time-lag on process control consideration and optimal control formulations be vigorously pursued in future studies.

Comparison of Dynamic Responses

Many models have been proposed in the past to fit experimental data, sometimes with the expressed objectives of explaining the oscillatory behaviors. Many variations of Monod's model exist, but they all give approximately the same response, at least too similar to show any statistically significant differences when they are fitted to experimental data corrupted with errors. A slightly better fit of one model does not at all imply that the model correctly describes the system, and one should be warned that choosing a model based only on data fitting is not a sound approach. If possible, the process should induced to express novel distinguishable dynamic features. Being drastically noticeable, oscillations and hysteresis are powerful tools that can be exploited to discriminate models.

Therefore, it is interesting to compare the dynamic response of a kernel-driven system with that predicted by other models. One of the modifications to the original chemostat system equations is to assume a non-constant substrate-to-biomass yield coefficient. The fact that the yield coefficient sometimes is not constant has long been recognized. The proposed remedies include the addition of a maintenance term in the biomass dynamic equation (Herbert, 1958) or a maintenance term in the substrate dynamic equation (Pirt, 1965). More recently, a yield coefficient that depends linearly on the limiting substrate concentration has been proposed (Essajee and Tanner, 1979). The effect of the yield expression on the stability of the system has been analyzed (Crooke *et al.*, 1980).

A relatively simple chemostat described by the following set of dynamic equations is considered.

$$\frac{dx(t)}{dt} = [-1 + \mu(s)]\, x(t) \tag{2.7.25a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s(s)}\mu(s)x(t) \tag{2.7.25b}$$

These equations are exactly the same as those considered before, with the exception that the yield coefficient is not necessarily a constant. It is found that the eigenvalues (roots of the characteristic equation) for this linearized system are:

$$\lambda = \frac{1}{2}\left[-\Gamma \pm \sqrt{\Gamma^2 - 4\mu_0'(1 - s_0)}\right], \tag{2.7.26}$$

where

$$\Gamma = 1 + \mu_0'(1 - s_0) - \frac{Y_0'}{Y_0}(1 - s_0) \tag{2.7.27}$$

The system becomes oscillatory (damped), when the eigenvalues have imaginary parts, or, equivalently, when the quantity inside the square root is less than zero.

$$\Gamma^2 - 4\mu_0'(1 - s_0) < 0 \tag{2.7.28}$$

The system becomes unstable with sustained oscillations (limit cycles) when the real parts of the eigenvalues become positive. Thus the necessary and sufficient condition for stability is $\lambda < 0$.

$$-\Gamma + \sqrt{\Gamma^2 - 4\mu_0'(1 - s_0)} < 0 \tag{2.7.29}$$

For simplicity, a Monod growth expression is substituted.

$$\mu(s) = \frac{\alpha s}{\beta + s}, \tag{2.7.30}$$

where $\alpha = \frac{\mu_m}{D}$ is the non-dimensionalized maximum specific growth rate and $\beta = \frac{K_s}{s_f}$ is the non-dimensionalized half-saturation constant. The steady-state values for this generalized system containing a variable yield coefficient can be calculated by:

$$s_0 = \frac{\beta}{\alpha - 1} \tag{2.7.31a}$$

$$x_0 = Y_s(s_0)[1 - s_0] \tag{2.7.31b}$$

Tanner's Model: The conditions for the existence of limit cycles are explicitly derived for three commonly encountered yield coefficient expressions. The first expression is linear in the limiting substrate concentration.

$$Y_s(s) = A + B's, \tag{2.7.32}$$

where $B' = Bs_f$ is the non-dimensionalized slope of the Y_s versus s line. The condition for limit cycles is:

$$\frac{A}{Bs_f} < \frac{\alpha\beta(1-\beta) - \beta(1+\beta)}{(\alpha-1)\left[(\alpha-1)^2 + \beta\right]}. \tag{2.7.33}$$

The stability region for this yield expression is shown in Figure 2.7.7, where the parameter space for limit cycles is under the respective curve. For parameters that lie in the region immediately outside these curves, the approach to the steady-state is (damped) oscillatory. For parameters even further away from the dotted oscillatory boundaries, the classical exponential approach is predicted.

The response of the system after a step increase in the dilution rate from 0.05 hr^{-1} to 0.1 hr^{-1} is shown in Figure 2.7.8 for different substrate concentrations in the feed stream. Here, s_f=200 g/l corresponds to the first dot at $\frac{K_s}{s_f} = 0.25$, which is within the unstable region in Figure 2.7.7; s_f=150 g/l corresponds to the second dot at $\frac{K_s}{s_f} = 0.33$, which is on the stability boundary; and s_f=100 g/l corresponds to the third dot at $\frac{K_s}{s_f} = 0.50$, which is outside the stability boundary. As s_f decreases, the parameters move away from the stability boundary, and the system becomes more stable after an upward shift the dilution rate.

Herbert's Model: The second commonly encountered expression for the yield coefficient is inversely linear in the limiting substrate concentration.

$$Y_s(s) = A - \frac{B'}{s}, \tag{2.7.34}$$

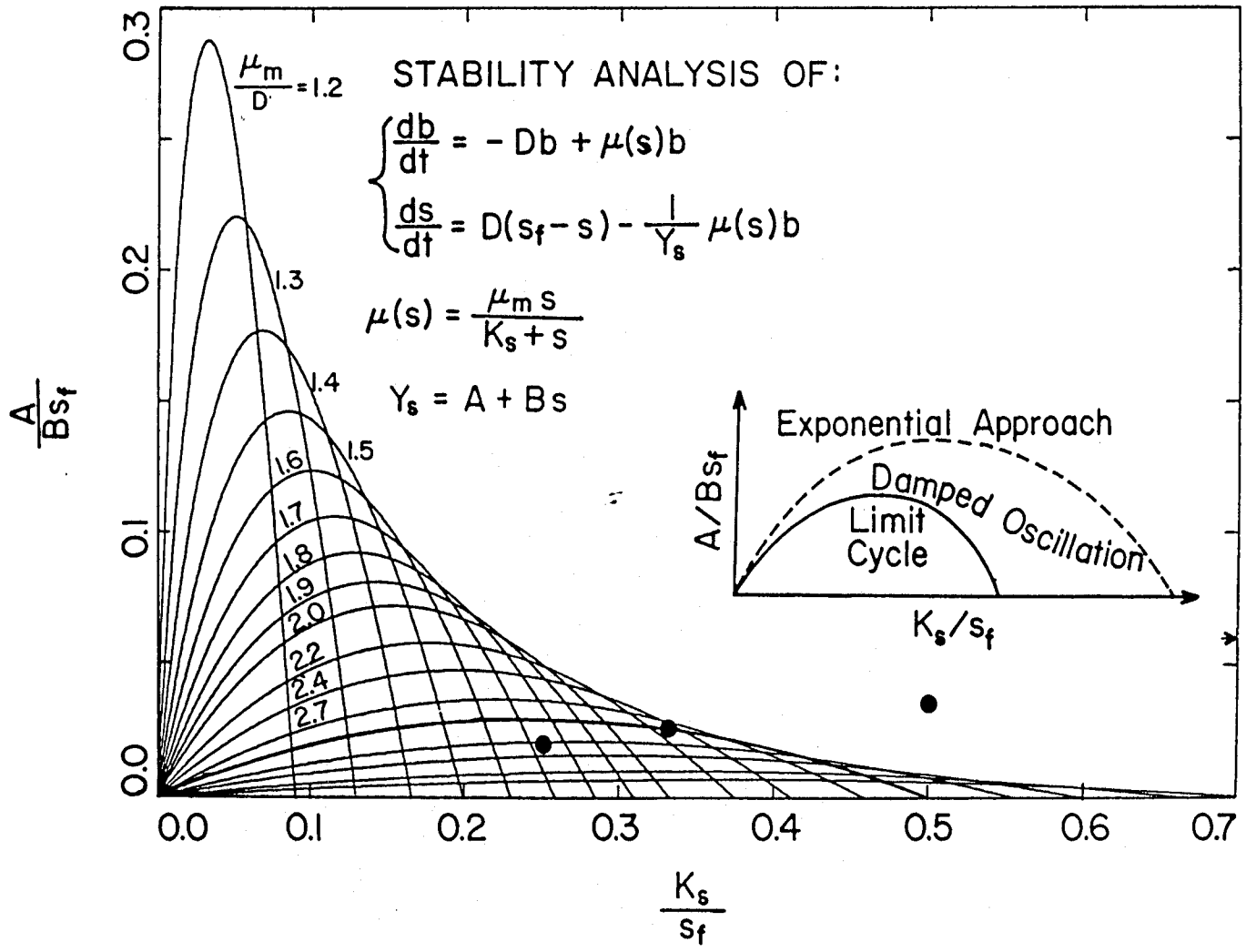


Figure 2.7.7. Stability boundaries for the yield coefficient linear in s:

$$Y_s = A + Bs_f s \quad \text{(non - dimensionalized)}.$$

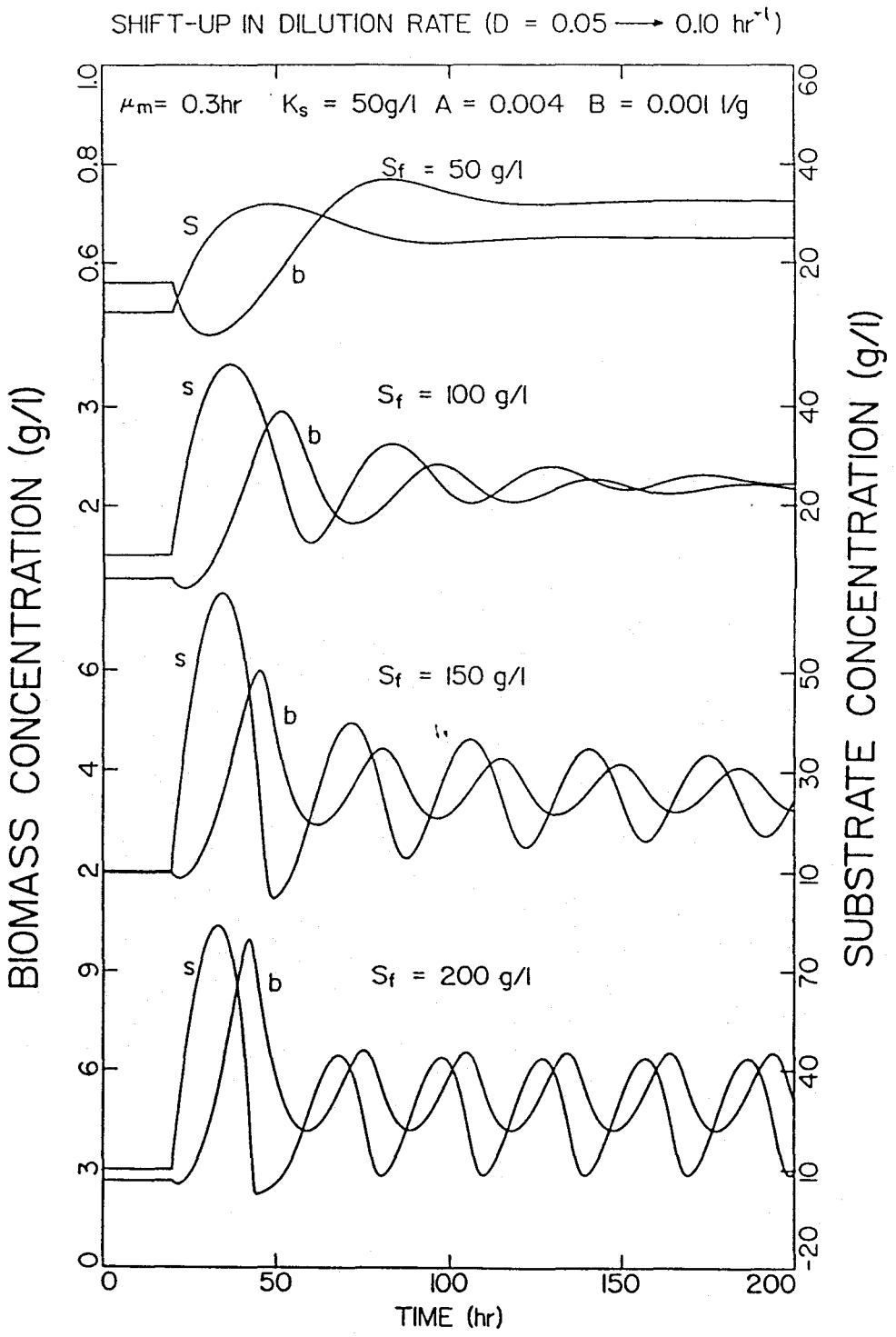


Figure 2.7.8. Simulated dynamic Response of a Monod model modified by

$$Y_s = A + Bs.$$

where $B' = \frac{B}{s_f}$ is the non-dimensionalized negative slope of Y_s versus $\frac{1}{s}$ line. The condition for limit cycles is:

$$\frac{s_f A}{B} < \frac{(\alpha-1)^2(2\alpha-1-\beta)}{\beta\left[(\alpha-1)^2+\beta\right]}. \tag{2.7.35}$$

The stability boundaries for this yield expression are plotted in Figure 2.7.9. The parameter space under the stability boundaries gives rise to limit cycle behaviors.

Pirt's Model: The third commonly used expression for a variable yield coefficient has the same saturation form as the Monod model.

$$Y_s(s) = \frac{A}{B'+s}, \tag{2.7.36}$$

where $B' = \frac{B}{s_f}$ is the non-dimensionalized half-saturation constant of the yield expression. The corresponding condition for limit cycles is:

$$\frac{s_f}{B} < \frac{(\alpha-1)\left[(\alpha(1-\beta)-1-\beta)\right]}{\beta\left[(\alpha-1)^2+\beta\right]}. \tag{2.7.37}$$

The stability boundaries of the above equation are plotted in Figure 2.7.10.

The above three functional forms of $Y_s(s)$ are plotted in Figure 2.7.11 along with the classical constant form. The first linear relationship is very useful because other expressions of $Y_s(s)$ all reduce to this linear form when they are linearized around the steady-state in a localized stability analysis.

The second form of yield coefficient $Y_s(s) = A - \frac{B'}{s}$ is encountered when there is a maintenance term in the biomass dynamic equation. The inclusion of this maintenance term changes the apparent yield coefficient. The modified dynamic equations are:

$$\frac{dx(t)}{dt} = \left[-1+\mu(s)\right]x(t) - mx(t) \tag{2.7.38a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s(s)}\mu(s)x(t), \tag{2.7.38b}$$

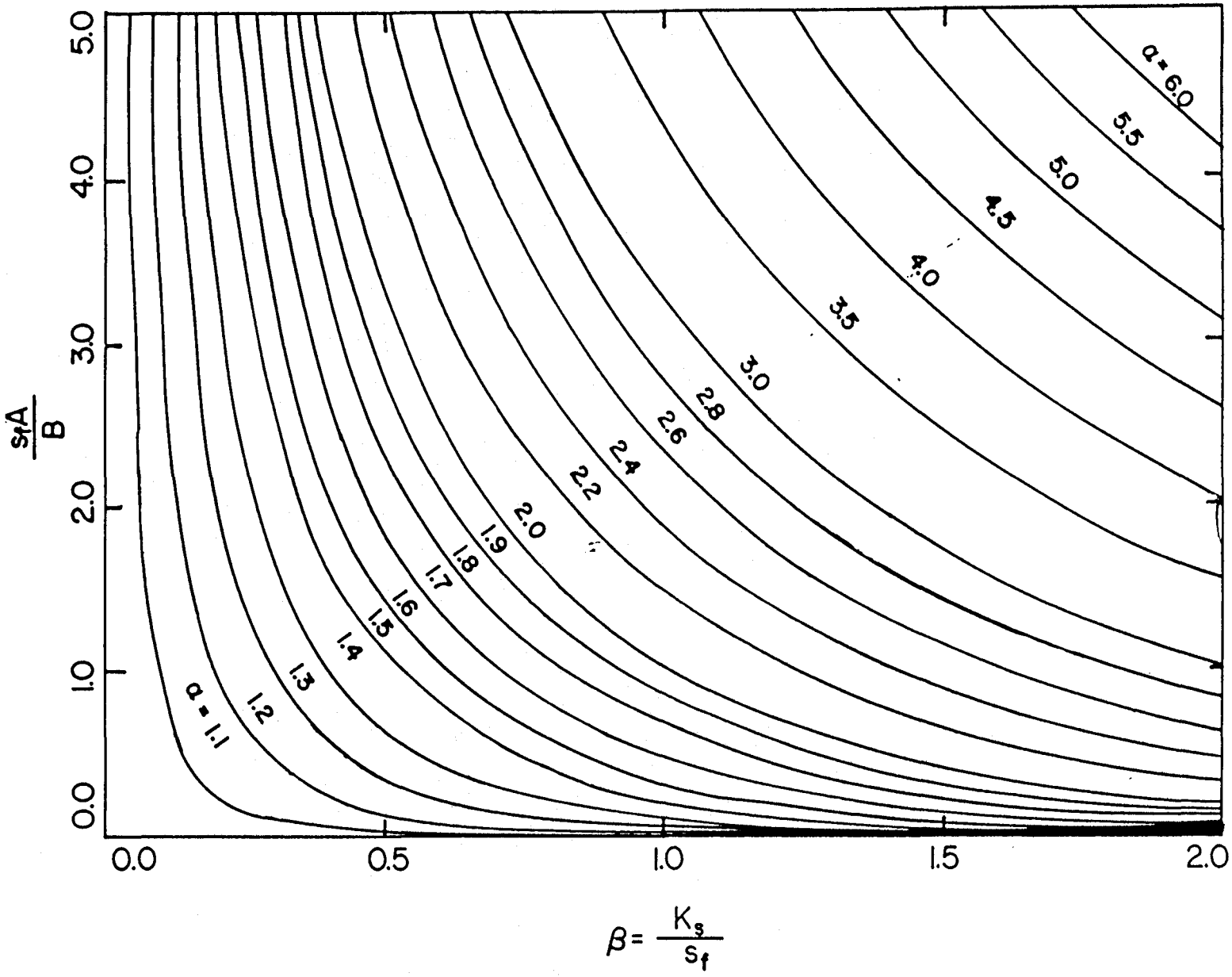


Figure 2.7.9. Stability boundaries for the yield coefficient:

$$Y_s(s) = A - \frac{B}{s_f s} \quad \text{(non - dimensionalized)}.$$

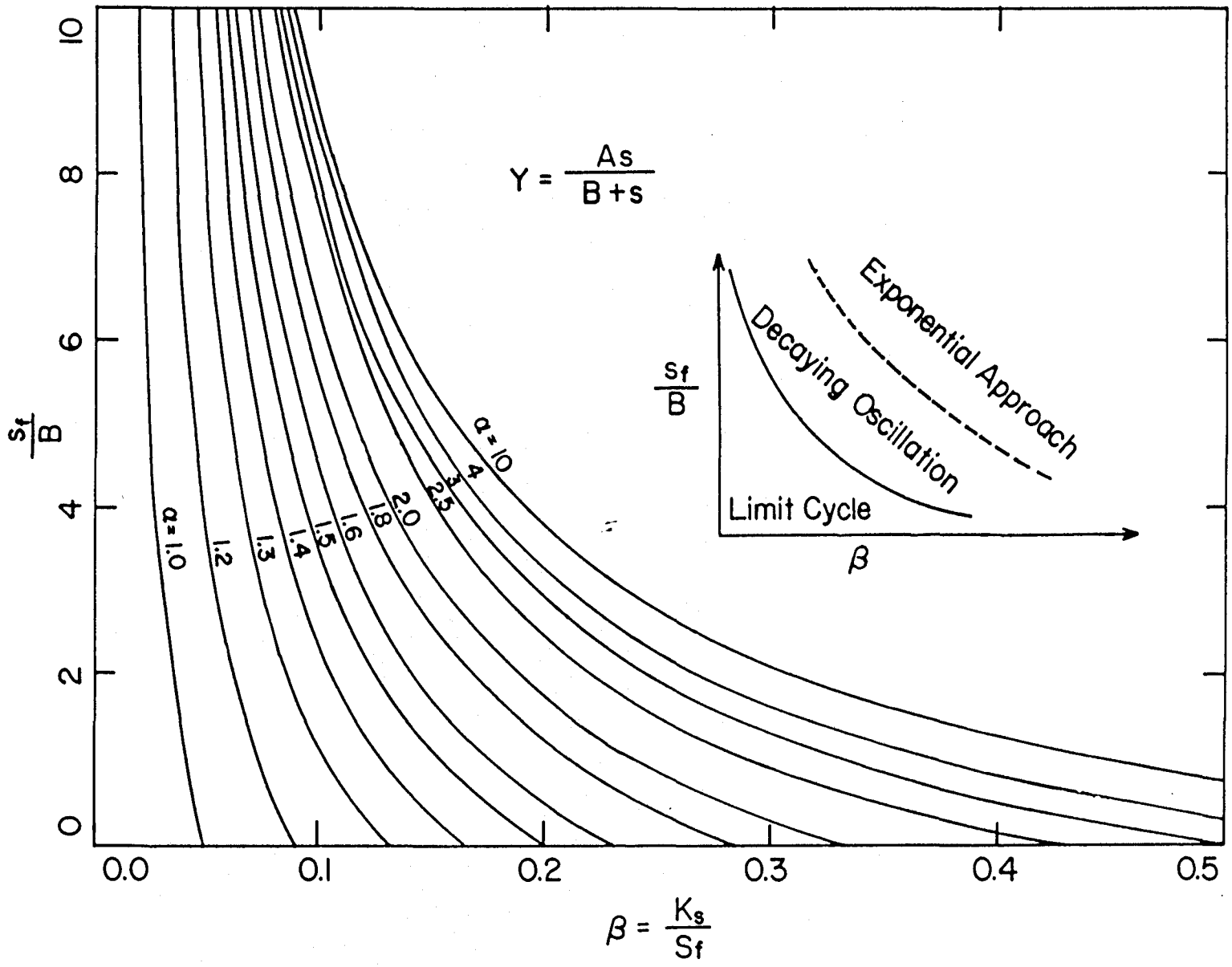


Figure 2.7.10. Stability boundaries for the yield coefficient:

$$Y_s(s) = \frac{A}{\frac{B}{s_f} + s} \quad \text{(non - dimensionalized)}.$$

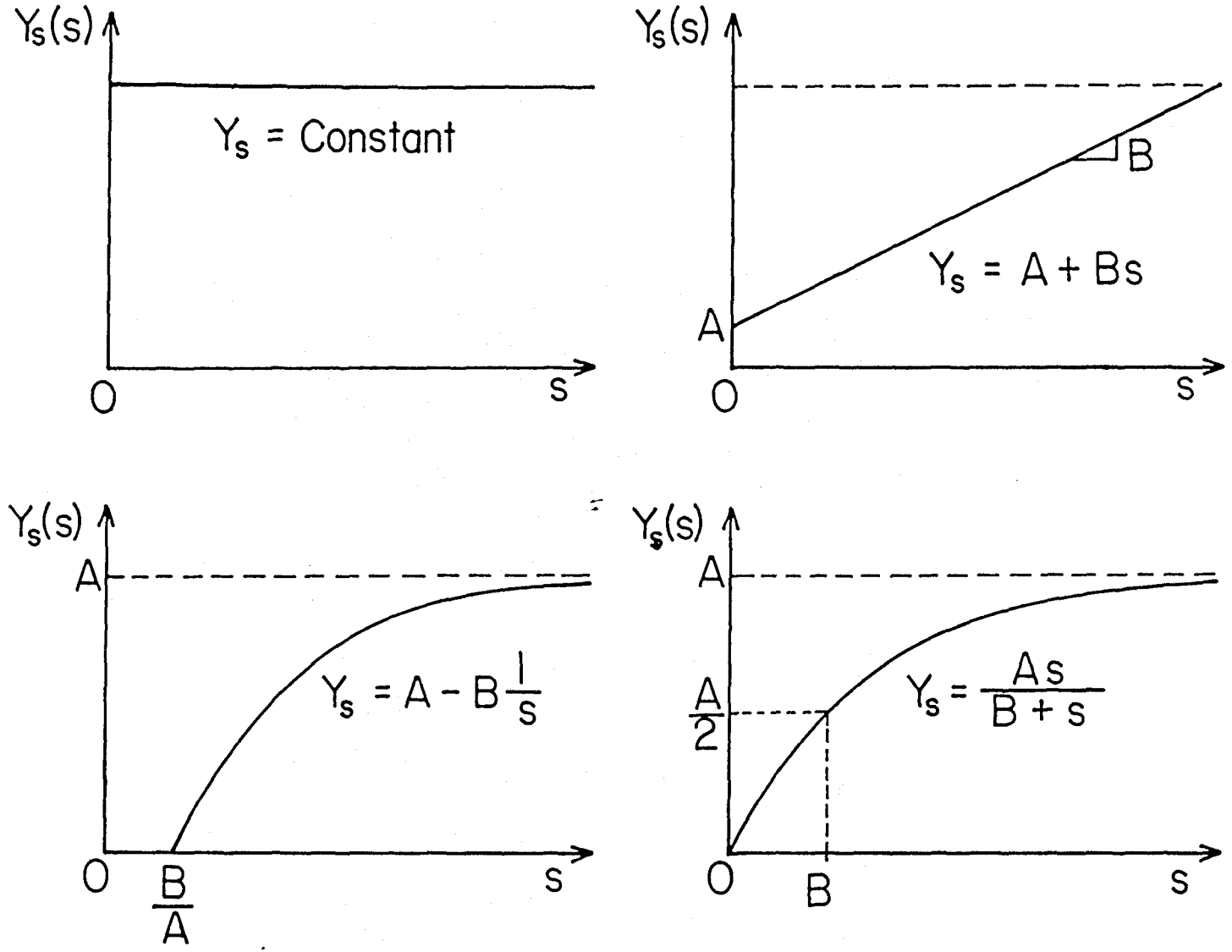


Figure 2.7.11. Four frequently encountered forms of $Y_s(s)$.

where $m = \frac{m'}{D}$ is the non-dimensionalized maintenance coefficient. The observed apparent yield coefficient is a function of the specific growth rate, which is a function of the substrate concentration.

$$Y_s^{\text{app}}(s) = Y_s^{\text{true}}\left(1 - \frac{m}{\mu(s)}\right) \overset{\text{Monod}}{\Longrightarrow} \overbrace{Y_s^{\text{true}}\left(1 - \frac{m}{\alpha}\right)}^{A} - \overbrace{\frac{Y_s^{\text{true}} m\beta}{\alpha}}^{B}\frac{1}{s} \qquad (2.7.39)$$
$$= A - \frac{B}{s}$$

The condition for damped oscillations is:

$$\frac{(\alpha - 1 - m)^2}{(1+m)\left[\alpha D_+ + (\alpha - 1 - m)\right]} < \beta < \frac{(\alpha - 1 - m)^2}{(1+m)\left[\alpha D_- - (\alpha - 1 - m)\right]}, \qquad (2.7.40)$$

where

$$D_\pm = (1 + 2m) \pm 2\sqrt{m(1+m)}. \qquad (2.7.41)$$

These boundaries are plotted in Figure 2.7.12. The condition for limit cycles is:

$$\Gamma = 1 + \frac{(\alpha - 1 - m)(\alpha - 1 - m - \beta - \beta m)}{\alpha\beta(1+m)} < 0, \qquad (2.7.42)$$

which is never true; therefore, limit cycle behaviors cannot be predicted by the addition of a maintenance term in the biomass dynamic equation.

The third form of yield coefficient $Y_s(s) = \frac{A}{B'+s}$ arises when a maintenance term is added to the substrate dynamic equation.

$$\frac{dx(t)}{dt} = \left[-1 + \mu(s)\right] x(t) \qquad (2.7.43a)$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s(s)}\mu(s)x(t) - mx(t), \qquad (2.7.43b)$$

where $m = \frac{m'}{D}$ is the non-dimensionalized maintenance coefficient. The presence of this term changes the apparent yield coefficient.

$$Y_s^{\text{app}}(s) = \frac{1}{\frac{1}{Y_s^{\text{true}}} + \frac{m}{\mu(s)}} \overset{\text{Monod}}{\Longrightarrow} \frac{\overbrace{\frac{\alpha}{\frac{\alpha}{Y_s^{\text{true}}} + m}}^{A} s}{\underbrace{\frac{m\beta}{\frac{\alpha}{Y_s^{\text{true}}} + m}}_{B} + s} \qquad (2.7.44)$$
$$= \frac{As}{B+s}$$

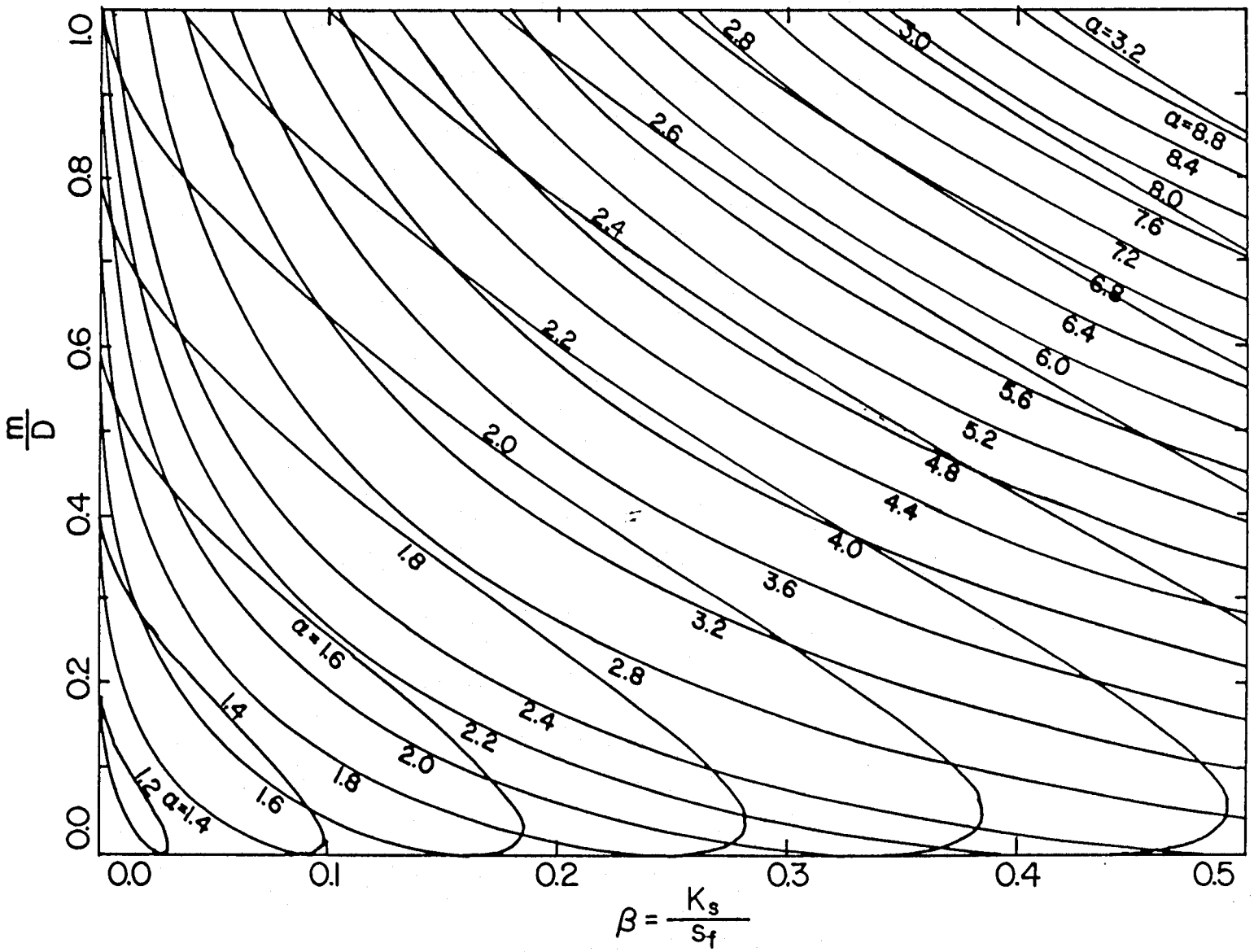


Figure 2.7.12. Damped oscillation boundaries when a maintenance term is included in the biomass dynamic equation (Herbert's model of maintenance).

Thus, the apparent Y_s always has the same form as $\mu(s)$. The condition for damped oscillations is:

$$\frac{1}{Y_s^{\text{true}}m} < \left[2\sqrt{\frac{(\alpha-1)(\alpha-1-\beta)}{\alpha\beta}} - 1\right] \frac{\alpha\beta}{(\alpha-1)^2+\beta}. \tag{2.7.45}$$

The parameter $\frac{1}{Y_s^{\text{true}}m}$ is basically the ratio of the amount of substrate utilized for growth to the amount of substrate used for maintenance. The oscillation boundaries are shown in Figure 2.7.12. Similarly, the condition for limit cycles is:

$$\frac{1}{Y_s^{\text{true}}m} < -\frac{\alpha\beta}{(\alpha-1)^2+\beta} < 0, \tag{2.7.46}$$

which again is never true. Typical responses of this system after a shift-up in the dilution rate from 0.05 hr^{-1} to 0.1 hr^{-1} is shown in figure 2.7.14.

A Literature Example

Finally, the dynamic response of a time-lag model with a mixed first-order kernel $k(t) = 0.2k_0(t) + 0.8k_1(t)$ is simulated in Figure 2.7.15. In these dynamic simulations, the parameter values of are chosen to be close to those observed by Lee, Tribe, and Rogers (1979), one of which is presented in Figure 2.1.8. (Rounded numbers are chosen for the parameters.) Rigorous data fitting was not attempted, because the objective here is to compare the main features of their experimental data and the predictions from various models. A range of models were simulated to see which ones could display the type of oscillations that are in agreement with the literature data.

Note that the various modifications of the yield coefficient to the basic model presented here are all capable of yielding approximately the correct magnitudes for both the biomass and substrate concentrations. In addition, all these models correctly predict a higher tendency for the system to become unstable when the substrate concentration in the feed stream is increased. However, neither of the

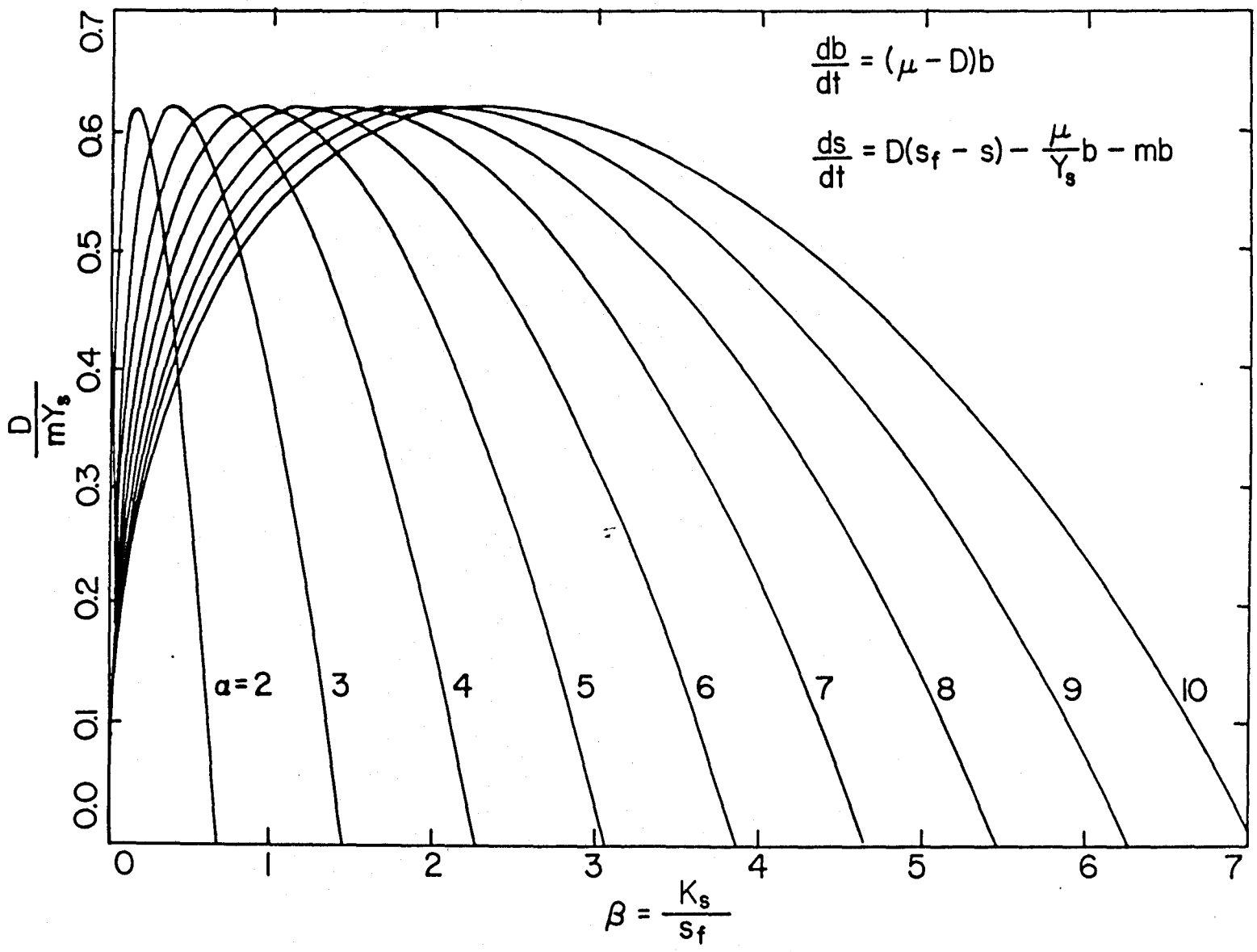


Figure 2.7.13. Damped oscillation boundaries when a maintenance term is included in the substrate dynamic equation (Pirt's model of maintenance).

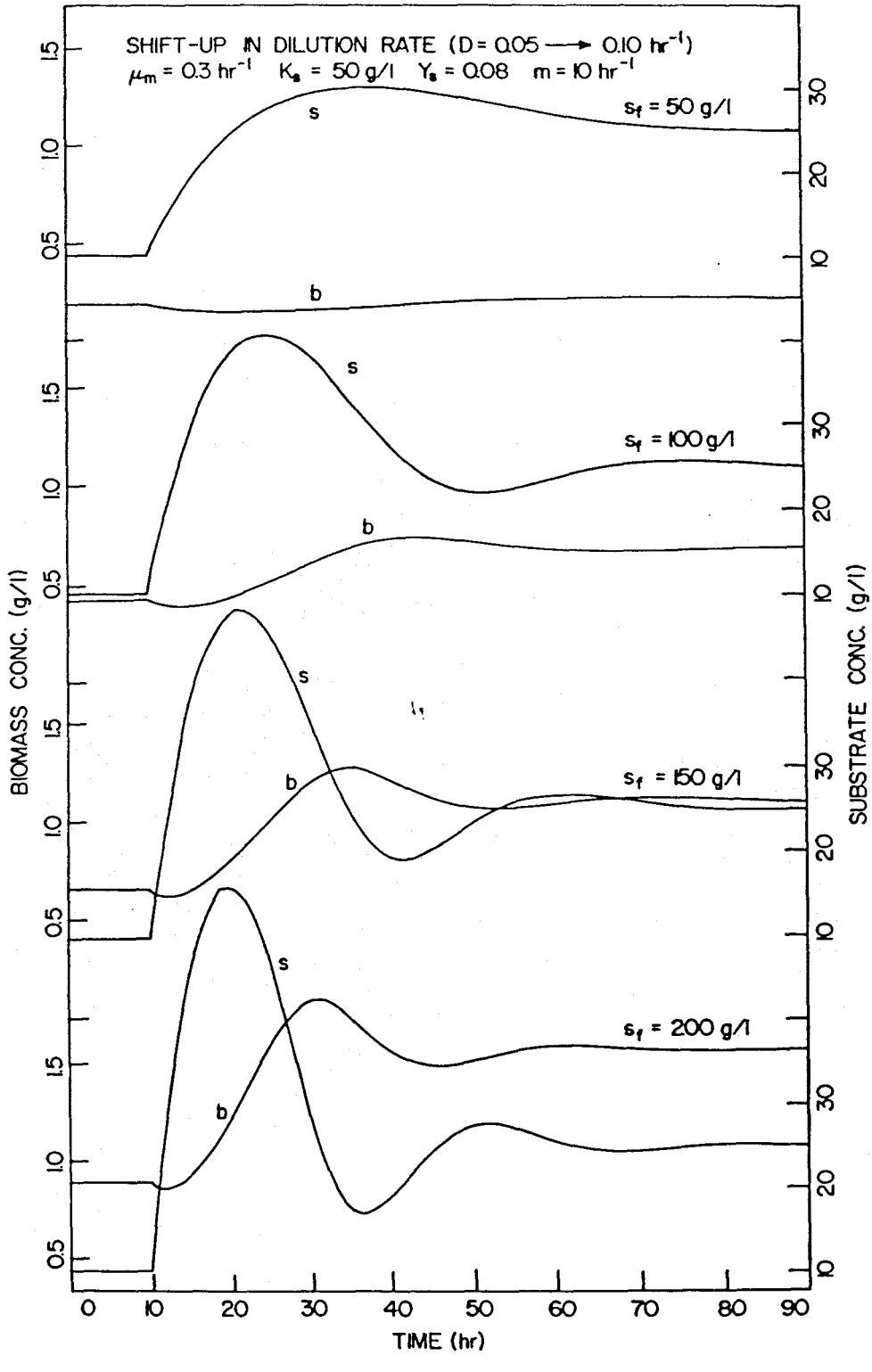


Figure 2.7.14. Simulated dynamic response of a Monod model modified by the inclusion of a maintenance term in the substrate dynamic equation.

maintenance models can exhibit the sustained oscillations that are reported by Lee *et al.*; thus, they are ruled out at this stage. Figure 2.7.8, simulating a linear yield coefficient, and Figure 2.7.15, simulating the inclusion of a time-lag kernel, appear to be very similar at first glance. Both predict approximately the correct period of about 40 hours, in addition to the amplitudes of oscillations for the biomass concentration (*e.g.*, 1~5 g/l at s_f=200g/l) and the substrate concentration (*e.g.*, 10~50g/l at s_f=200g/l). Actually, the linear yield coefficient model in the figure predicts a variation of the biomass concentration between approximately 3~4 g/l; whereas, the time-lag model simulation shows a variation of approximately 3~6 g/l. Nevertheless, these discrepancies can be largely eliminated by carefully choosing a better-fitting yield coefficient. However, at closer inspection, the phase difference between the oscillations in the biomass concentration and the substrate concentration for the linear yield coefficient is approximately 90 degrees, with the biomass concentration behind the substrate concentration; whereas, that for the time-lag kernel is approximately 180 degrees. Since the data from Lee *et al.* shows a phase difference of 180 degrees, one can conclude that the time-lag model is the superior one.

The significance of the illustration presented here is the demonstration that a time-lag model can withstand the more-critical test of oscillations where others have failed. If desired, the parameters can be refined to achieve a better fit. Furthermore, the product concentration is not considered because product formation can be described as a linear combination of a growth-related term and a nongrowth-related term. Superior fit of this third variable, including phase shift and amplitude, can be expected by appropriately assigning different weights to these two separate terms.

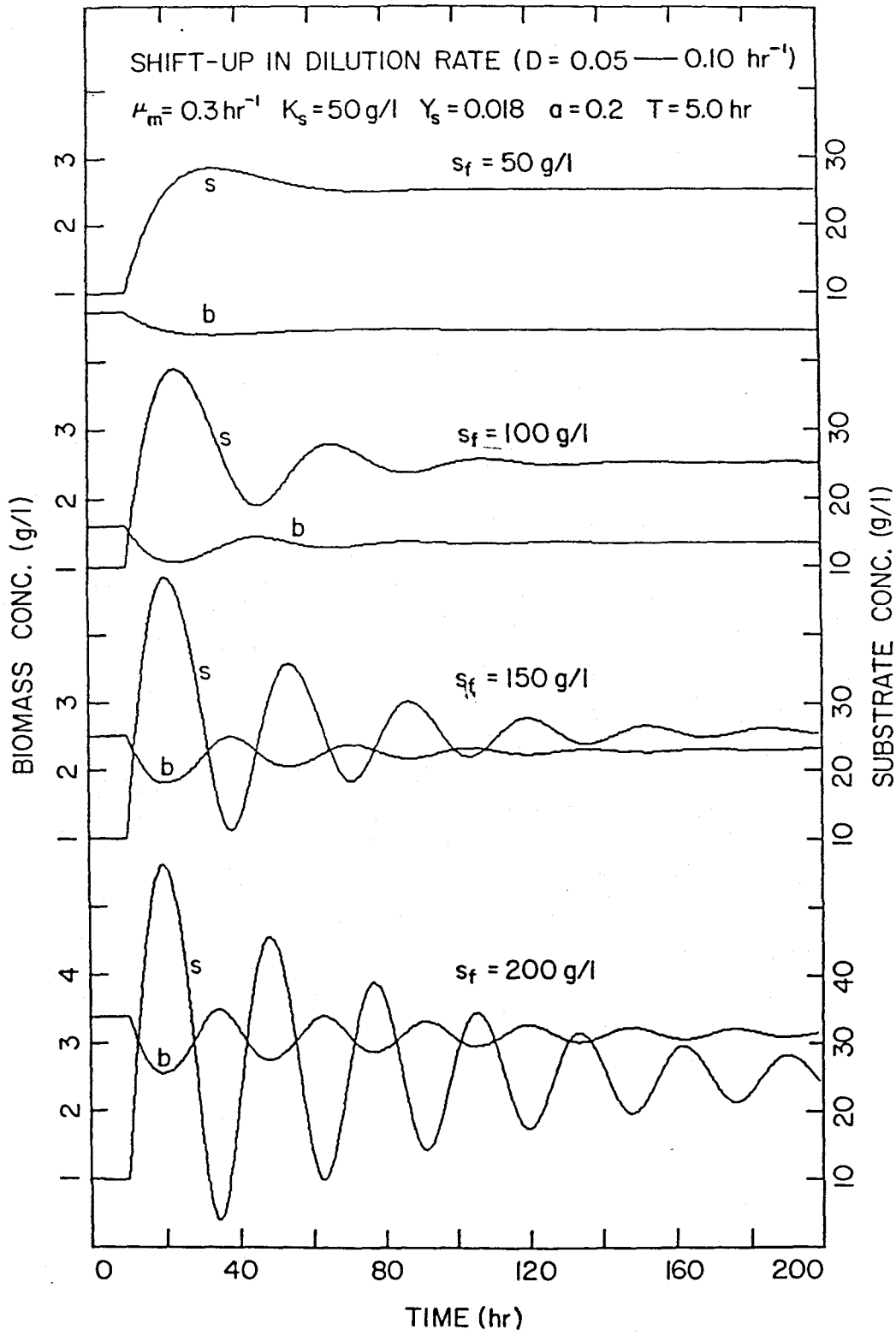


Figure 2.7.15. Simulated dynamic response of a time-lag model with a mixed first-order kernel $k(t) = 0.2k_0(t) + 0.8k_1(t)$.

2.8 FREQUENCY RESPONSE

One of the powerful tools of modern systems analysis is based on the frequency response from a forced sinusoidal perturbation. This is especially true for a dynamic system with time-lags.

Time-Lag Kernel Transfer Function

The characteristic equation of a time-lag system is discussed previously in the derivation of the kernel function in the section "Solutions to General Time-Lag Equations." The choice of the expression for the kernel base function is based on the desire that it should satisfy the following differential equation.

$$\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i k_n(t)}{dt^i} = 0 \qquad (2.8.1)$$

The elegance of this expression can be much appreciated in formulating the transfer function of the time-lag kernel. This can be easily accomplished by tracing backward the steps that have been taken to obtain the expression for the kernel. Basically, the reciprocal of the kernel's characteristic equation is equivalent to its transfer function. For the above nth-order kernel function, the Laplace transform is simply:

$$\left[\sum_{i=0}^{n+1} \binom{n+1}{i} T^i s^i\right] K(s) \;=\; D(s), \qquad (2.8.2)$$

where s is, unfortunately, the Laplace variable, not the substrate concentration. The use of s as the Laplace variable is well established, and this commonly accepted convention is followed in this section. Little confusion should arise since the distinction of these two terms can be quite clearly inferred from the usage. Following this notation, $K(s)$ and $D(s)$ in the above equation are the Laplace transform of $k(t)$ and $\delta(t)$ (Dirac delta function), respectively.

$$K(s) = \mathcal{L}\{k(t)\} \;=\; \int_0^\infty e^{-st} k(t)dt \;=\; \frac{1}{(1+Ts)^{n+1}} \qquad (2.8.3a)$$

$$D(s) = \mathcal{L}\{\delta(t)\} = \int_0^\infty e^{-st}\delta(t)dt = 1 \qquad (2.8.3b)$$

Thus, the transfer function of the kernel is simply:

$$\text{Impulse} \xrightarrow{D(s)} \boxed{G_k(s)} \xrightarrow{K(s)} \text{Kernel}$$

$$G_k(s) = \frac{K(s)}{D(s)} = K(s) = \frac{1}{(1+Ts)^{n+1}}, \qquad (2.8.4)$$

which is, of course, the same as that for a system of $n+1$ first-order dynamic blocks in series.

$$G_k(s) = \left[K_0(s)\right]^{n+1} \qquad (2.8.5)$$

where $K_0 = \frac{1}{(1+Ts)}$ is the transfer function for one single decoupled first-order system. From the above equation, it is clear that the kernel function is implicitly the response of the system to an impulse input. The claim that $k(t)$ is the impulse response function of the system is thus proved.

Because $k(t)$ is chosen to satisfy Equation (2.8.1), this in turn forces the variable that directly contains the time-lag, *e.g.*, $y_n(t) \equiv \int_{-\infty}^{t} \mu(h)k_n(t-h)dh$, to conform ideally to the following equation.

$$\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y_n(t)}{dt^i} = \mu(t) \qquad (2.8.6)$$

Applying the Laplace transform operator to both sides of the equation, one obtains:

$$\left[\sum_{i=0}^{n+1} \binom{n+1}{i} T^i s^i\right] Y(s) = M(s), \qquad (2.8.7)$$

where $Y(s)$ and $M(s)$ are the Laplace transforms of the functions $y(t)$ and $\mu(t)$, respectively.

$$Y(s) = \mathcal{L}\{y(t)\} = \int_0^\infty e^{-st}y(t)dt \qquad (2.8.8a)$$

$$M(s) = \mathcal{L}\{\mu(t)\} = \int_0^\infty e^{-st}\mu(t)dt \qquad (2.8.8b)$$

As before, the transfer function for an arbitrary input is

$$\text{System Input} \xrightarrow{M(s)} \boxed{G_y(s)} \xrightarrow{Y(s)} \text{System Output}$$

$$G_y(s) = \frac{Y(s)}{M(s)} = \frac{1}{\sum_{i=0}^{n+1} \binom{n+1}{i} T^i s^i} = \frac{1}{(1+Ts)^{n+1}}, \tag{2.8.9}$$

which is identical to the $G_k(t)$ just derived, as it should be, because the homogeneous part of the time-lag differential equations satisfied by $k(t)$ and $y(t)$ are the same. The meaning of the kernel is now ever clearer. The kernel describes how the system (*e.g.*, the observed specific growth rate) responds to a disturbance in the input (*e.g.*, the intrinsic growth rate that is directly related to the limiting substrate concentration.) Since the theories of frequency response and stability analysis are quite well developed for such a fundamental system, no further detailed discussion is warranted here.

A similar approach can be taken to obtain the transfer function for a generalized nth-order kernel that is a linear combination of the $n+1$ base functions.

$$k(t) = a_0 k_0(t) + a_1 k_1(t) + a_2 k_2(t) + \ldots + a_{n-1} k_{n-1}(t) + a_n k_n(t) \tag{2.8.10}$$

It has been proved that the complementary equation that must be satisfied by the time-lag variable $y(t)$ is:

$$\begin{aligned} \sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i} &= \sum_{i=0}^{n} \underbrace{\left[\sum_{j=i}^{n} \binom{j}{i} a_{n-j} \right]}_{\alpha_i} T^i \frac{d^i \mu(t)}{dt^i} \\ &= \sum_{i=0}^{n} \alpha_i T^i \frac{d^i \mu(t)}{dt^i}. \end{aligned} \tag{2.8.11}$$

The transfer function for this generalized system is:

$$\text{Intrinsic Sp. Growth Rate} \xrightarrow{M(s)} \boxed{G_y(s)} \xrightarrow{Y(s)} \text{Observed Sp. Growth Rate}$$

$$G_y(s) = \frac{Y(s)}{M(s)} = \frac{\sum_{i=0}^{n} \alpha_i T^i s^i}{\sum_{i=0}^{n+1} \binom{n+1}{i} T^i s^i} = \frac{\sum_{i=0}^{n} \alpha_i T^i s^i}{(1+Ts)^{n+1}}, \tag{2.8.12}$$

which can be further decomposed by a partial fraction expansion:

$$G_y(s) = \frac{a_n}{(1+Ts)^{n+1}} + \frac{a_{n-1}}{(1+Ts)^n} + \ldots + \frac{a_1}{(1+Ts)^2} + \frac{a_0}{(1+Ts)} = \sum_{i=0}^{n} \frac{a_i}{(1+Ts)^{i+1}}. \tag{2.8.13}$$

Such a system is equivalent to dividing the overall system disturbance $M(s)$ into smaller fractions and introducing each fraction at different points along the $n+1$ first-order system dynamic blocks shown in Figure 2.8.1. Note that the a_0 component passes through only one block; whereas, the a_n component passes through $n+1$ blocks in series.

Time-Lag System Transfer Functions

Our problem is more interesting. Note that the above transfer functions are formulated with the intrinsic specific growth rate $\mu(t)$ as the forcing input and the observed specific growth rate $y(t)$ as the output. Since a set of *nonlinear* state equations are under consideration, the various transfer functions for such a system cannot be easily obtained without first linearizing the state equations around the point of interest that is usually composed of the steady-state values.

In general, a system of differential equations containing a set of time-lag variables can be written as:

$$\frac{d\mathbf{x}_1(t)}{dt} = \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t) \tag{2.8.14a}$$

$$\frac{d\mathbf{x}_2(t)}{dt} = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t), \tag{2.8.14b}$$

where $\mathbf{x}_1(t)$ is the $n_1 \times 1$ vector of the original state variables, $\mathbf{x}_2(t)$ is the $n_2 \times 1$ vector of time-lag variables, and $\mathbf{u}(t)$ is the $m \times 1$ vector of control variables. The dynamics of the state and time-lag variables are described by the $n_1 \times 1$ and $n_2 \times 1$ vector functions $\mathbf{f}_1$ and $\mathbf{f}_2$, respectively. In our chemostat example, $\mathbf{x}_1(t)$ consists of the cell-biomass concentration $x(t)$ and the limiting substrate concentration $s(t)$. $\mathbf{x}_2(t)$ is the observed specific growth rate $y(t)$, the higher time derivatives of the

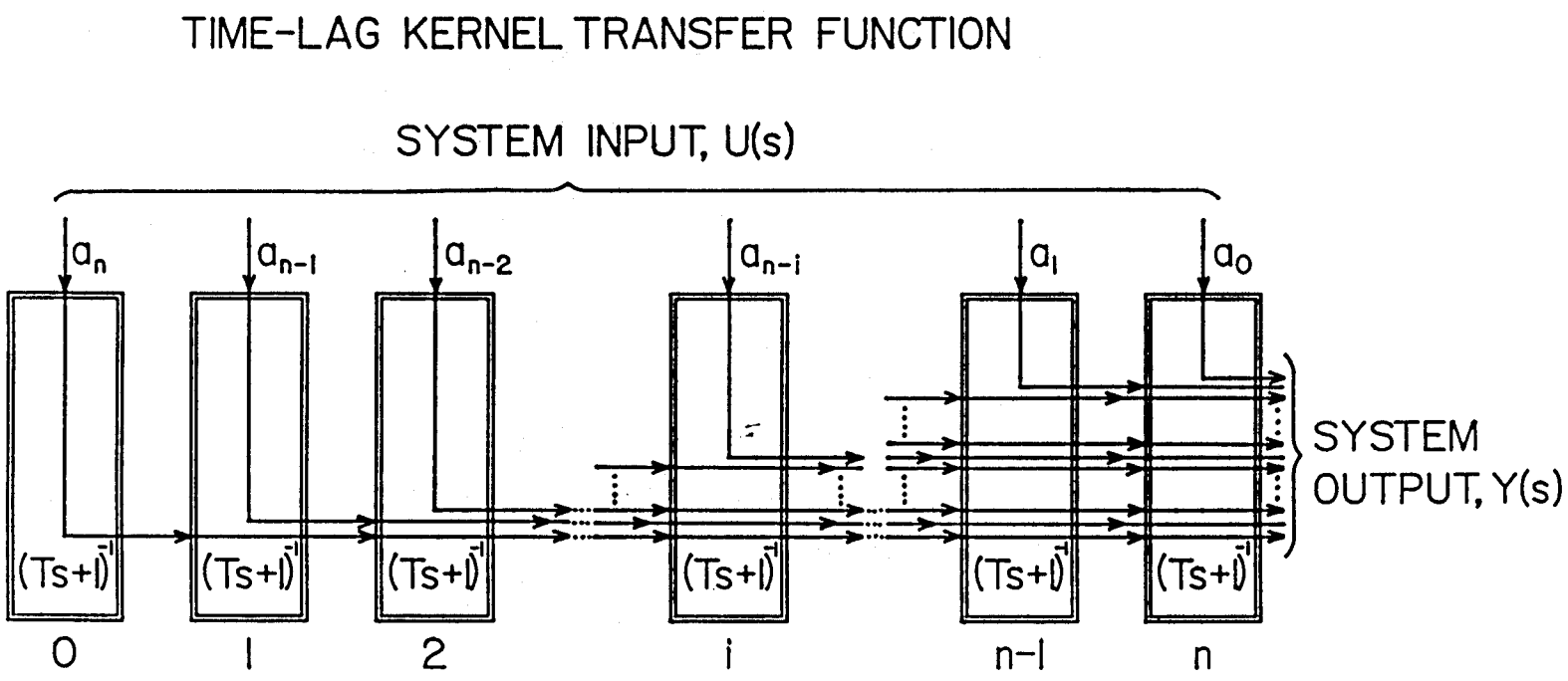


Figure 2.8.1. The overall kernel transfer function for multiple disturbance entries along n first-order system dynamic blocks.

observed specific growth rate $z(t) = \frac{dy(t)}{dt}$, and possibly the substrate to cell yield coefficient $Y_s(t)$, whose dependence on other state variables and time-lag though will not be extensively considered here. $\mathbf{u}(t)$ consists of the dilution rate $D(t)$ and/or the feed substrate concentration $s_f(t)$ in our chemostat example.

There are two related but distinct ways to obtain the transfer function for such a generalized system of dynamic equations. In the first method, the entire combined equation of dimension $n_t \times 1$, where $n_t = n_1 + n_2$,

$$\begin{bmatrix} \frac{d\mathbf{x}_1(t)}{dt} \\ \frac{d\mathbf{x}_2(t)}{dt} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t) \\ \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}, t) \end{bmatrix} \tag{2.8.14c}$$

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{2.8.14d}$$

may be linearized to yield:

$$\frac{d\mathbf{X}(t)}{dt} = \mathbf{A}\mathbf{X}(t) + \mathbf{B}\mathbf{U}(t). \tag{2.8.15}$$

Linear transformation has been performed by shifting the axes to coincide with the steady-state values: $[x_i(t) - x_{i0}] \Rightarrow X(t)$. In the above equation, $\mathbf{A} = \mathbf{f}_\mathbf{x}$ is the $n_t \times n_t$ square Jacobian matrix of the combined function $\mathbf{f}$. Similarly, $\mathbf{B} = \mathbf{f}_\mathbf{u}$ is the $n_t \times m$ control matrix. The following respective standard definitions are used consistently for the differentiation of a scalar L with respect to an m-dimensional column vector $\mathbf{v}$ and the differentiation of an n-dimensional column vector $\mathbf{g}$ with respect to another m-dimensional vector $\mathbf{v}$.

$$L_\mathbf{v}(\mathbf{v}) = \frac{dL(\mathbf{v})}{d\mathbf{v}} = \begin{bmatrix} \frac{\partial L}{\partial v_1} & \frac{\partial L}{\partial v_2} & \cdots & \frac{\partial L}{\partial v_m} \end{bmatrix} \quad \ldots \text{ Row Gradient Vector} \tag{2.8.16a}$$

$$\mathbf{g}_\mathbf{v} = \frac{d\mathbf{g}(\mathbf{v})}{d\mathbf{v}} = \begin{bmatrix} \frac{\partial g_1}{\partial v_1} & \frac{\partial g_1}{\partial v_2} & \cdots & \frac{\partial g_1}{\partial v_m} \\ \frac{\partial g_1}{\partial v_1} & \frac{\partial g_1}{\partial v_2} & \cdots & \frac{\partial g_1}{\partial v_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial v_1} & \frac{\partial g_1}{\partial v_2} & \cdots & \frac{\partial g_1}{\partial v_m} \end{bmatrix} \quad \ldots \text{ Jacobian Matrix} \tag{2.8.16b}$$

The fundamental matrix $\mathbf{A}$ in the above equation can be further diagonalized, or at least converted into a Jordan normal form, via the classical method of eigenvalue and eigenvector decomposition. This transformation yields:

$$\frac{d\tilde{\mathbf{X}}(t)}{dt} = \tilde{\mathbf{A}}\tilde{\mathbf{X}}(t) + \tilde{\mathbf{B}}\mathbf{U}(t), \tag{2.8.17}$$

where the tilde sign above each variable signifies that the quantities have been transformed by a nonsingular $n_t \times n_t$ similarity transformation matrix $\mathbf{T}$ composed of eigenvectors, or generalized eigenvectors in the case where only Jordan normal form can be achieved.

$$\tilde{\mathbf{X}}(t) = \mathbf{T}^{-1}\mathbf{X}(t) \tag{2.8.18a}$$

$$\tilde{\mathbf{B}} = \mathbf{T}^{-1}\mathbf{B} \tag{2.8.18b}$$

$$\tilde{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \tag{2.8.18c}$$

This diagonalized equation can be manipulated via Laplace transform to give:

$$(s\mathbf{I} - \tilde{\mathbf{A}})\tilde{\mathbf{X}}(s) = \tilde{\mathbf{B}}\mathbf{U}(s). \tag{2.8.19}$$

Or, equivalently, the effect of the system input $\mathbf{U}(s)$ on the transformed system output $\tilde{\mathbf{X}}(s)$ can be expressed compactly in an $n_t \times m$ transfer function matrix as:

$$\tilde{\mathbf{G}}(s) = \tilde{\mathbf{X}}(s)\mathbf{U}^{\mathrm{T}}(s)\left(\mathbf{U}(s)\mathbf{U}^{\mathrm{T}}(s)\right)^{-1} = \underbrace{(s\mathbf{I} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}}}_{\text{Transfer Function}} . \tag{2.8.20}$$

For a diagonalized matrix $\tilde{\mathbf{A}}$, the quantities to be inverted in the above equation are scalars rather than a full matrix. The separation of the original set of state variables into an equivalent set of *modes* makes the mathematical operation simple and elegant, because they are dynamically independent of each other. Standard methods of modal analysis can be applied to the modal dynamic equation of (2.8.17) to identify the dominant time constants or to predict the system stability. Although the decoupling afforded by the $\mathbf{T}$-transformed quantities is convenient to work with

mathematically, is is sometimes too abstract for the average person. Thus, the transformed equations are commonly inverse transformed by pre-multiplying with $\mathbf{T}$ and post-multiplying with $\mathbf{T}^{-1}$ once more to regain the original set of variables. In the original non-transformed notation, the above $n_t \times m$ transfer function matrix is equivalent to:

$$\begin{aligned}\mathbf{G}(s) = \mathbf{X}(s)\mathbf{U}^{\mathrm{T}}(s)\left(\mathbf{U}(s)\mathbf{U}^{\mathrm{T}}(s)\right)^{-1} &= \underbrace{\mathbf{T}(s\mathbf{I}-\tilde{\mathbf{A}})^{-1}\mathbf{T}^{-1}\mathbf{B}}_{\text{Transfer Function}} \\ &= \underbrace{(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}}_{\text{Transfer Function}} .\end{aligned} \tag{2.8.21}$$

In the second method, the transfer function is first formulated for the intrinsic case of no time-lags by considering the dynamics involving only the state variables. To this transfer function is then multiplied the kernel transfer function to obtain the overall system response to variations in the control variables. This decoupling approach is illustrated in Figure 2.8.2.

The linearized dynamic equation for the state variable is obtained from Equation (2.8.14a):

$$\frac{d\mathbf{X}_1(t)}{dt} = \mathbf{A}_{11}\mathbf{X}_1(t) + \mathbf{A}_{12}\mathbf{X}_2(t) + \mathbf{B}_1\mathbf{U}(t), \tag{2.8.22}$$

where $\mathbf{A}_{11} = \mathbf{f}_{1\mathbf{x}_1}$ (an $n_1 \times n_1$ matrix), $\mathbf{A}_{12} = \mathbf{f}_{1\mathbf{x}_2}$ (an $n_1 \times n_2$ matrix), and $\mathbf{B}_1 = \mathbf{f}_{1\mathbf{u}}$ (an $n_1 \times m$ matrix).

The kernel transfer function for the time-lag variables $\mathbf{x}_2$ can be directly extracted from our earlier discussion in this section. For the jth element of the $\mathbf{x}_2$ vector, the following time-lag differential equation:

$$\sum_{i=0}^{n_j+1} \binom{n+1}{i} T^i \frac{d^i x_{2j}(t)}{dt^i} = \sum_{i=0}^{n_j} \alpha_i T^i \frac{d^i z_j}{dt^i} \tag{2.8.23}$$

is translated to the Laplace domain as:

$$X_{2j}(s) = G_{2j}(s) Z_j(s), \tag{2.8.24}$$

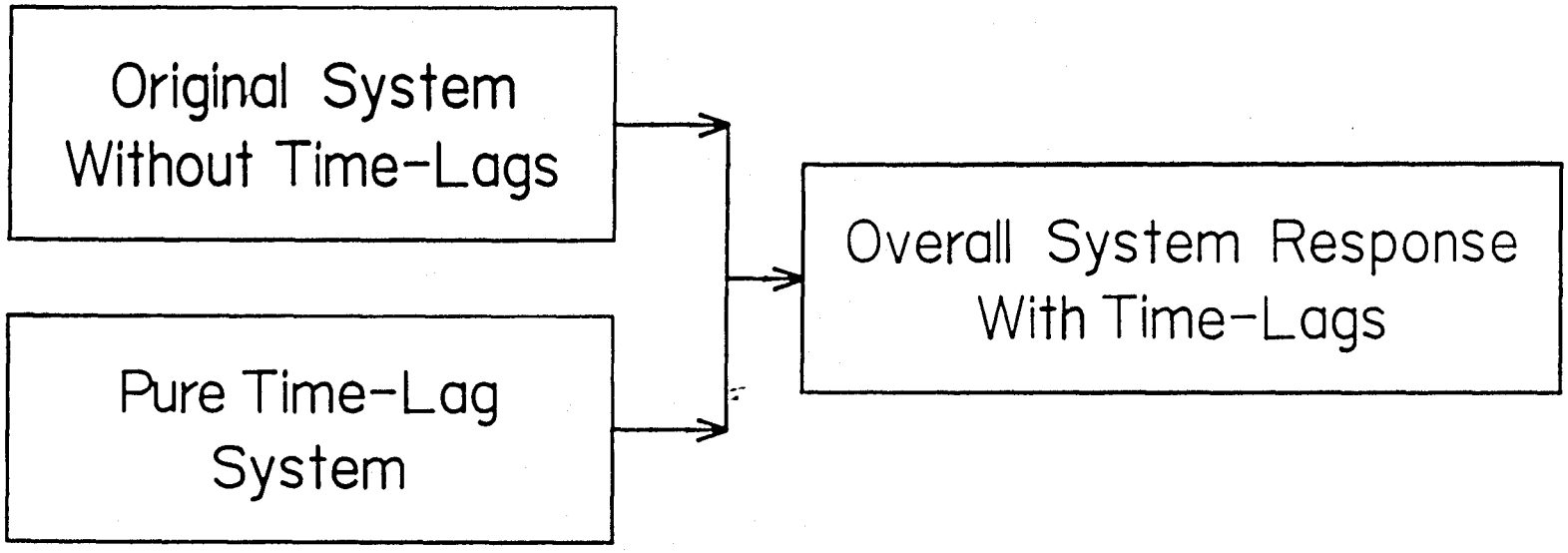


Figure 2.8.2. Decomposition of an overall time-lag system into two components. The original time-lag free system and the purely time-lag system can be solved separately to obtain the overall transfer function for the combined system.

where each kernel transfer function G_{2j} describes the dynamics of the individual time-lag variable $X_{2j}(s)$ as a response to changes in the corresponding intrinsic variable Z_j.

$$G_{2j}(s) = \sum_{i=0}^{n_j} \frac{a_{ji}}{(1+Ts)^{i+1}} \tag{2.8.25}$$

Thus, the following relation can be written for the entire set of non-interacting time-lag variables.

$$\underbrace{\begin{bmatrix} X_{21}(s) \\ X_{22}(s) \\ \vdots \\ X_{2k}(s) \end{bmatrix}}_{\mathbf{X}_2(s)} = \underbrace{\begin{bmatrix} G_{21}(s) & 0 & \dots & 0 \\ 0 & G_{22}(s) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G_{2k}(s) \end{bmatrix}}_{\mathbf{G}_2(s)} \underbrace{\begin{bmatrix} Z_1(s) \\ Z_2(s) \\ \vdots \\ Z_k(s) \end{bmatrix}}_{\mathbf{Z}(s)} \tag{2.8.26a}$$

$$\mathbf{X}_2(s) = \mathbf{G}_2(s)\mathbf{Z}(s) = \mathbf{G}_2(s)\mathbf{Z}_{\mathbf{x}_1}\mathbf{X}_1(s) \tag{2.8.26b}$$

The last relation in the above equation is obtained by using the chain rule of differentiation.

$$\frac{d\mathbf{Z}(t)}{dt} = \frac{d\mathbf{Z}}{d\mathbf{X}_1}\frac{d\mathbf{X}_1}{dt} \quad \Longrightarrow \quad \mathbf{Z}(s) = \mathbf{Z}_{\mathbf{x}_1}\mathbf{X}_1(s) \tag{2.8.27}$$

Each row of the linearization matrix $\mathbf{Z}_{\mathbf{x}_1}$, *i.e.*, essentially a row gradient vector, contains only one nonzero element if the intrinsic quantity depends on only one state variable. The corresponding row of $\mathbf{Z}_{\mathbf{x}_1}$ will contain more than one nonzero element if the intrinsic variable depends on more than one state variable. A common example is a specific growth rate expression that contains concentrations of multiple limiting substrates, products, and inhibitors. Furthermore, for an interacting time-lag system, the kernel transfer matrix $\mathbf{G}_2(s)$ will not be purely diagonal as above. The interaction will be manifested in some nonzero off-diagonal terms. An interacting system may result if, for example, the observed specific growth rate depends not only on the intrinsic specific growth rate μ but also on the intrinsic yield coefficient Y_s. Although only the simpler examples of non-interacting, single

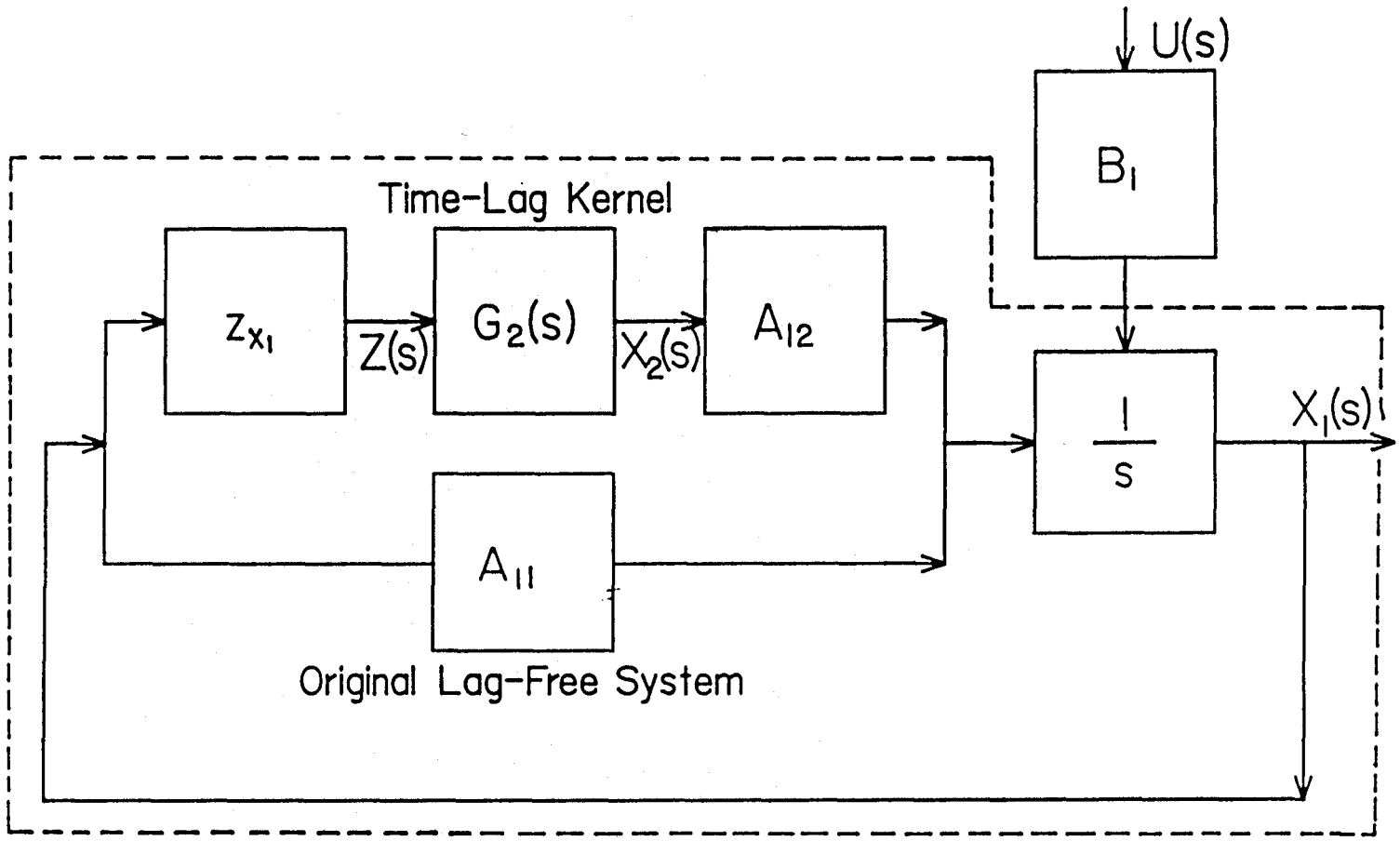
time-lag, single state dependent systems will be considered in this thesis, the formulation can be extended to more complicated systems with little or no significant modification.

The final relationship between the state variables $\mathbf{X}_1$ and the control variables $\mathbf{U}$ in the presence of time-lag effects results from combining Equations (2.8.22) and (2.8.26b) together.

$$\mathbf{G}_1(s) = \mathbf{X}_1(s)\mathbf{U}^{\mathrm{T}}(s)\left(\mathbf{U}(s)\mathbf{U}^{\mathrm{T}}(s)\right)^{-1} = \underbrace{\left[s\mathbf{I} - (\mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{G}_2(s)\mathbf{Z}_{\mathbf{x}_1})\right]^{-1}\mathbf{B}_1}_{\text{Transfer Function}} \tag{2.8.28}$$

The analogous transfer function expression with a diagonal transformation can also be derived similarly through eigenvalue and eigenvector analysis. The block diagram of this time-lag system is presented in Figure 2.8.3.

In this second method, one takes the Laplace transforms of Equations (2.8.14a) and (2.8.14b) separately to derive the respective transfer functions. One of the immediate advantages is the ease of manipulation afforded by a smaller system dimension. Another advantage is the fact that the analysis of the time-lag system is decomposed into two parts so that the individual results from each part can be easily derived from the classical systems theories on frequency response. The first part expresses the dependence of the system states (system output) on the control variables (system input) in the original time-lag free system, and the second part represents the additional contribution due to the presence of time-lags. Thus, the overall system response can be clearly attributed to these two separate effects. Note that the formulation of transfer function considered in this section is quite general in the sense that it allows the presence of multiple time-lag variables, each of which in turn may depend on multiple state variables. Multivariable extensions of the basic time-lag theory are discussed in detail in a later section.



$$G(s) = \left[sI - (A_{11} + A_{12}G_2 Z_{x_1})\right]^{-1} B_1$$

Figure 2.8.3. Block diagram of a time-lag system. Note that the original system is obtained when the time-lag branch is omitted.

Chemostat Example

For simplicity, possible time-lags in the yield coefficient will not be considered here. Likewise, forcing in the substrate feed concentration is not simulated in this example. Nonetheless, the effect of all these variables can be studied in exactly the same manner.

First, a simple chemostat system of Equations (2.2.5) and (2.2.6) is used to demonstrate the calculation of the transfer function with no time-lag. The starting linearized state equations are:

$$\underbrace{\begin{bmatrix} \frac{dX(t)}{dt} \\ \frac{dS(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{X}_1(t)}{dt}} = \underbrace{\begin{bmatrix} 0 & x_0\mu_0' \\ -\frac{1}{Y_s}\mu_0 & -D_0 - \frac{1}{Y_s}x_0\mu_0' \end{bmatrix}}_{\mathbf{A}_{11}} \underbrace{\begin{bmatrix} X(t) \\ S(t) \end{bmatrix}}_{\mathbf{X}_1(t)} + \underbrace{\begin{bmatrix} -x_0 \\ s_f - s_0 \end{bmatrix}}_{\mathbf{B}_1} \underbrace{[D(t)]}_{\mathbf{U}(t)}, \tag{2.8.29}$$

where the subscript "$_0$" signifies the steady-state values for the corresponding variables. Furthermore, $D_0 = \mu_0$ and $s_f - s_0 = \frac{1}{Y_s}x_0$. From the above equation, the transfer functions relating the states $X(s)$ and $S(s)$ to the sinusoidal disturbance in the dilution rate $D(s)$ are calculated.

$$\mathbf{G}(s) = \left[s\mathbf{I} - \mathbf{A}_{11}\right]^{-1}\mathbf{B}_1 \tag{2.8.30a}$$

$$\mathbf{G}(s) = \begin{bmatrix} \frac{X(s)}{D(s)} \\ \frac{S(s)}{D(s)} \end{bmatrix} = \begin{bmatrix} s & -x_0\mu_0' \\ \frac{1}{Y_s}\mu_0 & s + \mu_0 + C \end{bmatrix}^{-1} \begin{bmatrix} -x_0 \\ \frac{1}{Y_s}x_0 \end{bmatrix}, \tag{2.8.30b}$$

where, as before, $C = \frac{1}{Y_s}x_0\mu_0' = (s_f - s_0)\mu_0'$ is the stability variable. After the matrix inversion, the transfer functions in the absence of time-lag are:

$$G_{\frac{X}{D}}(s) = \frac{X(s)}{D(s)} = -x_0\frac{1}{s+c} \tag{2.8.31a}$$

$$G_{\frac{S}{D}}(s) = \frac{S(s)}{D(s)} = \frac{1}{Y_s}x_0\frac{1}{s+c}. \tag{2.8.31b}$$

Note that for the system to be locally stable, the denominator of the transfer function, which is equivalent to the characteristic equation of the linearized system, must have negative roots.

$$s + C = 0 \qquad \Longrightarrow s = -C \tag{2.8.32}$$

Thus, as derived previously, the system is stable only if C is non-negative.

To demonstrate the first method of calculating the transfer function for a system with a combined 0th- and 1st-order time-lag kernel, $k(t) = a_0 k_0(t) + a_1 k_1(t)$, in the specific growth rate, the linearized chemostat state equations with a sinusoidal forcing in the dilution rate are derived directly from Equations (2.2.10) and (2.2.11). The following equations result from incorporating a dilution rate forcing in the equations used in the preceeding section on chemostat stability analysis.

$$\frac{dX(t)}{dt} = x_0 Y(t) - x_0 D(t) \tag{2.8.33a}$$

$$\frac{dS(t)}{dt} = -\frac{1}{Y_s} y_0 X(t) - \underbrace{D_0}_{y_0} S(t) - \frac{1}{Y_s} x_0 Y(t) + \underbrace{(s_f - s_0)}_{\frac{1}{Y_s} x_0} D(t), \tag{2.8.33b}$$

$$\frac{dY(t)}{dt} = Z(t) \tag{2.8.33c}$$

$$\begin{aligned} \frac{dZ(t)}{dt} = &-a_0 \frac{1}{T} \frac{1}{Y_s} \mu_0' y_0 X(t) + \frac{1}{T^2}(1 - a_0 T y_0) \mu_0' S(t) \\ &- \frac{1}{T^2}(1 + C a_0 T) Y(t) - \frac{2}{T} Z(t) + C a_0 \frac{1}{T} D(t) \end{aligned} \tag{2.8.33d}$$

The Jacobian matrix $\mathbf{A} = \mathbf{f_x}$ for the above set of system dynamic equations is:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & x_0 & 0 \\ -\frac{1}{Y_s} y_0 & -y_0 & -\frac{1}{Y_s} x_0 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 \frac{1}{T} \frac{1}{Y_s} \mu_0' y_0 & \frac{1}{T^2}(1 - a_0 T y_0) \mu_0' & -\frac{1}{T^2}(1 + C a_0 T) & -\frac{2}{T} \end{bmatrix}. \tag{2.8.34}$$

And the corresponding control matrix $\mathbf{B} = \mathbf{f_u}$ is:

$$\mathbf{B} = \begin{bmatrix} -x_0 \\ \frac{1}{Y_s}x_0 \\ 0 \\ Ca_0\frac{1}{T} \end{bmatrix} \tag{2.8.35}$$

The transfer function for this system of dilution rate forcing is computed by substituting the above matrices **A** and **B** directly into Equation (2.8.21) .

$$\begin{aligned} \mathbf{G}(s) &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \\ &= \begin{bmatrix} s & 0 & -x_0 & 0 \\ \frac{1}{Y_s}y_0 & s+y_0 & \frac{1}{Y_s}x_0 & 0 \\ 0 & 0 & s & -1 \\ a_0\frac{1}{T}\frac{1}{Y_s}\mu_0' y_0 & -\frac{1}{T^2}(1-a_0Ty_0)\mu_0' & \frac{1}{T^2}(1+Ca_0T) & s+\frac{2}{T} \end{bmatrix}^{-1} \begin{bmatrix} -x_0 \\ \frac{1}{Y_s}x_0 \\ 0 \\ a_0\frac{1}{T}C \end{bmatrix} \end{aligned} \tag{2.8.36}$$

The determinant of $(s\mathbf{I} - \mathbf{A})$ in the above equation is exactly the same as the characteristic equation encountered in local stability analysis.

$$\begin{aligned} &\det|s\mathbf{I} - \mathbf{A}| \\ &= \begin{vmatrix} s & 0 & -x_0 & 0 \\ \frac{1}{Y_s}y_0 & s+y_0 & \frac{1}{Y_s}x_0 & 0 \\ 0 & 0 & s & -1 \\ a_0\frac{1}{T}\frac{1}{Y_s}\mu_0' y_0 & -\frac{1}{T^2}(1-a_0Ty_0)\mu_0' & \frac{1}{T^2}(1+Ca_0T) & s+\frac{2}{T} \end{vmatrix} \\ &= \frac{1}{T^2}\Big[s^4T^2 + s^3(2T + T^2y_0) + s^2(1 + 2Ty_0 + Ca_0T) + s(y_0 + C + Ca_0Ty_0) + Cy_0\Big] \\ &= \frac{(s+y_0)}{T^2}\Big[s(1+Ts)^2 + C(1+a_0Ts)\Big] \end{aligned} \tag{2.8.37}$$

The inverse of $(s\mathbf{I} - \mathbf{A})$ is:

$$(s\mathbf{I} - \mathbf{A})^{-1} = \frac{1}{\det|s\mathbf{I} - \mathbf{A}|T^2}\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix}, \tag{2.8.38}$$

where

$$q_{11} = (s + y_0)(1 + Ts)^2 + C + Ca_0Ts \tag{2.8.39a}$$

$$q_{12} = \mu_0' x_0(1 - a_0Ty_0) \tag{2.8.39b}$$

$$q_{14} = T^2 x_0 (s + y_0) \tag{2.8.39c}$$

$$q_{21} = -\frac{1}{Y_s} y_0 (1 + Ts)^2 \tag{2.8.39d}$$

$$q_{22} = s(1 + Ts)^2 + C a_0 T (s + y_0) \tag{2.8.39e}$$

$$q_{24} = -T^2 \frac{1}{Y_s} x_0 (s + y_0) \tag{2.8.39f}$$

$$q_{31} = -\frac{1}{Y_s} \mu_0' y_0 (1 + a_0 T s) \tag{2.8.39g}$$

$$q_{32} = \mu_0' s (1 - a_0 T y_0) \tag{2.8.39h}$$

$$q_{34} = T^2 s (s + y_0) \tag{2.8.39i}$$

Finally post multiplying $(s\mathbf{I} - \mathbf{A})^{-1}$ by $\mathbf{B}$ gives:

$$\mathbf{G}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \begin{bmatrix} G_{\frac{X}{D}}(s) \\ G_{\frac{S}{D}}(s) \\ G_{\frac{Y}{D}}(s) \\ G_{\frac{Z}{D}}(s) \end{bmatrix}, \tag{2.8.40}$$

where

$$\boxed{G_{\frac{X}{D}}(s) = \frac{X(s)}{D(s)} = -x_0 \frac{(1 + Ts)^2}{s(1 + Ts)^2 + C(1 + a_0 T s)}} \tag{2.8.41a}$$

$$\boxed{G_{\frac{S}{D}}(s) = \frac{S(s)}{D(s)} = \frac{1}{Y_s} x_0 \frac{(1 + Ts)^2}{s(1 + Ts)^2 + C(1 + a_0 T s)}} \tag{2.8.41b}$$

$$\boxed{G_{\frac{Y}{D}}(s) = \frac{Y(s)}{D(s)} = \frac{C(1 + a_0 T s)}{s(1 + Ts)^2 + C(1 + a_0 T s)}} \tag{2.8.41c}$$

Note that $X(s)$ and $S(s)$ are closely related by:

$$\frac{X(s)}{S(s)} = -\frac{1}{Y_s} \tag{2.8.42}$$

In addition, $\frac{Y(s)}{D(s)}$ can be expressed as the following products:

$$\frac{Y(s)}{D(s)} = \underbrace{\frac{(1 + a_0 T s)}{(1 + Ts)^2}}_{\frac{Y(s)}{M(s)}} \overbrace{\underbrace{\mu_0'}_{\frac{M(s)}{S(s)}} \frac{1}{Y_s} x_0}^{C} \underbrace{\frac{(1 + Ts)^2}{s(t + Ts)^2 + C(1 + a_0 T s)}}_{\frac{S(s)}{D(s)}} = \frac{Y(s)}{M(s)} \cdot \frac{M(s)}{S(s)} \cdot \frac{S(s)}{D(s)} \tag{2.8.43}$$

Because of the dimension of the system, the analytical solution of $(s\mathbf{I} - \mathbf{A})^{-1}$ is rather involved. For this four-dimensional system, an analytical solution can try one's patience. For a slightly higher-dimensional system, an analytical solution may be beyond an average person's level of perseverance, and a reduction in the dimension afforded by the following second method can be well appreciated because it greatly simplifies the algebra.

Based on the same set of chemostat state equations subject to a combined 0th- and 1st-order time-lag in the specific growth rate, the derivation of the transfer function is now demonstrated by using the second approach. The linearized state equations (2.8.33a) and (2.8.33b) used in the first method are expressed in an appropriate matrix form as:

$$\underbrace{\begin{bmatrix} \frac{dX(t)}{dt} \\ \frac{dS(t)}{dt} \end{bmatrix}}_{\frac{d\mathbf{X}_1(t)}{dt}} = \underbrace{\begin{bmatrix} 0 & 0 \\ -\frac{1}{Y_g}y_0 & -y_0 \end{bmatrix}}_{\mathbf{A}_{11}} \underbrace{\begin{bmatrix} X(t) \\ S(t) \end{bmatrix}}_{\mathbf{X}_1(t)} + \underbrace{\begin{bmatrix} x_0 \\ -\frac{1}{Y_g}x_0 \end{bmatrix}}_{\mathbf{A}_{12}} \underbrace{[Y(t)]}_{\mathbf{X}_2(t)} + \underbrace{\begin{bmatrix} -x_0 \\ \frac{1}{Y_g}x_0 \end{bmatrix}}_{\mathbf{B}_1} \underbrace{[D(t)]}_{\mathbf{U}(t)} \quad (2.8.44)$$

The transfer function for a combined 0th- and 1st-order time-lag kernel, $k(t) = a_0k_0(t) + a_1k_1(t)$, is decoupled from the rest of the state equations and is derived separately.

$$T^2\frac{d^2Y(t)}{dt^2} + 2T\frac{dY(t)}{dt} + Y(t) = M(t) + a_0T\frac{dM(t)}{dt} \quad (2.8.45a)$$

$$\Longrightarrow \quad T^2s^2Y(s) + 2TsY(s) + Y(s) = M(s) + a_0TsM(s) \quad (2.8.45b)$$

$$\Longrightarrow \quad \boxed{G_y(s) = \frac{Y(s)}{M(s)} = \frac{1 + a_0Ts}{(1 + Ts)^2}} \quad (2.8.45c)$$

The constitutive relationship of $\mu(s)$ can be similarly linearized around the steady-state value s_0 to yield:

$$\mu(s) \approx \mu_0's \Longrightarrow \quad M(s) = \mu_0'S(s) \quad (2.8.46)$$

Substituting the above relationship into the kernel transfer function, one obtains:

$$\underbrace{\frac{Y(s)}{X_2(s)}}_{} = \underbrace{\frac{1+a_0Ts}{(1+Ts)^2}}_{G_2}\underbrace{[0 \;\; \mu_0']}_{Z_{x_1}}\underbrace{\begin{bmatrix} X(s) \\ S(s) \end{bmatrix}}_{X_1(s)} \tag{2.8.47}$$

Finally, the overall transfer function is obtained by combining Equations (2.8.44) and (2.8.47) together.

$$\mathbf{G}_1(s) = \left[s\mathbf{I} - (\mathbf{A}_{11} + \mathbf{A}_{12}\mathbf{G}_2(s)\mathbf{Z}_{x_1})\right]^{-1}\mathbf{B}_1$$

$$= \left[\overbrace{\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}}^{s\mathbf{I}} - \overbrace{\begin{bmatrix} 0 & 0 \\ -\frac{1}{Y_s}y_0 & -y_0 \end{bmatrix}}^{\mathbf{A}_{11}} - \overbrace{\begin{bmatrix} x_0 \\ -\frac{1}{Y_s}x_0 \end{bmatrix}}^{\mathbf{A}_{12}}\overbrace{\begin{bmatrix} 1+a_0Ts \\ (1+Ts)^2 \end{bmatrix}}^{\mathbf{G}_2(s)}\overbrace{[0 \;\; \mu_0']}^{\mathbf{Z}_{x_1}}\right]^{-1} \cdot \overbrace{\begin{bmatrix} -x_0 \\ \frac{1}{Y_s}x_0 \end{bmatrix}}^{\mathbf{B}_1} \tag{2.8.48a}$$

$$\mathbf{G}_1(s) = \begin{bmatrix} \frac{X(s)}{D(s)} \\ \frac{S(s)}{D(s)} \end{bmatrix} = \begin{bmatrix} s & -\frac{x_0\mu_0'(1+a_0Ts)}{(1+Ts)^2} \\ \frac{1}{Y_s}y_0 & \frac{(s+y_0)(1+Ts)^2+C(1+a_0Ts)}{(1+Ts)^2} \end{bmatrix}^{-1}\begin{bmatrix} -x_0 \\ \frac{1}{Y_s}x_0 \end{bmatrix} \tag{2.8.48b}$$

After expansion, the first element of the above vector equation gives the effect of variations in the dilution rate on the biomass concentration.

$$\boxed{G_{\frac{X}{D}}(s) = \frac{X(s)}{D(s)} = -x_0\frac{(1+Ts)^2}{s(1+Ts)^2 + C(1+a_0Ts)}} \tag{2.8.49a}$$

Similar transfer function is also obtained to relate the substrate concentration to the dilution rate.

$$\boxed{G_{\frac{S}{D}}(s) = \frac{S(s)}{D(s)} = \frac{1}{Y_s}x_0\frac{(1+Ts)^2}{s(1+Ts)^2 + C(1+a_0Ts)}} \tag{2.8.49b}$$

Furthermore, the dependence of the observed specific growth rate on the dilution rate can be obtained by multiplying all the expressions together.

$$\frac{Y(s)}{D(s)} = \frac{Y(s)}{M(s)} \cdot \frac{M(s)}{S(s)} \cdot \frac{S(s)}{D(s)}$$
$$= \left[\frac{(1+a_0Ts)}{(1+Ts)^2}\right][\mu_0']\left[\frac{1}{Y_s}x_0\frac{(1+Ts)^2}{s(1+Ts)^2 + C(1+a_0Ts)}\right] \tag{2.8.49c}$$

$$\boxed{\frac{Y(s)}{D(s)} = \frac{C(1+a_0Ts)}{s(1+Ts)^2 + C(1+a_0Ts)}} \tag{2.8.49d}$$

Of course, the results of the second method are exactly the same as those derived through the first method.

As a special case, the transfer functions in the absence of time-lag effect are calculated by setting $T = 0$ in the above equations.

$$\lim_{T\to 0} G_{\frac{X}{D}}(s) = -x_0 \frac{1}{s+c} \tag{2.8.50a}$$

$$\lim_{T\to 0} G_{\frac{S}{D}}(s) = \frac{1}{Y_s} x_0 \frac{1}{s+c} \tag{2.8.50b}$$

Although it has been pointed out that $G_{\frac{X}{D}}(s)$ and $G_{\frac{S}{D}}(s)$ are related by a constant ratio, it is interesting to note that this conclusion is true whether or not there is time-lag:

$$\frac{G_{\frac{X}{D}}(s)}{G_{\frac{S}{D}}(s)} = \frac{\frac{X(s)}{D(s)}}{\frac{S(s)}{D(s)}} = \frac{X(s)}{S(s)} = -\frac{1}{Y_s}. \tag{2.8.51}$$

Note that $G_{\frac{X}{D}}(s)$, $G_{\frac{S}{D}}(s)$, and $G_{\frac{Y}{D}}(s)$ all have a common denominator $s(1+Ts)^2 + C(1+a_0Ts)$, which, when set to 0, is the same as the characteristic equation of (2.7.21).

$$s(1+Ts)^2 + C(1+a_0Ts) = 0 \tag{2.8.52}$$

As before, the local stability of this time-lag system can also be analyzed with the above equation. The roots to the above equation are negative and, thus, the chemostat system is stable provided that the following condition is satisfied:

$$CT < \frac{2}{1-2a_0}, \tag{2.8.53}$$

which is again the same as that derived previously. The above point corresponds to where the two complex roots cross the imaginary axis. A similar analysis is carried out for the fork point where two real roots become equal and branch into the complex plane. This is the boundary for damped oscillation. For a chemostat

with a 0th-order time-lag, an exponential approach to the nontrivial steady-state is possible if

$$CT < \frac{1}{4}. \tag{2.8.54}$$

If the chemostat has a purely 1st-order time-lag, the corresponding condition is:

$$CT < \frac{4}{27}. \tag{2.8.55}$$

For a general combined 0th and 1st-order time-lag kernel, the condition for an exponential approach to the steady-state is:

$$CT < \frac{-(27 - 36a_0 + 8a_0^2) + \sqrt{(27 - 36a_0 + 8a_0^2)^2 - 64a_0^3(a_0 - 1)}}{8a_0^3}. \tag{2.8.56}$$

How the linearized variables depend on each other is shown in Figure 2.8.4. Because both $G_{\frac{X}{D}}(s)$ and $G_{\frac{S}{D}}(s)$ in Equations (2.8.49a) and (2.8.49b) have a quadratic polynomial of s in the numerator and a cubic polynomial of s in the denominator, the functions approach $\frac{1}{s}$ for large values of s. Hence there is a first-order time-lag between the dilution rate input and the biomass/substrate concentration output. Because the intrinsic specific growth rate is assumed to be a direct function of the substrate concentration, it is in perfect synchronization with the substrate concentration. The time-lag between the intrinsic specific growth rate and the observed specific growth rate depends on the relative weight given to the order of the kernel as a_0. From the kernel transfer function of Equation (2.8.45c), it is clear that the order of lag with respect to the intrinsic specific growth rate or the substrate concentration is one for $a_0 = 1$ but increases to two for $a_0 = 0$. Thus, the order of lag in the observed specific growth rate with respect to the dilution rate forcing is two for $a_0 = 1$ and increases to three for $a_0 = 0$.

From these transfer functions, it is a simple matter to calculate the theoretical amplitude ratio (AR) and the phase angle ϕ by substituting the Laplace variable s

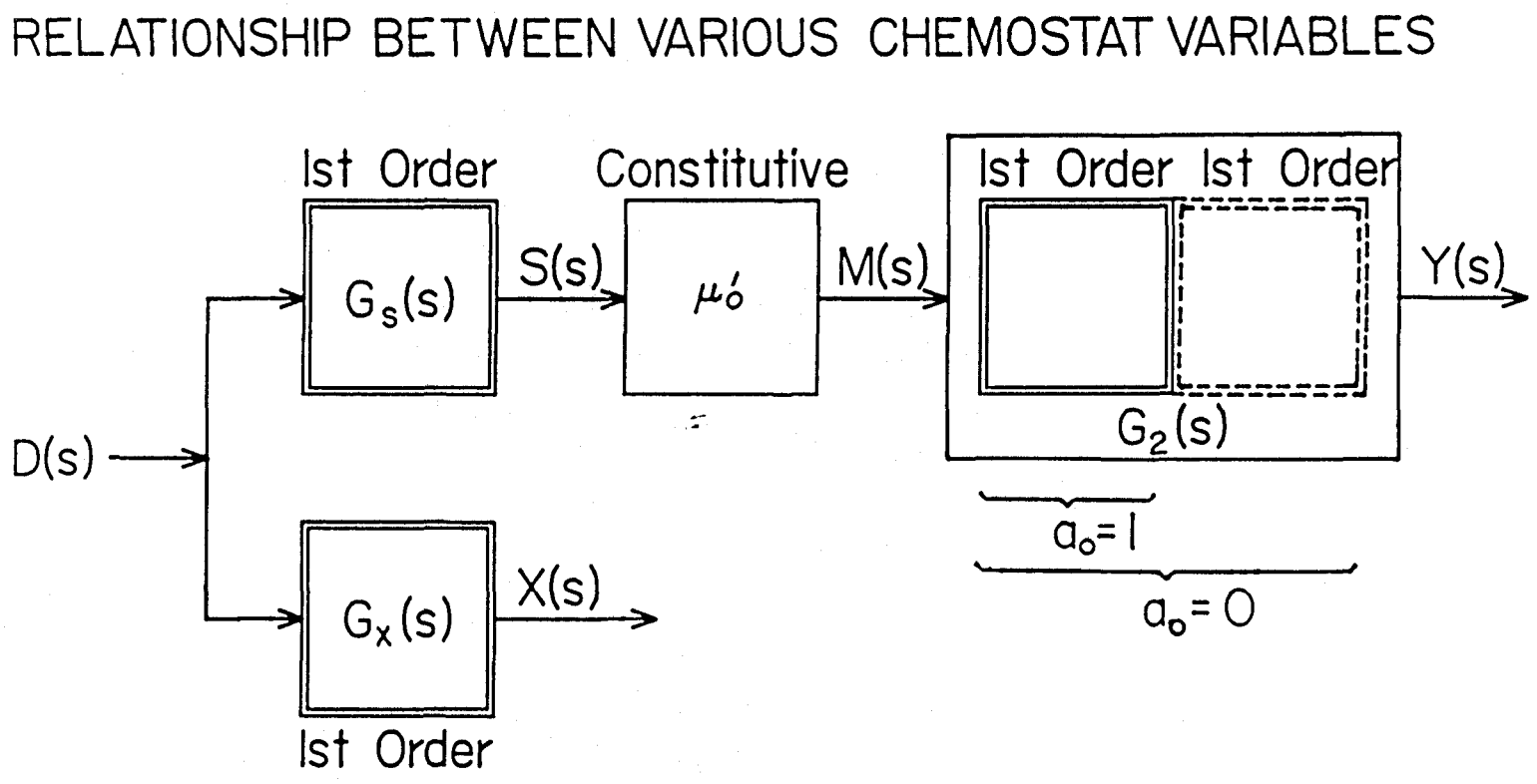


Figure 2.8.4. Time-lag relationship between various variables in a chemostat.

with $i\omega$.

$$AR = \left|G(i\omega)\right| \tag{2.8.57}$$

$$\phi = \angle G(i\omega) = \tan^{-1}\left(\frac{\Im m\left[G(i\omega)\right]}{\Re e\left[G(i\omega)\right]}\right) \tag{2.8.58}$$

Sinusoidal Dilution Rate Forcing for the Linearized Chemostat Example

Before proceeding with the discussion on the dependence of the system response as a function of the time-lag kernel, a brief comment is warranted here on the unfortunate use of "order" in two different contexts in this section. This thesis has consistently used a mathematical *time-lag kernel* order counter that starts from 0; however, a *dynamic system* order counter starts from 1, in accordance with the established usage. This is analogous to bit counting in computer science following either the bit order convention or the bit number convention, where the *first* bit (bit order) is referred to as the *0th* bit (bit number). Although there should be little confusion from the context, ordinal numbers will be spelled out (first, second, etc.) when the order of a dynamic system is being referenced; they will remain numeral (1st, 2nd, etc.) when the order of the kernel function is the subject. Thus, a *0th*-order kernel function gives rise to a *first*-order system lag, and a purely *1st*-order kernel function gives rise to two *first*-order system lags in series, which is also commonly referred to as a *second*-order system.

Following the previous chemostat example, the effect of changes in the time-lag parameters on the theoretical frequency response of the linearized chemostat system is studied with a generalized 1st-order kernel function: $k(t) = a_0 k_0(t) + a_1 k_1(t)$. The stability variable used in this simulation is 0.5 hr^{-1}, and all the parameters and the combinations thereof are well within the stability limit of $CT < \frac{1}{0.5 - a_0}$. Thus, one is assured that the transfer functions are not stretched beyond the valid regime in obtaining the Bode diagrams. The parameters whose effects are to be studied

are the lag time constant, T, the relative weight of the kernel between the 0th-order base function and the 1st-order base function, a_0, and the stability variable C. Representative points chosen from the kernel parameter space are indicated in Figure 2.8.5.

Three separate Bode plots are presented in each of the following amplitude ratio figures and phase angle figures as functions of the forcing frequency. These plots are derived from the respective transfer functions described by Equation (2.8.49a) for the biomass concentration with respect to oscillations in the dilution rate, Equation (2.8.45c) for the observed specific growth rate with respect to oscillations in the intrinsic specific growth rate, and Equation (2.8.49d) for the observed specific growth rate with respect to oscillations in the dilution rate. The substrate concentration is not plotted because its oscillation is always in the opposite direction of the biomass concentration. Furthermore, the last plot of the observed specific growth rate with respect to the dilution rate is redundant, because it can be calculated from the first two. Namely, for a linearized system, the amplitude ratio of the last plot is the product of the amplitude ratios of the first two plots, and the phase angle of the last plot is the sum of the phase angles of the first two plots.

For the amplitude ratio plots, the magnitudes have been normalized with respect to the steady-state values, *i.e.*, $A.R._0 = \lim_{\omega \to 0} A.R.(\omega)$, which are the same as those obtained by directly setting $s \to 0$ in the respective transfer functions. For example, the biomass/D $A.R.$ plot is normalized with $-x_0 \frac{1}{C}$.

$$\lim_{s \to 0} G_{\frac{X}{D}}(s) = \lim_{s \to 0} -x_0 \frac{(1+Ts)^2}{s(1+Ts)^2 + C(1+a_0 Ts)} = -x_0 \frac{1}{C} \tag{2.8.59}$$

Similarly, the phase angles have also been referenced to the steady-state values, *e.g.*, 180° has been added to the actual phase angle for the biomass/D and Y/D plots.

Effect of Lag Time Constant with a_0=0: Figures 2.8.6 and 2.8.7 are used to compare the effect of time-lag constant for a purely 1st-order time-lag kernel, *i.e.*,

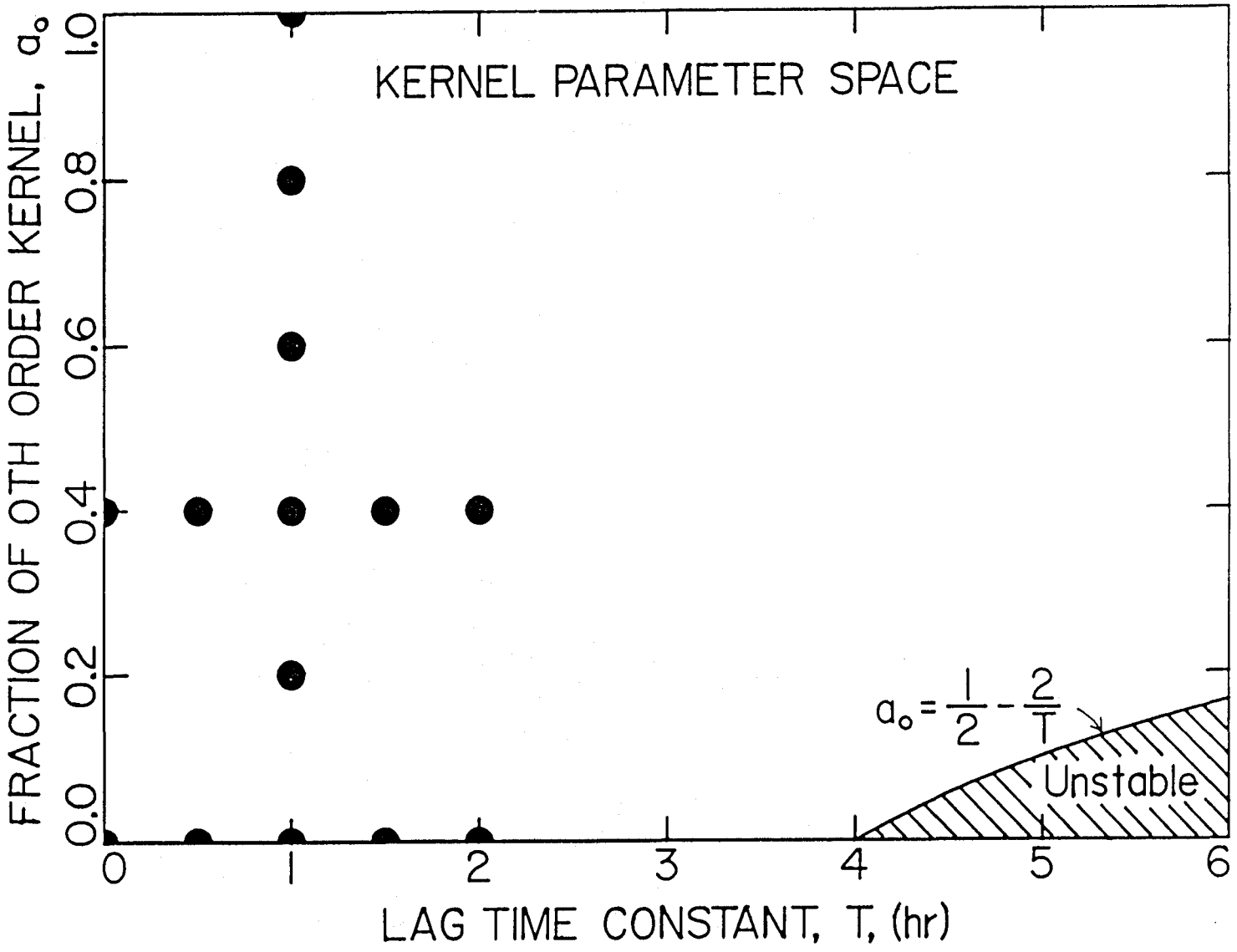


Figure 2.8.5. Parameter space used for linearized frequency response analysis. All the simulation points are within the stability limit.

$a_0 = 0$. Except for the case of $T = 0$, *i.e.*, no time-lag at all, all the high frequency asymptotes in the *A.R.* plot of Y/M have slopes of -2, and the corner frequencies are $\omega_c = \frac{1}{T}$. This is to be expected from a system with two equivalent first-order lags in series, where each lag contributes -1 to the high frequency asymptotic slope. With each lag contributing -90° to the high frequency phase angle, the same conclusion that a purely 1st-order time-lag kernel is equivalent to two first-order CSTRs in series is also confirmed in the corresponding phase angle plot of Y/M, where the phase angle decreases from 0° to -180°.

The *A.R.* and phase angle plots for biomass/D, however, are considerably more complicated. Because Equation (2.8.49a) is not a simple first-order system with the general form of $\frac{1}{1+Ts}$, it does not yield asymptotes with readily identifiable corner frequencies. This can be demonstrated by decomposing the normalized Equation (2.8.49a) into either five first-order dynamic blocks for

$$CT \leq \begin{cases} \frac{-(27-36a_0+8a_0^2)+\sqrt{(27-36a_0+8a_0^2)^2-64a_0^3(a_0-1)}}{8a_0^3} & \text{for } a_0 \neq 0 \\ \frac{4}{27} & \text{for } a_0 = 0 \end{cases} \tag{2.8.60}$$

or four first-order and one second-order dynamic blocks otherwise.

$$G_{\frac{X}{D}}(s)_{\text{normalized}} = \frac{(1+Ts)^2}{\frac{1}{C}s(1+Ts)^2 + (1+a_0Ts)} \tag{2.8.61a}$$

$$G_{\frac{X}{D}}(s)_{\text{normalized}} = \begin{cases} \frac{(1+Ts)^2}{(1+T_1s)(1+T_2s)(1+T_3s)} \\ \frac{(1+Ts)^2}{(1+T_1s)(1+2\xi T_2s+T_2^2s^2)} \end{cases} \tag{2.8.61b}$$

In general, an nth-order polynomial can be decomposed into n first-order factors, one for each real root of the equation. If two roots occur as complex conjugate pairs, then these two factors can be combined to yield an equivalent real second-order factor.

Note that when $\xi < 0.707$ in the above equation, there exists a maximum in the decomposed amplitude ratio that corresponds to the factor $(1 + 2\xi T_2 s + T_2^2 s^2)$

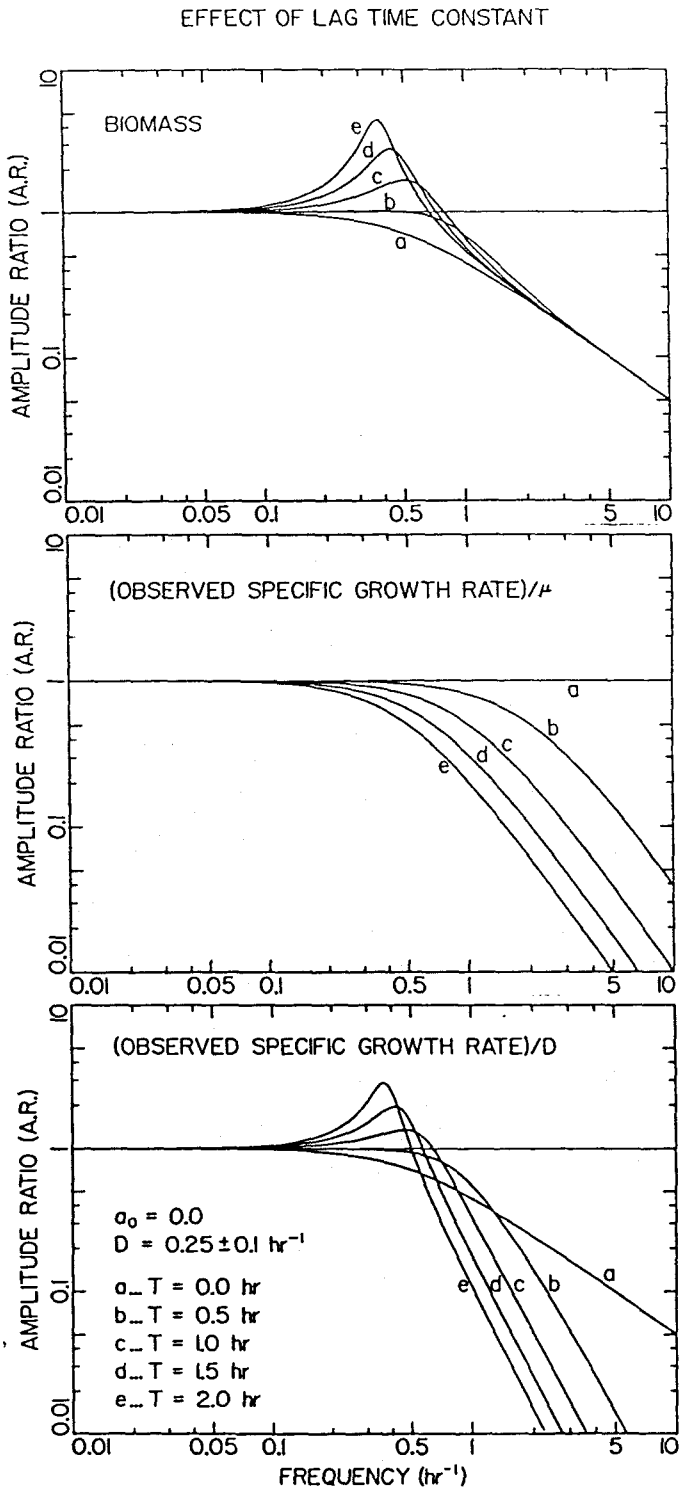


Figure 2.8.6. Effect of the lag time constant, T, on the amplitude ratios for linearized chemostat state equations subject to a sinusoidal dilution rate forcing and a purely 1st-order time-lag kernel. (a_0=0.0, C=0.5 hr^{-1}.)

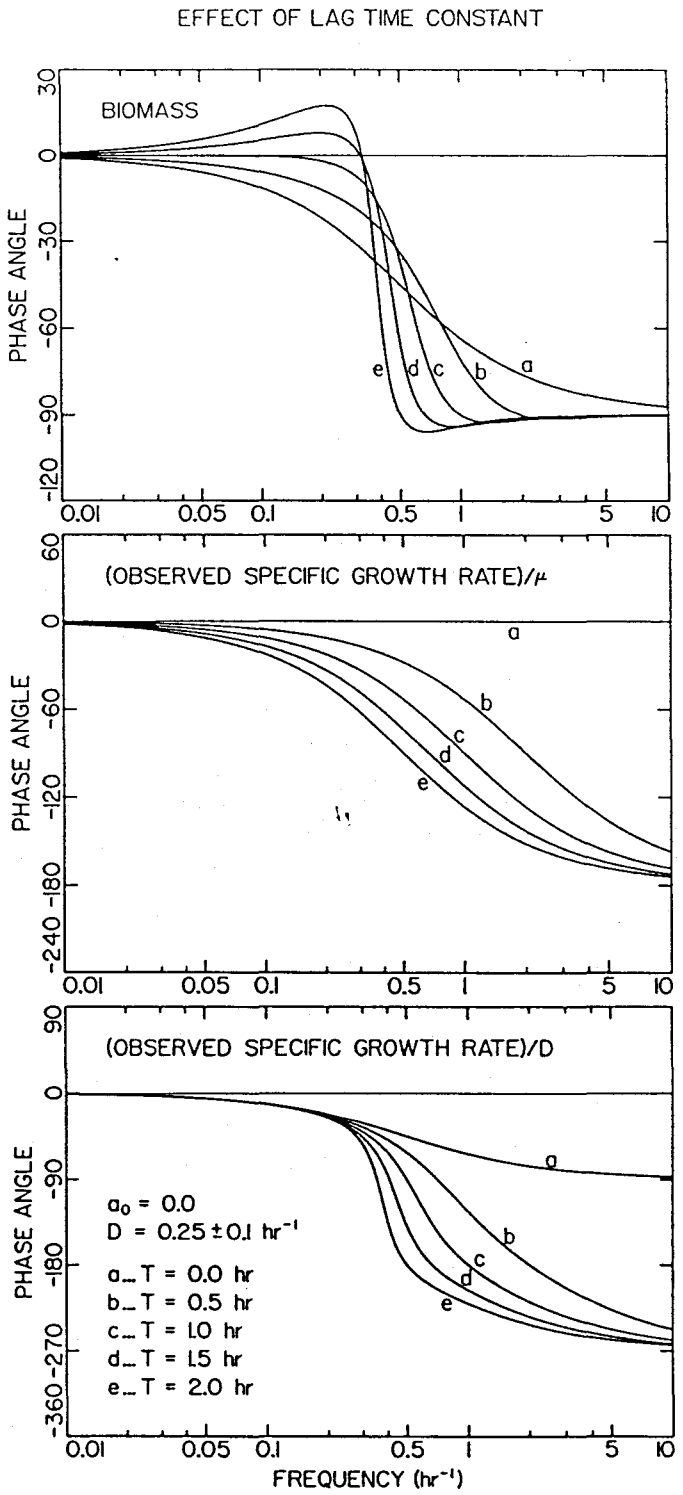


Figure 2.8.7. Effect of the lag time constant, T, on the phase angles for linearized chemostat state equations subject to a sinusoidal dilution rate forcing and a purely 1st-order time-lag kernel. (a_0=0.0, C=0.5 hr^{-1}.)

in the denominator of $G_{\frac{X}{D}}(s)$. The amplitude ratio of this component can rise above unity only when there is a maximum. This maximum is attained at $\omega_{\max}$ described by the following equation:

$$\omega_{\max} = \frac{1}{T_2}\sqrt{1-2\xi^2}. \tag{2.8.62}$$

As a rule of thumb, the maximum occurs in the vicinity of $\omega_{\max} = \frac{1}{T_2}$ because the factor $\sqrt{1-2\xi^2}$ is usually close to unity. The corresponding maximum amplitude is described by:

$$A.R._{\max} = \frac{1}{2\xi\sqrt{1-2\xi^2}}. \tag{2.8.63}$$

Otherwise, $A.R.$ decreases monotonically as ω is increased. As the damping coefficient $\xi \to 0$, $A.R._{\max} \to \infty$. The normalized transfer function at this point of zero damping can be obtained by substituting $(CT)_{\text{crit}} = \frac{1}{0.5-a_0}$ into Equation (2.8.61).

$$G_{\frac{X}{D}}(s)_{\text{normalized,crit}} = \frac{(1+T_{\text{crit}}s)^2}{(1+\frac{T_{\text{crit}}}{2}s)\left[1+(1-2a_0)T_{\text{crit}}^2 s^2\right]}. \tag{2.8.64}$$

This is the point where the two complex roots of the characteristic equation cross the imaginary axis at $\pm i\frac{1}{\sqrt{1-2a_0}T_{\text{crit}}} = \pm i\frac{\sqrt{1-2a_0}}{2}C_{\text{crit}}$.

As an example, for $a_0 = 0$, T=2.0 hr, and C=0.5 hr^{-1}, which is the same as the value used to generate the Bode diagram, one gets $CT = 1 > \frac{4}{27}$; thus, the normalized transfer function is decomposed into four first-order and one second-order dynamic blocks:

$$G_{\frac{X}{D}}(s)_{\text{normalized}} = \frac{(1+2s)^2}{8s^3+8s^2+2s+1} = \frac{(1+2s)^2}{(1+1.14s)(1+2\cdot\underbrace{0.162}_{\xi}\cdot\underbrace{2.65}_{T_2}s+\underbrace{2.65}_{T_2^2}s^2)} \tag{2.8.65}$$

The decomposed Bode diagrams are shown in Figures 2.8.8 and 2.8.9.

Note that the factors in the numerator of $G_{\frac{X}{D}}(s)$ tend to pull the overall $A.R.$ curve up; whereas, the factors in the denominator produce the opposite effect. As T is decreased, the corner frequency for the factor $(1+Ts)^2$ shifts toward the

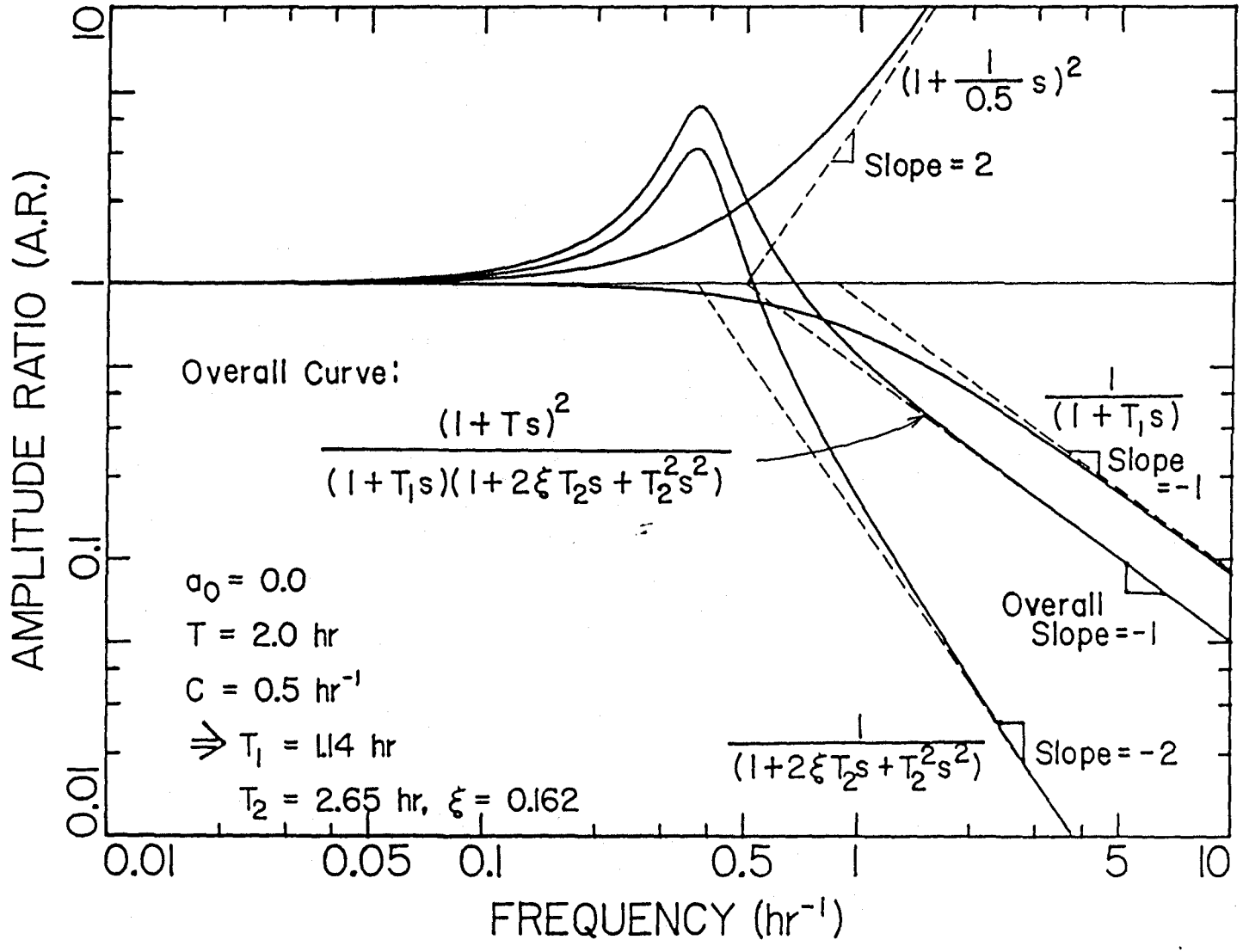


Figure 2.8.8. Construction of the overall biomass/D frequency response curve of the amplitude ratio from the decomposed factors. ($a_0 = 0$, T=2.0 hr, C=0.5 hr^{-1}.)

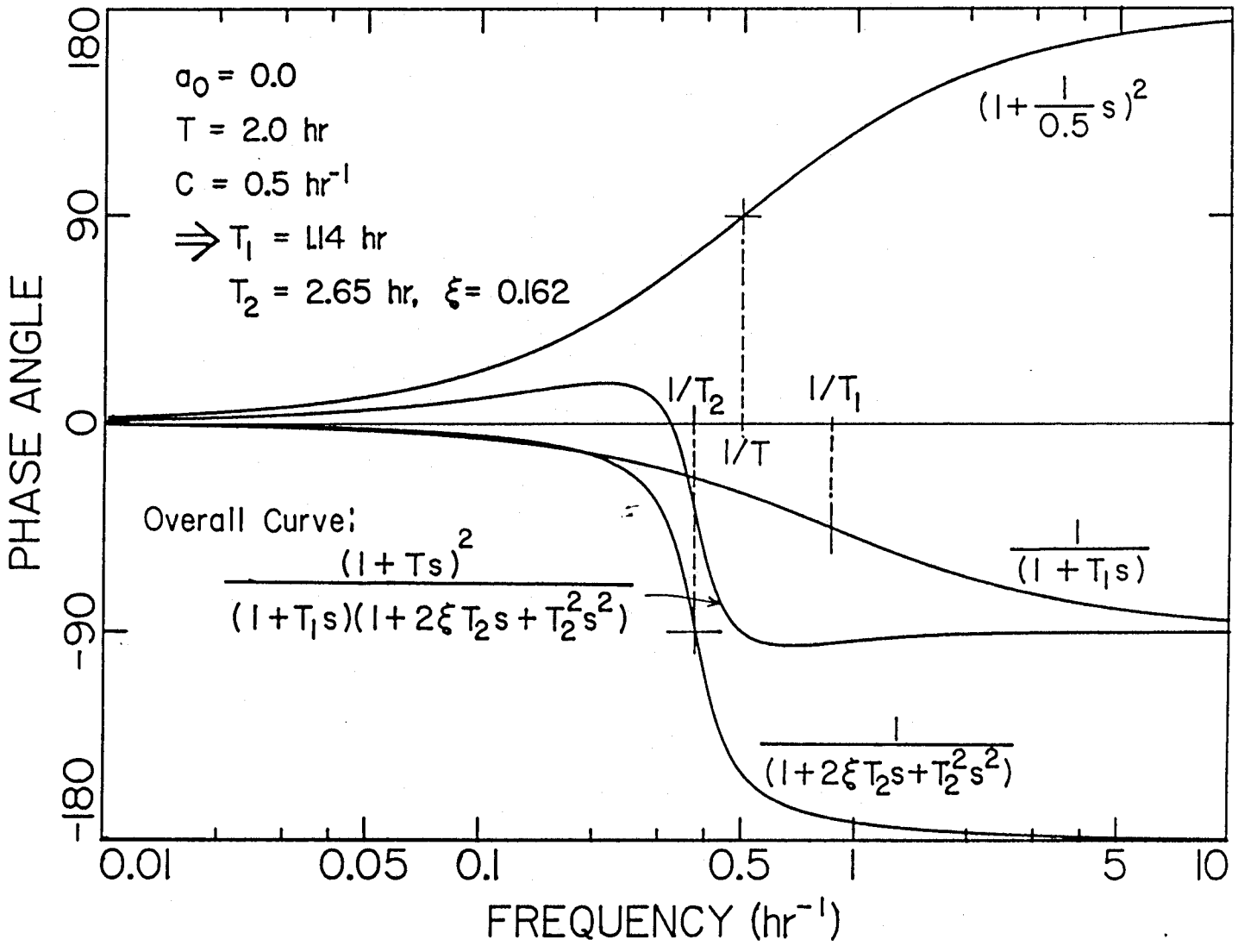


Figure 2.8.9. Construction of the overall biomass/D frequency response curve of the phase angle from the decomposed factors. ($a_0 = 0$, T=2.0 hr, C=0.5 hr^{-1}.)

right. Concurrently, T_2 in the factor $(1 + 2\xi T_2 s + T_2^2 s^2)$ in the denominator is also decreased, which forces the resonant frequency $\omega_{\max}$ toward the right. In addition, ξ becomes larger and depresses $A.R._{\max}$. Eventually a point is reached where all the factors work in concert to dissipate completely the maximum in the amplitude ratio. The biomass/D plot in Figure 2.8.6 shows that in the absence of time-lag effects, *i.e.*, when $T = 0$, the response becomes purely first-order with the $\omega_c = C$.

On the other hand, as T approaches the stability limit values of $T_{\text{crit}} = \frac{1}{C(0.5-a_0)}$ with $a_0 = 0$, the normalized transfer function $G_{\frac{X}{D}}(s)$ becomes:

$$G_{\frac{X}{D}}(s) = \frac{(1+Ts)^2}{(1+\frac{T}{2}s)(1+T^2s^2)}. \tag{2.8.66}$$

Thus, for $a_0 = 0$, the roots of the characteristic equation cross the imaginary axis at $\frac{1}{T_{\text{crit}}} = \frac{C}{2}$. On the Bode diagram, the appearance of this unstable steady-state is manifested in $A.R._{\max} \to \infty$ as $\omega \to \omega_{\text{crit}} = \frac{1}{T_{\text{crit}}}$ for $T_{\text{crit}} = \frac{2}{C}$. The biomass/$D$ plot in Figure 2.8.6 has ω_{crit}=0.25 hr^{-1}, because the value of C used in this simulation is 0.5 hr^{-1}. Although the overall order of the system of biomass/D is one, as shown by the asymptotic slope of -1, a maximum in the $A.R.$ can be induced with a nonzero T. The maximum amplitude ratios of biomass/D curves in Figure 2.8.6 are plotted in Figure 2.8.10 as a function of T for $a_0 = 0$. The corresponding maximum frequencies as a function of T are plotted in Figure 2.8.11, and changes in the values of T_2 and ξ as T is increased are demonstrated in Figure 2.8.12. It is apparent that it is most strongly influenced by the second-order factor of $(1 + 2\xi T_2 s + T_2^2 s^2)$ in the denominator of the transfer function.

Similar interesting behavior for nonzero T can also be observed in the phase angle diagram. There is a very sudden drop in the phase angle near $\omega_{\max}$. Note that at first glance, phase angle for biomass/D appears to become positive for large values of T. Since a positive phase angle means that disturbances are observed before the perturbation is imposed, such anticipatory behavior is obviously physically

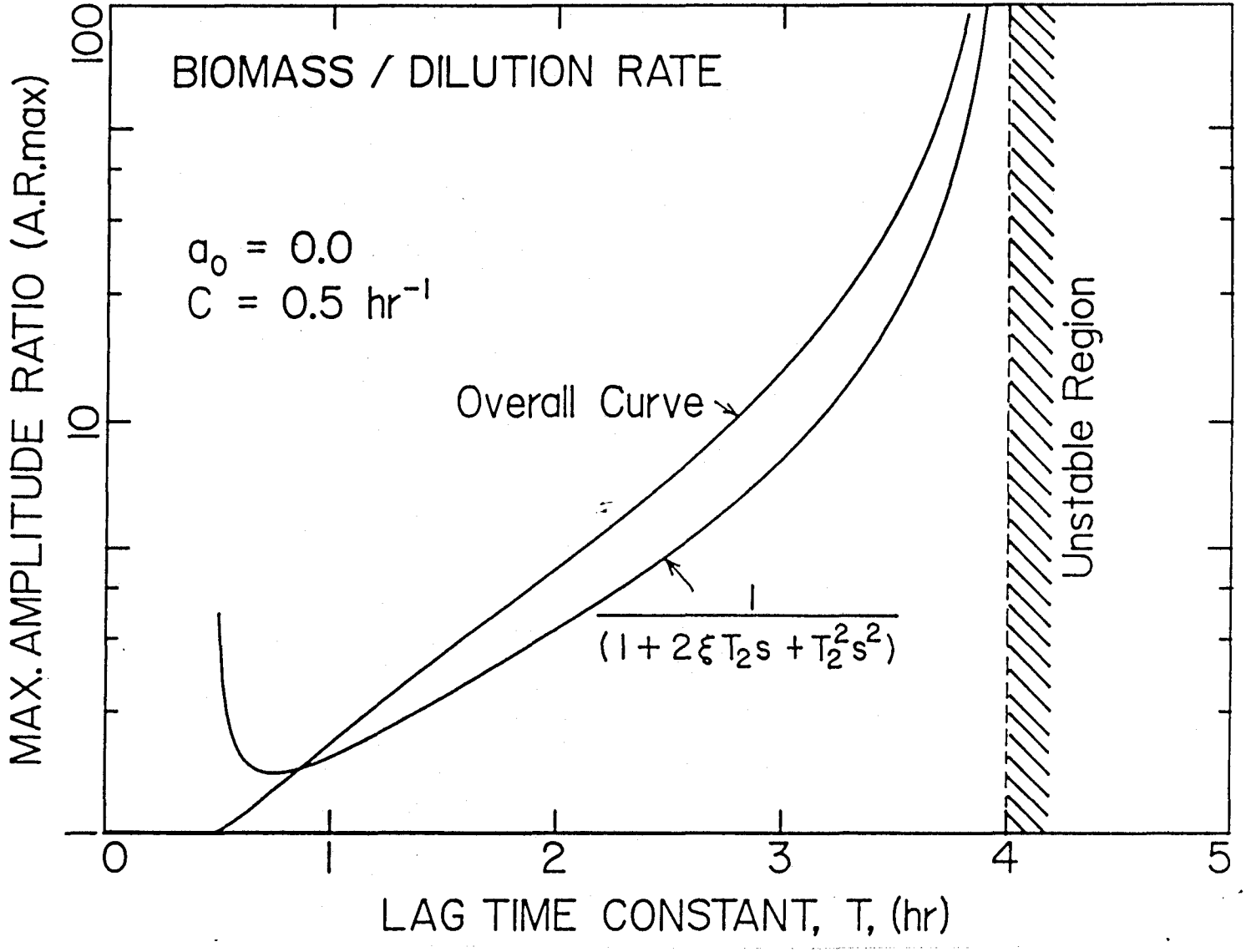


Figure 2.8.10. Maximum amplitude ratios for the factor $\frac{1}{1+2\xi T_2 s+T_2^2 s^2}$ and for the overall curve. ($a_0 = 0$, C=0.5 hr^{-1}.)

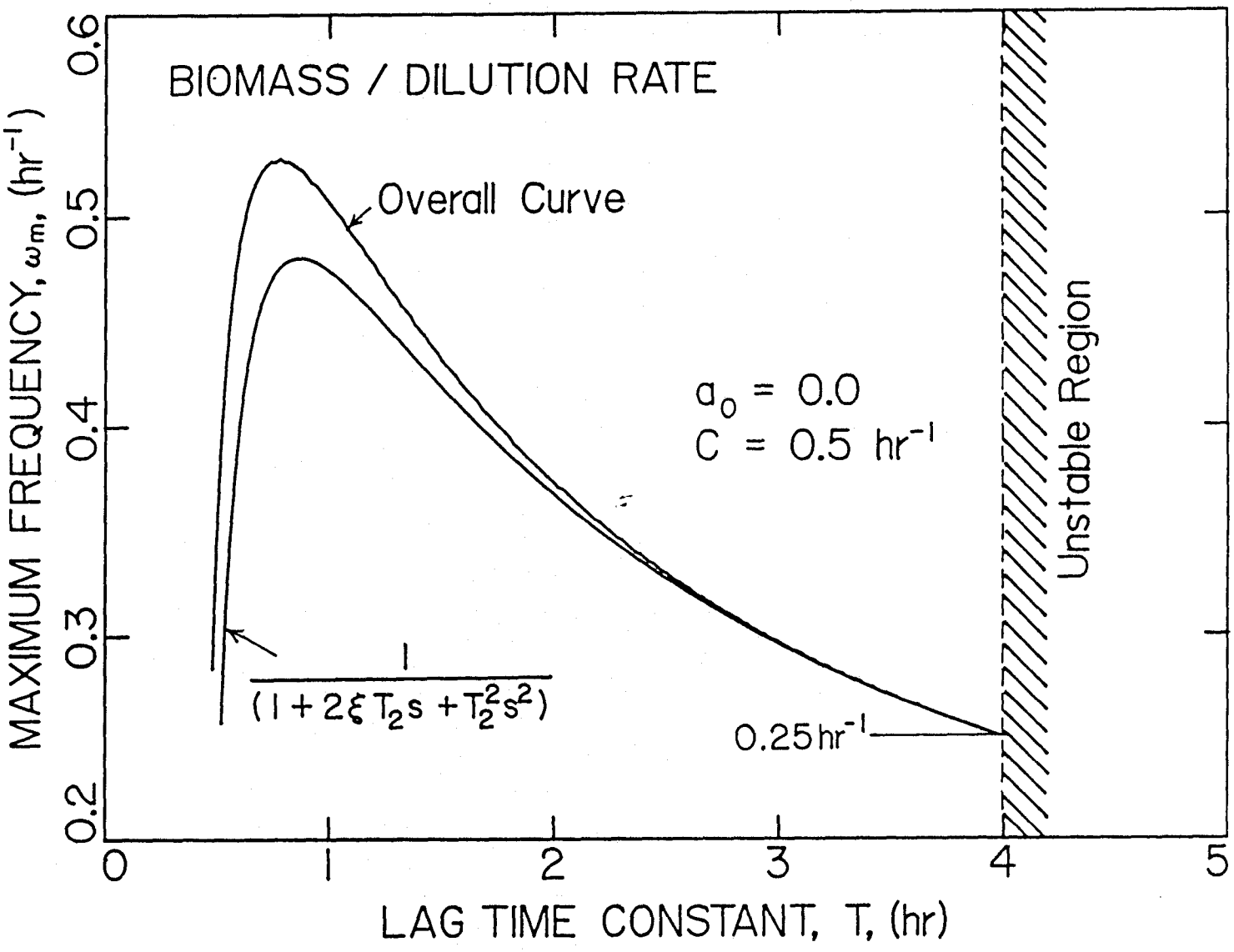


Figure 2.8.11. Frequencies at which maximum amplitudes are reached for the factor $\frac{1}{1+2\xi T_2 s+T_2^2 s^2}$ and for the overall curve. ($a_0 = 0$, C=0.5 hr^{-1}.)

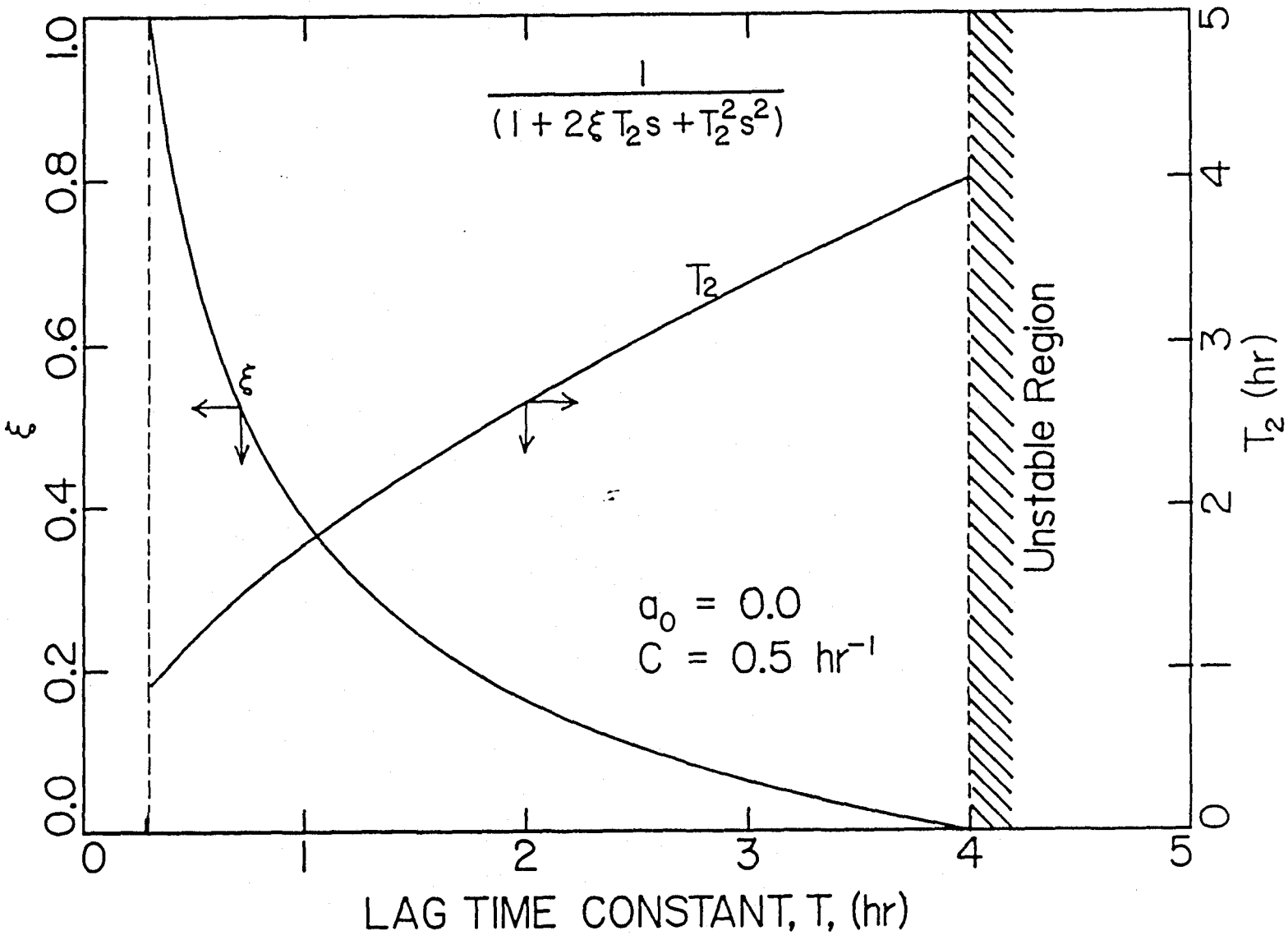


Figure 2.8.12. Changes in ξ and T_2 for the factor $\frac{1}{1+2\xi T_2 s+T_2^2 s^2}$ in the biomass / plot. ($a_0 = 0$, C=0.5 hr^{-1}.)

impossible. One should be reminded that the phase angle curve for biomass/D has been shifted up by 180° so that all angles are referenced to 0°as $\omega \rightarrow 0$. Finally, the phase angle for biomass/D decreases from 0° to -90°, which is consistent with the asymptotic slope of -1 in the corresponding $A.R.$ plot.

Effect of Lag Time Constant with a_0=0.4: The effects of the time-lag constant in a combined 0th and 1st-order kernel are studied with an example of $a_0 = 0.4$. The Bode diagrams for this case are displayed in Figures 2.8.13 and 2.8.14.

The asymptotic slopes for the biomass/D plots are -1. Aside from the generally slightly lower $A.R._{\max}$, there is basically not much difference in the biomass/D plots between these figures and the previous ones with a_0=0.

The asymptotic slopes of the Y/M plots are theoretically -1, but they are difficult to measure accurately unless ω is very large. The decomposed Bode diagrams for a_0=0.4 and T=1.0 hr are shown in Figures 2.8.15 and 2.8.16. As the frequency is increased, the $A.R.$ curve initially bends down toward an asymptote with a slope of -2 corresponding to the factor of $\frac{1}{(1+Ts)^2}$. The effect of $(1 + a_0 Ts)$ then becomes more pronounced, and the $A.R.$ curve is pulled up toward the high frequency asymptote with an overall slope of -1. In an attempt to gain more information from the Bode diagrams, the theoretical $A.R.$ plot for Y/M is derived from the transfer function $G_y = \frac{Y(s)}{M(s)}$.

$$A.R. = \frac{\sqrt{1 + a_0^2 T^2 \omega^2}}{(1 + T^2 \omega^2)} \tag{2.8.67}$$

Slope of the Bode diagram is:

$$\text{Slope} = \frac{d \log A.R.}{d \log \omega} = \omega \frac{d \log A.R.}{d\omega} = -\frac{(a_0^2 T^2 \omega^2 + 2 - a_0^2) T^2 \omega^2}{(1 + a_0^2 T^2 \omega^2)(1 + T^2 \omega^2)} \tag{2.8.68}$$

The slope of the curve is -1 at:

$$\omega_t = \frac{1}{T\sqrt{1 - 2a_0^2}} \qquad \text{for } a_0 < \frac{1}{\sqrt{2}} = 0.707. \tag{2.8.69}$$

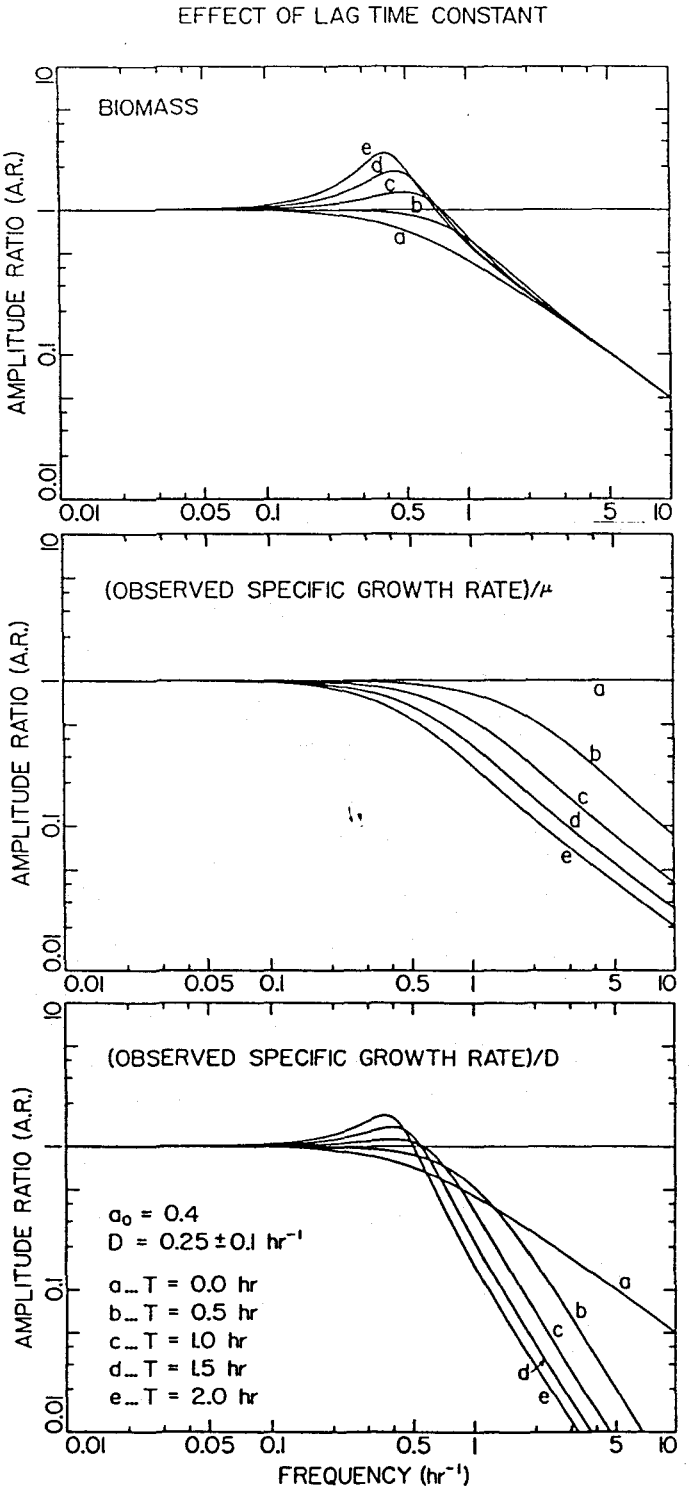


Figure 2.8.13. Effect of the lag time constant, T, on the amplitude ratios for linearized chemostat state equations subject to a sinusoidal dilution rate forcing and a combined 0th-order and 1st-order time-lag kernel. (a_0=0.4, C=0.5 hr^{-1}.)

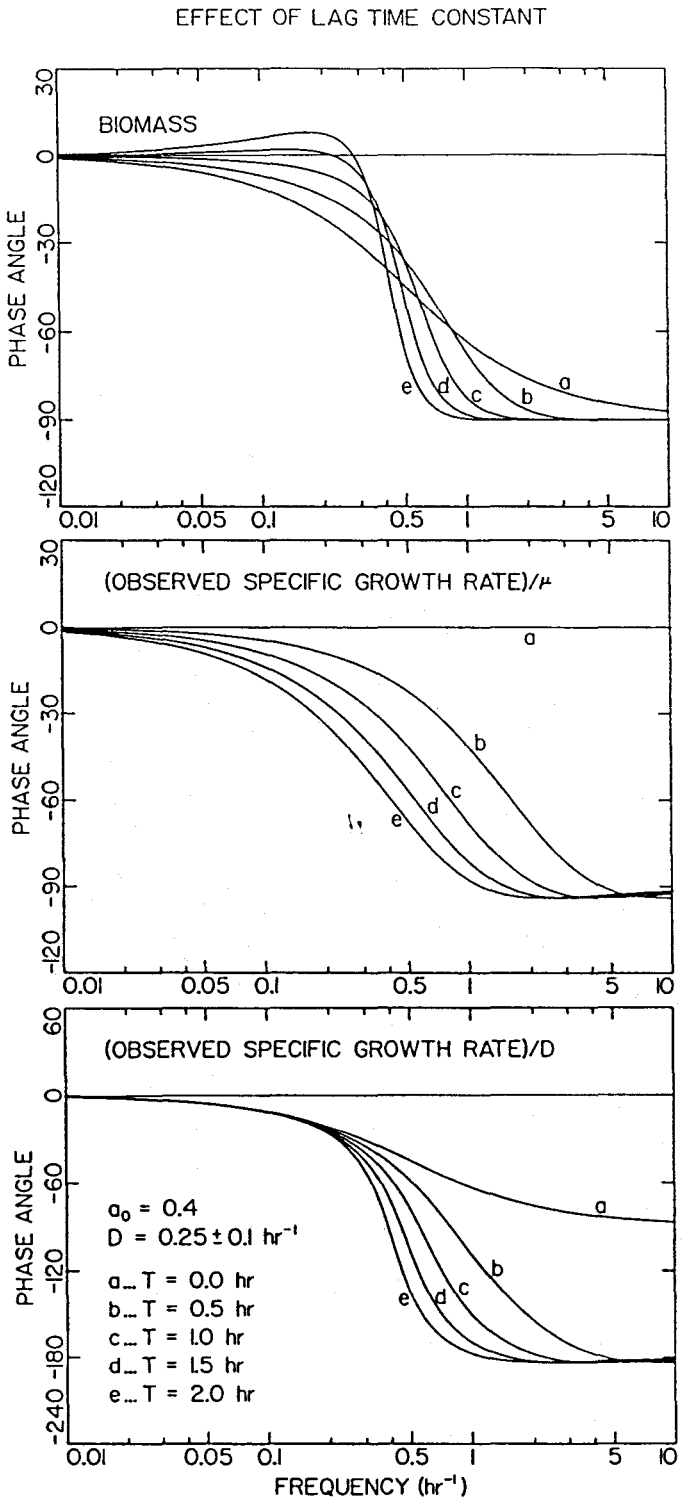


Figure 2.8.14. Effect of the lag time constant, T, on the phase angles for linearized chemostat state equations subject to a sinusoidal dilution rate forcing and a combined 0th-order and 1st-order time-lag kernel. (a_0=0.4, C=0.5 hr^{-1}.)

A slope of -1 does not exist for $a_0 \geq 0.707$. The tangent line passing through this point intersects $A.R. = 1$ at the corner frequency of:

$$\omega_{-1} = \frac{1}{2T\sqrt{1-a_0^2}} \quad \text{for } a_0 <= 0.707. \tag{2.8.70}$$

For a_0=0.4, these corner frequencies are $\frac{1}{1.833T}$, *e.g.*, ω_{-1}=1.091, 0.546, 0.364, and 0.273 hr^{-1} for T=0.5, 1.0, 1.5, and 2.0 hr, respectively. Furthermore, the high frequency asymptote is described by:

$$\log A.R._{\infty} = \begin{cases} \log \frac{a_0}{T} - \log \omega & \text{for } a_0 \neq 0 \\ \log \frac{1}{T^2} - 2\log \omega & \text{for } a_0 = 0\,. \end{cases} \tag{2.8.71}$$

The intercept of the high frequency asymptote with $A.R. = 1$, thus, occurs at:

$$\omega_c = \begin{cases} \frac{a_0}{T} & \text{for } a_0 \neq 0 \\ \frac{1}{T} & \text{for } a_0 = 0\,. \end{cases} \tag{2.8.72}$$

Because of the same interactions between the divided components, the phase angle curve initially sets out toward -180° but is eventually pulled back to -90°. Based on the denominator of the transfer function, the system order of biomass/D is technically 2, but the system order based on the high frequency asymptote behaviors of $A.R.$ and phase angle is 1. Because of this complication, the asymptotic $A.R.$ slope is difficult to obtain accurately from the Y/D plot.

Effect of a_0 with T=1.0 hr: The effects of the weight distribution between the 0th and 1st-order kernel base functions are studied in Figures 2.8.17 and 2.8.18. This example assumes T=1.0 hr. One of the significant features of these figures is that the asymptotic slopes for the Y/M plot are one for $a_0 > 0$ but shifts to two for $a_0 = 0$. As indicated by Equation (2.8.72), for T=1 hr, the asymptotes cross $A.R. = 1$ at a_0 for $a_0 \neq 0$ and at 1 for $a_0 = 0$. The $A.R.$ curves fan out as ω is increased. At a decade right of $\frac{1}{T}$, *i.e.*, $\omega = \frac{10}{T}$, one obtains $A.R. = \frac{\sqrt{1+100a_0}}{101} \approx 0.1a_0$.

Similarly, the asymptotic phase angles for the Y/M plot are -90° for $a_0 > 0$ but drops to -180° for $a_0 = 0$. Thus, it can be used to determine the presence of the

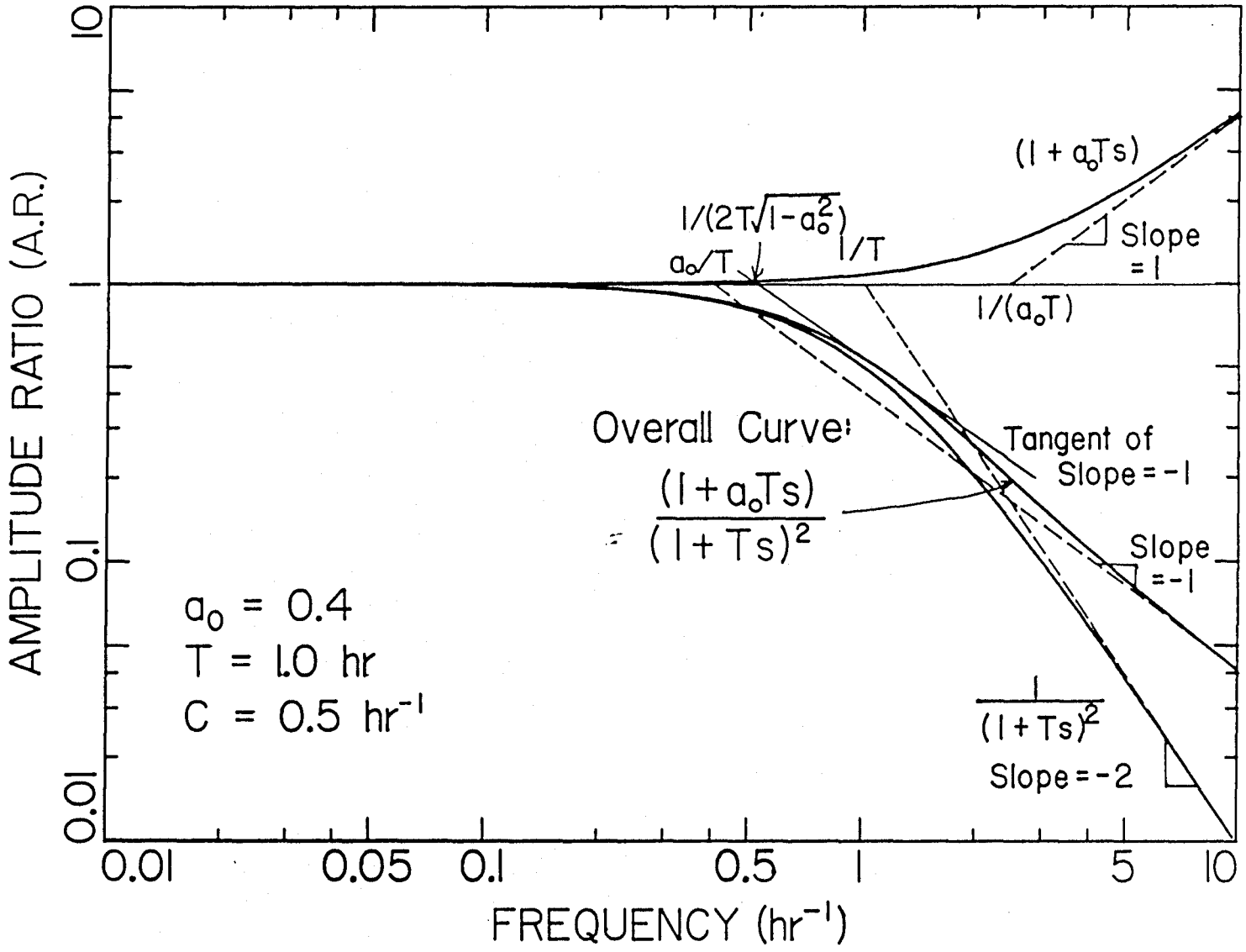


Figure 2.8.15. Construction of the overall Y/M frequency response curve of the amplitude ratio from the decomposed factors. ($a_0 = 0.4$, T=1.0 hr, C=0.5 hr^{-1}.)

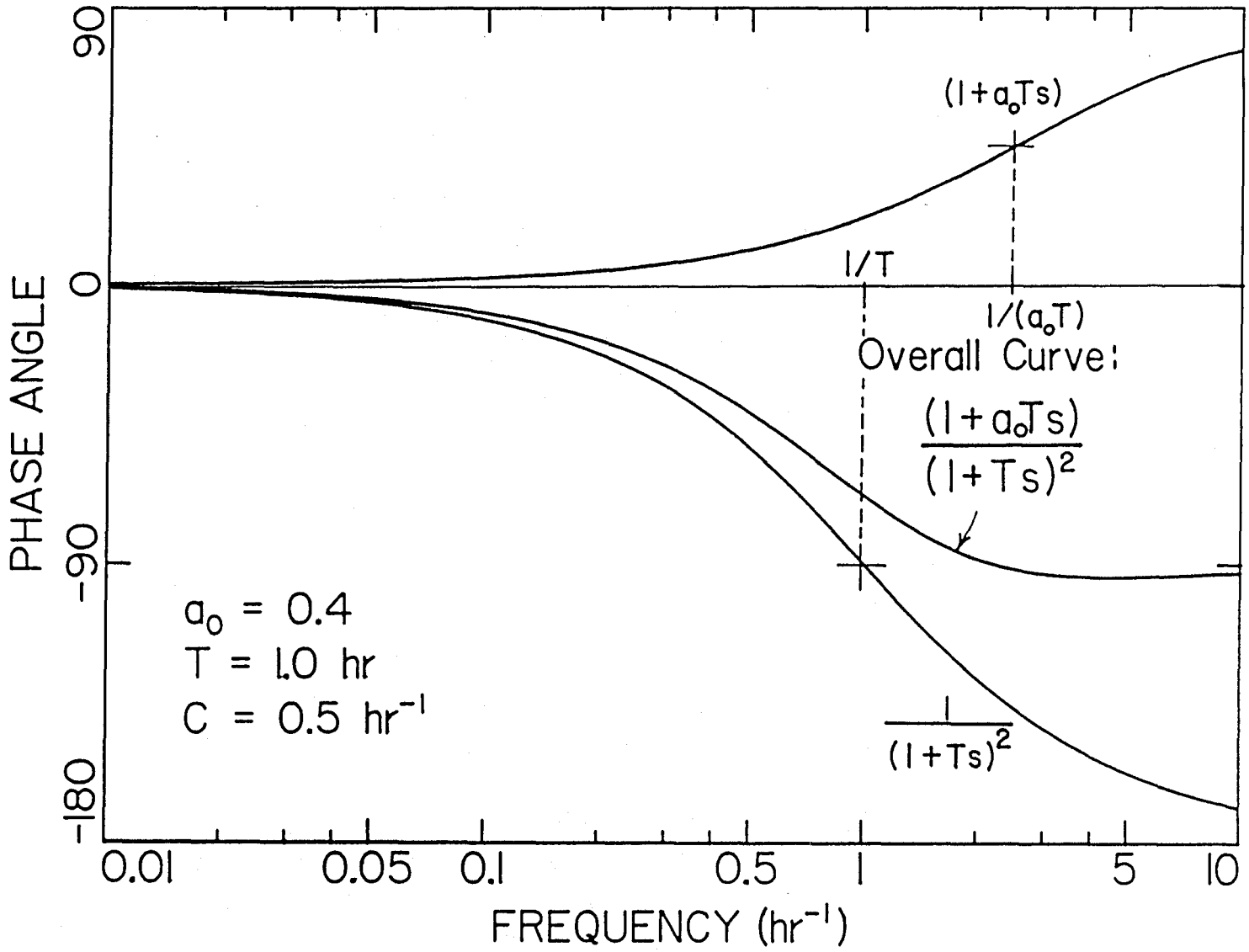


Figure 2.8.16. Construction of the overall Y/M frequency response curve of the phase angle from the decomposed factors. ($a_0 = 0.4$, T=1.0 hr, C=0.5 hr^{-1}.)

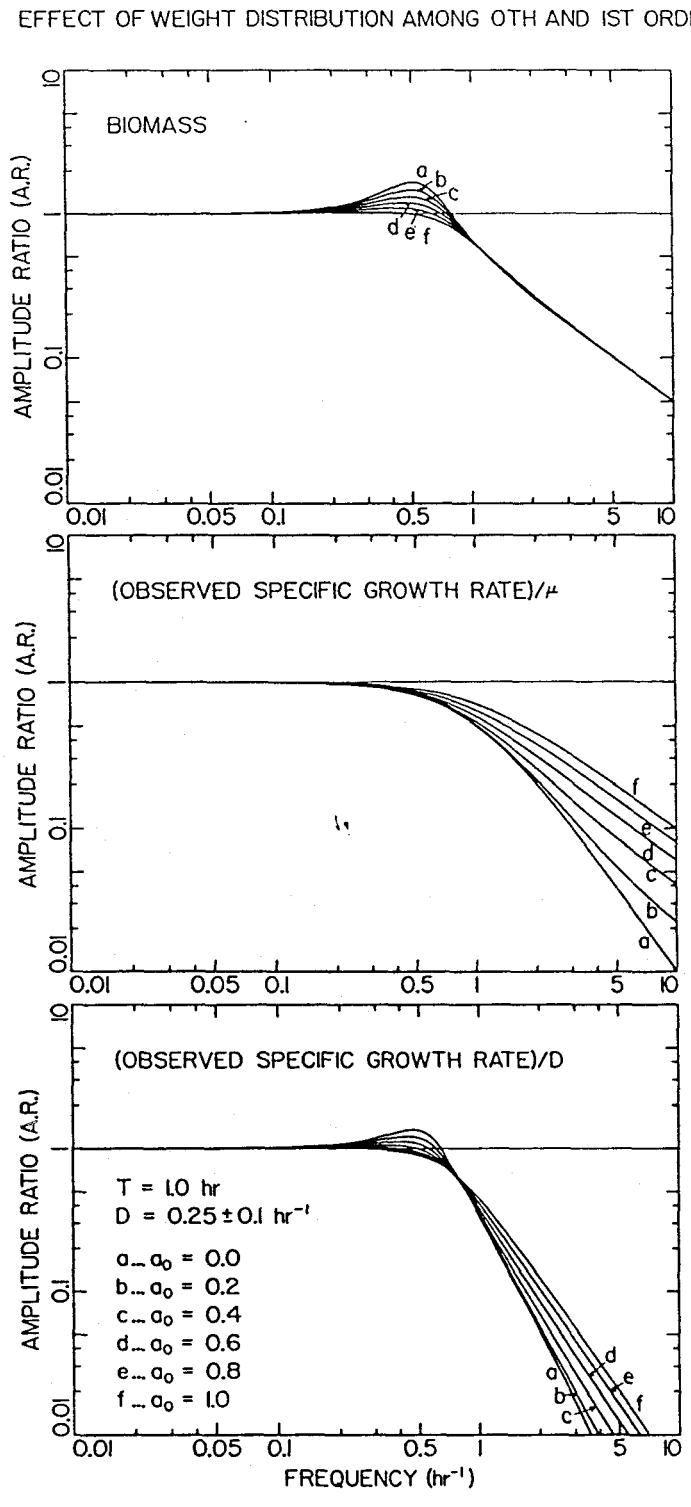


Figure 2.8.17. Effect of the 0th-order fraction in the kernel function, a_0, on the amplitude ratios for linearized chemostat state equations subject to a sinusoidal dilution rate forcing. (T=1.0 hr, C=0.5 hr^{-1}.)

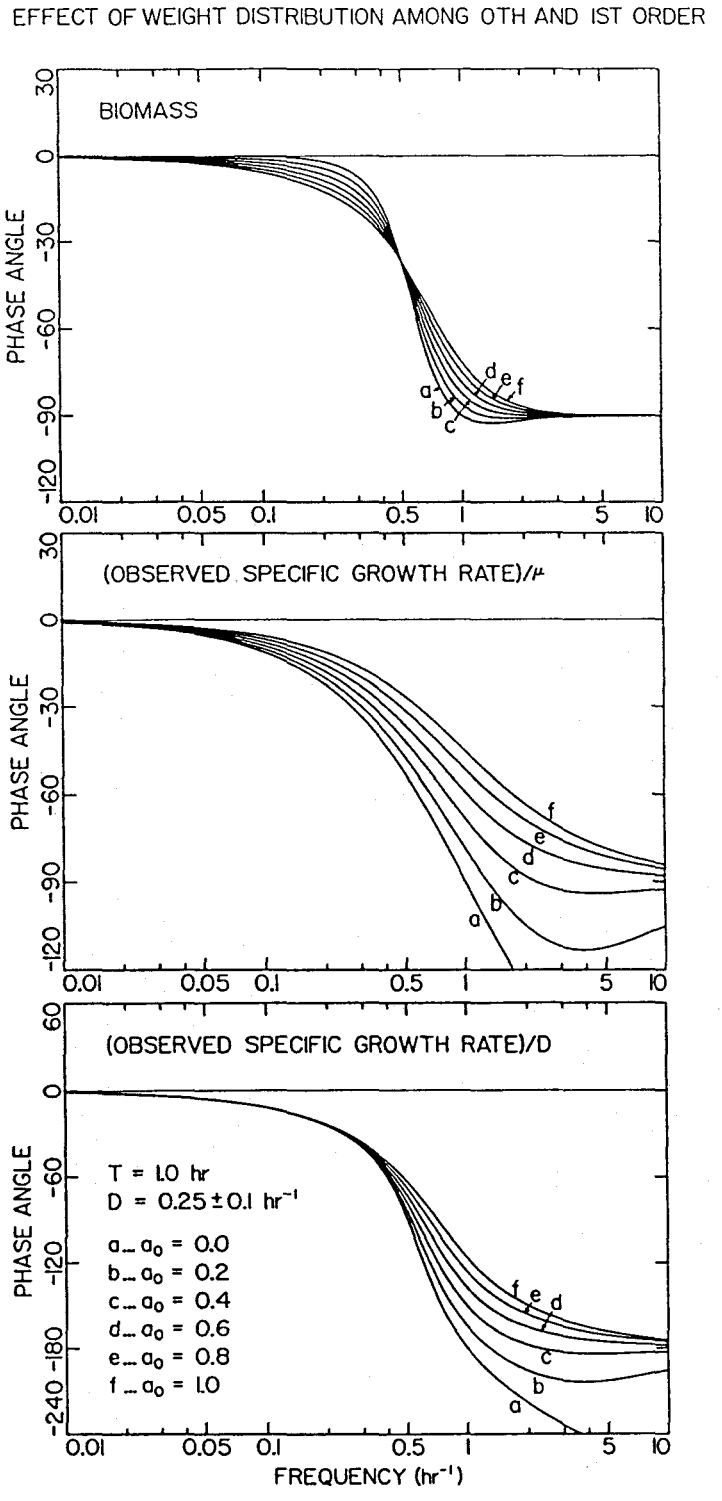


Figure 2.8.18. Effect of the 0th-order fraction in the kernel function, a_0, on the phase angles for linearized chemostat state equations subject to a sinusoidal dilution rate forcing. (T=1.0 hr, C=0.5 hr^{-1}.)

1st-order kernel component. Note that there exist minima in the phase angle curves for $0 < a_0 < 0.5$. The frequency at which the minimum occurs can be calculated explicitly.

$$\text{Phase Angle} = \phi = 2\tan^{-1}(-T\omega) + \tan^{-1}(a_0 T\omega) \tag{2.8.73}$$

$$\frac{d\phi}{d\omega_{\min}} = 0 \qquad \Longrightarrow \omega_{\min} = \frac{1}{T}\sqrt{\frac{2-a_0}{a_0(1-2a_0)}} \quad \text{for } 0 < a_0 < 0.5 \tag{2.8.74}$$

Thus, for a_0=0.2 and 0.4, the above equation gives $\omega_{\min} = \frac{1}{T}3.873$ and $\frac{1}{T}4.472$, respectively.

Effect of Stability Variable with a_0=0.4 and T=1.0 hr: In the linearized chemostat state equations, different operating conditions and model parameters can all be conveniently grouped into a single stability variable C. The effects of the stability variable on the frequency response are shown in Figures 2.8.19 and 2.8.20. The most interesting feature of the $A.R.$ plot for biomass/D is the sharp increase in $A.R._{\max}$ as C is increased. Note that with a_0=0.4 and T=1.0 hr, the critical value of C is 10 hr^{-1}, at which point $A.R._{\max} \to \infty$. This limiting $\omega_{\max}$ is $\frac{1}{T\sqrt{1-2a_0}}$=2.236 hr^{-1}. The plot of $A.R._{\max}$ as a function of C is shown in Figure 2.8.21, $\omega_{\max}$ is shown in Figure 2.8.22, and changes in T_2 and ξ in the transfer function decomposition are plotted in Figure 2.8.23. Note that in contrast to changes in T that is shown in Figure 2.8.11, $\omega_{\max}$ increases along with C. the direction of changes in T_2 shown in Figure 2.8.23 is also the opposite of that shown in Figure 2.8.12.

Dynamic Simulation of Sinusoidal Dilution Rate Forcing:

It is emphasized that the above frequency analysis is based strictly on transfer functions that are derived from linearized state equations. Because the state equations are actually nonlinear, the response of the system subjected to a similar sinusoidal dilution rate forcing is further simulated more realistically by directly integrating the nonlinear dynamic equations. The basic model parameters used in

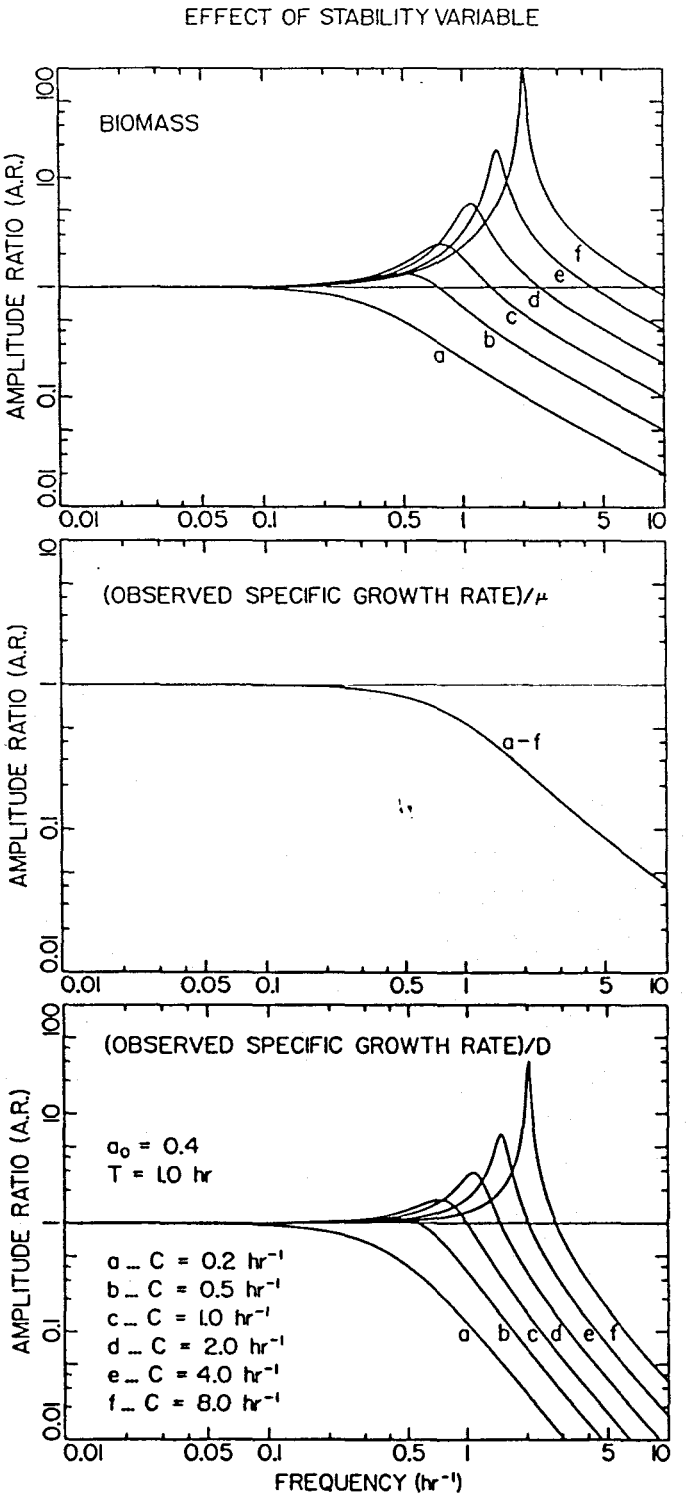


Figure 2.8.19. Effect of the stability variable, C, on the amplitude ratios for linearized chemostat state equations subject to a sinusoidal dilution rate forcing. (a_0=0.4, T=1.0 hr.)

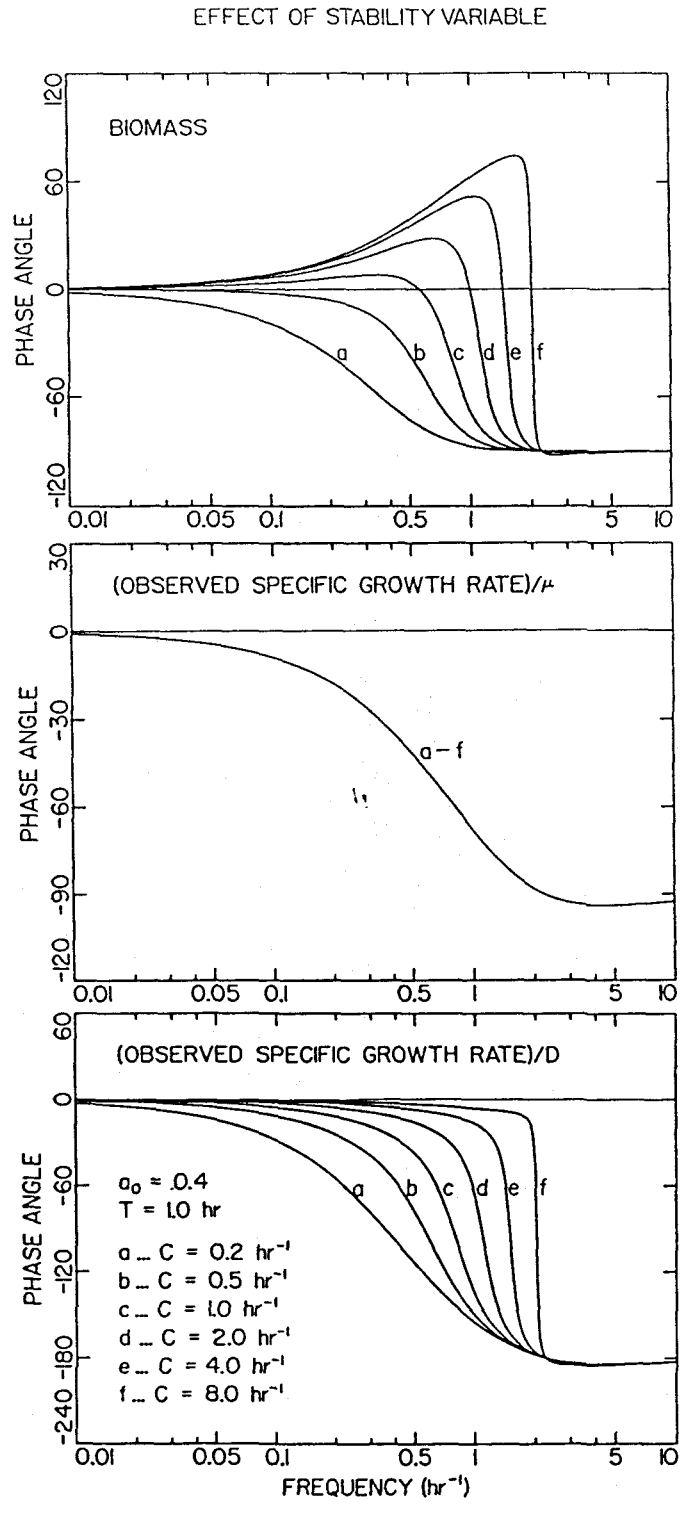


Figure 2.8.20. Effect of the stability variable, C, on the phase angles for linearized chemostat state equations subject to a sinusoidal dilution rate forcing. (a_0=0.4, T=1.0 hr.) state equations. (a_0=0.4, T=1.0 hr.)

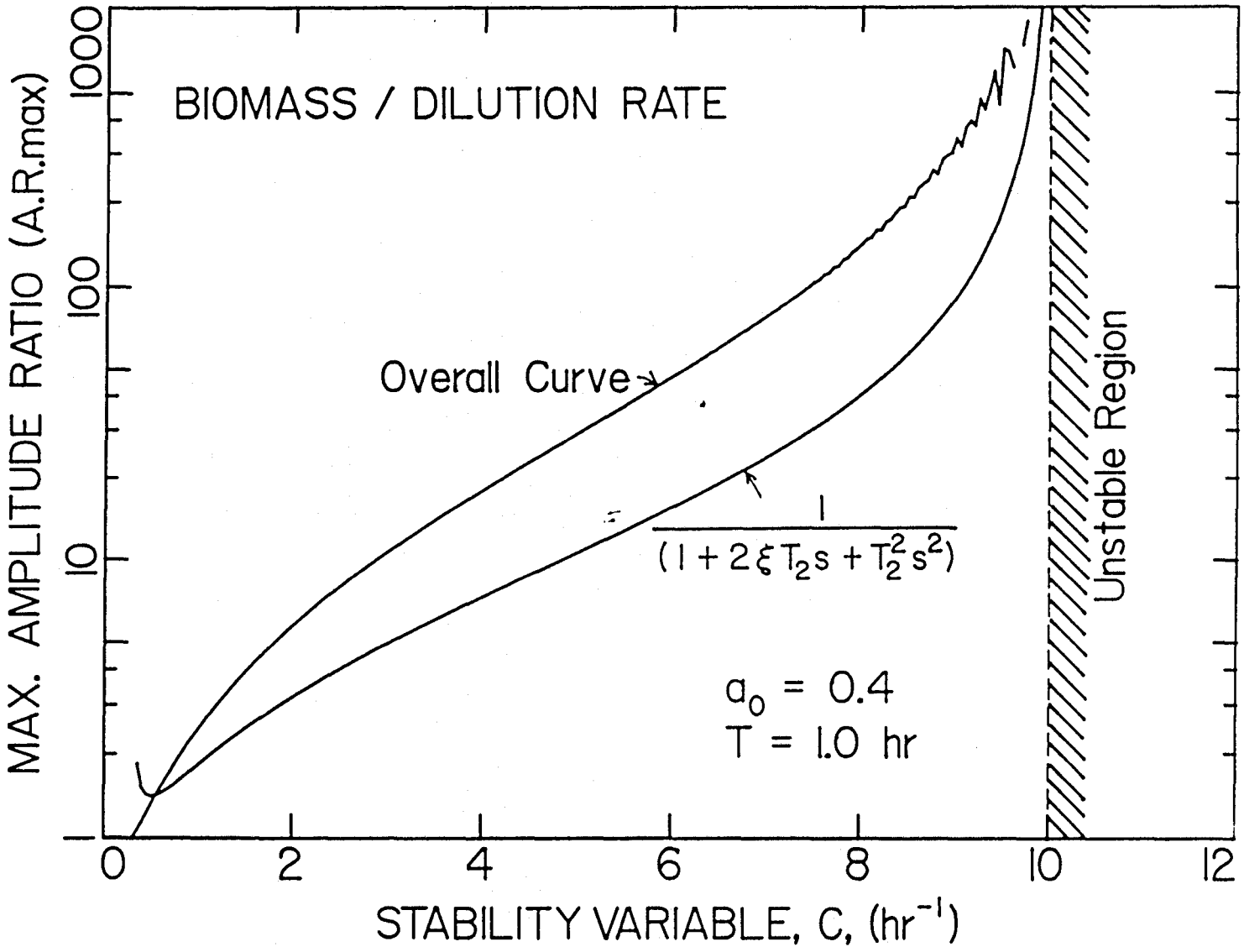


Figure 2.8.21. The maximum amplitude ratios for the factor $\frac{1}{1+2\xi T_2 s+T_2^2 s^2}$ and for the overall curve. ($a_0 = 0.4$, T=1.0 hr.)

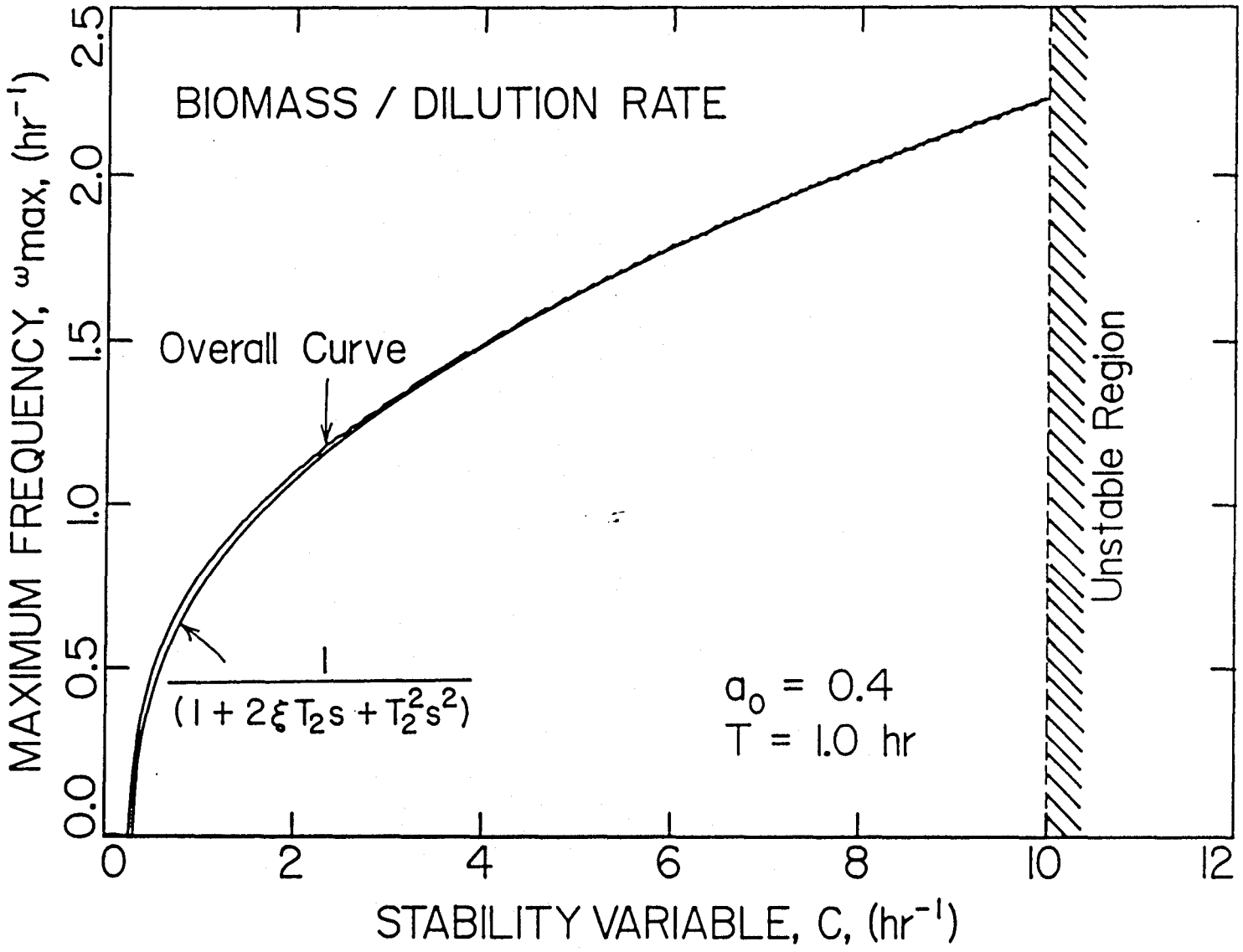


Figure 2.8.22. Frequencies at which maximum amplitudes are reached for the factor $\frac{1}{1+2\xi T_2 s+T_2^2 s^2}$ and for the overall curve. ($a_0 = 0.4$, T=1.0 hr.)

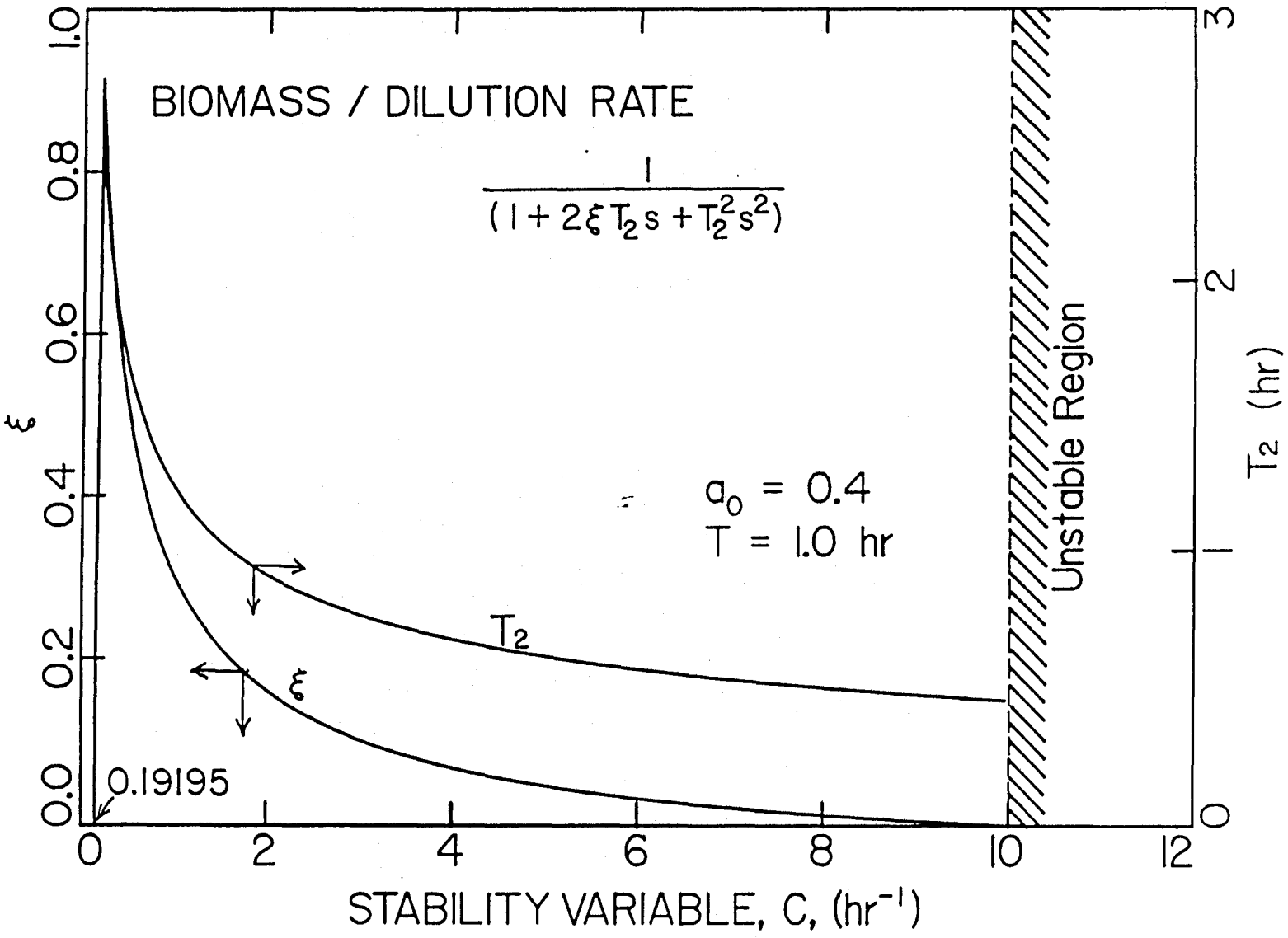


Figure 2.8.23. Changes in ξ and T_2 for the factor $\frac{1}{1+2\xi T_2 s+T_2^2 s^2}$ in the biomass / D plot. ($a_0 = 0.4$, T=1.0 hr.)

this simulation are: μ_m=0.5 hr, K_s=1 g/l, Y_s=0.5, and S_f=5 g/l. The mean dilution rate is 0.25 hr^{-1}, and its amplitude of oscillation is 0.1 hr^{-1}. These values give rise to a steady-state biomass concentration of 2 g/l and a steady-state substrate concentration of 1 g/l at the mean dilution rate. Thus, based on the linearized chemostat equations, the stability variable C 0.5 hr^{-1} is the same as that used consistently in generating the previous Bode diagrams based on a strictly linearized analysis. As before, the effects of the lag time constant, T, the relative weight of the kernel, a_0, the mean dilution rate and the forcing amplitude, A, are studied from this set of simulations, and those effects associated with nonlinearities will be identified.

The Bode diagrams presented hereafter are the result of dynamical system simulation by integrating the chemostat equations for at least five cycles. Once the initial transient component vanishes and the "steady-state" solution is established, the time and magnitudes that correspond to the maxima and minima in the biomass concentration, the substrate concentration, the intrinsic specific growth rate, and the observed specific growth rates are recorded for each frequency. Because of the nonlinear relationships, responses of these variables are not purely sinusoidal, although the forcing dilution rate is strictly sinusoidal.

The results from one of the dynamic simulations used to construct the Bode diagrams are shown in Figure 2.8.24 for the biomass concentrations. The kernel parameters used to generate these plots are a_0=0.4 and T=1.0 hr, and the period of dilution rate oscillation is 20 hr with an amplitude of 0.1 hr^{-1} centered around 0.25 hr^{-1}. Figure 2.8.24 shows that even without the time-lag, the biomass concentration as a function of time is not sinusoidal; the entire curve is shifted somewhat downward from the calculated steady-state concentration, based on the mean dilution rate. With time-lag, the curve turns around its minima at a much slower

rate than it does around its maxima. There also exist some inflection points in the response curve during the downward cycle. In this example, the amplitude is amplified slightly, and the biomass response with a time-lag is generally ahead of that without a time-lag.

Figure 2.8.25 shows the various specific growth rates for the above example. Note that the observed specific growth rate is consistently behind the corresponding intrinsic specific growth rate with time-lag. The oscillation amplitude of the observed specific growth rate is smaller than that of the intrinsic specific growth rate, which is in turn even smaller than the specific growth rate obtainable with no time-lag. This successive decrease in the amplitude is expected because the time-lag relationship is mathematically equivalent to a first-order low pass filter for a 0th-order kernel function. Furthermore, the oscillations in both $y(t)$ and $\mu(t)$ are larger than that in the dilution rate, which is not possible for a chemostat system with no time-lag.

It should be warned that a direct comparison of the phase angles between the specific growth rate with time-lag and that with no time-lag is not valid. For instance, Figure 2.8.25 shows that during the upward cycle, the specific growth rate in the absence of time-lag appears to be ahead of that with a time-lag. It must be stressed that the correct reference curve must be selected in order for the comparison to be valid. In this case, the specific growth rate with no lag should be referenced to the corresponding substrate response curve, and phase angle calculation of the intrinsic and observed specific growth rates should also be based on similar variables.

Because the rate of change in the biomass concentration depends on $y(t) - D(t)$, the biomass concentration curve changes its direction at the point where $y(t)$ and $D(t)$ cross each other. In other words, a local maximum in the biomass concentration corresponds to where $y(t)$ dips below $D(T)$, and a minimum point

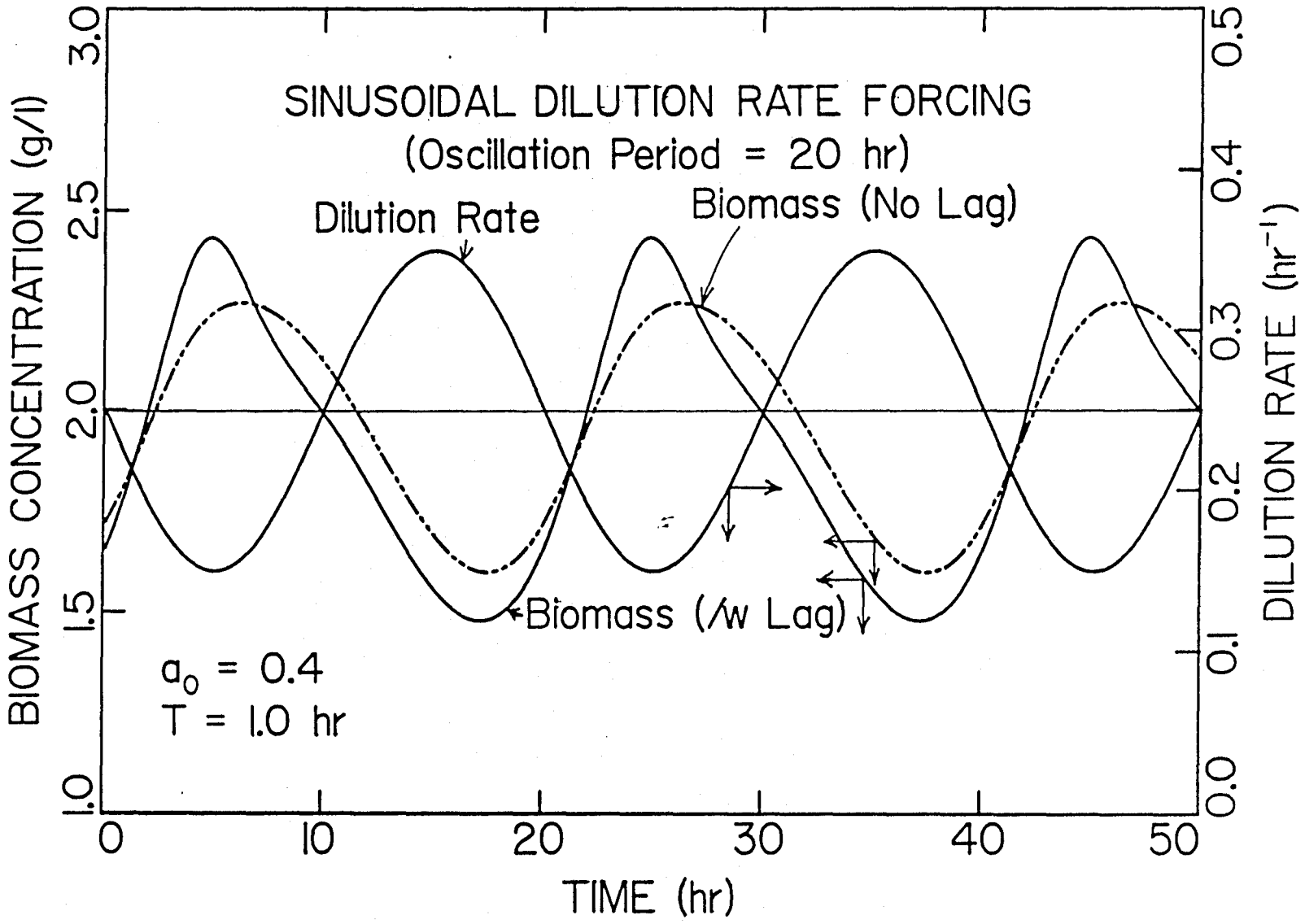


Figure 2.8.24. Response of the biomass concentration to the sinusoidal dilution rate forcing. (D=0.25±0.1 hr^{-1}, period=20 hr, a_0=0.4, T=1.0 hr.)

is reached when $y(t)$ overtakes $D(t)$. The logarithmic biomass concentration is the same as the area between the two curves $y(t)$ and $D(t)$.

$$\ln \frac{x(t)}{x(0)} = \int_0^t y(t') - D(t')dt' \tag{2.8.75}$$

In comparison to the biomass concentration, nonlinearities are more pronounced for the specific growth rate. The reason for this is that the model variables in effect are subjected to double nonlinearities. The first set arises in the state dynamic equations and yield distortions in the state variables, *e.g.*, the substrate concentrations. As the model variables, *e.g.*, the intrinsic specific growth rate, are calculated by another nonlinear constitutive relationship, they become further distorted.

The dynamic trajectories of this example are shown in Figure 2.8.26 in a three-dimensional phase plane. Of course, being on paper in reality, it is actually a two-dimensional representation thereof. The approach of the initial state to the "steady-state" periodic trajectory that appears as a loop can be seen. The entire curve lies within the positive octant. Because an nth-order time-lag kernel adds $n+1$ orders to the system dynamics, this example, based on biomass and substrate state equations, is effectively a 4th order system. Therefore, it is only a projection of a four-dimensional curve into a three-dimensional space.

Because of the reduction in the dimensional space during plotting, foldings that are not possible in a purely two-dimensional dynamic system can often appear for a certain range of model parameters. One such example is created by increasing the oscillation period of the dilution rate forcing to 42 hr, while all other parameters remain unchanged from the previous example.

The biomass concentration shown as a function of time in Figure 2.8.27 has developed small notches at its maxima. This is due to the fact that $y(t)$ and $D(t)$

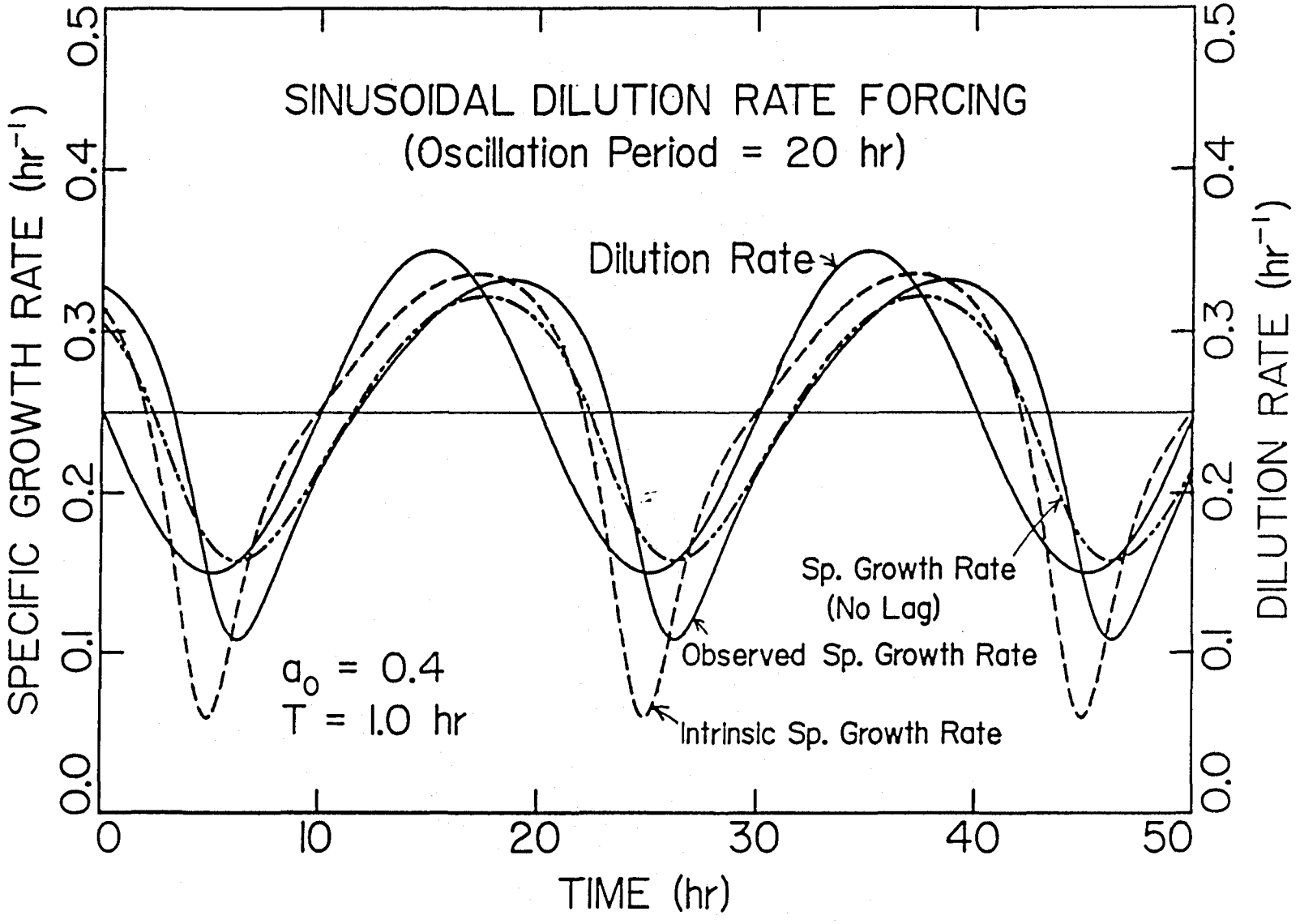


Figure 2.8.25. Response of the specific growth rates to the sinusoidal dilution rate forcing. (D=0.25±0.1 hr^{-1}, period=20 hr, a_0=0.4, T=1.0 hr.)

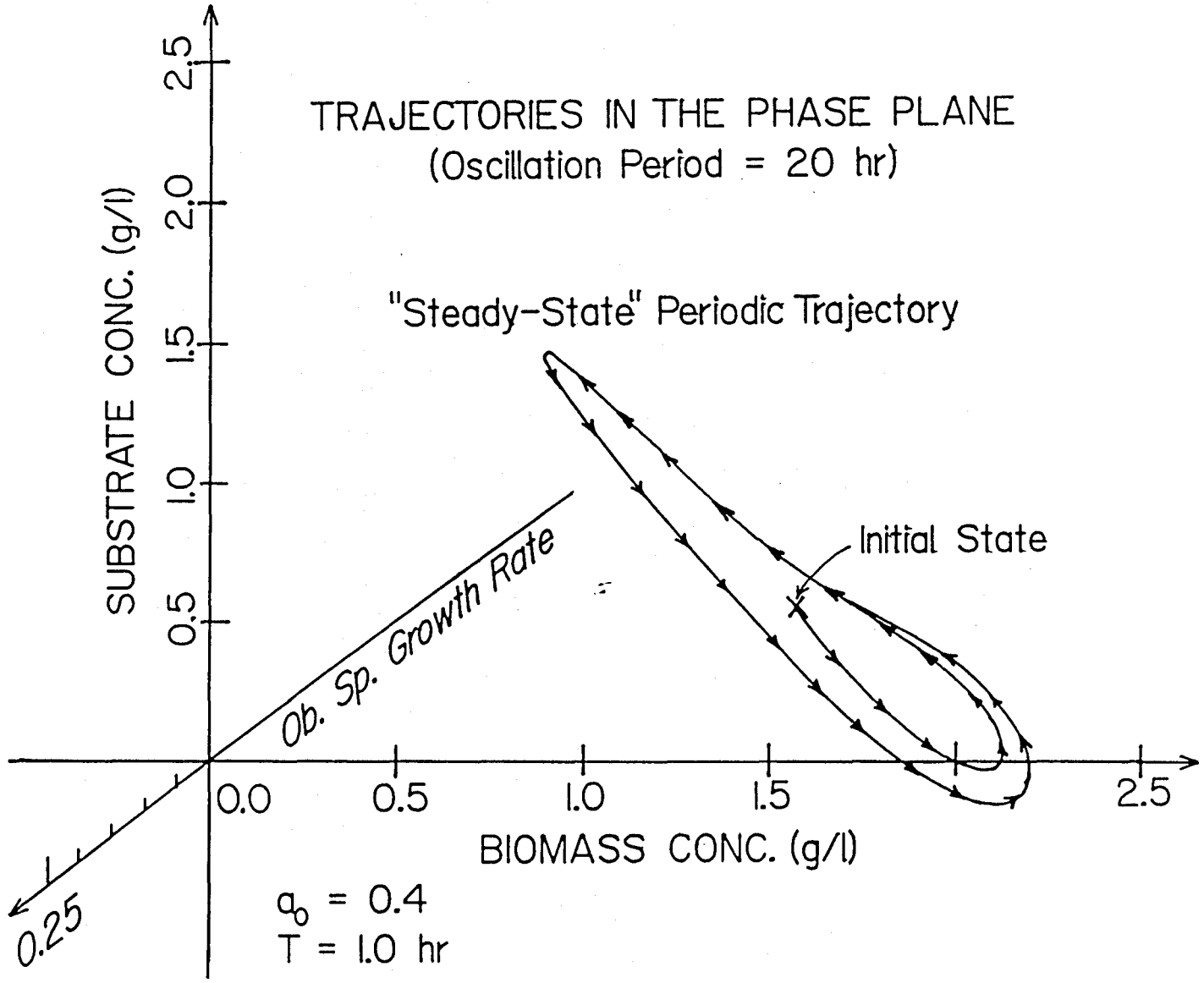


Figure 2.8.26. two-dimensional representation of the state trajectory in a three-dimensional phase plane for a chemostat subject to the sinusoidal dilution rate forcing. (D=0.25±0.1 hr^{-1}, period=20 hr, a_0=0.4, T=1.0 hr.)

cross each other three times near their minima, as shown in Figure 2.8.28. This intriguing behavior can be explained as follows. First, the observed specific growth rate remains high during the downward cycle of the dilution rate due to the presence of time-lag. A high specific growth rate fast depletes the substrate and causes the substrate concentration to overshoot slightly. The substrate concentration, in turn, translates into the intrinsic specific growth rate. Responding to the intrinsic growth rate with time-lag, the observed specific is brought down and crosses below the dilution rate curve. At this point, the biomass concentration reaches a maximum, and the substrate concentration reaches a minimum and swings back up. The rate of decrease in the observed specific growth rate is sharply reduced, and it again crosses the dilution rate curve, this time going above. Such micro-oscillation of the observed specific growth rate around the dilution rate can occur for a range of forcing periods. The overall effect is the appearance of local minima and maxima, or notches/indentations, in both the biomass and substrate concentration curves. Because of the longer oscillation period in this example, the system can better adapt to the slowly changing operating conditions, and the distortions in the specific growth rate curves become smaller. Similarly, the amplitude ratios of the specific growth rates are not as great as in the previous example.

The state trajectories for this system are plotted in Figure 2.8.29. The trajectories cross themselves at certain points in the projected space. In a fully dimensioned space, state trajectories cannot cross themselves, for doing so implies that there exists more than one solution for the same initial value problem. The well known Existence and Uniqueness Theorem in ordinary differential equations denies the existence of such cross points. Finally, the notch is seen in the lower right hand side of the "steady-state" loop.

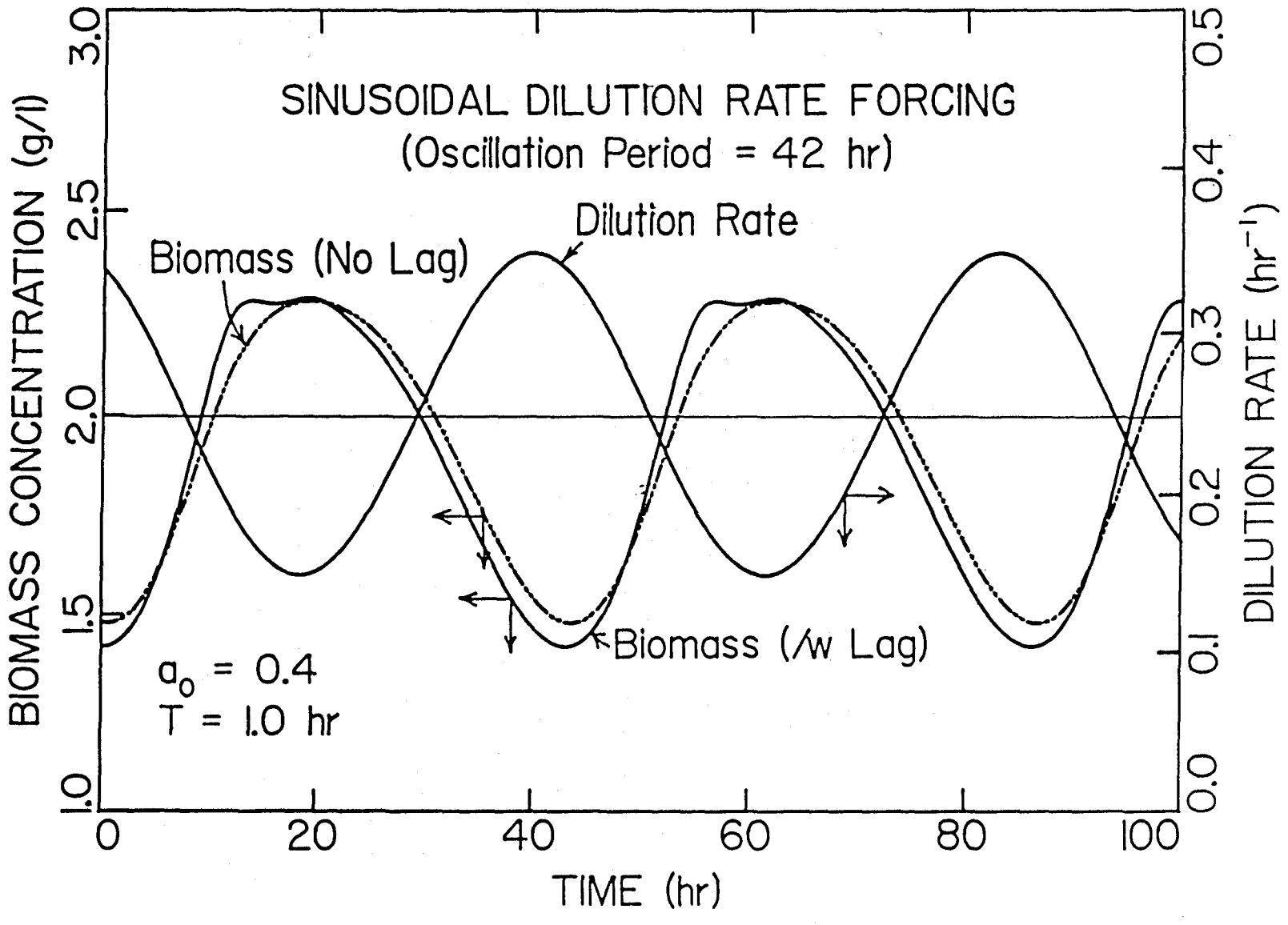


Figure 2.8.27. Response of the biomass concentration to the sinusoidal dilution rate forcing. (D=0.25±0.1 hr^{-1}, period=42 hr, a_0=0.4, T=1.0 hr.)

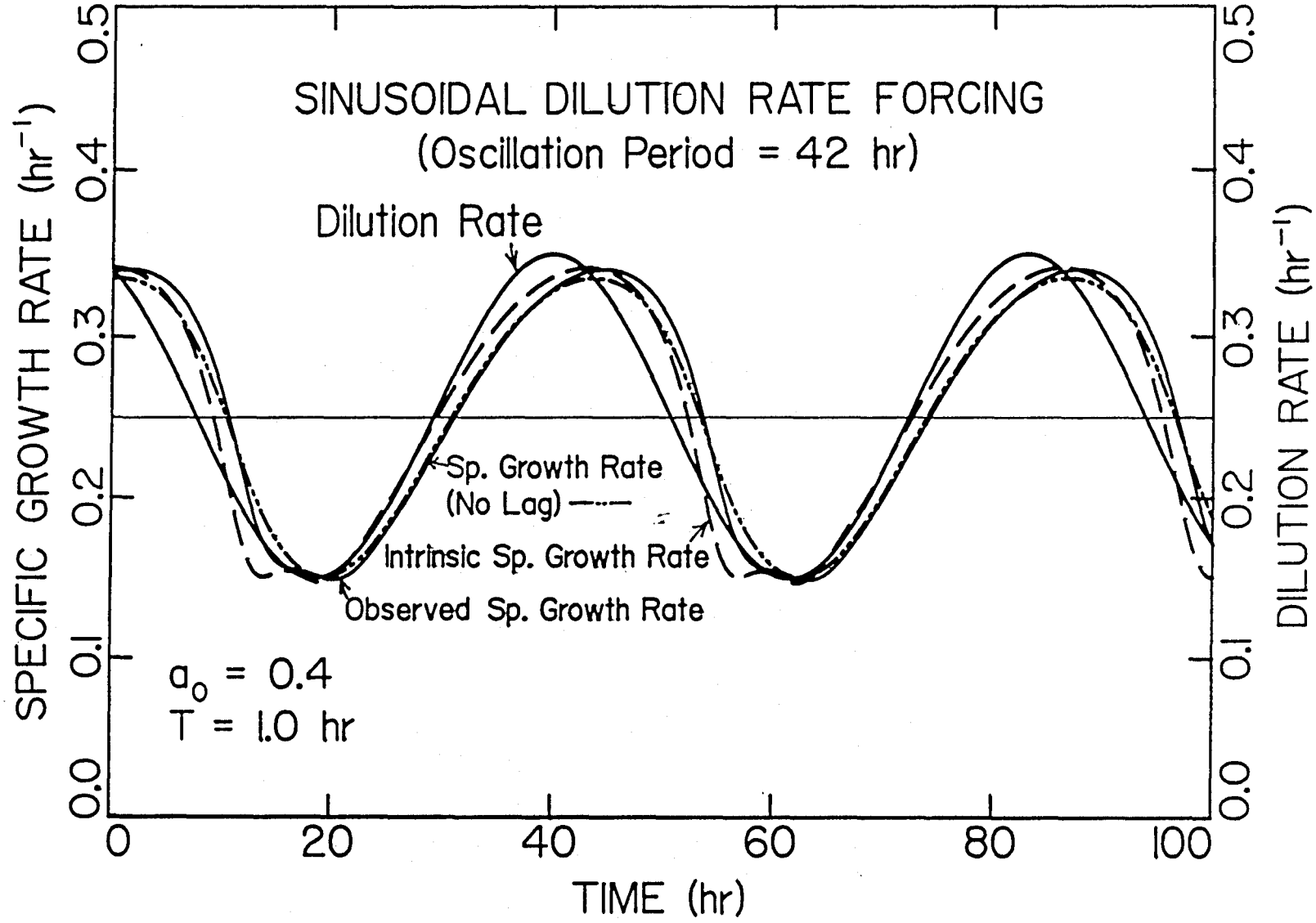


Figure 2.8.28. Response of the specific growth rates to the sinusoidal dilution rate forcing. (D=0.25±0.1 hr^{-1}, period=42 hr, a_0=0.4, T=1.0 hr.)

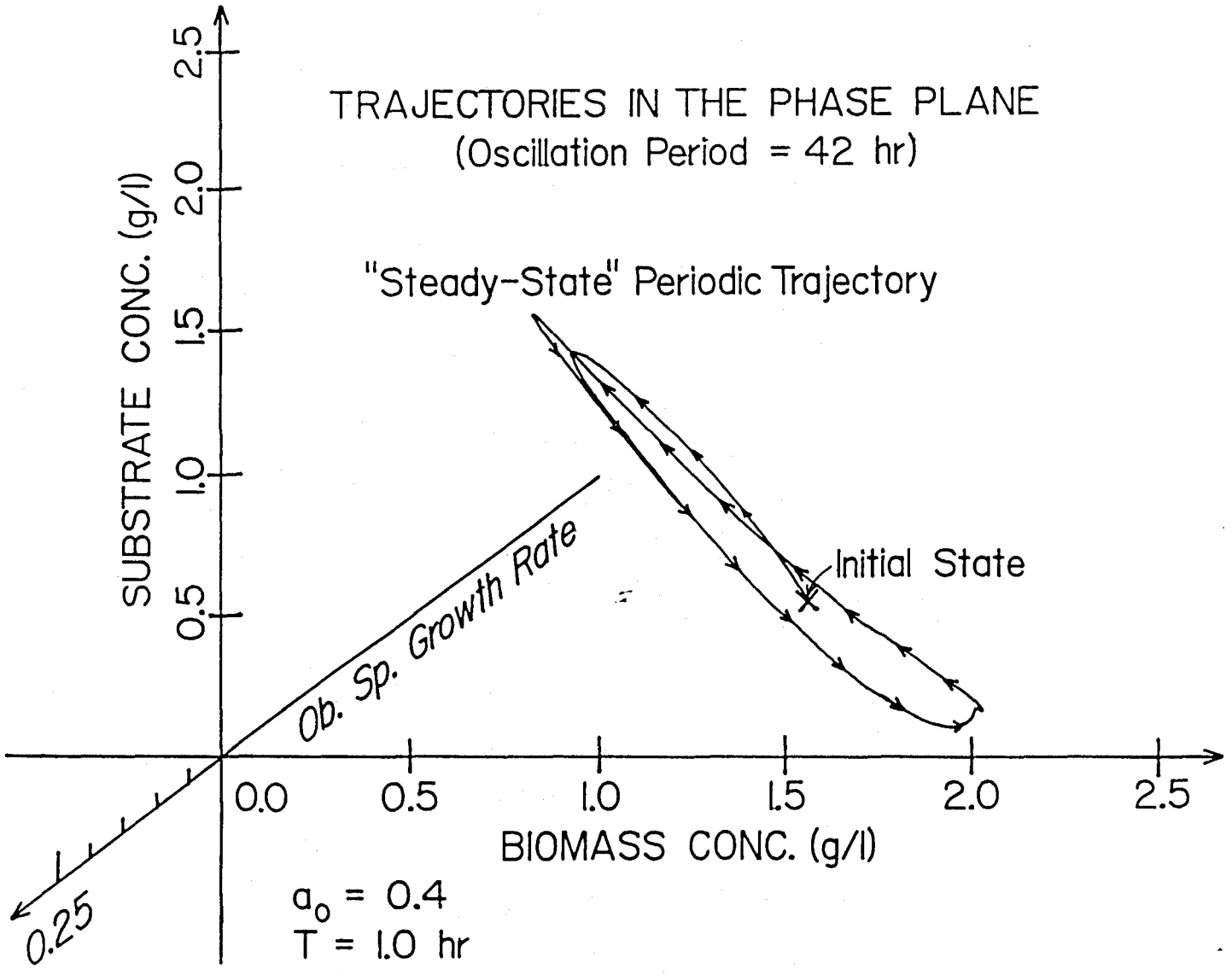


Figure 2.8.29. two-dimensional representation of the state trajectory in a three-dimensional phase plane for a chemostat subject to the sinusoidal dilution rate forcing. (D=0.25±0.1 hr^{-1}, period=42 hr, a_0=0.4, T=1.0 hr.)

The significance of this second example is that it is possible to explain such an interesting behavior in terms of classical chemostat equations with only the addition of a time-lag kernel. However, no attempt is made to induce and observe such "notch" behaviors in a fermentor experimentally.

Bode Diagrams from Direct Dynamic Simulation:

Although the absolute maxima and minima in each cycle are relatively easy to identify, because of the distortion in the response curves, the conventional definitions of amplitude ratio and phase angle are not directly applicable in our system. This difficulty is demonstrated in the previous example, where multiple local maxima/minima are possible. Thus, these nonlinear effects necessitate constructing new definitions for the amplitude ratios and phase angles for the biomass concentration and the specific growth rates. The definitions are shown graphically in Figure 2.8.30 for the biomass concentration, Figure 2.8.31 for the observed specific growth rate with respect to the intrinsic specific growth rate, and Figure 2.8.32 for the observed specific growth rate with respect to the dilution rate.

In each of the three diagrams, six amplitudes are identified. In Figure 2.8.30, for example, these are:

$$A_{x,\max} = x_{\max} - x_0 \tag{2.8.76a}$$

$$A_{x,\min} = x_0 - x_{\min} \tag{2.8.76b}$$

$$A_{x,\mathrm{mm}} = x_{\max} - x_{\min} \tag{2.8.76c}$$

$$A^*_{x,\max} = x^*_{\max} - x_0 \tag{2.8.76d}$$

$$A^*_{x,\min} = x_0 - x^*_{\min} \tag{2.8.76e}$$

$$A^*_{x,\mathrm{mm}} = x^*_{\max} - x^*_{\min} \tag{2.8.76f}$$

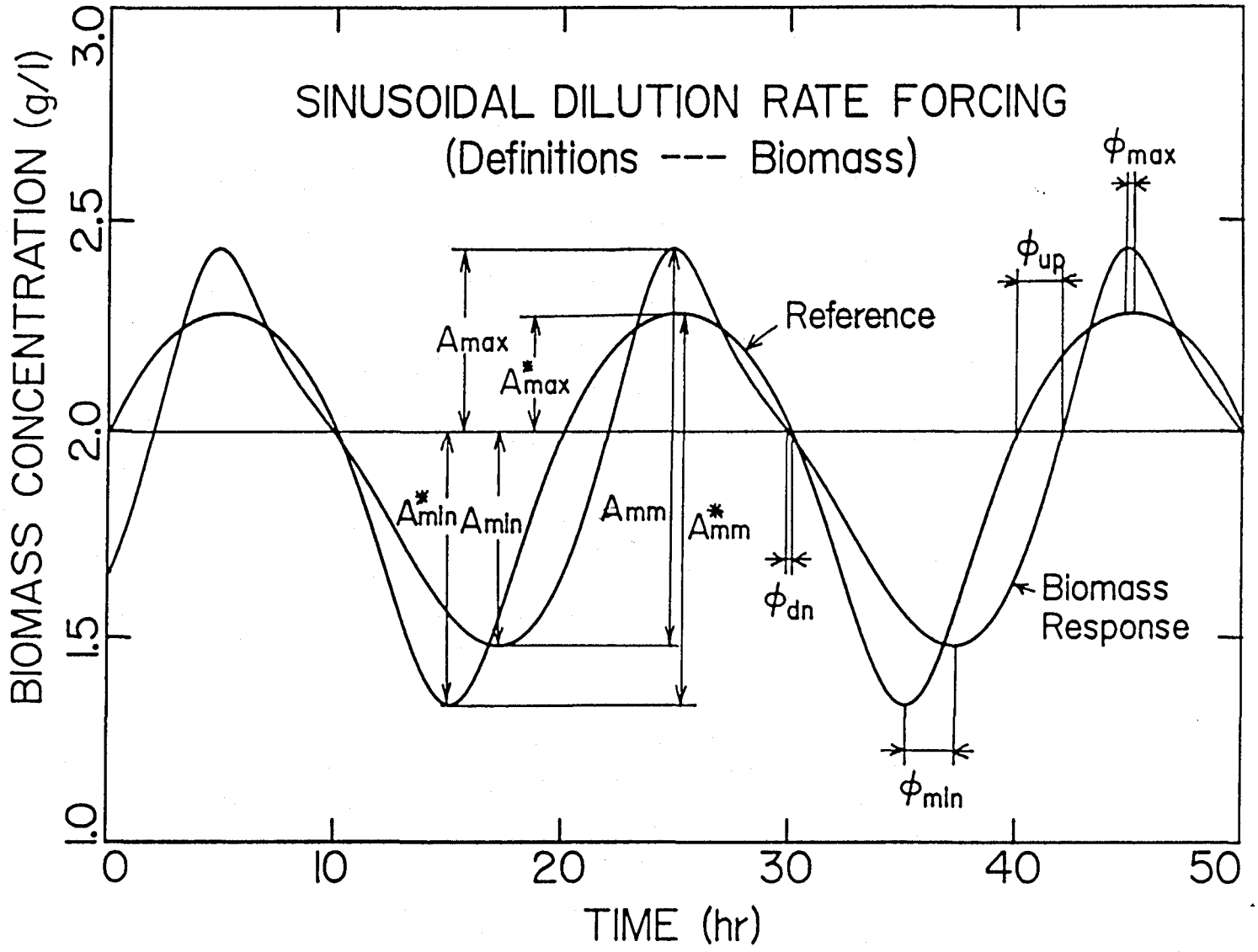


Figure 2.8.30. Graphical representation of the definitions for the amplitude ratios and phase angles for the biomass concentration.

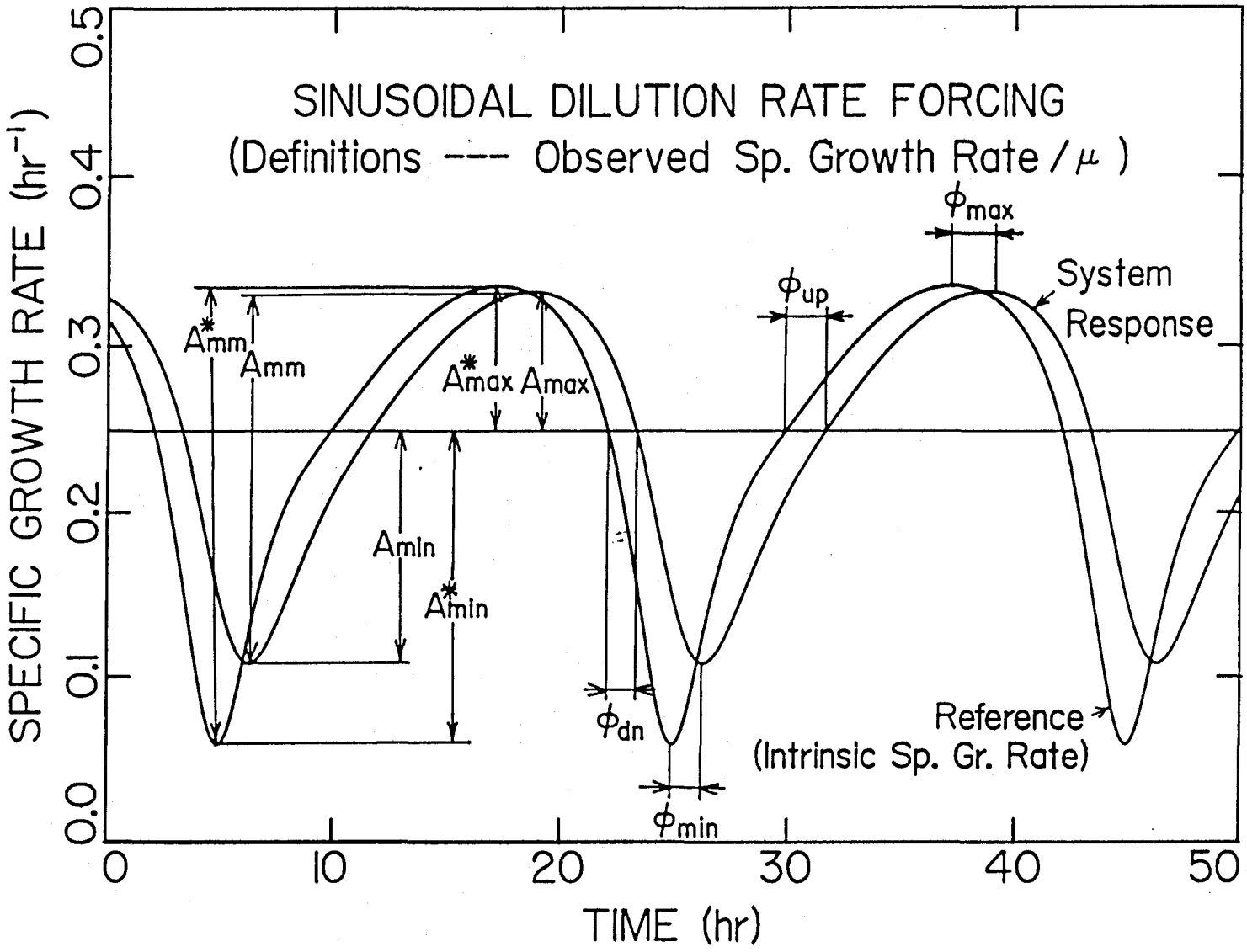


Figure 2.8.31. Graphical representation of the definitions for the amplitude ratios and phase angles for the observed specific growth rate with respect to the intrinsic specific growth rate.

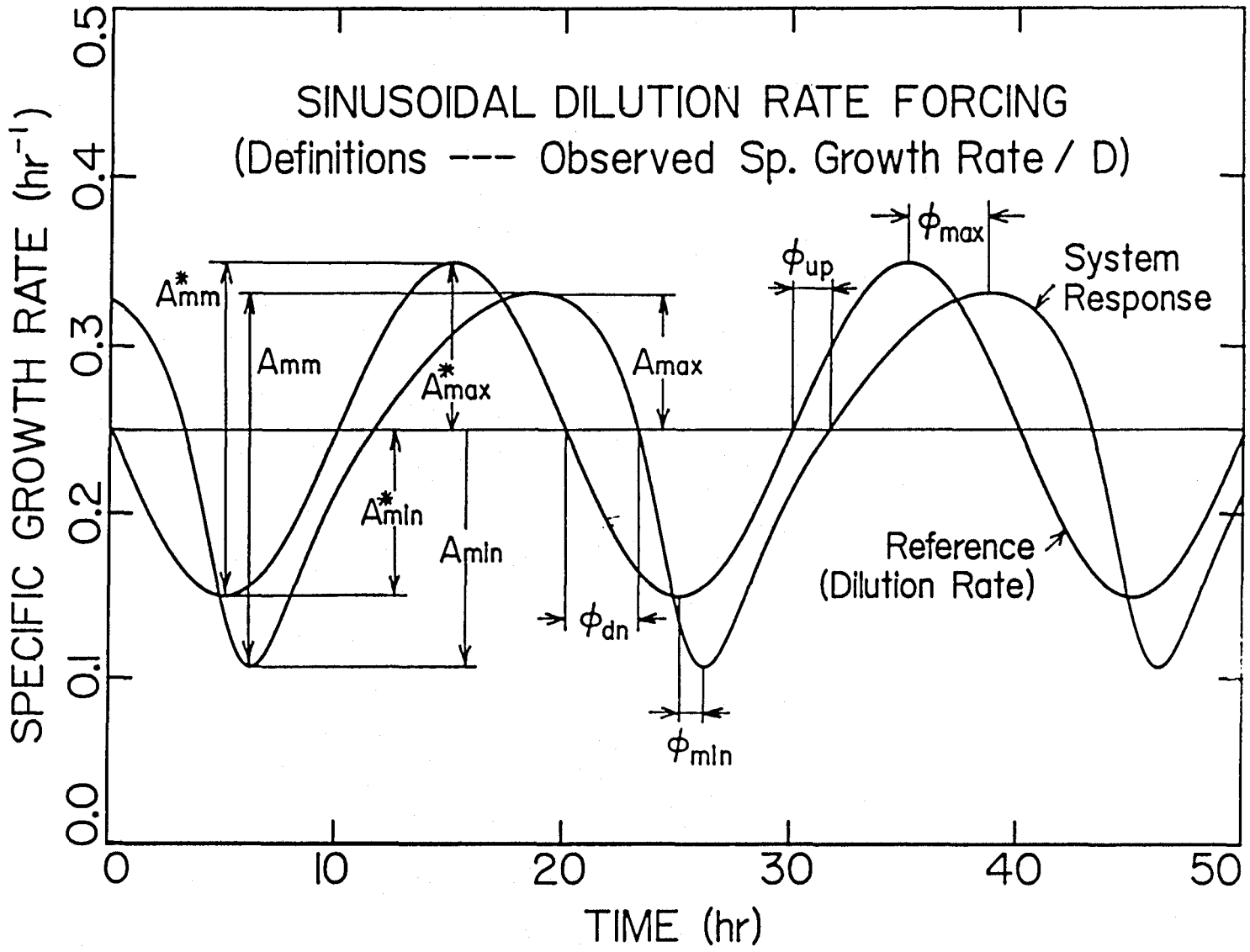


Figure 2.8.32. Graphical representation of the definitions for the amplitude ratios and phase angles for the observed specific growth rate with respect to the excitation dilution rate.

where x is the dynamically simulated sinusoidal response curve of the biomass concentration, $x^\star$ is the theoretical steady-state value of biomass concentration corresponding to the dilution rate at that instant. The subscripts "max" and "min" signify that the values are the absolute maximum and minimum, respectively, in each cycle, and x_0 is the steady-state biomass concentration at a constant mean dilution rate of D_0. From these individual amplitudes the following three amplitude ratios are constructed:

$$A.R._{x,\max} \equiv \frac{A_{x,\max}}{A^*_{x,\max}} \tag{2.8.77a}$$

$$A.R._{x,\min} \equiv \frac{A_{x,\min}}{A^*_{x,\min}} \tag{2.8.77b}$$

$$A.R._{x,\mathrm{mm}} \equiv \frac{A_{x,\mathrm{mm}}}{A^*_{x,\mathrm{mm}}} \tag{2.8.77c}$$

Because of the upward vertical shift in the biomass concentration from the instantaneous steady state values, the following inequality holds:

$$A.R._{x,\min} < A.R._{x,\mathrm{mm}} < A.R._{x,\max} \tag{2.8.78}$$

In a similar manner, eight critical points are identified in each cycle for each of the three combinations of response curves and reference curves. In Figure 2.8.30 for the biomass, for example, these eight points are used to calculate the following four phase angles:

$$\phi_{x,\max} \equiv \left(t_{x,\max} - t^*_{x,\max}\right)\frac{360^\circ}{T_f} \tag{2.8.79a}$$

$$\phi_{x,\min} \equiv \left(t_{x,\min} - t^*_{x,\min}\right)\frac{360^\circ}{T_f} \tag{2.8.79b}$$

$$\phi_{x,\mathrm{up}} \equiv \left(t_{x,\mathrm{up}} - t^*_{x,\mathrm{up}}\right)\frac{360^\circ}{T_f} \tag{2.8.79c}$$

$$\phi_{x,\mathrm{dn}} \equiv \left(t_{x,\mathrm{dn}} - t^*_{x,\mathrm{dn}}\right)\frac{360^\circ}{T_f} \tag{2.8.79d}$$

where $t_{x,\max}$ and $t_{x,\min}$ are the times at which the dynamically simulated biomass concentration reaches the highest and lowest values, respectively. Similarly, $t^*_{x,\max}$

and $t^{\star}_{x,\min}$ are the times of maxima and minima, respectively, for the theoretical instantaneous steady-state values of biomass concentration corresponding to the dilution rate. The times at which the curves cross x_0 are identified as $t_{x,\mathrm{up}}$, $t^{\star}_{x,\mathrm{up}}$, $t_{x,\mathrm{dn}}$, $t^{\star}_{x,\mathrm{dn}}$. Finally, in the above equations $T_f = \frac{2\pi}{\omega}$ is the period of the dilution rate.

In Figure 2.8.31, similar definitions are constructed from the dynamically simulated response curves of the observed specific growth rate and intrinsic specific growth rate, which is the reference. In Figure 2.8.32, the reference curve is replaced by the dilution rate. Incidentally, these reference curves allow both the amplitude ratios to be normalized to unity and the phase angles to approach 0° at low frequencies.

The following general observations are made from a series of dynamic simulations. The maxima in the biomass concentration always coincide with the minima in the substrate concentration, and vice versa. The normalized amplitude ratios of these two states as functions of the forcing frequencies are also identical. Because the intrinsic specific growth rate, $\mu(t)$, is a direct monotonically increasing function of $s(t)$, its maxima and minima are also synchronized with the substrate concentration, even though the shapes of the response curves do not coincide, again due to the nonlinear constitutive relationship of $\mu(s)$. The order in which maxima and minima are reached is summarized below:

$$D(t) \Longrightarrow \underbrace{s(t), \mu(t) \Longrightarrow y(t) \Longrightarrow x(t)}_{180^\circ}$$

This sequence is in agreement with Figure 2.8.4.

The effects of nonlinearities are manifested in Figures 2.8.33 and 2.8.34. All the curves in these figures should coincide with the linearized model in the absence of distortions from the ideal sinusoidal behavior. From the biomass *A.R.* diagram,

it can be seen that $A.R._{x,\max}$ is consistently above the linearized model and that $A.R._{x,\min}$ is consistently below the linearized model. $A.R._{x,\mathrm{mm}}$, being the weighted average of $A.R._{x,\max}$ and $A.R._{x,\min}$ and lying in between these two values, is considerably closer to the linearized model.

$$\frac{A.R._{x,\mathrm{mm}} = A_{x,\max} + A_{x,\min}}{A^*_{x,\max} + A^*_{x,\min} = \frac{A.R._{x,\max}}{1+\frac{1}{r}} + \frac{A.R._{x,\min}}{1+r}}, \tag{2.8.80}$$

where $r = \frac{A^*_{x,\max}}{A^*_{x,\min}}$, which is unity if $A^*_{x,\max} = A^*_{x,\min}$.

It is generally true that $A.R._{\frac{y}{\mu},\min} < A.R._{\frac{y}{\mu},\min}$. Note also that there is a region in which $A.R._{\frac{y}{D},\max}$ and $A.R._{\frac{y}{D},\min}$ deviate significantly from each other. Due to the extremely small amplitudes and periods, results in the high frequency region, *i.e.*, $\omega > 5$ hr^{-1} are not very reliable. Furthermore, sudden jumps of phase angles near the mid region, *i.e.*, 0.2 hr^{-1} $< \omega <$0.3 hr^{-1} are caused by the difficulties encountered in identifying the minima and maxima when there are notches in the response curves.

In parallel to the linearized analysis, the effects of the lag time constant, T, for a purely 1st-order kernel and a combined 0th/1st-order kernel are studied in the next four frequency response diagrams (Figures 2.8.35, 2.8.36, 2.8.37, and 2.8.38). The effects of the 0th-order fractions in the kernel function, a_0, are shown in Figures 2.8.39 and 2.8.40. The amplitude ratios used to plot these diagrams are based on the difference between the maxima and minima, and the phase angles are the algebraic average of $\phi_{\max}$ and $\phi_{\min}$. The overall behavior in terms of corner frequencies and slopes is similar to the linearized model, except that the dynamically simulated amplitude ratios are consistently lower than those of the linearized model. In general, there is wider deviation for larger amplitude ratios. This is mainly due to the physical constraint that concentrations cannot be increased indefinitely nor

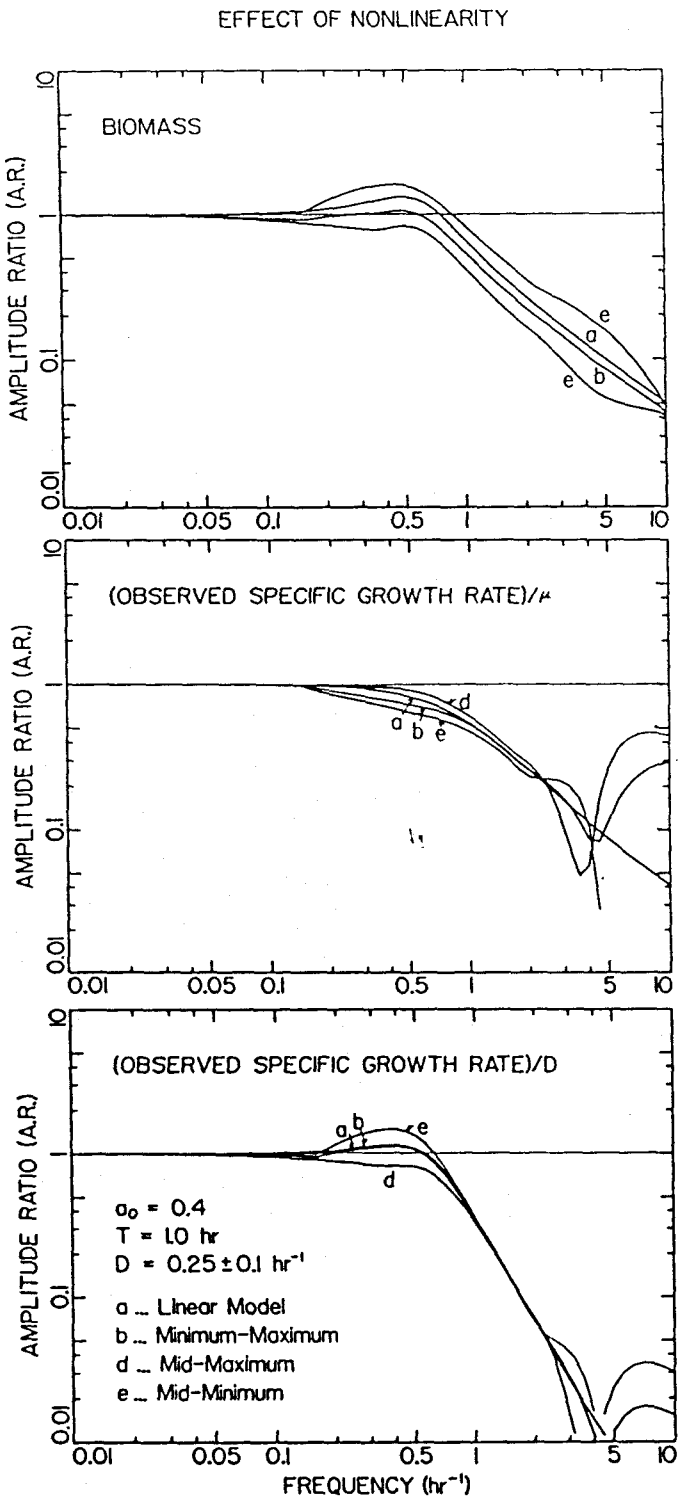


Figure 2.8.33. Effect of nonlinearities in the chemostat state equations on the amplitude ratios with a sinusoidal dilution rate forcing (a_0=0.4, T=1.0 hr.)

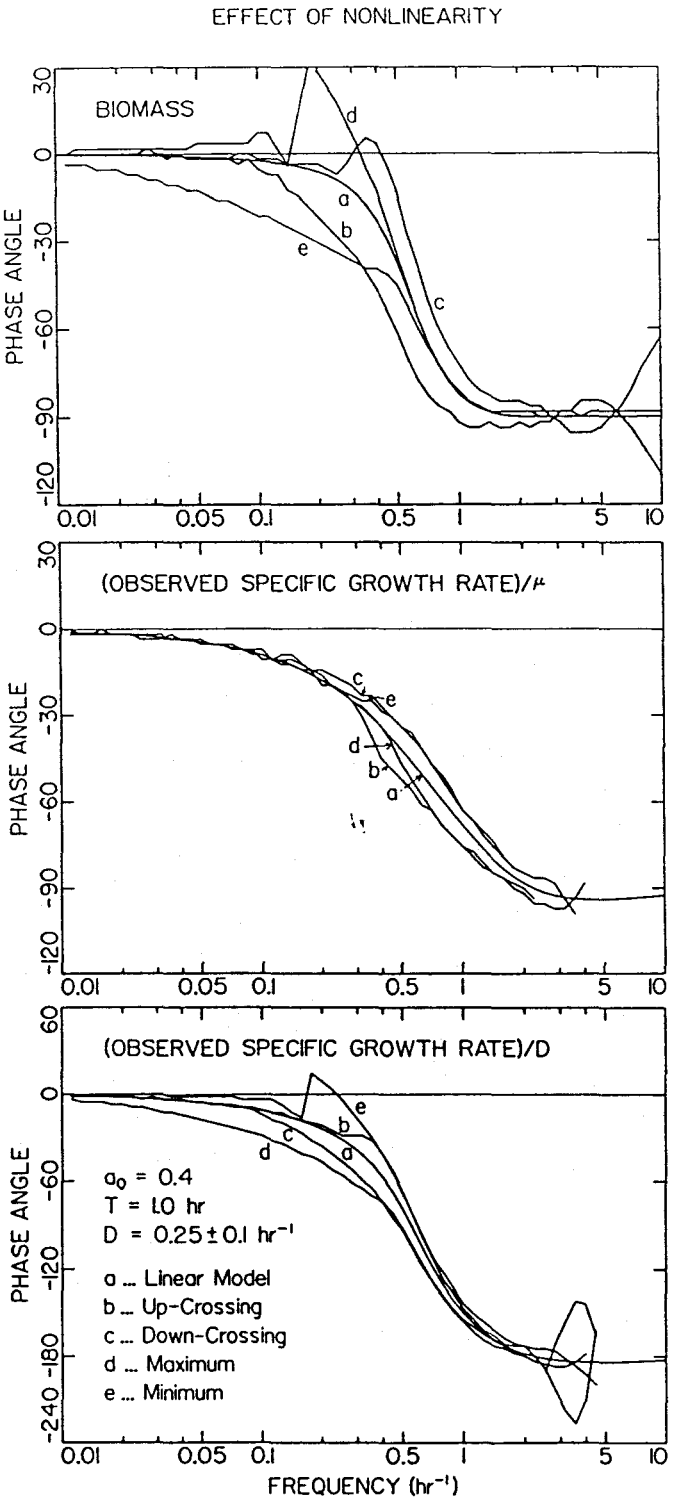


Figure 2.8.34. Effect of nonlinearities in the chemostat state equations on the phase angles with a sinusoidal dilution rate forcing (a_0=0.4, T=1.0 hr.)

negative. Thus, the amplitude ratios often cannot match those values achievable using the linearized analysis.

Finally, the effects of forcing amplitudes and the mean dilution rates are studied with the following oscillations:

Curve	Dilution Rate	x_0	s_0	μ_0'	C
	(hr^{-1})	(g/l)	(g/l)	(l/g-hr)	(hr^{-1})
a (Ref)	0.250±0.100	2.00	1.00	0.125	0.500
b (Mid)	0.250±0.150	2.00	1.00	0.125	0.500
c (Hi)	0.325±0.075	1.57	1.86	0.061	0.193
d (Low)	0.175±0.075	2.23	0.54	0.211	0.943

For a linear system, the response should not depend on the amplitude of the forcing variable. As indicated by Figure 2.8.41, the amplitude ratios for biomass are strongly influenced by the amplitude of the dilution rate oscillation, and there are some small differences in the middle frequency range for Y/M; whereas, the observed specific growth rate with respect to the dilution rate is not influenced. This is also generally true for the phase angle plots presented in Figure 2.8.42, although there are some shifts in the biomass plots. The last two curves in the above table are used to study the effects of the stability variable C, and the dynamic simulation shows the same behavior as predicted by the linearized model. Again, the high amplitude ratios are generally lower than those given by the linearized model.

Numerical Problems:

Even with the classical Monod chemostat equations, there exist numerical problems when the substrate concentration is approaching depletion very quickly. One such case is encountered near the end of a batch run. In certain situations, this makes the changes in the substrate concentration quite "stiff" in the sense that

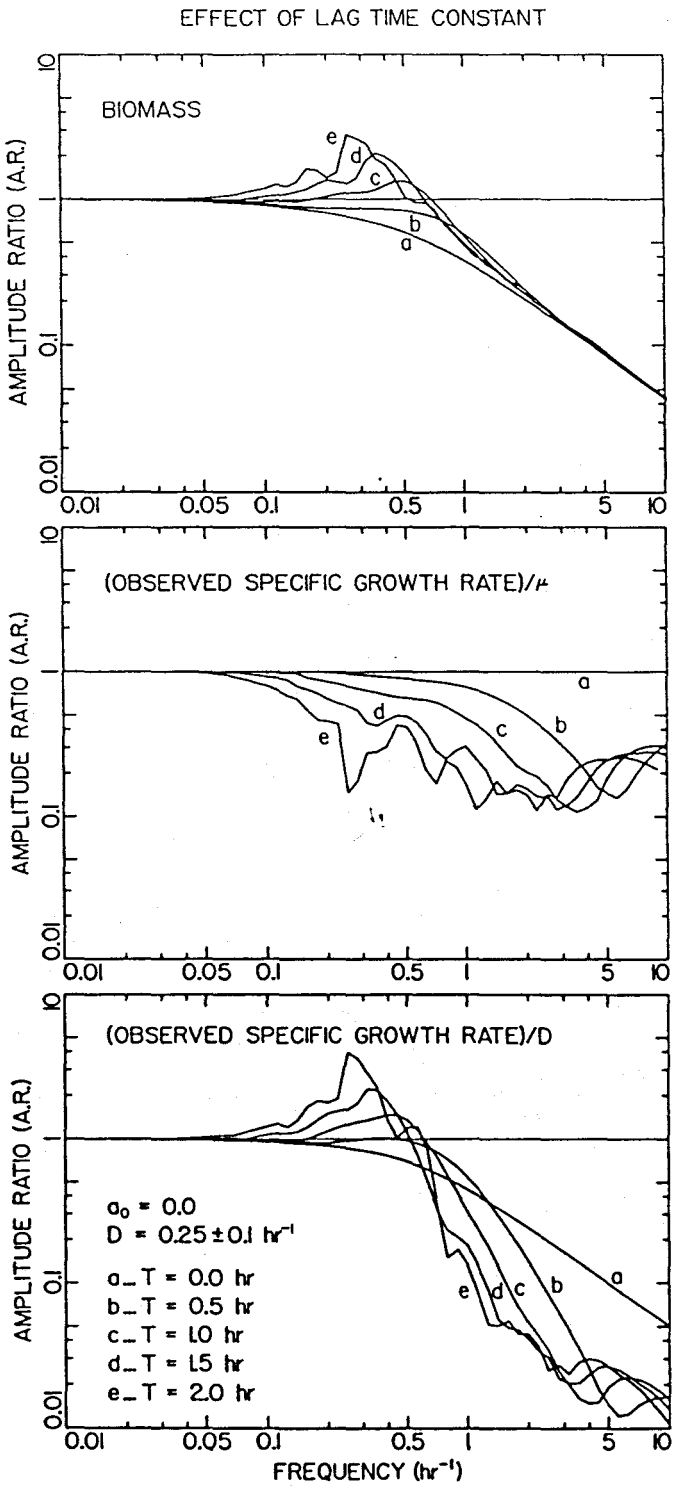


Figure 2.8.35. Effect of the lag time constant, T, on the amplitude ratios for a chemostat system subject to a sinusoidal dilution rate forcing, dynamically simulated with a purely 1st-order time-lag kernel. (a_0=0.0, See text for chemostat model parameters used.)

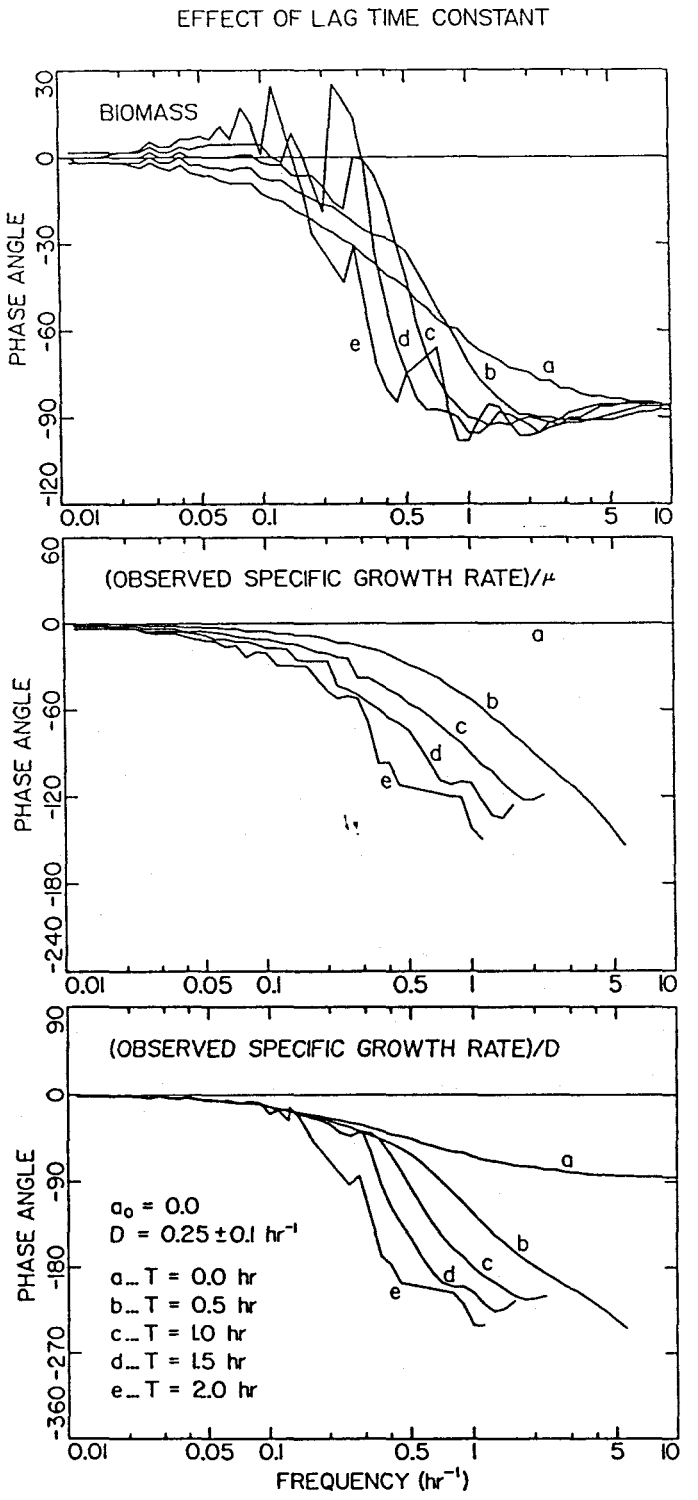


Figure 2.8.36. Effect of the lag time constant, T, on the phase angles for a chemostat system subject to a sinusoidal dilution rate forcing, dynamically simulated with a purely 1st-order time-lag kernel. (a_0=0.0, See text for chemostat model parameters used.)

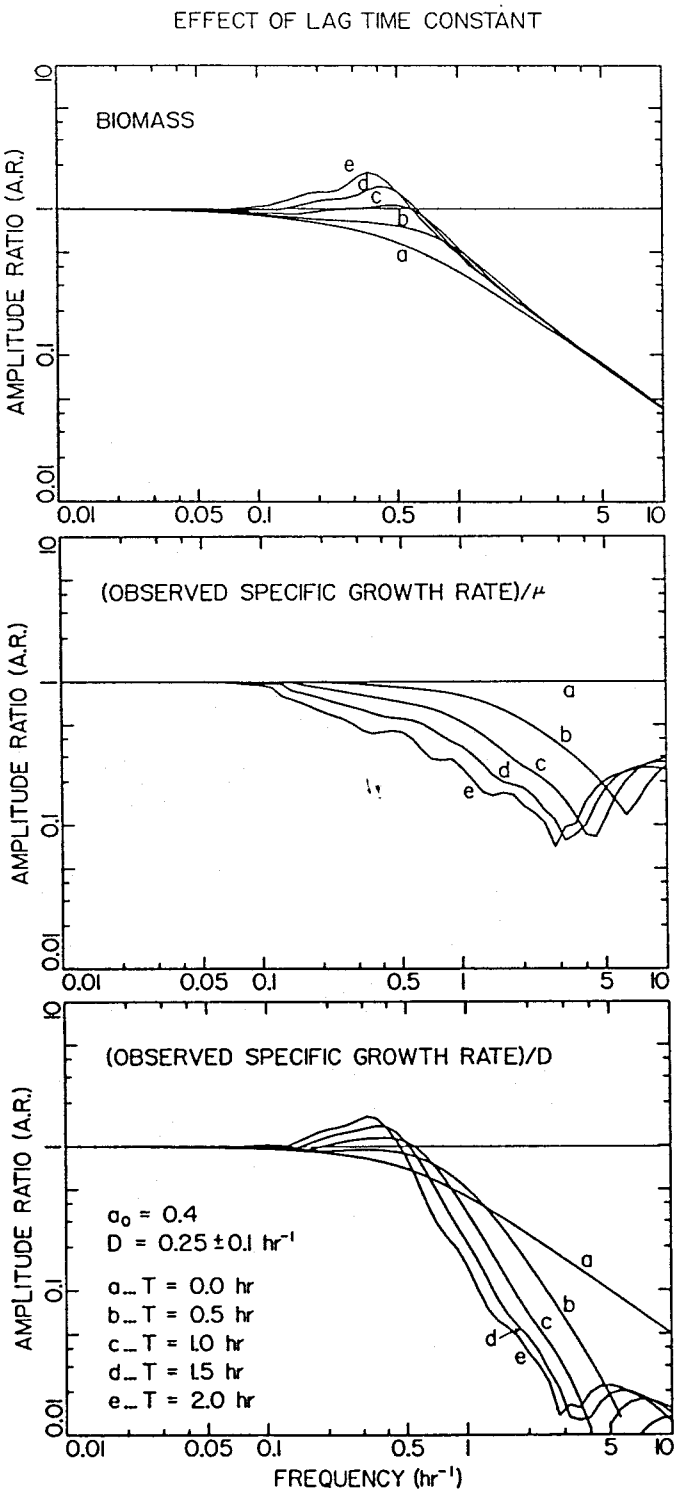


Figure 2.8.37. Effect of the lag time constant, T, on the amplitude ratios for a chemostat system subject to a sinusoidal dilution rate forcing, dynamically simulated with a combined 0th-order and 1st-order time-lag kernel. (a_0=0.4)

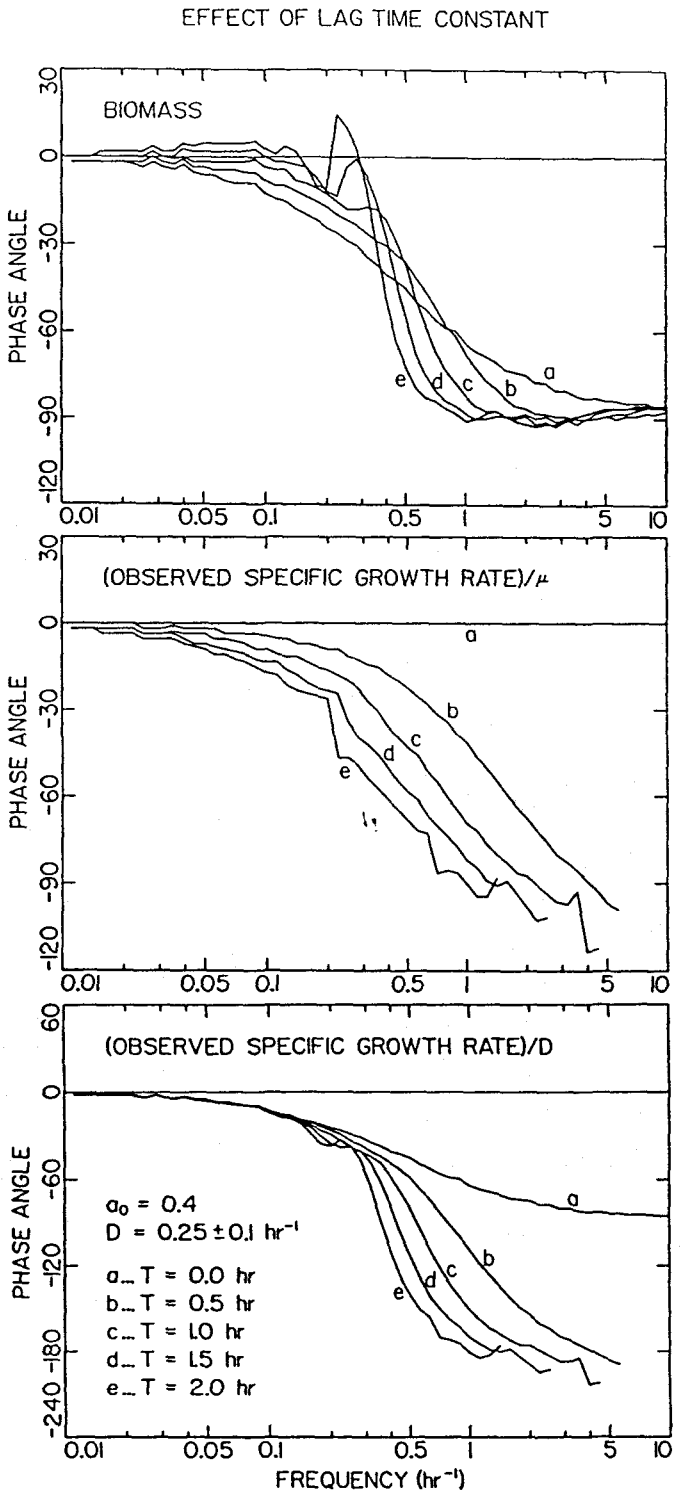


Figure 2.8.38. Effect of the lag time constant, T, on the phase angles for a chemostat system subject to a sinusoidal dilution rate forcing, dynamically simulated with a combined 0th-order and 1st-order time-lag kernel. (a_0=0.4)

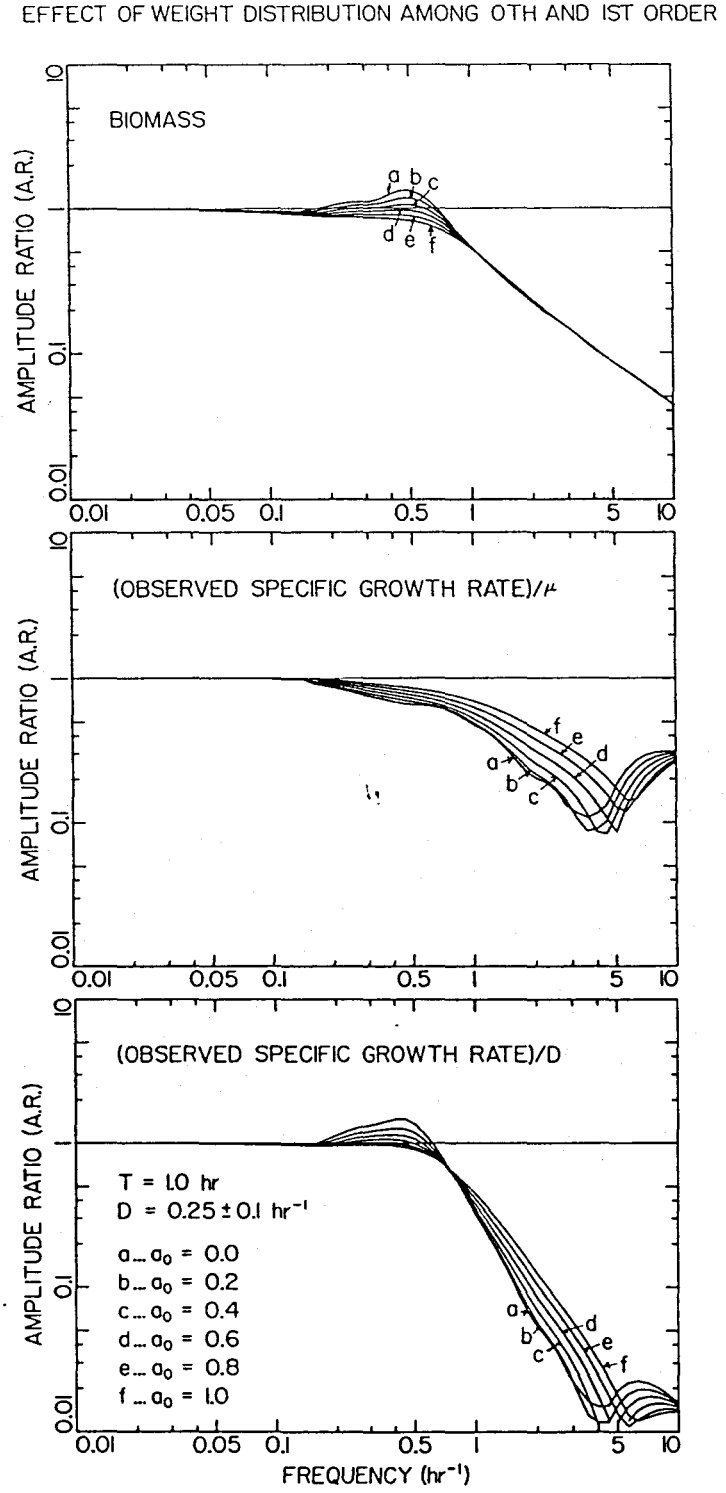


Figure 2.8.39. Effect of the 0th-order fraction in the kernel function, a_0, on the amplitude ratios for a dynamically simulated chemostat system subject to a sinusoidal dilution rate forcing. (T=1.0 hr.)

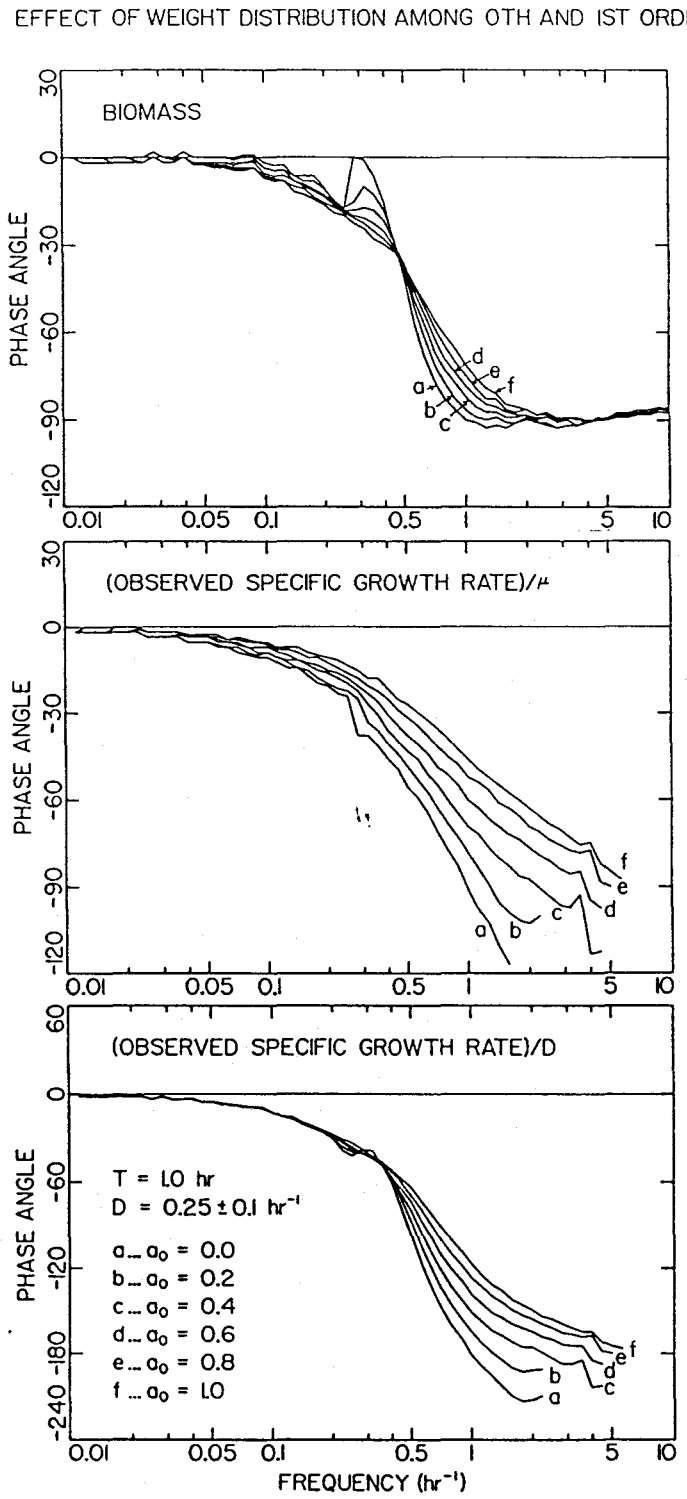


Figure 2.8.40. Effect of the 0th-order fraction in the kernel function, a_0, on the phase angles for a dynamically simulated chemostat system subject to a sinusoidal dilution rate forcing. (T=1.0 hr.)

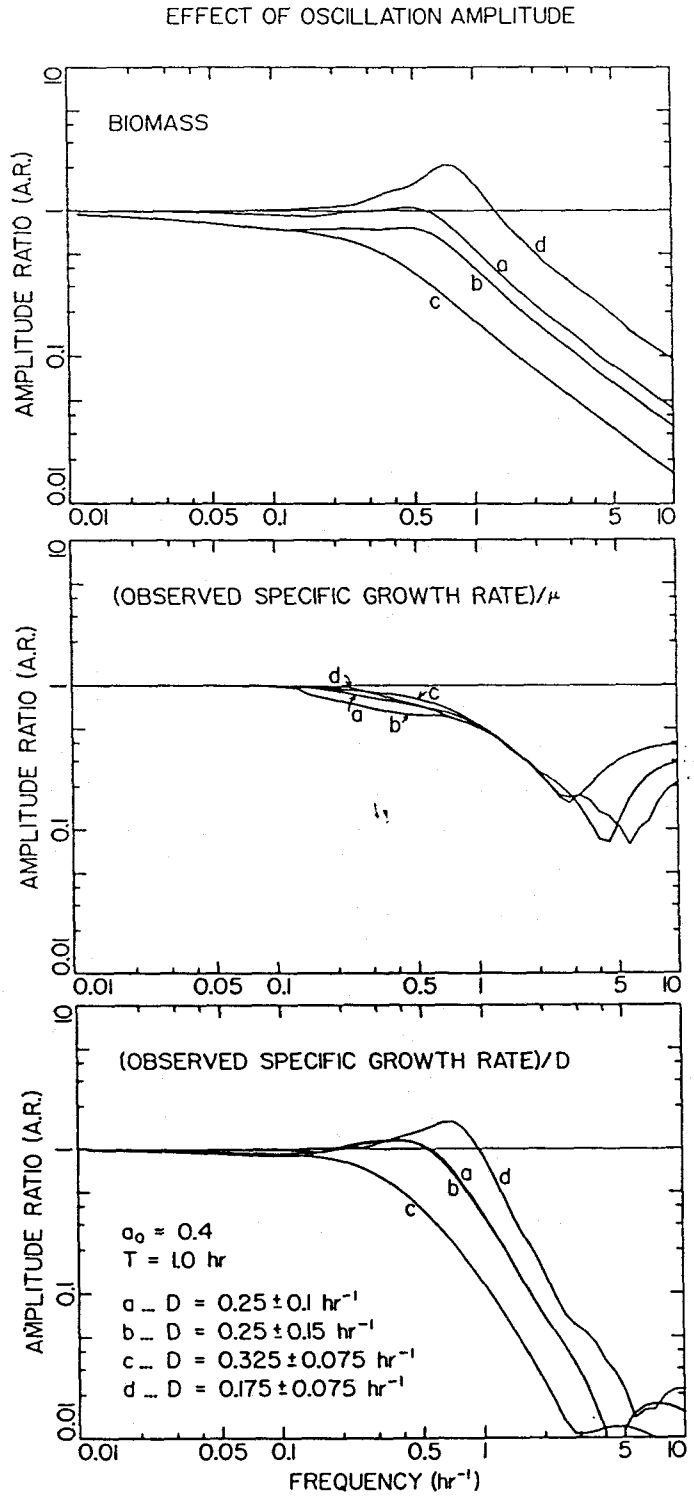


Figure 2.8.41. Effect of the oscillation amplitude of the sinusoidal dilution rate on the amplitude ratios for a dynamically simulated chemostat system. (a_0=0.4, T=1.0 hr.)

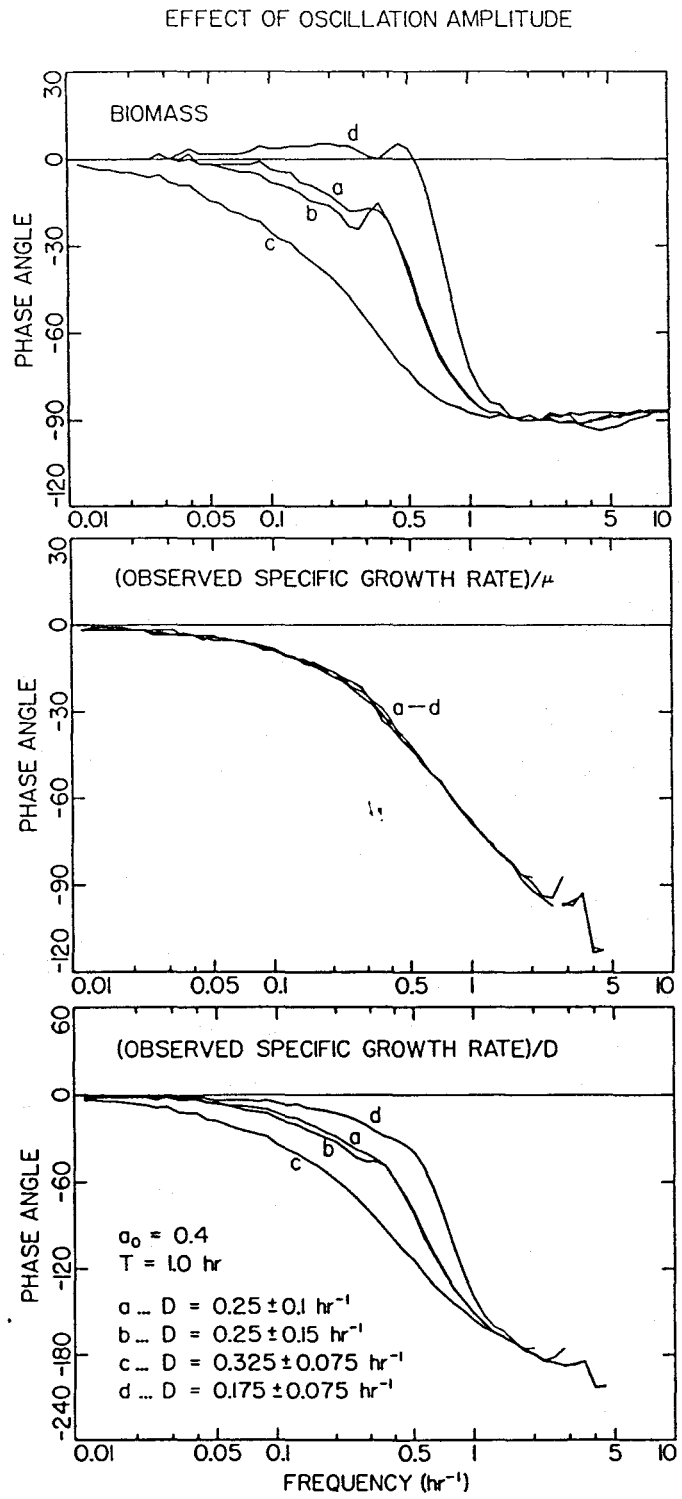


Figure 2.8.42. Effect of the oscillation amplitude of the sinusoidal dilution rate on the phase angles for a dynamically simulated chemostat system. (a_0=0.4, T=1.0 hr.)

throughout most of a fermentation run the specific growth rate stays at a high value close to μ_m, and the substrate is quickly consumed. Because in practice the half-saturation constant K_s is usually relatively small, the specific growth rate does not reflect the effect of K_s until the last moment before the substrate is totally exhausted. Suddenly, the specific growth rate drops to zero. This type of problem can be effectively handled with a good stiff numerical integration method.

With the time-lag model, to this numerical difficulty are added the problems mentioned in association with a discrete time-delay. Thus, although not as problematic, the numerical difficulties associated with integrating a set of differential equations with a discrete time-delay is also present in a time-lag system. Because the observed specific growth rate lags behind the intrinsic specific growth rate, which is in direct synchronization with the limiting substrate concentration, one needs to introduce a shutoff factor that can automatically cut off the substrate consumption when the substrate concentration drops to values very close to zero. Therefore, the physically impossible occurrence of a negative substrate concentration is prevented. There are a small number of places this shutoff factor can be inserted in the system dynamic equation. The following is one example.

$$\frac{dx(t)}{dt} = \Big[-1 + y(t) f_s(s) \Big] x(t) \tag{2.8.81a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s} y(t) f_s(s) x(t) \tag{2.8.81b}$$

When placed at these points, $f_s(s)$ should be 0 for $s \leq 0$ to cut off any residual $y(t)$, and many expressions exist for this slight modification. For example, the substrate concentration can be prevented from overshooting into the negative region by installing a shutoff factor f_s that corresponds to an on-off switch.

$$f_s = \begin{cases} 0 & \text{for } s \leq 0 \\ 1 & \text{for } s > 0 \end{cases} \tag{2.8.82}$$

Physically, this equation means that there is no growth when there is no substrate present. Although this logically simple solution effectively prevents substrate concentration from becoming negative, it is numerically too drastic for most integration routines to handle properly without inducing convergence problems within each integration step. To soften the stiffness, a saturation expression may be used.

$$f_s = \begin{cases} 0 & \text{for } s \leq 0 \\ \frac{s}{0.001K_s+s} & \text{for } s > 0 \end{cases} \tag{2.8.83}$$

Motivation for Frequency Response:

In closing, it is appropriate to reiterate the objectives of this section. On the surface, it may seem that few truly new concepts are contained in this section. Because of the special care taken during the formulation of the time-lag theory in previous sections, much of the work has indeed already been done. Precisely, the beauty of the time-lag approach lies in its simplicity and the ease with which the well developed systems theories can be directly applied with practically no modification. This simplicity is demonstrated through the derivation of the transfer functions and in the prediction of response to sinusoidal disturbances in the dilution rate. Thus, a researcher need not waste effort on model evaluation, and a typical process control engineer is not required to possess a mathematical knowledge beyond what has already been mastered.

2.9 EXTENSIONS

Extensions to Time-lags in the Yield Coefficient

Although throughout this thesis the presence of time-lag is considered only in the specific growth rate, it is not restricted to that variable. For example, another variable in the basic set of chemostat dynamic equations is the cell-biomass to substrate yield coefficient Y_s. This yield coefficient may change if there is more than one pathway that can lead the starting reactant or some intermediates to

different end products, including the cell-biomass, or if the reaction stoichiometry is variable.

The simplest theoretical example is illustrated below where the substrate can be directed toward either biomass synthesis or another non-biomass product formation.

$$\xi_1 : \quad S \longrightarrow \varsigma_1 \text{ Biomass} \tag{2.9.1a}$$

$$\xi_2 : \quad S \longrightarrow \varsigma_2 \text{ Product} \tag{2.9.1b}$$

where ξ_1 and ξ_2 are the extents of the respective reactions, and ς_1 and ς_2 are the simplified stoichiometry coefficients. It is quite common that the enzymes that catalyze the alternative metabolic pathways are induced by the presence of certain substrates or inducers. This leads to the dependence of the steady-state level of alternative metabolic enzymes on, for instance, the substrate concentration. Because the synthesis or even the activation of the enzymes for the newly induced pathway cannot be achieved instantaneously, the existence of dynamic lags is inevitable. It is apparent that a shift in the substrate concentration will not immediately increase the rate of material flux in the newly available pathway. The rate of the alternate metabolic reaction must be synchronized with the finite rate of accumulation of active enzymes. As a result, there is a time-lag in the adaptation of the extents of reaction #2 to changing environmental conditions.

If the product formation is not rigorously considered, for example, the production of CO_2 due to energy waste that is dependent on the substrate concentration, then the apparent yield coefficient is defined as:

$$Y_s(s) = \varsigma_1 \frac{\mathrm{MW}_{\mathbf{biomass}}}{\mathrm{MW}_{\mathbf{substrate}}} \frac{\xi_1(s)}{\xi_1(s) + \xi_2(s)}. \tag{2.9.2}$$

Thus, it is clear how the yield coefficient may depend on the substrate concentration and how the *observed* apparent yield coefficient in a dynamic experiment may lag behind the *intrinsic* apparent yield coefficient measured in steady-state experiments.

With the inclusion of a time-lag in the yield coefficient, the set of Equations (2.2.5) and (2.2.6) can be written as:

$$\frac{dx(t)}{dt} = [-1 + \mu(s)]\, x(t) \tag{2.9.3a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{\int_{-\infty}^{t} Y_s[s(h)]k_Y(t-h)dh}\mu(s)x(t), \tag{2.9.3b}$$

where $\int_{-\infty}^{t} Y_s[s(h)]k_Y(t-h)dh$ is the observed yield coefficient that lags behind the intrinsic yield coefficient in accordance with the description of the time-lag kernel $k_Y(t)$. Following the same approach as that adopted for a time-lag in the specific growth rate, the last equation can be expressed in terms of the time-lag variable $y_s(t)$.

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{y_s(t)}\mu(s)x(t), \tag{2.9.4}$$

where $y_s(t)$ is defined as:

$$y_s(t) \equiv \int_{-\infty}^{t} Y_s[s(h)]k_Y(t-h)dh. \tag{2.9.5}$$

Analogous to a time-lag in the specific growth rate, for a general n_Yth-order time-lag kernel $k_Y(t)$ that is a linear combination of the base exponential distribution functions $k_i(t)$,

$$k_Y(t) = b_0k_0(t) + b_1k_1(t) + b_2k_2(t) + \ldots + b_{n_Y}k_{n_Y}(t), \tag{2.9.6}$$

the observed yield coefficient $y_s(t)$ satisfies the same type of differential equation as before.

$$\sum_{i=0}^{n_Y+1}\binom{n+1}{i}T^i\frac{d^iy_s(t)}{dt^i} = \sum_{i=0}^{n_Y}\beta_iT^i\frac{d^iY_s(t)}{dt^i}; \qquad \text{where } \beta_i = \sum_{j=i}^{n_Y}\binom{j}{i}b_{n_Y-j} \tag{2.9.7}$$

The solution to this case will not be further elaborated. It suffices to note that all the steps closely parallel those considered previously for a time-lag in the specific growth rate.

Extensions to Multiple Time-Lags

Because of the versatile manner in which the time-lag concept has been formulated, the extension of the methodology to more than one time-lag variable is now a straightforward matter of performing more routine mathematical exercises, even though the resulting behaviors can be quite interesting. As an example, a non-interacting formulation is presented below for time-lags in both $\mu(t)$ and $Y_s(t)$ simultaneously.

$$\frac{dx(t)}{dt} = [-1 + y(t)]\, x(t) \tag{2.9.8a}$$

$$\frac{ds(t)}{dt} = 1 - s(t) - \frac{1}{Y_s(t)} y(t)x(t). \tag{2.9.8b}$$

As before,

$$y(t) \equiv \int_{-\infty}^{t} \mu[s(h)] k_\mu(t-h)dh \tag{2.9.9a}$$

satisfies:

$$\sum_{i=0}^{n_\mu+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{n_\mu} \alpha_i T^i \frac{d^i \mu(t)}{dt^i}; \qquad \text{where } \alpha_i = \sum_{j=i}^{n_\mu} \binom{j}{i} a_{n_\mu - j} \tag{2.9.9b}$$

for a time-lag in the specific growth rate that is of the form:

$$k_\mu(t) = a_0 k_0(t) + a_1 k_1(t) + a_2 k_2(t) + \ldots + a_{n_Y} k_{n_\mu}(t). \tag{2.9.9c}$$

Similarly the observed yield coefficient:

$$Y_s(t) \equiv \int_{-\infty}^{t} Y_s[s(h)] k_Y(t-h)dh \tag{2.9.10a}$$

satisfies:

$$\sum_{i=0}^{n_Y+1} \binom{n+1}{i} T^i \frac{d^i Y_s(t)}{dt^i} = \sum_{i=0}^{n_Y} \beta_i T^i \frac{d^i Y_s(t)}{dt^i}; \qquad \text{where } \beta_i = \sum_{j=i}^{n_Y} \binom{j}{i} b_{n_Y - j} \tag{2.9.10b}$$

for a time-lag in the yield coefficient which has the following general form:

$$k_Y(t) = b_0 k_0(t) + b_1 k_1(t) + b_2 k_2(t) + \ldots + b_{n_Y} k_{n_Y}(t). \tag{2.9.10c}$$

Note that the order of the time-lag kernel for the specific growth rate and that for the yield coefficient need not be the same.

Finally, while cases where the time-lag variables are interacting may be fully analyzed, this complication is not considered in this thesis.

Extensions to Time-Lags with Multivariate Dependence

Throughout this thesis, the time-lag variable is assumed to depend on only one variable, for example, the dependence of the intrinsic specific growth rate on the limiting substrate concentration. Our formulation of the time-lag concept is general enough to absorb the time-lag variable's dependence on more than one independent variable with practically no modification. For example, if the intrinsic specific growth rate is dependent on an array of state variables, x_1, x_2, ..., x_m, then the time-lagged response of the observed specific growth rate is:

$$y(t) \equiv \int_{-\infty}^{t} \mu[x_1(h), x_2(h), \ldots, x_m(h)]k(t-h)dh = \int_{-\infty}^{t} \mu(h)k(t-h)dh \quad (2.9.11a)$$

It can be seen from a careful re-examination of the derivation that the conversion of the above time-lag integral into an equivalent differential equation is not affected by the actual functional form of the intrinsic specific growth rate nor its dependence on other variables. In general, a time-lag variable may be a function of multiple independent variables, but no modification is necessary at this step. Because of the powerful and general nature of the formulation, the time-lag differential equation to be satisfied for a generalized nth-order kernel is exactly the same as the special example, in which the specific growth rate is a function of only one independent variable.

$$\sum_{i=0}^{n+1} \binom{n+1}{i} T^i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{n} \alpha_i T^i \frac{d^i \mu(t)}{dt^i}; \qquad \text{where } \alpha_i = \sum_{j=i}^{n} \binom{j}{i} a_{n-j} \quad (2.9.11b)$$

Although the time-lag differential equation remains unchanged, the intrinsic specific growth rate's dependence on more than one independent variable is reflected

in the expanded expression for the derivatives of $\frac{d^i\mu(t)}{dt^i}$ in the above equation. The first time derivative can be calculated by using the chain rule of differentiation:

$$\frac{d\mu(t)}{dt} = \frac{\partial\mu(x_1, x_2, \ldots, x_m)}{\partial x_1}\frac{dx_1}{dt} + \frac{\partial\mu(x_1, x_2, \ldots, x_m)}{\partial x_2}\frac{dx_2}{dt} + \ldots + \frac{\partial\mu(x_1, x_2, \ldots, x_m)}{\partial x_m}\frac{dx_m}{dt} \tag{2.9.12}$$

Or equivalently,

$$\frac{d\mu(t)}{dt} = \underbrace{\begin{bmatrix}\frac{\partial\mu}{\partial x_1} & \frac{\partial\mu}{\partial x_2} & \cdots & \frac{\partial\mu}{\partial x_m}\end{bmatrix}}_{\mu_x} \underbrace{\begin{bmatrix}\frac{dx_1}{dt}\\ \frac{dx_2}{dt}\\ \vdots\\ \frac{dx_m}{dt}\end{bmatrix}}_{\frac{dx}{dt}} = \mu_x\frac{dx}{dt} \tag{2.9.13}$$

The second time derivative is:

$$\frac{d^2\mu(t)}{dt^2} = \frac{d}{dt}\left[\mu_x\frac{dx}{dt}\right]$$

$$= \underbrace{\begin{bmatrix}\frac{\partial\mu}{\partial x_1} & \frac{\partial\mu}{\partial x_2} & \cdots & \frac{\partial\mu}{\partial x_m}\end{bmatrix}}_{\mu_x} \underbrace{\begin{bmatrix}\frac{d^2x_1}{dt^2}\\ \frac{d^2x_2}{dt^2}\\ \vdots\\ \frac{d^2x_m}{dt^2}\end{bmatrix}}_{\ddot{x}}$$

$$+ \underbrace{\begin{bmatrix}\frac{dx_1}{dt} & \frac{dx_2}{dt} & \cdots & \frac{dx_m}{dt}\end{bmatrix}}_{\dot{x}^T} \underbrace{\begin{bmatrix}\frac{\partial^2\mu}{\partial x_1\partial x_1} & \frac{\partial^2\mu}{\partial x_2\partial x_1} & \cdots & \frac{\partial^2\mu}{\partial x_m\partial x_1}\\ \frac{\partial^2\mu}{\partial x_1\partial x_2} & \frac{\partial^2\mu}{\partial x_2\partial x_2} & \cdots & \frac{\partial^2\mu}{\partial x_m\partial x_2}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2\mu}{\partial x_1\partial x_m} & \frac{\partial^2\mu}{\partial x_2\partial x_m} & \cdots & \frac{\partial^2\mu}{\partial x_m\partial x_m}\end{bmatrix}}_{\mu_{xx}} \underbrace{\begin{bmatrix}\frac{dx_1}{dt}\\ \frac{dx_2}{dt}\\ \vdots\\ \frac{dx_m}{dt}\end{bmatrix}}_{\dot{x}}$$

$$= \mu_x\frac{d^2x}{dt^2} + \left(\frac{dx}{dt}\right)^T\mu_{xx}\left(\frac{dx}{dt}\right) \tag{2.9.14}$$

The third time derivative is:

$$\frac{d^3\mu(t)}{dt^3} = \frac{d}{dt}\left[\mu_x\frac{d^2x}{dt^2} + (\frac{dx}{dt})^T\mu_{xx}(\frac{dx}{dt})\right]$$
$$= \mu_x\frac{d^3x}{dt^3} + 3(\frac{dx}{dt})^T\mu_{xx}(\frac{d^2x}{dt^2}) + (\frac{dx}{dt})^T\left(\frac{d}{dt}\mu_{xx}\right)(\frac{dx}{dt}), \tag{2.9.15}$$

where

$$\frac{d}{dt}[\mu_{xx}] = \begin{bmatrix} \frac{dx_1}{dt} & \frac{dx_2}{dt} & \cdots & \frac{dx_m}{dt} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial^3\mu}{\partial x_1\partial x_1\partial x_1} & \frac{\partial^3\mu}{\partial x_2\partial x_1\partial x_1} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_1\partial x_1} \\ \frac{\partial^3\mu}{\partial x_1\partial x_1\partial x_2} & \frac{\partial^3\mu}{\partial x_2\partial x_1\partial x_2} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_1\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^3\mu}{\partial x_1\partial x_1\partial x_m} & \frac{\partial^3\mu}{\partial x_2\partial x_1\partial x_m} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_1\partial x_m} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{dx_1}{dt} & \frac{dx_2}{dt} & \cdots & \frac{dx_m}{dt} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial^3\mu}{\partial x_1\partial x_2\partial x_1} & \frac{\partial^3\mu}{\partial x_2\partial x_2\partial x_1} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_2\partial x_1} \\ \frac{\partial^3\mu}{\partial x_1\partial x_2\partial x_2} & \frac{\partial^3\mu}{\partial x_2\partial x_2\partial x_2} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_2\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^3\mu}{\partial x_1\partial x_2\partial x_m} & \frac{\partial^3\mu}{\partial x_2\partial x_2\partial x_m} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_2\partial x_m} \end{bmatrix}$$
$$+ \cdots$$
$$+ \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dx_1}{dt} & \frac{dx_2}{dt} & \cdots & \frac{dx_m}{dt} \end{bmatrix} \begin{bmatrix} \frac{\partial^3\mu}{\partial x_1\partial x_m\partial x_1} & \frac{\partial^3\mu}{\partial x_2\partial x_m\partial x_1} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_m\partial x_1} \\ \frac{\partial^3\mu}{\partial x_1\partial x_m\partial x_2} & \frac{\partial^3\mu}{\partial x_2\partial x_m\partial x_2} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_m\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^3\mu}{\partial x_1\partial x_m\partial x_m} & \frac{\partial^3\mu}{\partial x_2\partial x_m\partial x_m} & \cdots & \frac{\partial^3\mu}{\partial x_m\partial x_m\partial x_m} \end{bmatrix}. \tag{2.9.16}$$

Although the mathematics quickly becomes intractable because the order of the time derivative increases with the order of the kernel, the concept is simple. After applying the chain rule, the appropriate state dynamic equations are substituted for the $\frac{dx}{dt}$ terms in the time derivatives. At this point, stability analysis or dynamic simulations can be conducted with the same methods as used previously. Thus, this time-lag methodology can be extended effortlessly to specific growth rate expressions that contain, among other effects, inhibitor, product, or multiple limiting substrate terms. Temperature and pH effects on the specific growth rate can also be accommodated. Other time-lag variables' dependence on more than one variable can also be treated similarly.

Extensions to Nonlinear Kernel Differential Equations:

Only linear operators have been treated in the previous sections. The kernel dynamic equation has been linear, and the kernel function has been assumed to be expressible as a linear combination of the base functions. Likewise, when dynamic equations are nonlinear, we have used the method of quasi-linearization. Linearizing a set of nonlinear dynamic equations around the point of interest is attractive because the results can be analyzed easily with the well-established mathematical theories. However, if the global behavior is to be analyzed more precisely, the nonlinear process dynamics should be retained to reflect the process structure. This, in turn, requires the formulation of a nonlinear time-lag kernel. The general steps taken are briefly illustrated on following pages without an extensive analysis.

With a slight modification, it is possible to extend the time-lag kernel methodology to a nonlinear system. In contrast to the standard linear treatment, there is no analogous all-powerful method to handle a nonlinear system. The method of *Volterra series* is used in this discussion because it is a logical extension of the linear kernel representation. The following treatment has the advantage that a linear system can be viewed as a specialized case of the nonlinear system.

The input and output relationship for a dynamic nonlinear process can be represented with the following series of time-lag kernel integrals that are similar to

the original time-lag kernel already considered.

$$
\begin{aligned}
y(t) = & \int_{-\infty}^{t} \mu(h) k_1(t-h)\,dh \\
& + \int_{-\infty}^{t}\int_{-\infty}^{t} \mu(h_1)\mu(h_2) k_2(t-h_1, t-h_2)\,dh_1 dh_2 \\
& + \int_{-\infty}^{t}\int_{-\infty}^{t}\int_{-\infty}^{t} \mu(h_1)\mu(h_2)\mu(h_3) k_3(t-h_1, t-h_2, t-h_3)\,dh_1 dh_2 dh_3 \\
& + \dots \\
& + \underbrace{\int_{-\infty}^{t} \dots \int_{-\infty}^{t}}_{l \text{ times}} \mu(h_1)\mu(h_2)\dots\mu(h_l) k_l(t-h_1, t-h_2, \dots, t-h_l)\,dh_1 dh_2 \dots dh_l \\
= & \int_{0}^{\infty} \mu(t-h) k_1(h)\,dh \\
& + \int_{0}^{\infty}\int_{0}^{\infty} \mu(t-h_1)\mu(t-h_2) k_2(h_1, h_2)\,dh_1 dh_2 \\
& + \int_{0}^{\infty}\int_{0}^{\infty}\int_{0}^{\infty} \mu(t-h_1)\mu(t-h_2)\mu(t-h_3) k_3(h_1, h_2, h_3)\,dh_1 dh_2 dh_3 \\
& + \dots \\
& + \underbrace{\int_{0}^{\infty} \dots \int_{0}^{\infty}}_{l \text{ times}} \mu(t-h_1)\mu(t-h_2)\dots\mu(t-h_l) k_l(h_1, h_2, \dots, h_l)\,dh_1 dh_2 \dots dh_l
\end{aligned}
\tag{2.9.17}
$$

This can be considered as a series of convolution integrals, each representing the degree of interaction between the input impulses introduced into the system at different times. The first term groups all the linear, noninteracting elements together. In this term, the contribution to $y_{\text{linear}}(t)$ from two individual inputs $\mu_1(t)$ and $\mu_2(t)$ can be computed separately and later added. In short, the signals can be superposed for the linear term.

$$
y_{\text{linear}}(t) = \int_{-\infty}^{t} \mu(h) k_1(t-h)\,dh \tag{2.9.18a}
$$

However, when there are quadratic, cubic, or higher-degree interactions between the signals, higher-degree convolution integrals should be included in the description of

the output as a function of the input.

$$y_{\text{quadratic}}(t) = \int_{-\infty}^{t}\int_{-\infty}^{t} \mu(h_1)\mu(h_2)k_2(t-h_1,t-h_2)dh_1dh_2 \qquad (2.9.18b)$$

$$y_{\text{cubic}}(t) = \int_{-\infty}^{t}\int_{-\infty}^{t}\int_{-\infty}^{t} \mu(h_1)\mu(h_2)\mu(h_3)k_3(t-h_1,t-h_2,t-h_3)dh_1dh_2dh_3 \qquad (2.9.18c)$$

$$\vdots$$

$$y_{l\text{th degree}}(t) = \underbrace{\int_{-\infty}^{t}\cdots\int_{-\infty}^{t}}_{l \text{ times}} \mu(h_1)\mu(h_2)\ldots\mu(h_l)k_l(t-h_1,t-h_2,\ldots,t-h_l)dh_1dh_2\ldots dh_l, \qquad (2.9.18l)$$

where $k_l(t_1,t_2,\ldots,t_l)$ is an impulse response of the lth degree, and the time-lag kernel integral containing $k_l(\ldots)$ is a functional of the lth degree. Thus, the overall output is the summation of all these linear and nonlinear effects. This is analogous to an lth degree polynomial.

$$y(t) = y_{\text{linear}}(t) + y_{\text{quadratic}}(t) + y_{\text{cubic}}(t) + \ldots + y_{l\text{th degree}} \qquad (2.9.19)$$

If the disturbance is imposed on an established steady-state, *i.e.*, if $u(t) = 0$ for $t < 0$, then the kernel integrals can be simplified slightly by replacing the limits of

integration from $-\infty \Rightarrow t$ to $0 \Rightarrow t$.

$$
\begin{aligned}
y(t) = & \int_0^t \mu(h)k_1(t-h)dh \\
& + \int_0^t \int_0^t \mu(h_1)\mu(h_2)k_2(t-h_1, t-h_2)dh_1dh_2 \\
& + \int_0^t \int_0^t \int_0^t \mu(h_1)\mu(h_2)\mu(h_3)k_3(t-h_1, t-h_2, t-h_3)dh_1dh_2dh_3 \\
& + \dots \\
& + \underbrace{\int_0^t \dots \int_0^t}_{l \text{ times}} \mu(h_1)\mu(h_2)\dots\mu(h_l)k_l(t-h_1, t-h_2, \dots, t-h_l)dh_1dh_2\dots dh_l \\
= & \int_0^t \mu(t-h)k_1(h)dh \\
& + \int_0^t \int_0^t \mu(t-h_1)\mu(t-h_2)k_2(h_1, h_2)dh_1dh_2 \\
& + \int_0^t \int_0^t \int_0^t \mu(t-h_1)\mu(t-h_2)\mu(t-h_3)k_3(h_1, h_2, h_3)dh_1dh_2dh_3 \\
& + \dots \\
& + \underbrace{\int_0^t \dots \int_0^t}_{l \text{ times}} \mu(t-h_1)\mu(t-h_2)\dots\mu(t-h_l)k_l(h_1, h_2, \dots, h_l)dh_1dh_2\dots dh_l
\end{aligned}
\tag{2.9.20}
$$

Each of the kernel functions satisfies:

$$k_i(t_1, t_2, \dots, t_i) = 0 \quad \text{for } t_i < 0 \tag{2.9.21}$$

$$\lim_{t_1\to\infty} k_i(t_1, t_2, \dots, t_i) = \lim_{t_2\to\infty} k_i(t_1, t_2, \dots, t_i) = \dots = \lim_{t_i\to\infty} k_i(t_1, t_2, \dots, t_i) = 0 \tag{2.9.22}$$

$$k_{i,\text{sym}}(t_1, t_2, \dots, t_i) = \frac{1}{n!}\Big[k_{i,\text{asym}}(t_1, t_2, \dots, t_i) + \dots + k_{i,\text{asym}}(t_i, t_{i-1}, \dots, t_1)\Big] \tag{2.9.23}$$

The first of the above three conditions physically means that the current states are not affected by future events. The second condition is needed for stability and convergence. The third one is a symmetry condition which states that a high-degree time-lag kernel is either symmetrical with respect to interchanging of the

time variables or can be reduced to such a symmetrical form by combining the asymmetrical ones.

The original linear time-lag formulation can be obtained by setting all higher-degree terms in the nonlinear time-lag kernel equation (2.9.17) to 0, *i.e.*, $y_{\text{quadratic}}(t) \equiv 0$, $y_{\text{cubic}}(t) \equiv 0$, ..., and $y_{l\text{th degree}}(t) \equiv 0$. Equation (2.9.17) is a truncated functional power series analogous to a truncated power series. This analogy can be clarified by considering a limiting case of $k_1(t) = b_1\delta(t)$, $k_2(t_1,t_2) = b_2\delta(t_1)\delta(t_2)$, $k_3(t_1,t_2,t_3) = b_3\delta(t_1)\delta(t_2)\delta(t_3)$, ..., etc. This is equivalent to the special case of a system with no dynamic elements. Substituting these relationships into Equation (2.9.17) results in a simple input-output relationship described by a polynomial of degree l:

$$y(t) = b_1\mu(t) + b_2\mu(t)\mu(t) + b_3\mu(t)\mu(t)\mu(t) + \ldots + b_l\,[\mu(t)]^l \tag{2.9.24}$$

As $l \to \infty$, one obtains:

$$y(t) = \sum_{i=1}^{\infty} \underbrace{\int_{-\infty}^{t} \cdots \int_{-\infty}^{t}}_{i \text{ times}} \mu(h_1)\mu(h_2)\ldots\mu(h_i)k_i(t-h_1, t-h_2, \ldots, t-h_i)dh_1dh_2\ldots dh_i \tag{2.9.25}$$

Although, in principle, any continuous nonlinear dynamic process can be represented this way, just as any continuous function can be represented by an infinitely dimensioned power series expansion:

$$f(t) = a_0 + a_1t + a_2t^2 + a_2t^3 + a_2t^4 + \ldots \tag{2.9.26}$$

Similarly, most continuous functions cannot be exactly represented by a finitely dimensioned polynomial, an arbitrary functional generally cannot be expressed exactly with a finite number of kernel integrals. Just as one seldom uses more than a few terms in the power expansion to evaluate the function $f(t)$, only a few terms

are usually needed to give a reasonably accurate system response for a nonlinear system, especially when the nonlinear expansion is carried out around the point of interest. In fact, because mathematical manipulations can become cumbersome, it is not advisable to use more than the first two or three terms.

The dynamic equation and initial conditions satisfied by the kernel function can be derived if the corresponding nonlinear dynamic equation for the process is known, as indicated by the following differential equation having the general form:

$$D\{y(t)\} + N\left\{y(t), \frac{dy(t)}{dt}, \ldots, \frac{dy^l(t)}{dt^l}\right\} = D\{\mu(t)\} + M\left\{\mu(t), \frac{d\mu(t)}{dt}, \ldots, \frac{d\mu^l(t)}{dt^l}\right\}, \tag{2.9.27}$$

where $D\{\cdot\}$ is the linear differential operator, and $N\{\cdot\}$ and $M\{\cdot\}$ are the residual nonlinear differential operators. The first step is to take the derivatives of Equation (2.9.17), using Leibniz's rule of differentiation of a functional. The resulting expressions for $y(t)$, $\frac{dy(t)}{dt}$, ..., $\frac{dy^l(t)}{dt^l}$ are substituted into the above equation. Following the same procedure used in Section 2.2, similar terms are collected and compared to arrive at the differential equation and initial conditions that describe the kernel function.

One of the advantages of using a Volterra series, besides the similarities in the form of kernel integrals, is that the nonlinear dynamic equation can be easily manipulated in a transformed domain, either Fourier transform or Laplace transform. As an example, the use of multidimensional Laplace transforms will be demonstrated. An l-dimensional Laplace transform $\mathcal{L}\{\cdot\}$ for a function $f(t_1, t_2, \ldots, t_l)$ is defined as:

$$\begin{aligned} F_l(s_1, s_2, \ldots, s_l) &= \mathcal{L}\{f(t_1, t_2, \ldots, t_l)\} \\ &\equiv \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{l \text{ times}} f(t_1, t_2, \ldots, t_l) \exp\left[\sum_{i=1}^{l} s_i t_i\right] dt_1 dt_2 \ldots dt_l. \end{aligned} \tag{2.9.28}$$

Thus, the lth degree kernel integral:

$$\begin{aligned} & y_{l\text{th degree}}(t) \\ &= \underbrace{\int_{-\infty}^{t} \cdots \int_{-\infty}^{t}}_{l \text{ times}} \mu(h_1)\mu(h_2)\ldots\mu(h_l)k_l(t-h_1, t-h_2, \ldots, t-h_l)dh_1dh_2\ldots dh_l \\ &= \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{l \text{ times}} \mu(t-h_1)\mu(t-h_2)\ldots\mu(t-h_l)k_l(h_1, h_2, \ldots, h_l)dh_1dh_2\ldots dh_l \end{aligned} \tag{2.9.29}$$

can be transformed into:

$$Y_{l\text{th degree}}(s) = K_{l\text{th degree}}(s_1, s_2, \ldots, s_l)M(s_1)M(s_2)\ldots M(s_l), \tag{2.9.30}$$

where

$$Y_{l\text{th degree}}(s) = \mathcal{L}\{y_{l\text{th degree}}(t)\} \tag{2.9.31a}$$

$$K_{l\text{th degree}}(s_1, s_2, \ldots, s_l) = \mathcal{L}\{k_{l\text{th degree}}(t_1, t_2, \ldots, t_l)\} \tag{2.9.31b}$$

$$M(s_1) = \mathcal{L}\{\mu(t_1)\} \tag{2.9.31c}$$

A common technique used to simplify the manipulation of a multidimensional Laplace transform is the *association of variables*. In place of a single real time variable t, this technique temporarily uses a multiple of time variables t_1 for s_1, t_2 for s_2, ..., and t_l for s_l. At the end, all these intermediate time variables are set to be t because time is only one-dimensional:

$$t_1 = t_2 = \ldots = t_l = t. \tag{2.9.32}$$

Likewise, the variable in the transformed domain that corresponds to t is s, which can be obtained by setting:

$$s_1 + s_2 + \ldots + s_l = s. \tag{2.9.33}$$

By using this shortcut, the inverse Laplace transform $\mathcal{L}^{-1}\{\cdot\}$ of a two-dimensional function can be obtained by first working with two separate time variables t_1 and t_2.

$$f_2(t_1,t_2)=\frac{1}{(2\pi i)^2}\int_{c_1-i\infty}^{c_1+i\infty}\int_{c_2-i\infty}^{c_2+i\infty}F_2(s_1,s_2)\exp\left(s_1t_1+s_2t_2\right)ds_1ds_2. \quad (2.9.34)$$

These time variables are then changed to t ($t_1 \Rightarrow t$ and $t_2 \Rightarrow t$), and $s_1+s_2=s$:

$$\begin{aligned} f_2(t) =& \mathcal{L}^{-1}\{F_2(s)\} \\ =& \frac{1}{2\pi i}\int_{c_2-i\infty}^{c_2+i\infty}\underbrace{\left\{\frac{1}{2\pi i}\int_{c_1-i\infty}^{c_1+i\infty}F_2(s-s_2,s_2)ds_2\right\}}_{F_2(s)}\exp(st)ds \\ =& \frac{1}{2\pi i}\int_{c_2-i\infty}^{c_2+i\infty}F_2(s)\exp(st)ds. \end{aligned}, \quad (2.9.35)$$

where

$$F_2(s)=\frac{1}{2\pi i}\int_{c_1-i\infty}^{c_1+i\infty}F_2(s-s_2,s_2)ds_2. \quad (2.9.36)$$

A similar process can be repeated to reduce $F_l(s_1,s_2,\ldots,s_l)$ to $F_l(s)$. It should be cautioned that other variations in the definitions of the Laplace transform pairs exist, and these definitions should be followed consistently.

One of the simplest nonlinearities is the multiplication of two factors:

$$\begin{aligned} y(t) &= \left[\int_{-\infty}^{t}\mu(h)k_a(t-h)dh\right]\cdot\left[\int_{-\infty}^{t}\mu(h)k_b(t-h)dh\right] \\ &= \int_{-\infty}^{t}\int_{-\infty}^{t}\mu(h_1)\mu(h_2)k_a(t-h_1)k_b(t-h_2)dh_1dh_2. \end{aligned} \quad (2.9.37)$$

The Laplace transform representation is:

$$Y(s_1,s_2)=K_a(s_1)K_b(s_2)M(s_1)M(s_2), \quad (2.9.38a)$$

which is the same as:

$$Y(s)=\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}K_a(s-s_2)K_b(s_2)M(s-s_2)M(s_2)ds_2, \quad (2.9.38b)$$

where

$$K_a(s_1) = \mathcal{L}\{k_a(t_1)\} \tag{2.9.39b}$$

$$K_b(s_2) = \mathcal{L}\{k_b(t_2)\} \tag{2.9.39c}$$

$$K_{ab}(s_1, s_2) = K_a(s_1)K_b(s_2) = \mathcal{L}\{k_a(t_1)k_b(t_2)\} \tag{2.9.39c}$$

An example of the dynamic equation that contains the above quadratic term is:

$$\frac{dy(t)}{dt} + y(t) + a[y(t)]^2 = \mu(t) \tag{2.9.40}$$

The differential equation satisfied by the time-lag kernel is simply the homogeneous portion of the above equation:

$$\frac{dk(t)}{dt} + k(t) + a[k(t)]^2 = 0 \tag{2.9.41}$$

The Laplace transform of the kernel differential equation is:

$$\underbrace{(s_1+s_2)K_2(s_1,s_2)}_{\frac{dk(t)}{dt}} + \underbrace{K_2(s_1,s_2)}_{k(t)} + \underbrace{a\left(\frac{1}{s_1+1}\right)\left(\frac{1}{s_2+1}\right)}_{\text{Quadratic term}} = 0 \tag{2.9.42}$$

After some algebraic manipulations, the above Laplace transformed nonlinear time-lag kernel is reduced to:

$$K_2(s_1, s_2) = -\frac{a}{(s_1+s_2+1)(s_1+1)(s_2+1)} \tag{2.9.43}$$

The above kernel is inverse-transformed to yield:

$$k_2(t_1, t_2) = a\left\{\exp\{-(t_1+t_2)\} - \exp\left\{-\frac{1}{2}[(t_1+t_2) - |t_1-t_2|]\right\}\right\} \tag{2.9.44}$$

Finally, t_1 and t_2 are substituted with t to give:

$$k_2(t) = a\{e^{-2t} - e^{-t}\} \tag{2.9.45}$$

In summary, the kernel function in the time domain becomes complicated for most nonlinear processes. However, the corresponding expressions in the transformed domain are generally simple if a Volterra series is used to describe the nonlinear behavior.

2.10 DISCUSSION

Microbial behavior depends not only on the present state of the environment but on past histories as well. This is the main reason for the inadequacy of the simple set of Equations (2.2.5) and (2.2.6). The dependence of a culture on its past history is manifested in the presence of a lag phase in the beginning of a batch cultivation. It is also present during the transient of continuous fermentors resulting from, among others, a shift-up of nutrient concentration. Such lag has often been explained in terms of the need to synthesize the necessary pools of enzymes and intermediates before the rate of substrate utilization is adjusted to the changed conditions.

The time-lag model, in its general form, is quite flexible and is capable of explaining various observed phenomena. In addition to those mentioned in previous sections, it describes the asymmetric response of a biological process after the dilution rate or the feed concentration of the limiting nutrient is shifted up or down.

The behavior of a chemostat after a shift-up or a shift-down in the dilution rate is simulated with a Monod model and the corresponding time-lag model proposed in the preceeding sections. Figures 2.10.1–2.10.3 show the theoretical response of a chemostat after a shift-up in the dilution rate from 0.2 hr^{-1} to 0.4 hr^{-1}. Representative values found in a typical fermentation are used in this simulation: $\mu_m = 0.5$ hr^{-1}, $K_s = 1.0$ g/l, $Y_s = 0.5$ g/g, $a_0 = 0.2$, $T = 1.0$ hr and $S_f = 5.0$ g/l. As indicated in Figure 2.10.1, the biomass concentration predicted by the time-lag model is decreasing faster than that predicted by the classical Monod model. At the same time, Figure 2.10.2 shows that the limiting substrate concentration of the time-lag model is increasing faster than that of the Monod model. Because the instantaneous intrinsic specific growth rate is expressed as a function of the limiting substrate concentration existing at that time incident, the intrinsic specific growth

rate predicted by the time-lag model also leads that predicted by the Monod model. Note, however, that the observed specific growth rate, which is an integral of all the past growth rates weighed by the kernel function, lags considerably behind that calculated in the Monod model.

The corresponding response of the same chemostat after a shift-down in the dilution rate from 0.4 hr^{-1} to 0.2 hr^{-1} is shown in Figures 2.10.4–2.10.6. Again, after the perturbation, the biomass concentrations predicted by both models increase toward the same steady-state value, while the substrate concentrations of both models decrease toward the corresponding steady-state value. As in the shift-up simulation, the time-lag model resulted in a faster response in the changes in concentrations. Similarly, the intrinsic specific growth rate in the time-lag model leads the specific growth rate in the Monod model, which, in turn, is followed by the observed specific growth rate in the time-lag model.

Thus, the word "lag" in the time-lag model does not describe the quickness in the response in concentrations, which usually constitutes the primary measurements. Rather, it describes the sluggish response in the specific growth rate when the time-lag kernel function is associated with the specific growth rate. In general, the time-lag model predicts a quicker initial response of the biomass and substrate concentrations toward the eventual steady-state values in both cases where the dilution rate is either shifted up or shifted down.

The concentration response in a shift-up or shift-down is made much more quickly as the average time-lag of the kernel function is increased. As the initial response is made quicker, the system tends to overshoot, resulting in an oscillatory approach toward the eventual steady-state values. Sustained oscillation can result if the average time-lag exceeds the critical value. This is analogous to the response of a controller under proportional action, where a large controller gain brings a

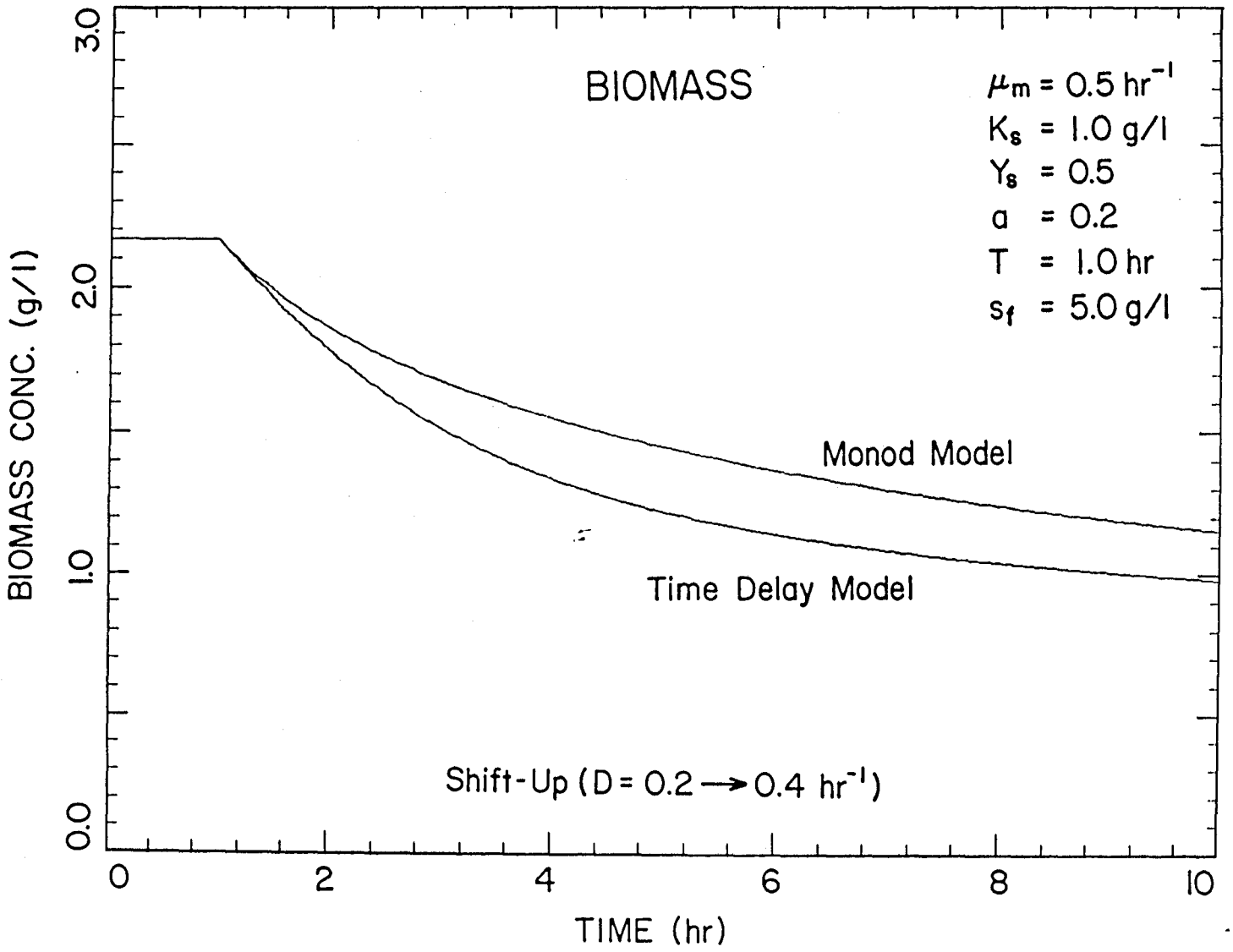


Figure 2.10.1. Biomass concentrations predicted by the time-lag model and Monod model after a shift-up in the dilution rate from 0.2 hr^{-1} to 0.4 hr^{-1} at 1.0 hr. The system is originally at a steady-state.

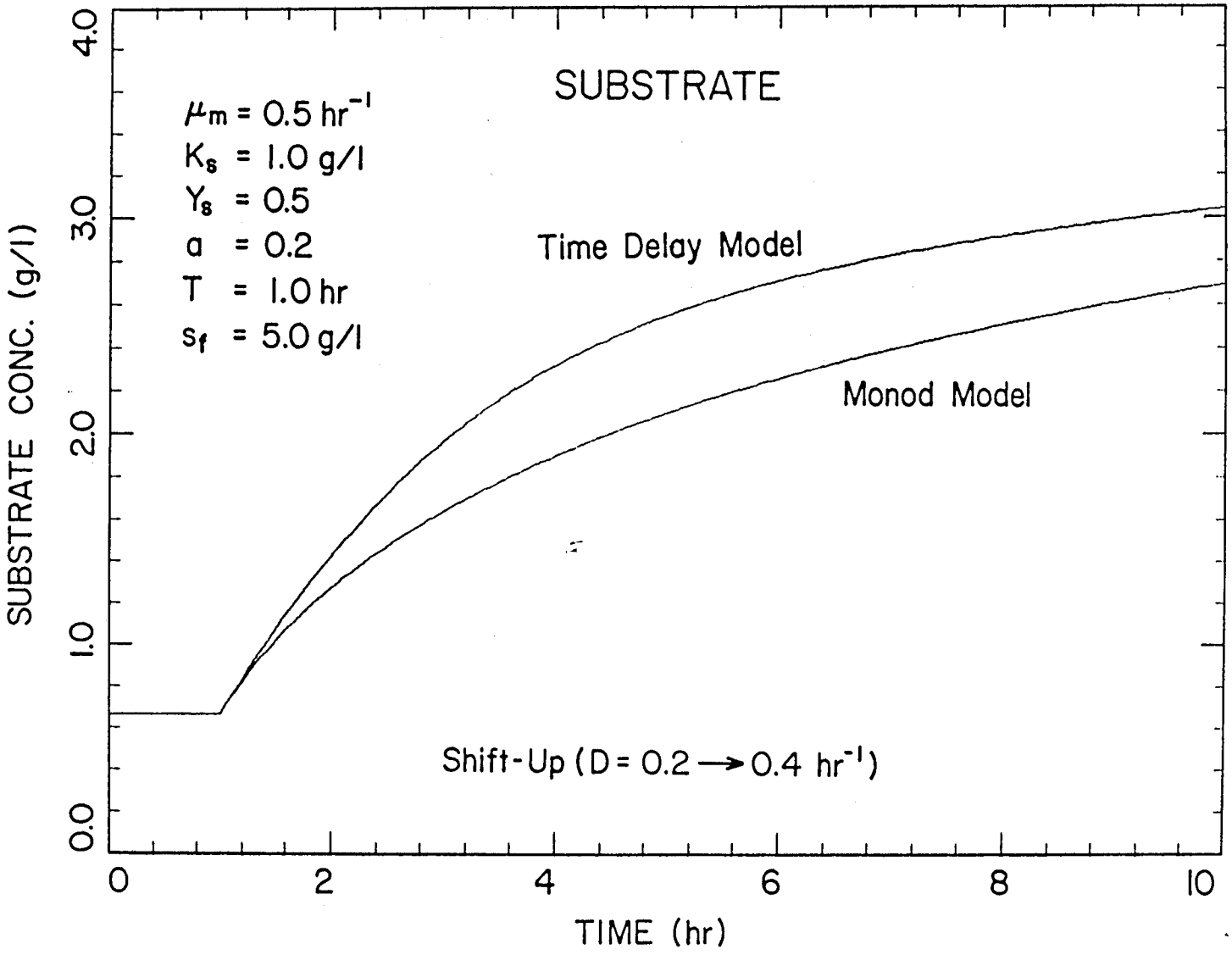


Figure 2.10.2. Substrate concentrations predicted by the time-lag model and Monod model after a shift-up in the dilution rate from 0.2 hr^{-1} to 0.4 hr^{-1} at 1.0 hr.

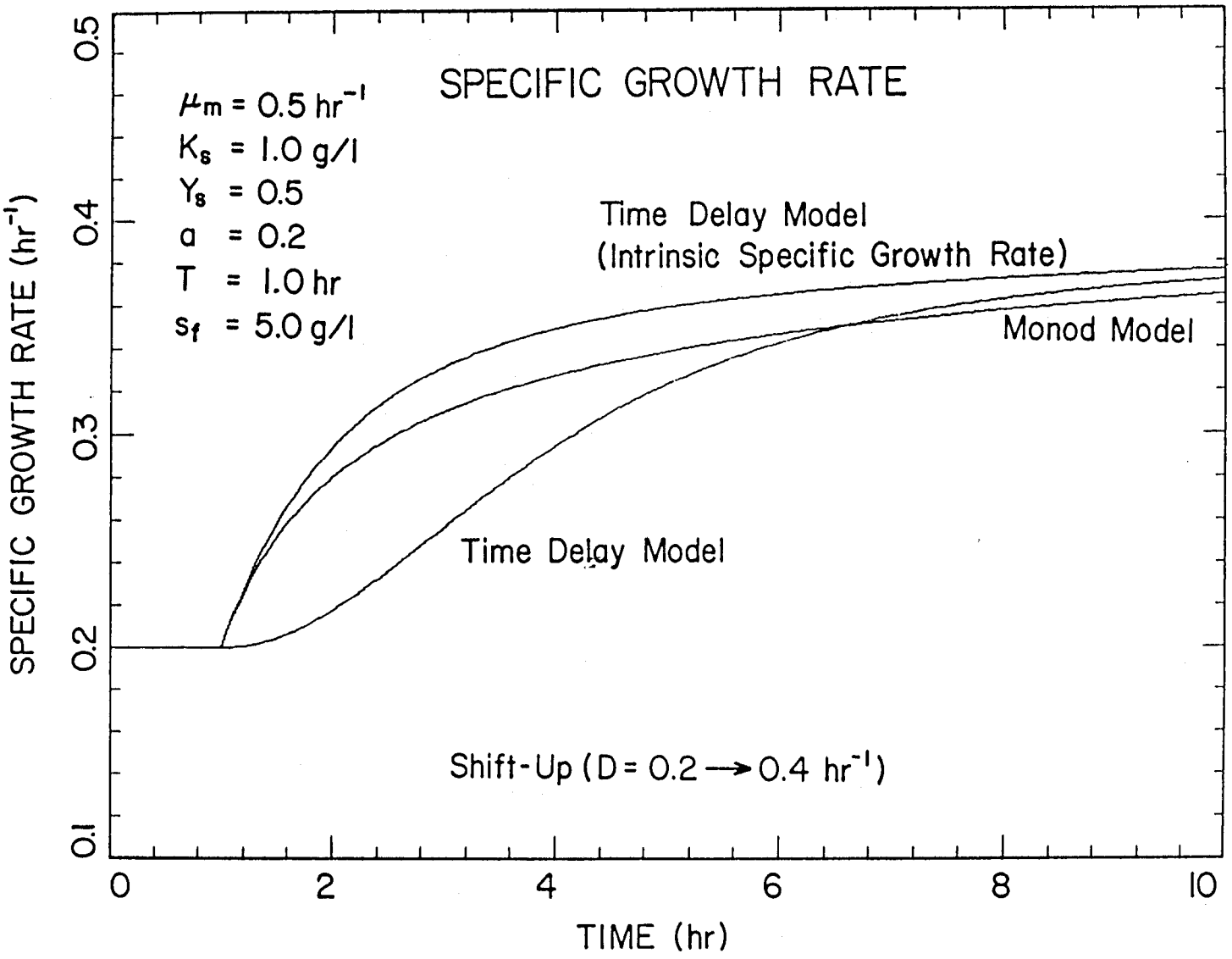


Figure 2.10.3. Specific growth rates predicted by the time-lag model and Monod model after a shift-up in the dilution rate from 0.2 hr^{-1} to 0.4 hr^{-1} at 1.0 hr. Note the initial order of response: μ (time-lag model) $\rightarrow$ μ (Monod model) $\rightarrow$ y (time-lag model).

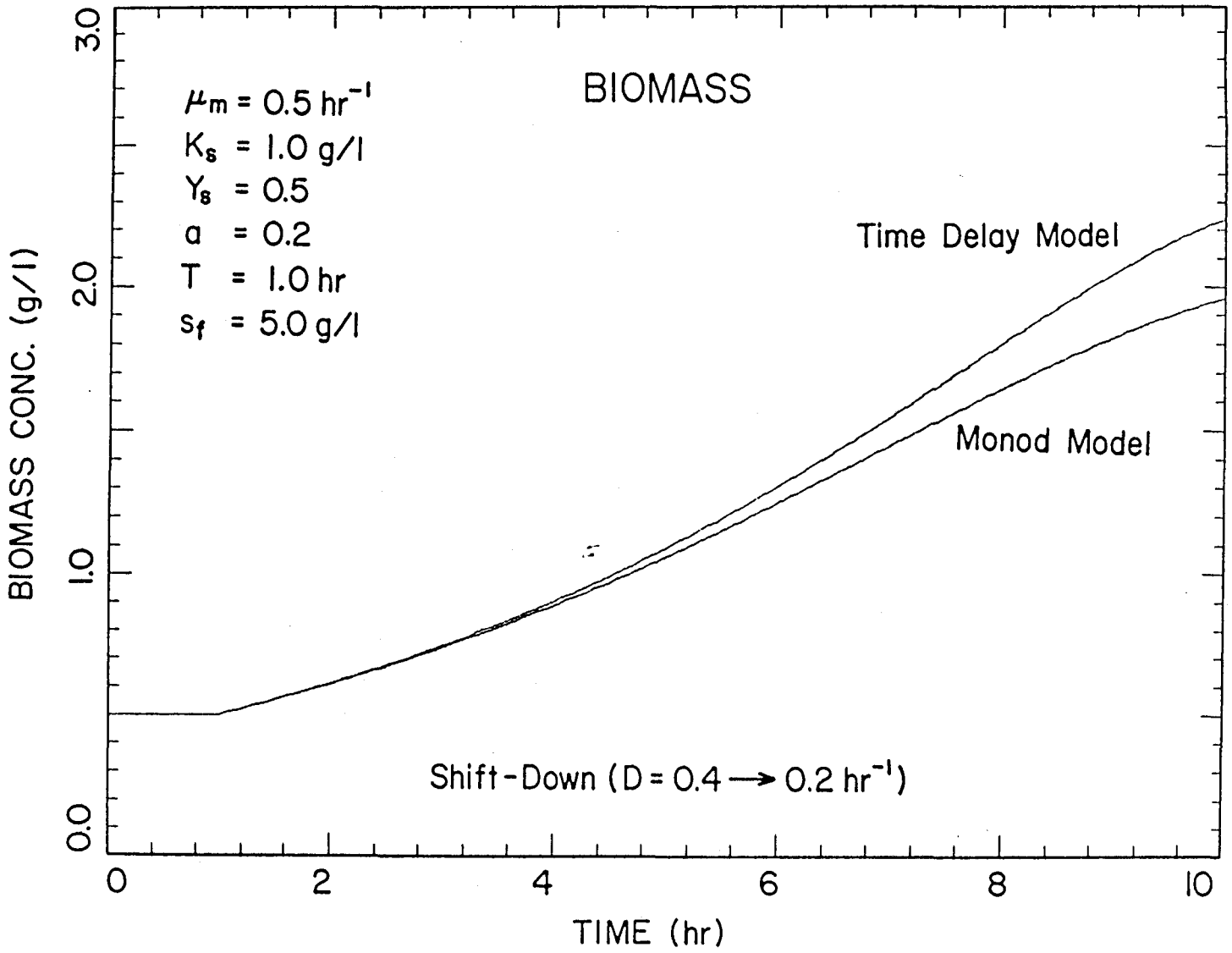


Figure 2.10.4. Biomass concentrations predicted by the time-lag model and Monod model after a shift-down in the dilution rate from 0.4 hr^{-1} to 0.2 hr^{-1} at 1.0 hr. The system is originally at a steady-state.

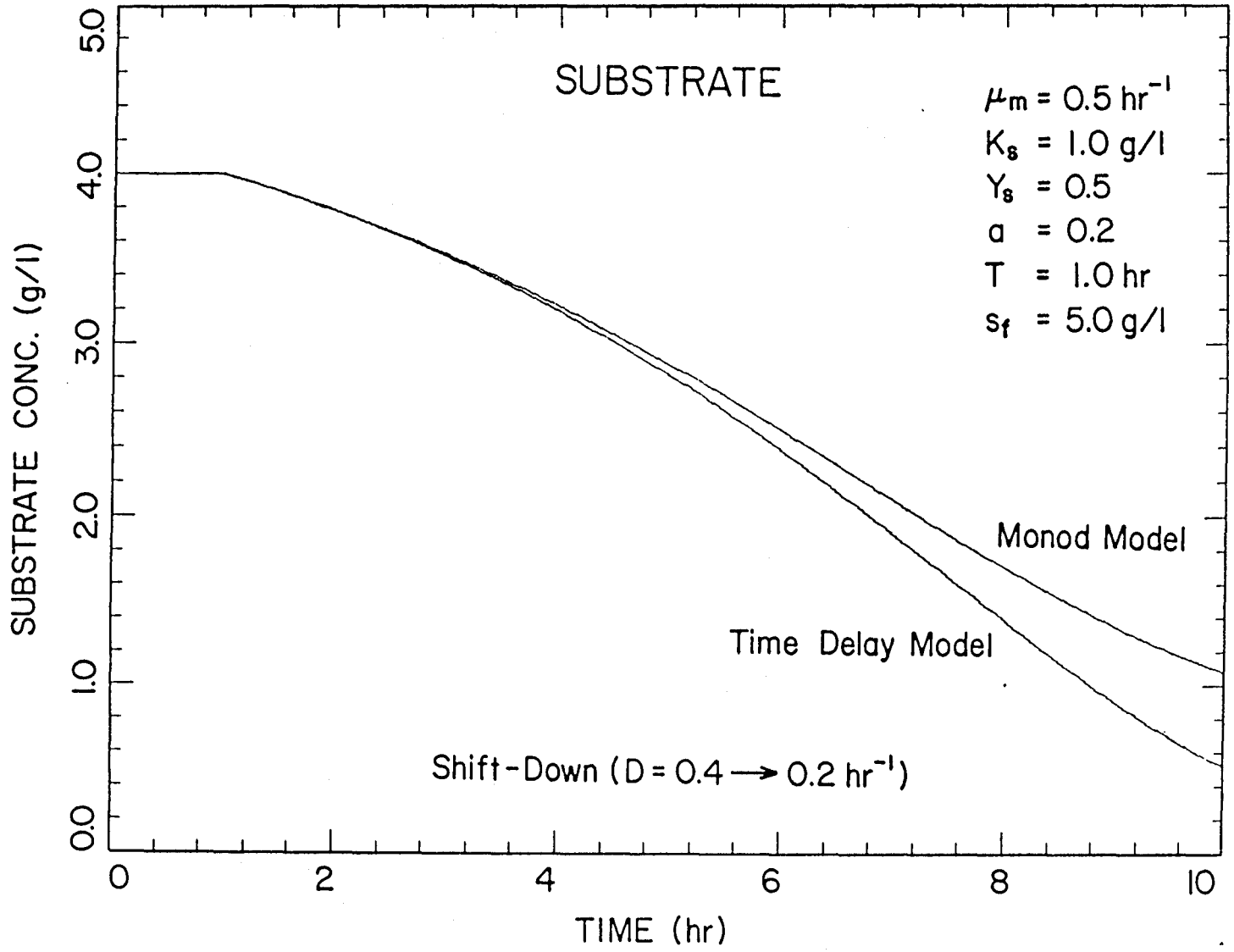


Figure 2.10.5. Substrate concentrations predicted by the time-lag model and Monod model after a shift-down in the dilution rate from 0.4 hr^{-1} to 0.2 hr^{-1} at 1.0 hr.

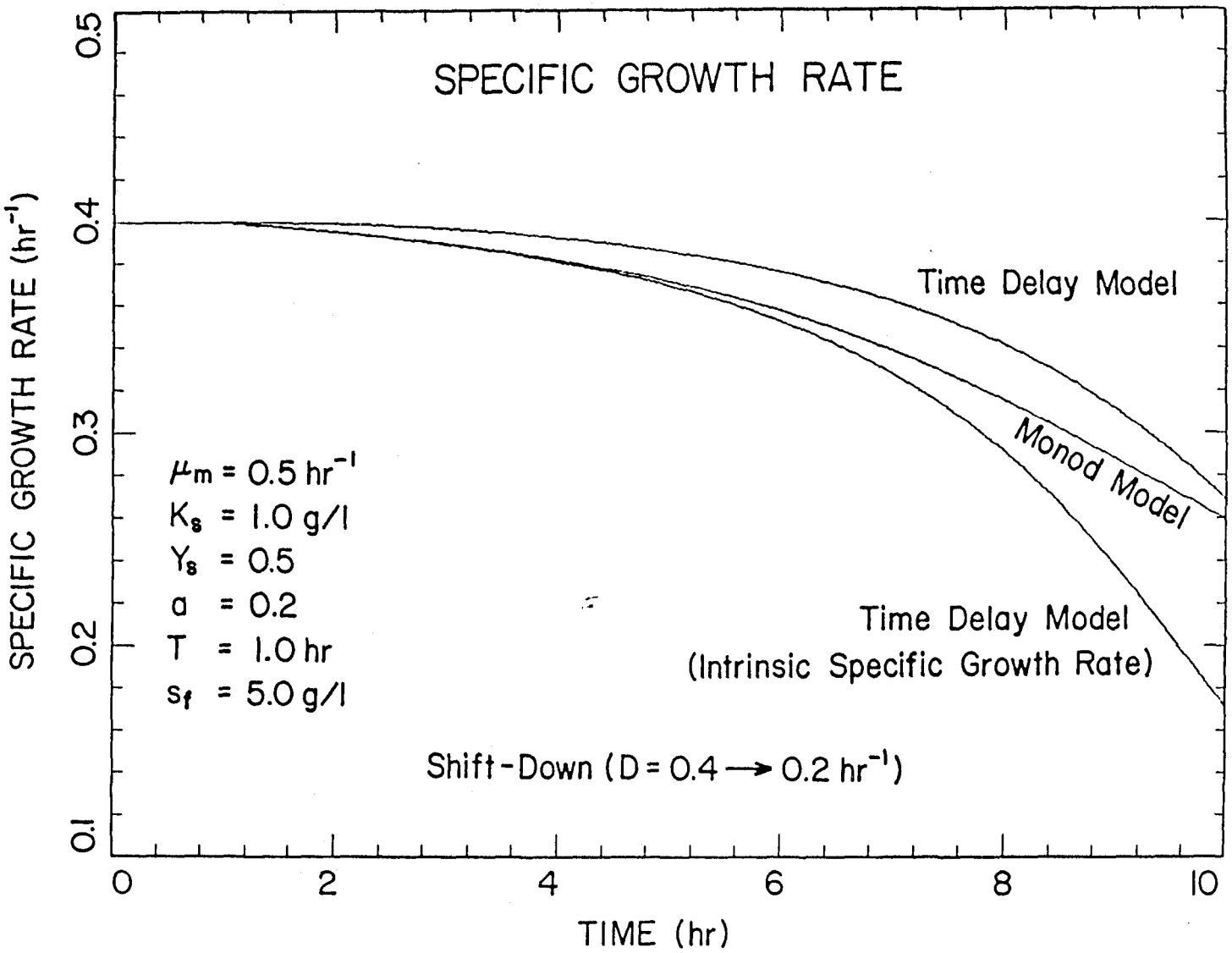


Figure 2.10.6. Specific growth rates predicted by the time-lag model and Monod model after a shift-down in the dilution rate from 0.4 hr^{-1} to 0.2 hr^{-1} at 1.0 hr. Note the initial order of response: μ (time-lag model) $\rightarrow$ μ (Monod model) $\rightarrow$ y (time-lag model).

quicker response and simultaneously makes the system more prone to overshoot and oscillation.

If the observed specific growth rate is plotted against the limiting substrate concentration, a hysteresis behavior can be observed. Figure 2.10.7 shows that the transient path followed during a shift-up operation by a system exhibiting time-lag in the specific growth rate does not coincide with the path predicted based on steady-state data, *i.e.*, the Monod model in this simulation study. Similarly, the system does not retrace the same path followed earlier as the dilution rate is decreased to the original value. The overall effect is the creation of an area enclosed by the shift-up and shift-down paths. Thus, the hysteresis behavior can also be easily explained with the inclusion of a time-lag kernel in the specific growth rate.

In summary, much benefit can be derived from the recognition of time-lag. It is a well-known fact that time-lag can cause, among other undesirable problems, serious instability difficulties if it is neglected in a control strategy. Furthermore, an optimal control scheme may not be optimal if time-lag is not properly considered. Figure 2.10.8 shows a hypothetical run in a bioreactor. The data collected during the short transient period after the start-up can be used to update the shape of the kernel and other model parameters. Based on the updated model, model parameters, and objective functions, an optimal path can be calculated by an on-line computer. Occasionally, deliberate excursions can be introduced to update the kernel and model parameters if they are suspected of gradual changes during a long steady-state run. A simple, powerful model, such as the one proposed herein, and the state of parameter estimation scheme discussed in the previous sections can be used in the combined forward and feedback control of a bioreactor.

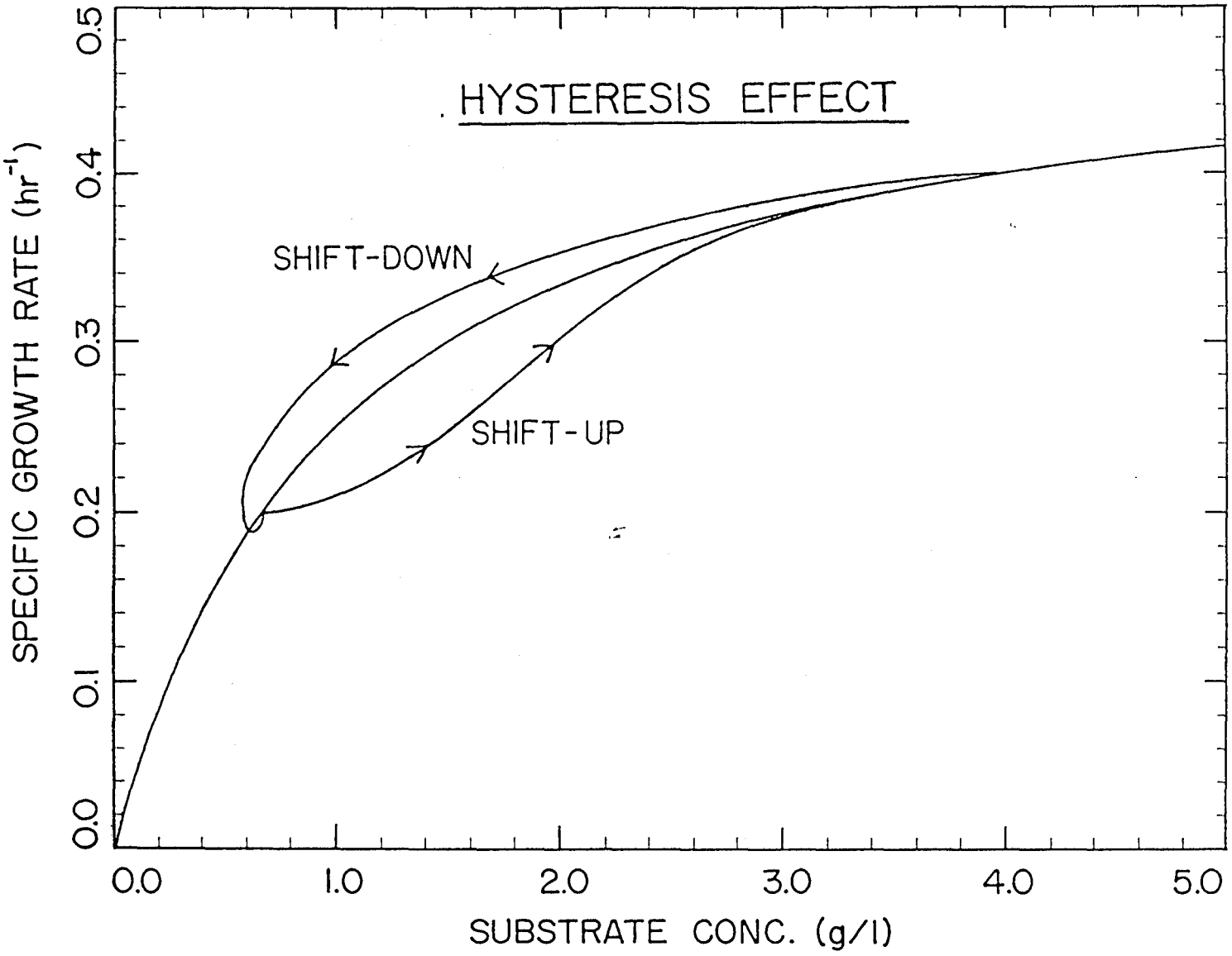


Figure 2.10.7. Hysteresis followed during a shift-up and a shift-down in the dilution rate.

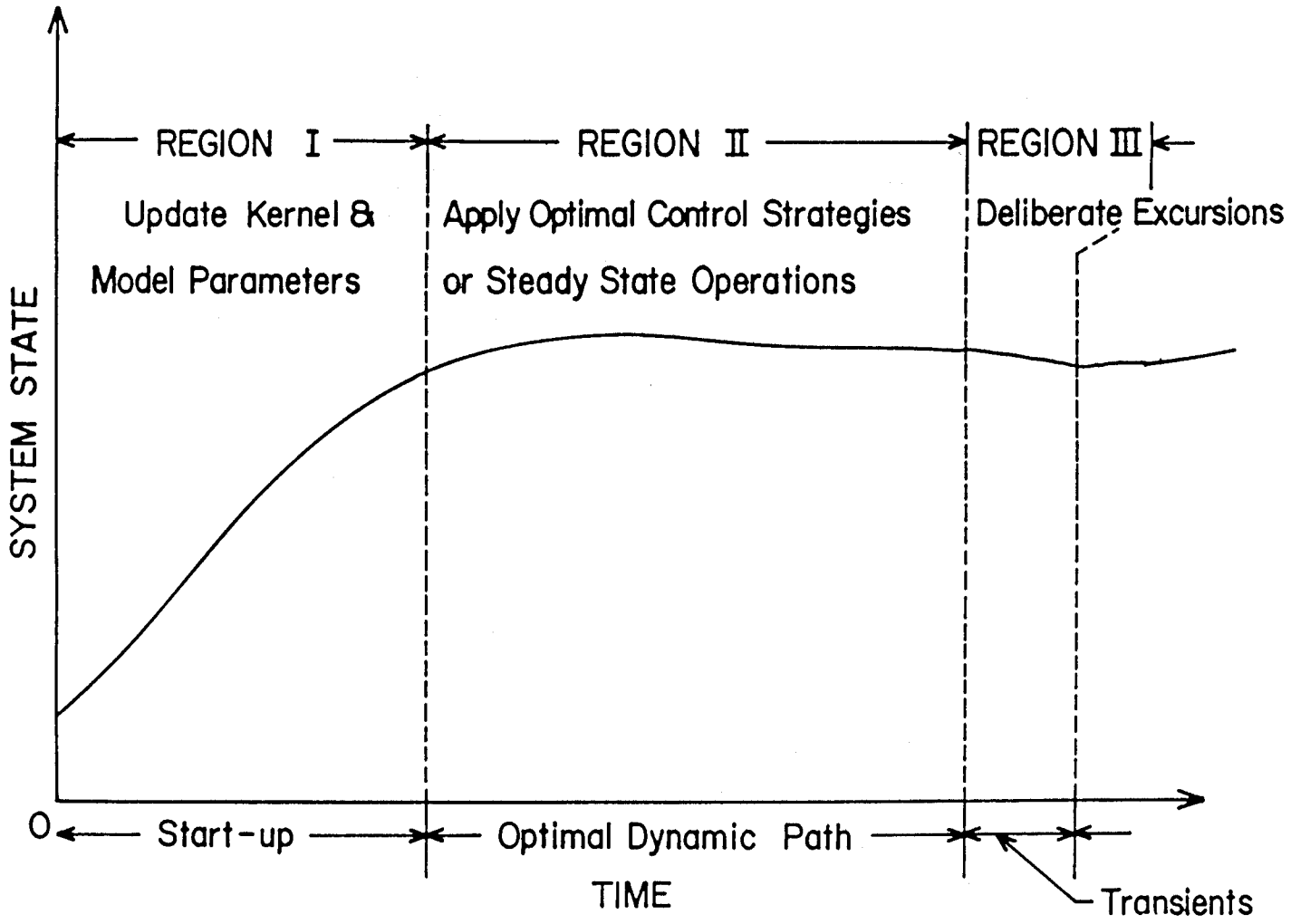


Figure 2.10.8. Use of transient data for the determination of kernel and model parameters during the start-up of a bioreactor and the subsequent utilization of model in control and optimization.

CHAPTER 3

EXPERIMENTAL METHODS AND MATERIALS

3.1 FERMENTOR AND GENERAL INSTRUMENTATION

The study of the dynamics of *Saccharomyces cerevisiae* was conducted in a modified New Brunswick Microferm fermentor at 30 °C and ambient pressure. The schematic of the entire experimental apparatus is shown in Figure 3.1.1. The general aseptic techniques so vital to the successful operation of a fermentor will not be discussed here. Rather, only those components unique to our system will be described in more detail in the following sections. This section will concentrate on the various on-line measurements, and Section 3.5 will concentrate on the off-line measurements used in this study.

Fermentor: The fermentor unit has factory built-in temperature and stirrer speed controllers. Either a 14-liter jar or a 5-liter jar can be fitted on this fermentor for the cultivation of microorganisms. Most of the fermentation studies were performed with a 5-liter jar; this volume was deemed most appropriate because it offered sufficient working volume so that the disturbance to the system caused by repeated sampling was not significant, and yet it is small enough to be handled easily with low nutrient consumption rate, especially under continuous operations. Various aspects of the fermentor were carefully controlled so that as many extraneous physical influences as possible were eliminated. This was done to ensure that the intrinsic biological behavior was selected. Wall growth, as checked visually, was nonexistent in all the experiments; this was accomplished by using a relatively high stirring rate (500 rpm).

The original fermentor head plate was heavily modified to allow for the installation of all the necessary ports. The following ports populated the head plate: a pH electrode port, a dissolved oxygen electrode port, an inoculation port, three separate wells for a thermometer, a temperature control thermistor, and a temperature recorder thermistor, an inlet and an outlet for the cooled/heated water used

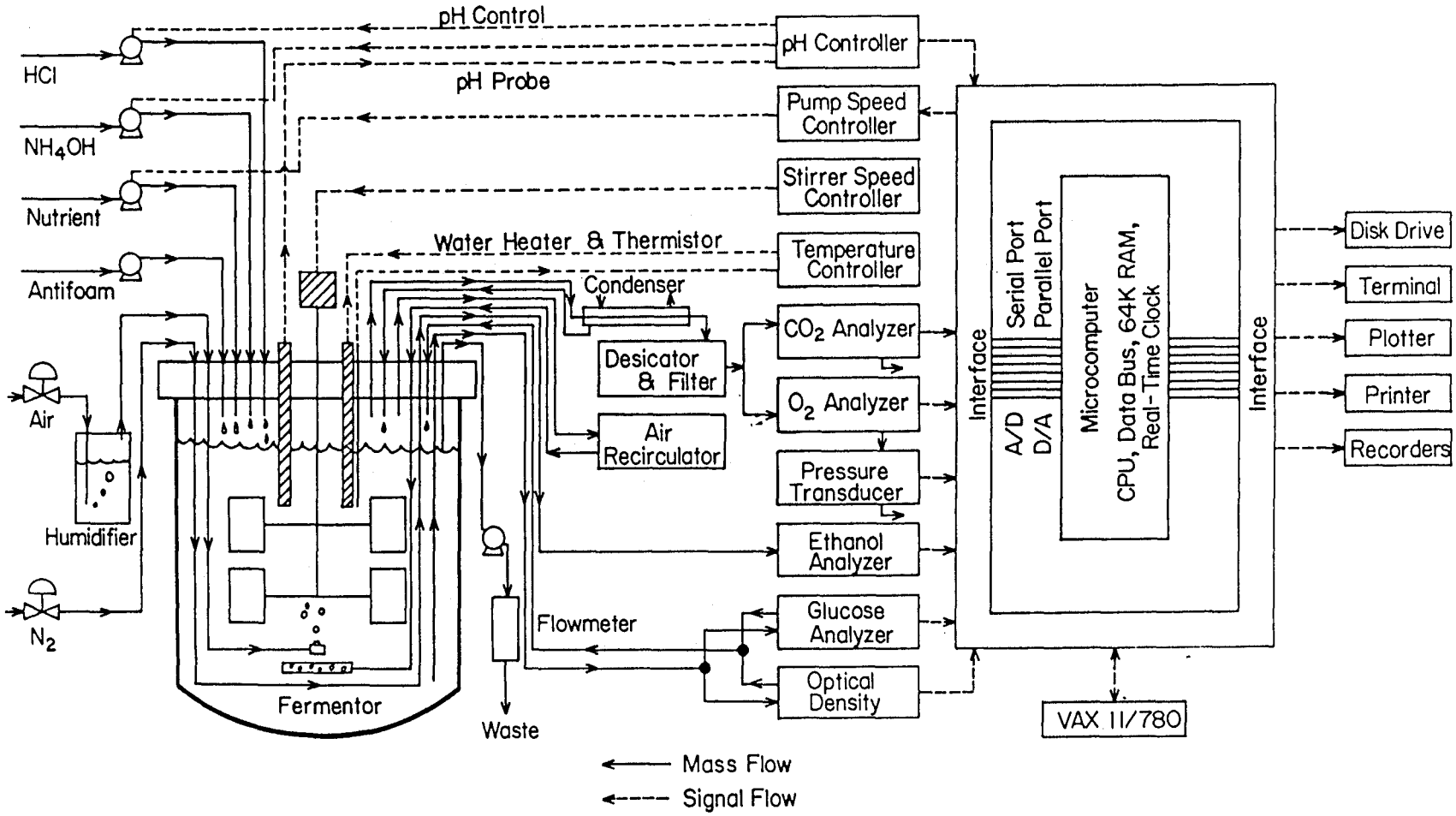


Figure 3.1.1. Schematic of the experimental apparatus.

in temperature control, a fresh air inlet, an exhaust air outlet with a condenser and the associated inlet and outlet for the continuously flowing cooling water, an inlet and an outlet for the air recirculator, an acid addition port, a base addition port, an antifoam addition port, a nutrient addition port, a continuous withdrawal port that was used to maintain a constant volume, an off-line sampling port, an inlet (return) port and an outlet (withdrawal) port to provide the recirculation of fermentation broth needed both by the continuously operated flow filter as part of the on-line glucose analyzer and by the optical flow-through cell of the spectrophotometer as part of the optical density measurement, an inlet and an outlet port for the Teflon tubing as part of the on-line ethanol analyzer. There was a total of twenty-four ports. Finally, a magnetic stirrer coupler also occupied a large part of the head plate.

Sampling: The off-line sampling port, shown in Figure 3.1.2, was secured onto the fermentor head plate by a 316 stainless steel Swagelok assembly. All other ports were also constructed similarly, except for the continuous withdrawal port, which had silicone ferrules instead of the usual 316 stainless-steel ones. This allowed the withdrawal tubing to slide along the swagelok assembly and made it possible to adjust the tip position of the outlet, thus, the level of the fermentation broth, while maintaining a sterile tight seal. By pumping the withdrawal line at a rate slightly larger than the nutrient feed rate, it was possible to maintain the working volume at a constant level. This was accomplished rather easily by coupling the waste withdrawal line and the nutrient feed line to the same peristaltic pump with two different sized pump heads. The working volume was initially approximately 4 liters for batch runs, and it was held at 2 liters for continuous runs. The broth levels were marked on the side of the fermentor jar at various points during a fermentation

run, and the actual volume was determined afterwards by transferring the entire contents into a graduated cylinder.

Waste Flow Rate: The waste flow rate was measured as often as necessary with a 100-ml graduated cylinder attached to the waste line as shown in Figure 3.1.3. After each measurement, the content of the graduated cylinder was emptied by hanging the cylinder in the inverted position. Because the entire flow rate measurement unit was autoclaved along with the fermentor, the risk of contamination arising from the waste flow rate measurement was totally eliminated. When not in use, the graduated cylinder was kept in the inverted position, and the waste stream was simply collected in a 23-liter waste jar. The waste jar was aseptically replaced with another sterilized one when it became full.

Foaming: Foaming was controlled by the continuous addition of autoclaved 0.2 g/l silicone antifoam (General Electric) by a peristaltic pump at such a rate that the antifoam concentration in the fermentor was maintained at approximately 2-10 mg/l.

Nutrient Addition: The filtration-sterilized nutrient was passed through another 0.2 μm filter and two breaker units before finally being added to the fermentor by a peristaltic pump (Cole-Parmer). The nutrient feed pump was controlled by a pump speed controller (Cole-Parmer) that was modified to be commanded either by a manually adjusted set-point or by a computer generated variable set-point that was sent through a D/A converter and an isolation amplifier energized by a 12-volt power supply. The electrical modification is described in Appendix A. For example, one of the calibration curves of the dilution rate versus the voltage output from the D/A converter is shown in Figure 3.1.4. Such calibration curves were generated for every run, because the actual flow rate delivered by the pump depended not only on the pump speed and the size of the pump head used but also on such artifacts as

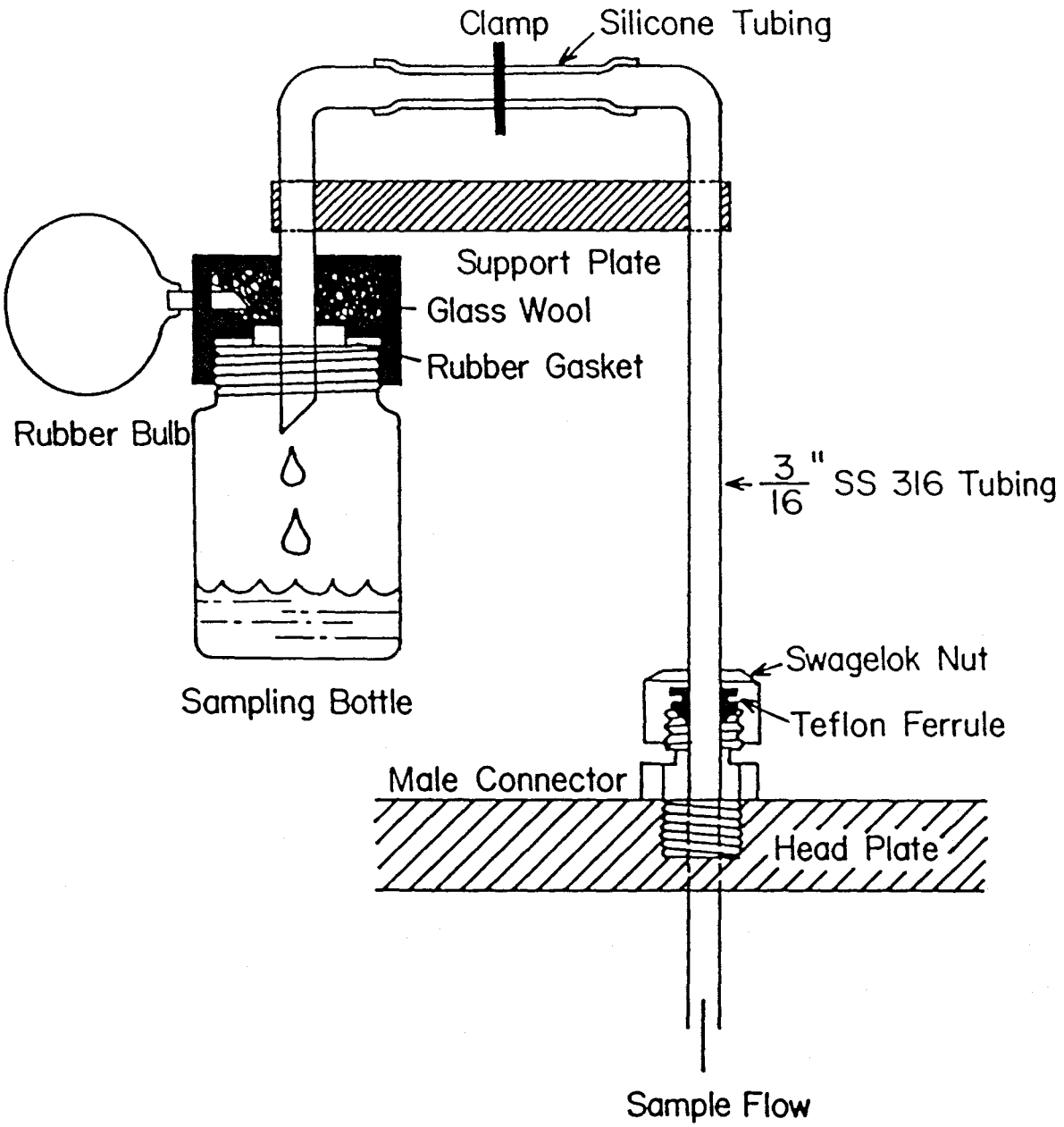


Figure 3.1.2. Construction of the sampling port. The rubber bulb is squeezed to force a fraction of the air in the sampling bottle into the fermentor. When the rubber bulb is released, the sample is withdrawn from the fermentor to replace the displaced air.

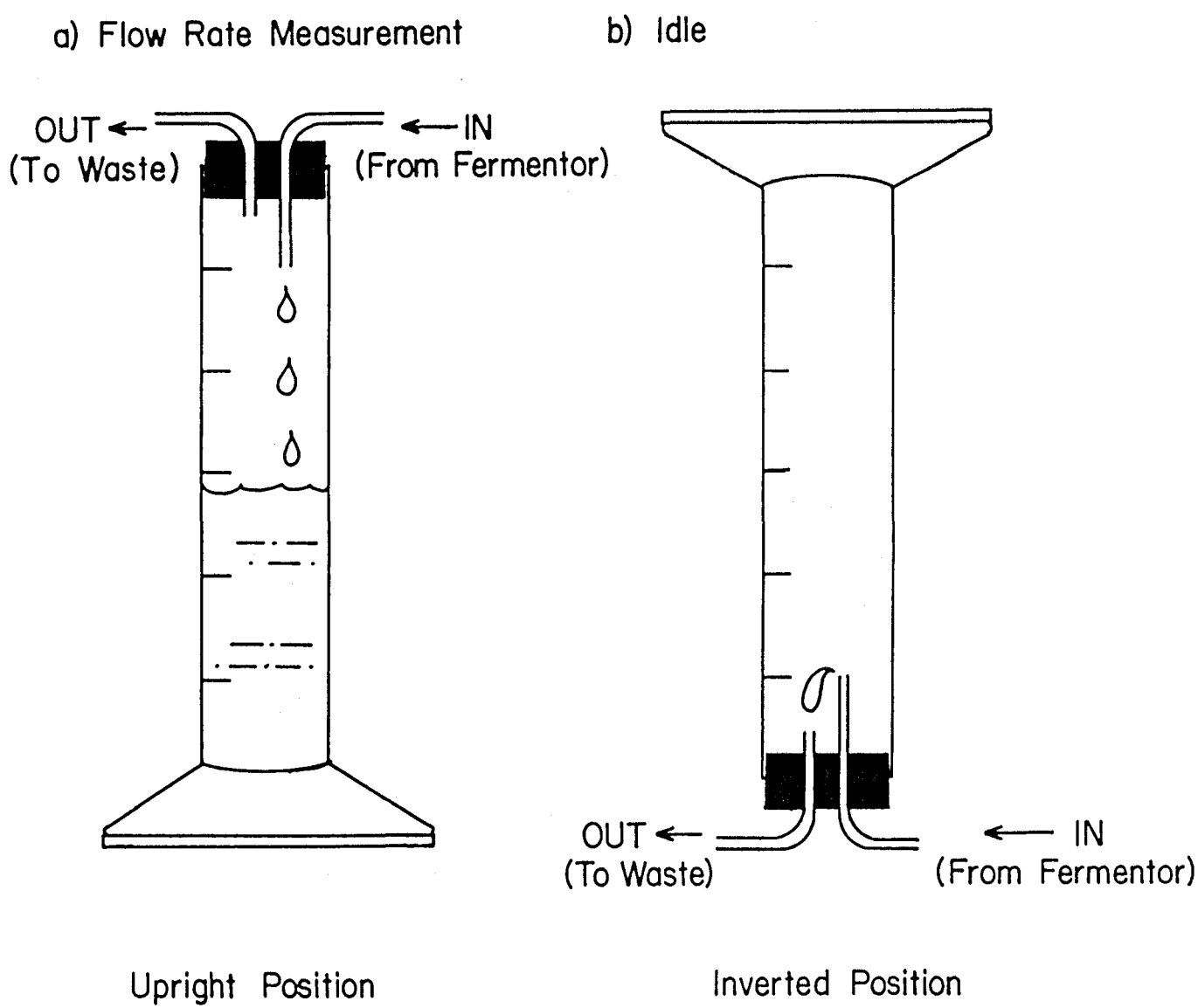


Figure 3.1.3. Aseptic flow rate measurement of the waste stream by using a steam-sterilized 100-ml graduated cylinder.

the tubing material, the length of service, the tension stress applied to the tubing section within the pump head, and the tubing installation procedure.

pH Measurement: A glass combination pH probe (Ingold) was used to monitor the pH in the fermentor. This electrode's response was found to be very stable, with a drift of only 0.02 pH unit for a duration of three months. Because the electrolyte in the pH electrode, in the absence of a pressurized electrode holder, boiled over during autoclaving, the electrolyte solution was withdrawn from the pH electrode prior to autoclaving and refilled afterward. The pH probe's signal was sensed by a pH controller (Chemtrix), which had the capability of maintaining the pH in the fermentor to 0.02 pH units within the set-point when operated in the expanded scale. In order to achieve this accuracy, the pH controller was slightly modified. For example, the original 1-turn potentiometers were replaced with respective 10-turn high-quality potentiometers and locking dials. HCl and NH_4OH solutions of known concentrations (approx. 0.1 N) were added by two separate small peristaltic pumps (Markson) to achieve a constant pH. The pH of the fermentor was set at 5.00 throughout this entire work, except for a few runs that were designed to follow the microorganism's transient response to a shift in the pH. The pH controller's on/off actions were monitored by measuring the voltages across the switch relays that energized the power outlets for acid and base addition pumps. A voltage drop of greater than 0.5 volt across the relay signaled that the pump was on; whereas, a voltage drop of less than 0.5 volt meant that the pump was off. The pH readings were also tapped; however, this proved to be unnecessary because the excellent controller performance practically ensured that the pH value was constant.

Dissolved Oxygen Measurement: A dissolved oxygen electrode (New Brunswick) was initially installed in the fermentor. However, in the absence of an *in situ* sterile calibration procedure, the drift of the electrode was too severe to

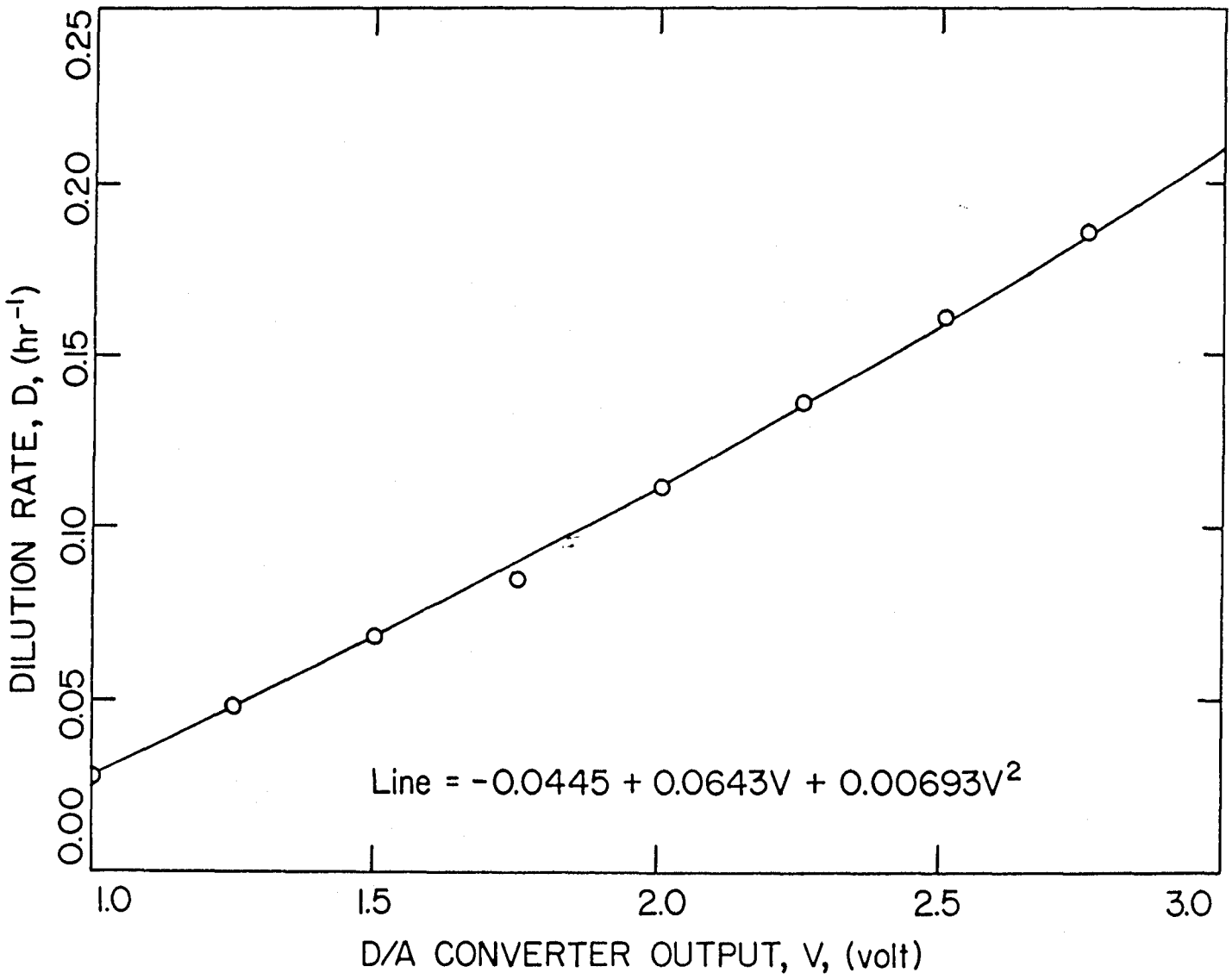


Figure 3.1.4. Calibration curve for dilution rate as a function of the D/A output to the computer controlled nutrient pump.

be of any practical use in a long-lasting fermentation experiment, and its use was quickly abandoned. It is suggested that research be continued on the automatic *in-situ* calibration of sensors in the future.

Sparging: The original design of the stirrer for the 5-liter jar was such that either a very high rate of air sparging (¿5 liter per minute) or a relatively high level of agitation (¿700 rpm) was required in order to achieve the level of aeration needed to provide a sufficient amount of oxygen to the culture so that oxygen availability did not affect the growth. Because the uncertainties in the oxygen uptake rate and carbon dioxide evolution rate were proportional to the air flow rate, a lower rate of air sparging was desired. Similarly, gentler stirring is preferred due to the possibility of the adverse effects arising from a high shear rate on the cell wall at a high rpm. The stirrer shaft was extended and the stirrers were strategically placed so that a fine swarm of air bubbles could be distributed throughout the whole fermentation broth at 300 rpm. This also eliminated the possibility of an incomplete mixing and the presence of a dead zone in the fermentor. In addition, an air recirculation pump was constructed from an aquarium air pump. This air recirculation pump was tested to be leak free, and the air flow rate delivered by it was adjusted with a Variac. After passing through a condenser to rid excess water, part of the exhaust gas from the fermentor was diverted to the recirculation pump. It reentered the fermentor through a newly constructed multi-channeled sparger separately located below the original single-hole sparger. See Figure 3.1.5. The recirculation air flow rate is adjusted with a Variac and measured with an air flow meter. A calibration curve for the air recirculation pump is shown in Figure 3.1.6. Air filters (0.2 μm) were placed on each side of the recirculation pump so that the pump did not need to be autoclaved. The purpose of installing this air recirculating pump was to increase the gas-liquid mass transfer characteristic of the fermentor without increasing the

net air flow rate. The addition of the air recirculation pump did not affect the calculation of OUR and CER.

Air Flow Rate: The air was supplied by compressed air available from Caltech's Central Warehouse. Each cylinder was analyzed for the oxygen and carbon dioxide content (averaging 20.946% and 0.033%, respectively), and the result was used in the calculation of OUR and CER. The air flow rate was carefully controlled by a high-precision mass flow controller (Tylan) and calibrated with the water displacement method. The air flow rate was set at 1.0 liter/min. The compressed air, after passing through a two-step pressure regulator (Matheson), was filtered to prevent any dust particles or oil droplets from impairing the normal operation of this sensitive controller. The principle of operation of this controller can be found in Appendix G and will not be discussed here.

Gaseous Oxygen and Carbon Dioxide Measurement: The inlet air was first humidified by bubbling it through a $CuSO_4$ solution. It was subsequently passed through a heated glass wool filter provided in NBS's original equipment for sterilization and finally circulated through a series of 0.2 μm gas filters to ensure absolute sterility before being sparged into the culture broth. The exhaust gas stream from the fermentor was stripped of excess water by passing it through a condenser filled with ceramic pellets. It was further dried in two columns of anhydrous $CaSO_4$ (Drylite) with color indicators. The oxygen and carbon dioxide concentrations in the dried exhaust gas were subsequently determined with a Beckman Model 755 Paramagnetic Oxygen Analyzer in the zero suppressed range (19-21%) and a Horiba Model PIR-2000 infrared gas analyzer, respectively. Both gas analyzers were calibrated before and after every run with a range of standard gases of known concentrations of oxygen or carbon dioxide. To evaluate the drift in the instruments' response, the calibration procedures were also carried out during those runs

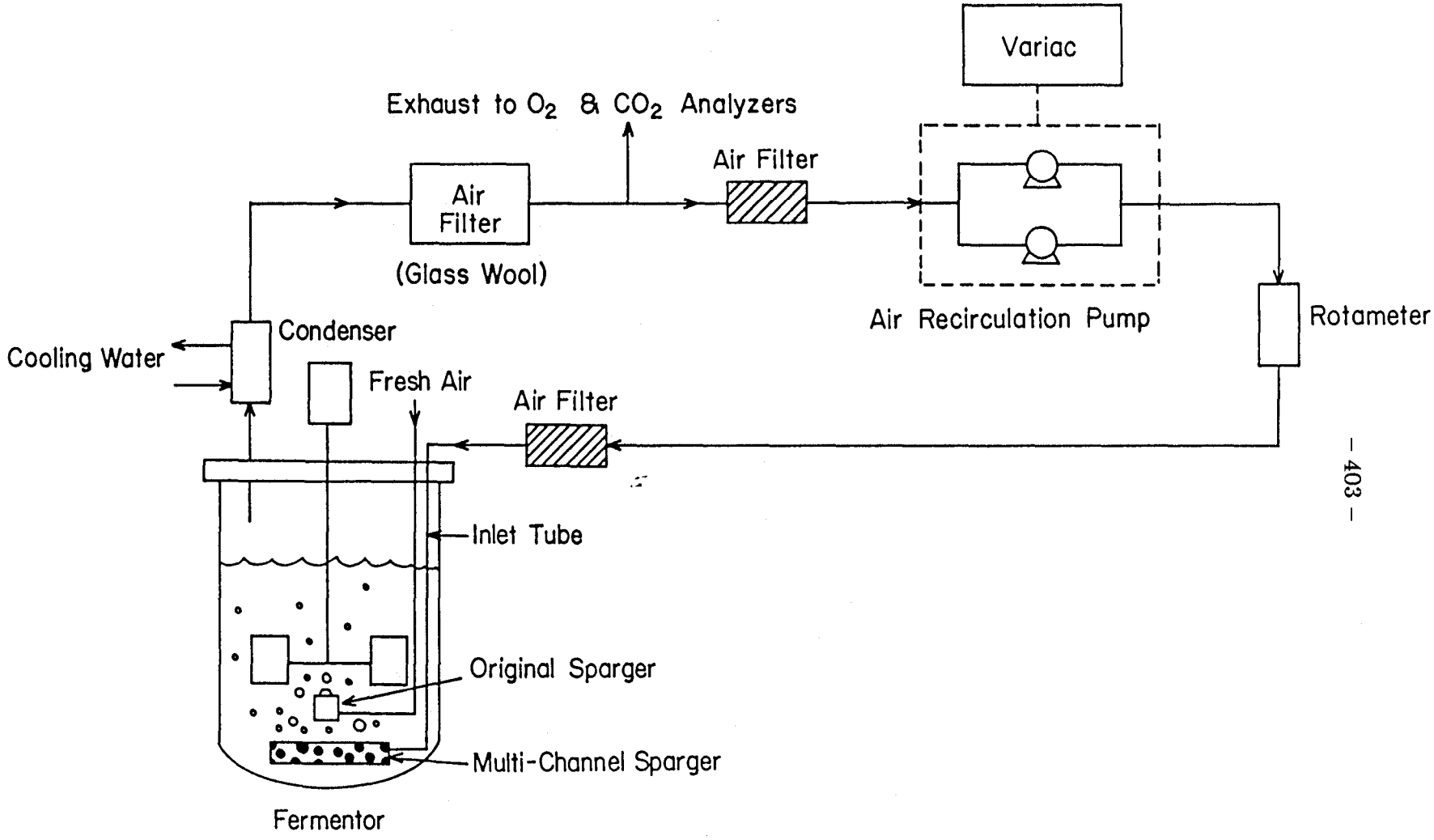


Figure 3.1.5. Schematic of the air recirculation system installed to increase the mass transfer coefficient at the same *net* gas flow rate.

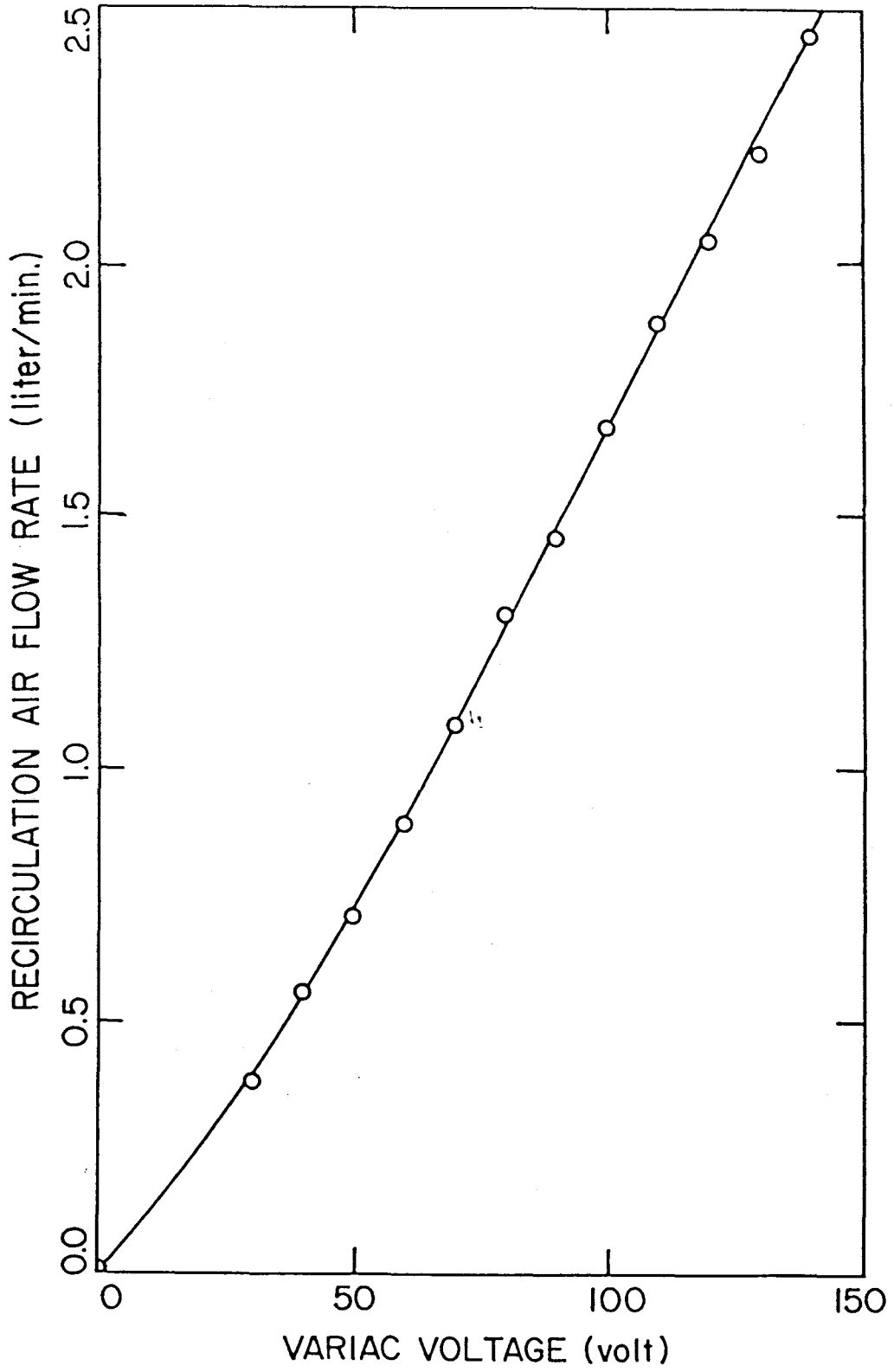


Figure 3.1.6. Air recirculation pump's capacity as a function of the Variac output.

that lasted longer than 24 hours. Each calibration procedure took 5 minutes to execute manually. One typical calibration curve of the CO_2 analyzer is shown in Figure 3.1.7. Because the paramagnetic oxygen analyzer senses the partial pressure of oxygen in the sample, not the percent concentration, the total external gauge pressure has a profound influence on the readings of the analyzer. Thus, the effect of external pressure fluctuations on the oxygen analyzer's response was corrected by monitoring the ambient pressure on-line with a pressure transducer (Setra System). A typical calibration curve for the pressure transducer is shown in Figure 3.1.8. Because the gas analyzers, especially the oxygen analyzer, were also sensitive to the gas flow rates, they were monitored by air flow meters (Linde Specialty Gas) and regulated manually with rotameters at 250 ml/min for the oxygen analyzer and 400 ml/min for the carbon dioxide analyzer.

Biomass Concentration Measurement: A stream of broth was continuously withdrawn from the fermentor and fed into a bubble trap. A peristaltic pump was used to circulate the sample through a 0.2 mm flow cell in a spectrophotometer (Spectronic 21, Bausch & Lomb) to measure the optical density at 660 nm. The flow cell had no dead-zone, and the velocity of the sample flowing through the cell was high enough to discourage any possible wall growth. The sample was then joined with the overflow from the bubble trap and returned to the fermentor. As shown in Figure 3.1.9, instead of the fermentor sample, the filtration-sterilized water stored in a separate reservoir could be pumped through the flow cell as often as necessary to recalibrate the optical flow cell. The actual procedure was extremely simple; it consisted of merely moving the clamps to appropriate positions in the flow lines.

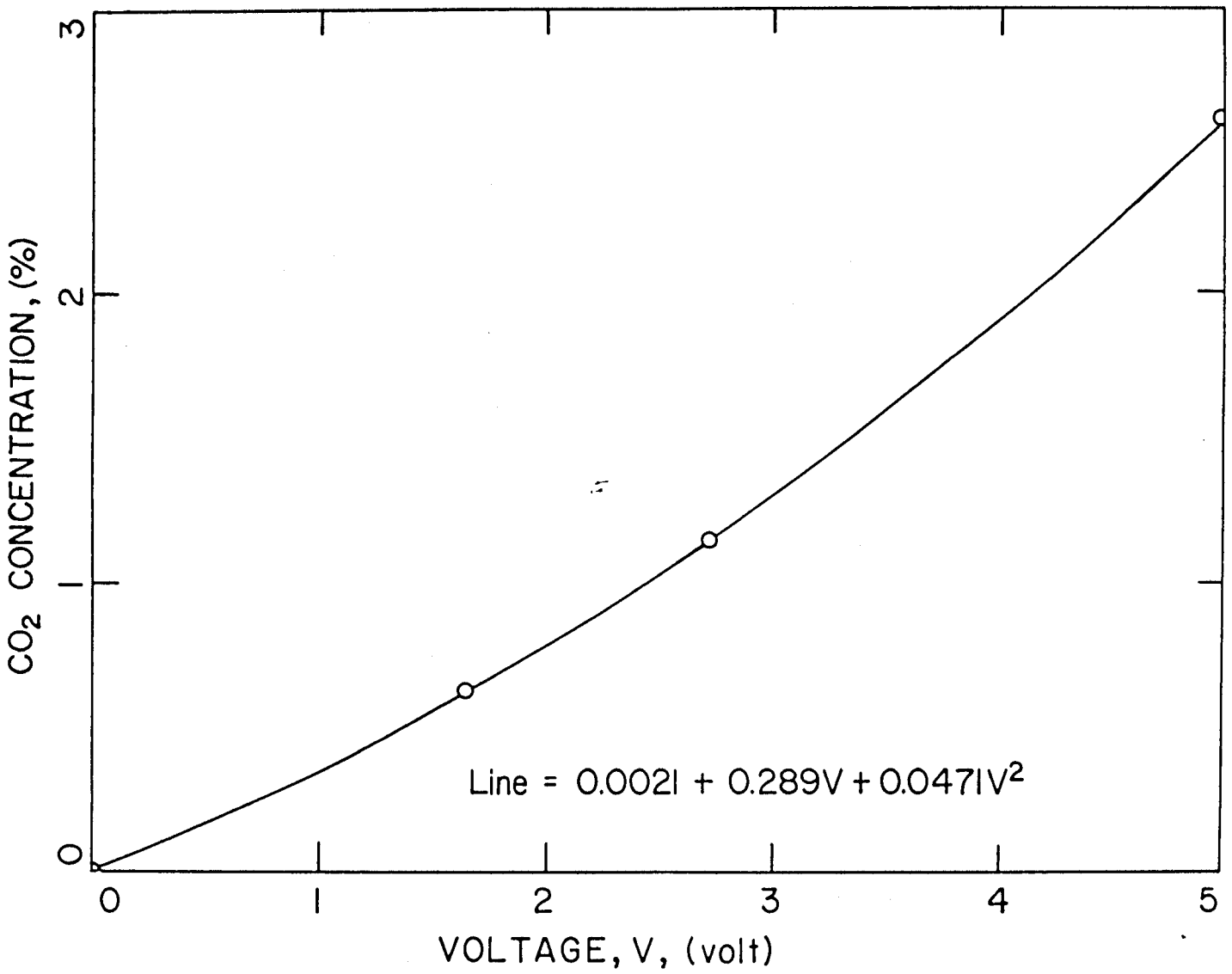


Figure 3.1.7. CO_2 concentration as a function of the A/D voltage reading.

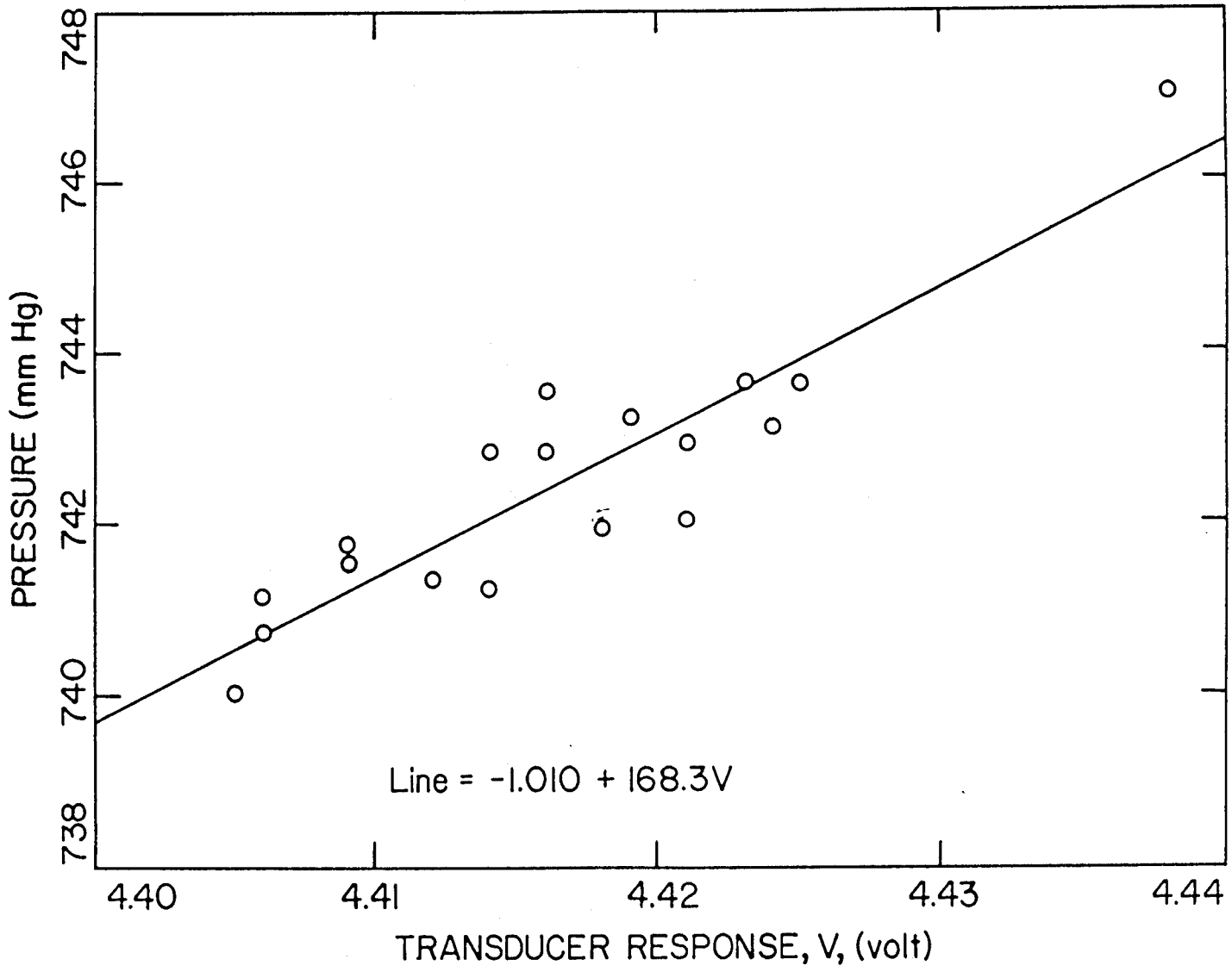


Figure 3.1.8. Atmospheric pressure as a function of the pressure transducer's output.

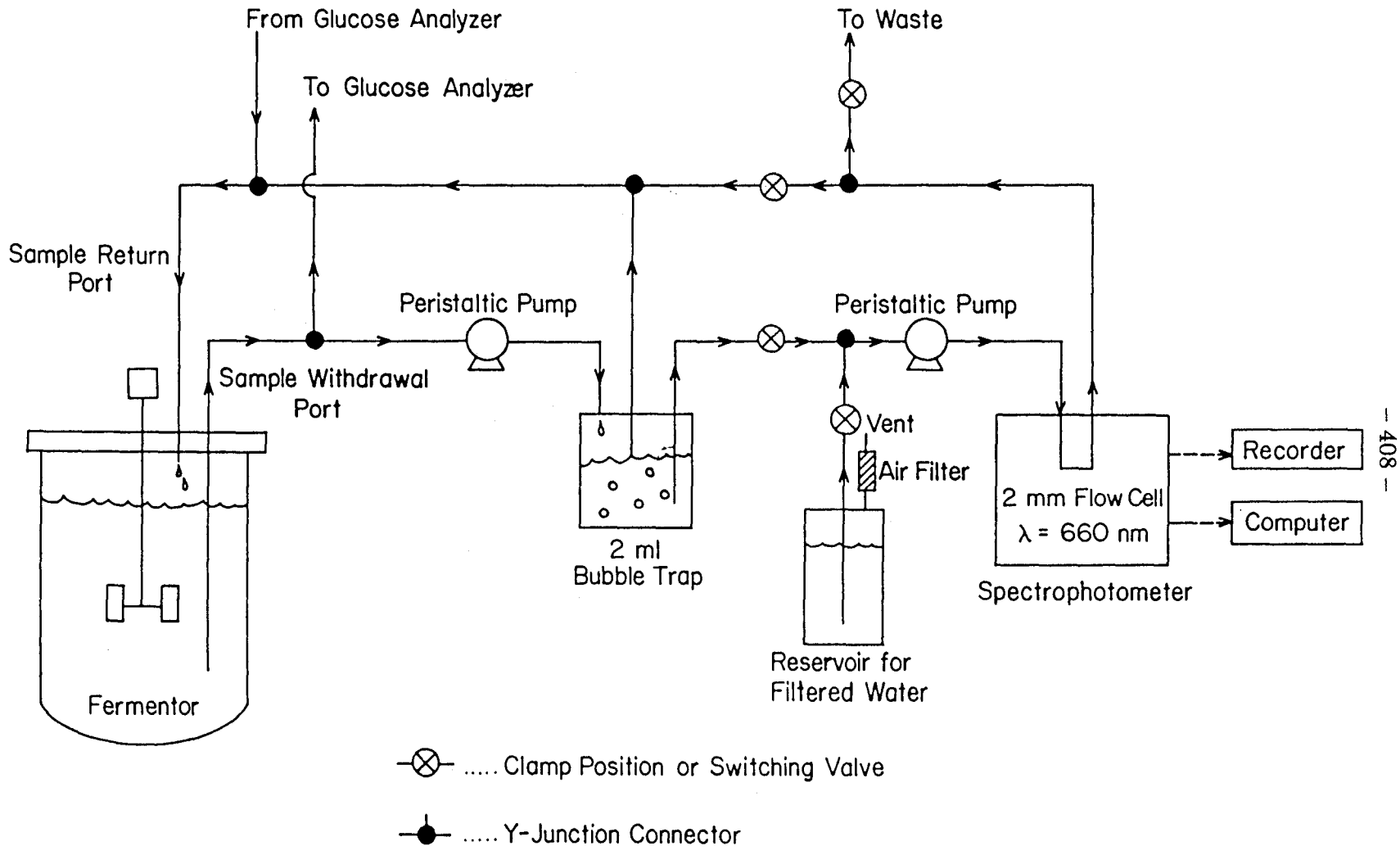


Figure 3.1.9. Schematic of the continuous optical density measurement apparatus.

3.2 ON-LINE GLUCOSE ANALYZER

An on-line glucose analyzer was constructed, and the schematic of this device is shown in Figure 3.2.1. The principle of detection of this analyzer is based on the oxidation of glucose by some oxidants. For many decades, this wet chemistry method has been used to analyze the glucose level in blood serum and plasma samples (Fingerhut *et al.*, 1963). This is also the widely used flow method with AutoAnalyzers. However, its use in the fermentation field is not widespread due to the nonselective nature of the method. Many components are oxidized if the oxidant is highly reactive. Conversely, glucose oxidation will not proceed unless the oxidant possesses adequate oxidation power. Although the enzymatic method of glucose oxidase-peroxidase is the most specific, it is relatively expensive, and, as in any enzyme-based method, the ease of reagent preparation and the stability consideration must also be weighed.

In our study, Fe^{+3} was selected because of its moderate oxidation-reduction potential. The oxidation of glucose was catalyzed in the presence of cyanide ions and under alkaline conditions at an elevated temperature of 90 °C. The reaction proceeds as follows:

Oxidation: $$\text{Glucose} + \tfrac{1}{2}\ O_2 \longrightarrow \text{Gluconic acid} \qquad (3.2.1)$$

Reduction: $$2\ Fe(CN)_6^{-3}(III) + 2\ OH^- \longrightarrow 2\ Fe(CN)_6^{-4}(II) + H_2O + \tfrac{1}{2}\ O_2 \qquad (3.2.2)$$

Alkaline conditions were required to handle safely the cyanide solutions to prevent the evolution of the deadly cyanide gases under acidic conditions. Under this condition, other potentially oxidizable components typically present in the fermentation broth, such as ethanol and ammonium ions, were tested for interference with the glucose measurement, and they were found to exert insignificant effect on the response. The glucose concentration was determined photometrically due to

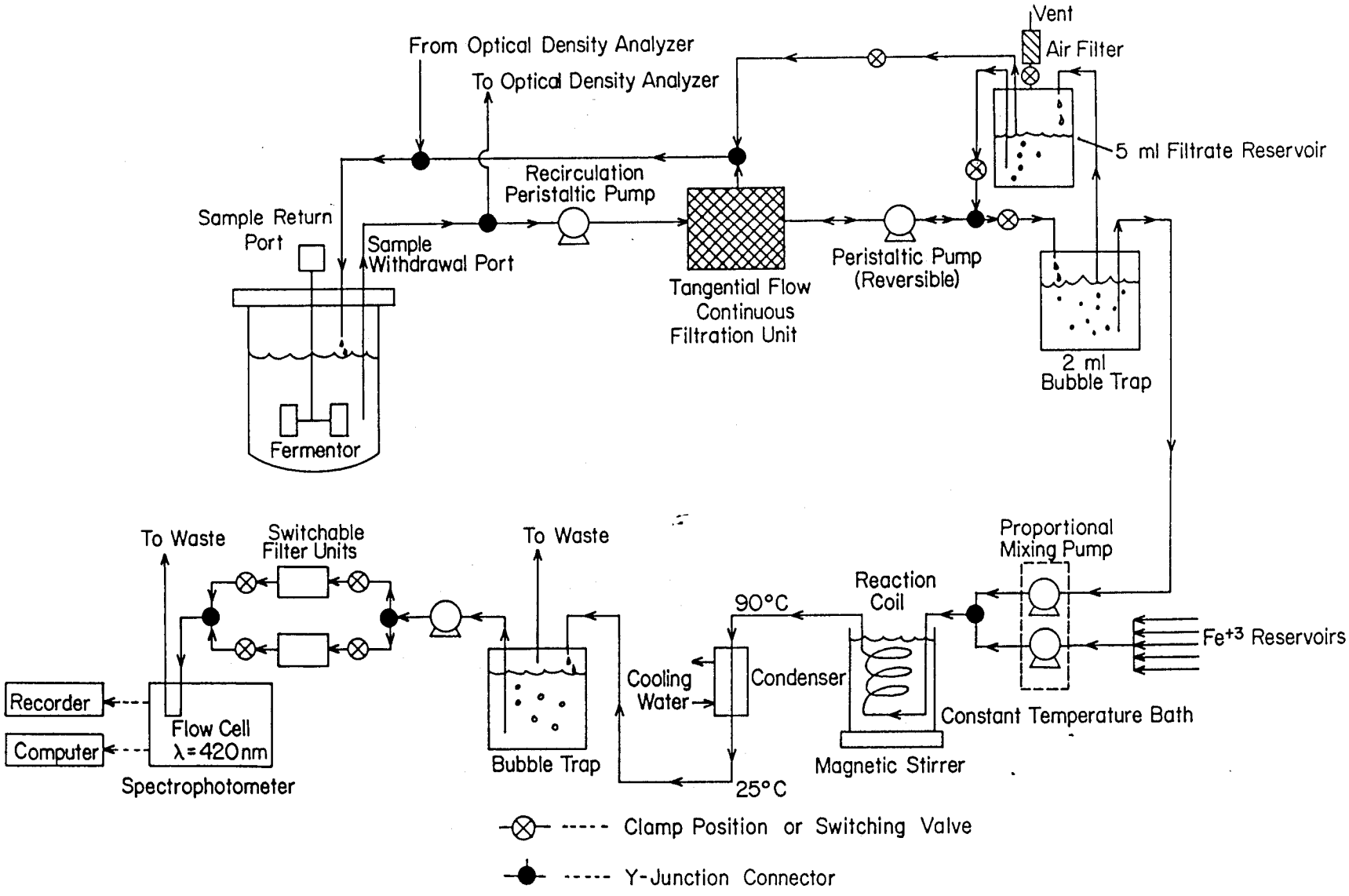


Figure 3.2.1. Schematic of the on-line glucose analyzer working under the oxidation of glucose by Fe^{+3} to Fe^{+2} at 90 °C.

the change in color upon the reduction of Fe^{+3} (yellow) to Fe^{+2} (colorless). As indicated in the absorbance spectrum shown in Figure 3.2.2, the ferricyanide(III) solution had a maximum absorbance at a wavelength of 420 nm; this setting was used in the measurement of Fe^{+3} concentration. Figure 3.2.3 shows that a duration of 5 minutes is needed for the reaction to reach completion.

Calibration curves were routinely generated for the on-line glucose analyzer, and one such curve is shown in Figure 3.2.4. The response is quite linear (in the inverted sense) below the saturation point. A disadvantage of this method is that it works well only in the limited glucose range below the saturation point; thus, the concentration of the ferricyanide in the feed solution must be changed according to the range of glucose concentration in the fermentor. This is caused by the fact that the measurement indicates the level of the residual ferricyanide, rather than the level of the reduced ferrocyanide that is ultimately proportional to the glucose concentration. There has been an effort by various investigators (Fingerhut *et al.*, 1963; Fingerhut *et al.*, 1966; Klein *et al.*, 1966) to couple the ferrocyanide formed in Reaction (3.2.2) to another color reaction of blue molybdate.

Another commonly used wet chemistry method employing dinitrosalicylic acid was also found to be highly suitable for the on-line determination of glucose concentrations (Miller, 1959). This method tests for the presence of free carbonyl group (C=O), the so-called reducing sugars. This involves the oxidation of the aldehyde functional group present in, for example, glucose and the ketone functional group in fructose. Simultaneously, 3,5-dinitrosalicylic acid (DNS) is reduced to 3-amino,5-nitrosalicylic acid under alkaline conditions:

Reduction: Aldehyde Group $\longrightarrow$ Carboxyl Group (3.2.3)

Oxidation: 3,5–Dinitrosalicylic Acid $\longrightarrow$ 3–amino, 5–nitrosalicylic Acid (3.2.4)

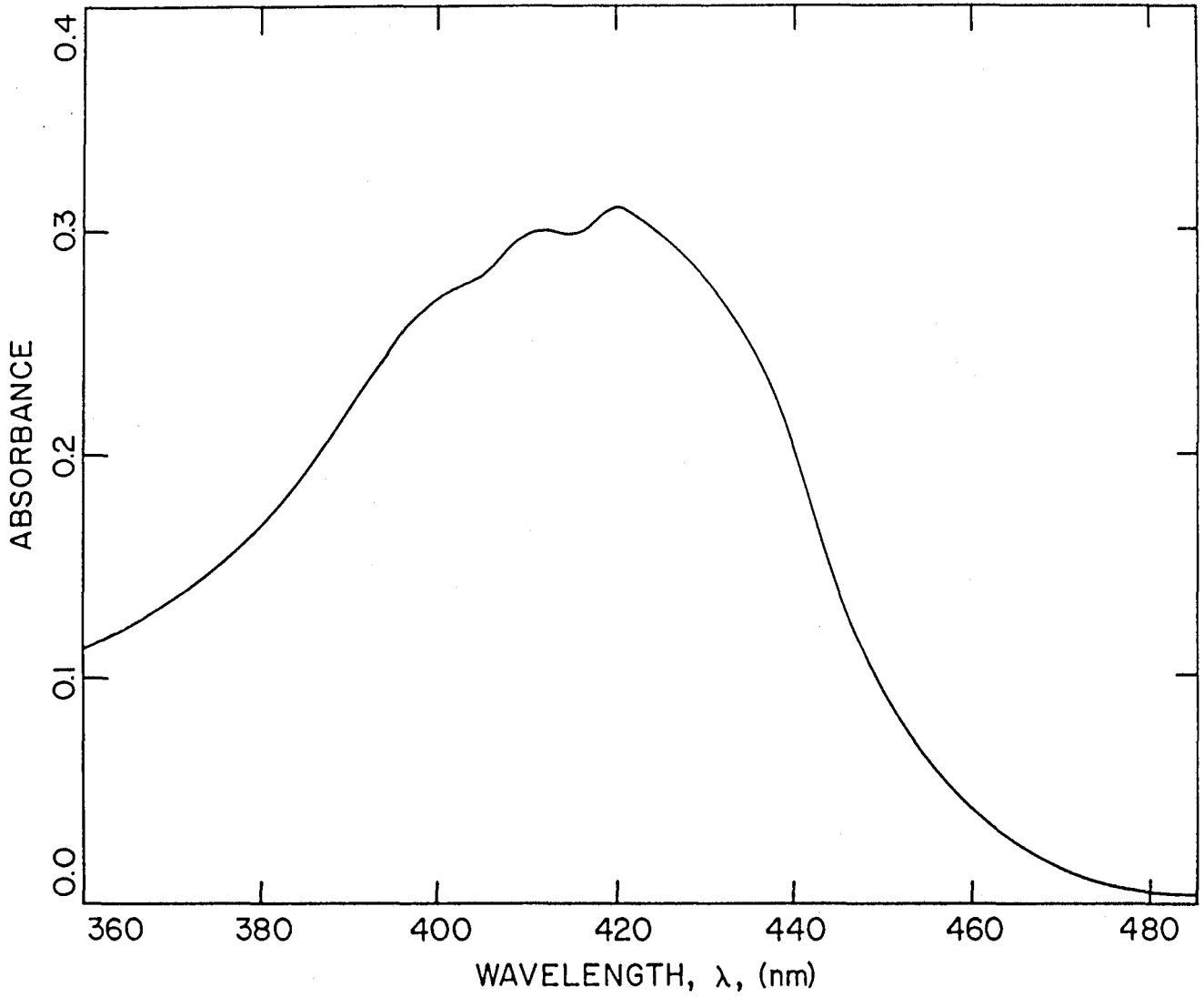


Figure 3.2.2. Absorbance spectrum of potassium ferricyanide(iii) in the visible range.

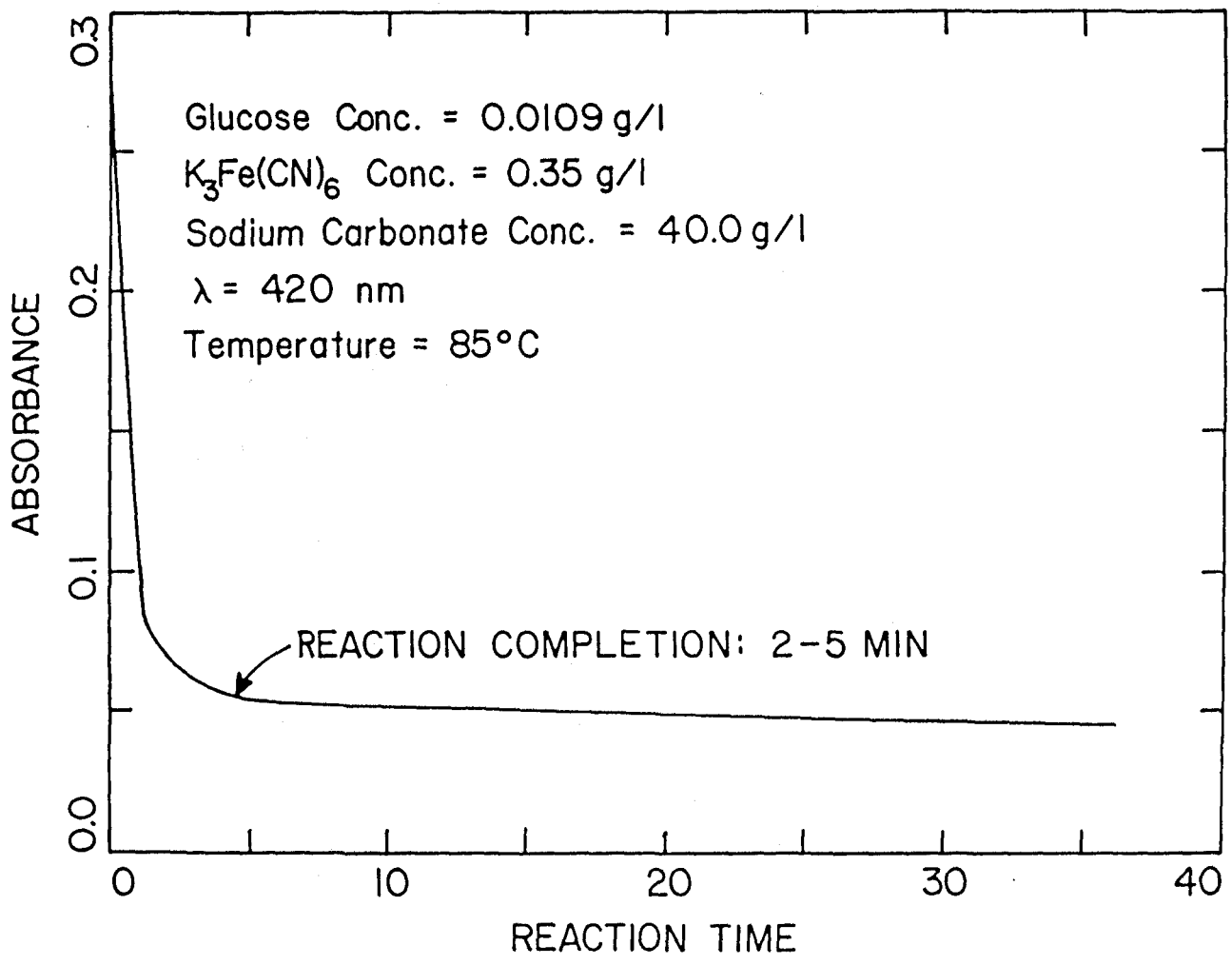


Figure 3.2.3. Response of the glucose analyzer due to the oxidation of glucose by Fe^{+3}. Note that the reaction is essentially complete at 2–5 minutes after the reactants are mixed together.

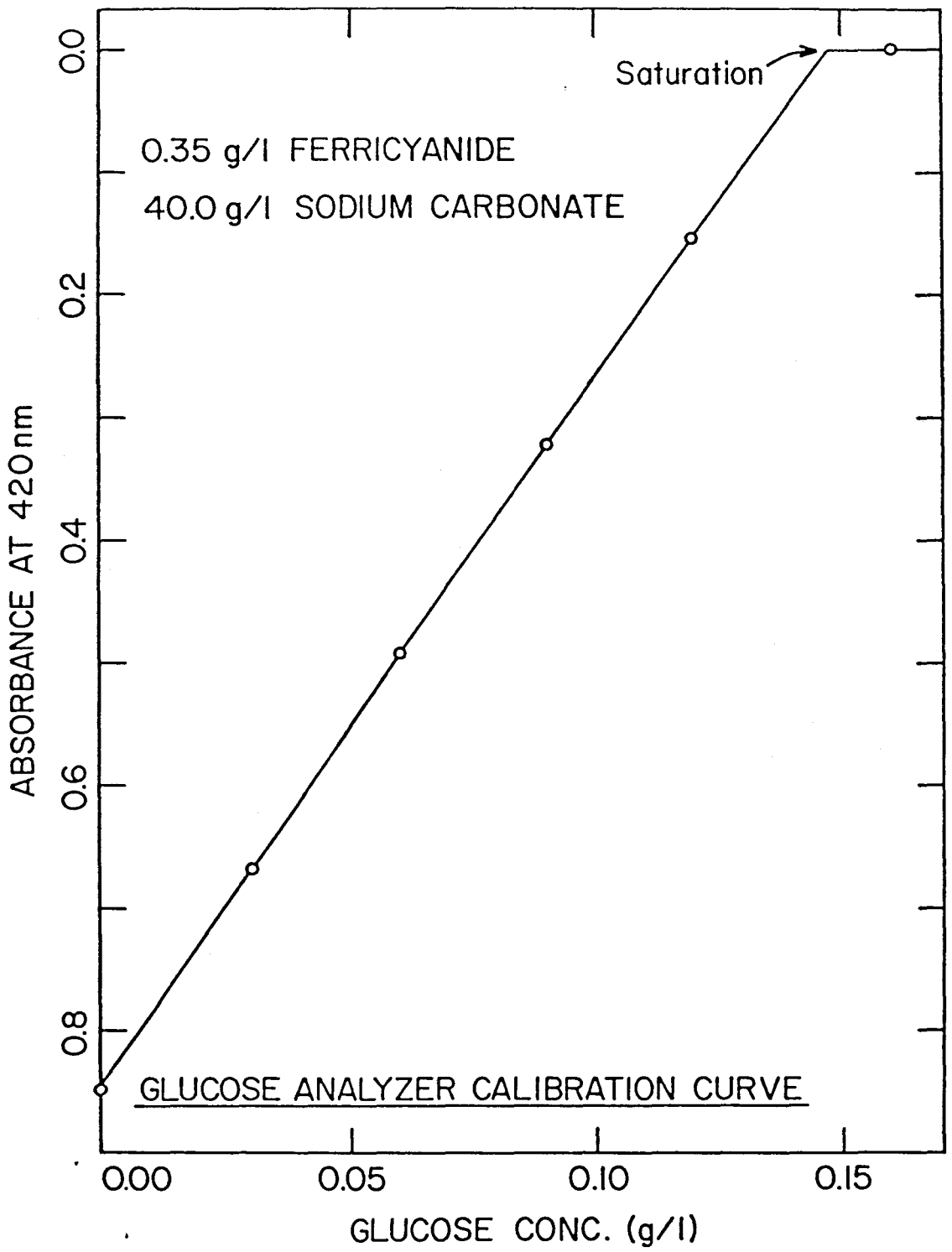


Figure 3.2.4. Glucose analyzer's response as a function of glucose concentration.

In the on-line automated scheme, an equal volume of the DNS reagent solution (10 g/l DNS, 2 g/l phenol, 0.5 g/l Na_2SO_3, 10 g/l NaOH) is mixed with the cell-free sample. The development of red-brown color after heating the reaction mixture at 90°C for 5 minutes is continuously detected with a spectrophotometer at 575 nm. Because dissolved oxygen can interfere with the above oxidation-reduction reactions, sulfite, which itself is not necessary for the color reaction, is added in the reagent to absorb the dissolved oxygen. In addition, phenol, up to 2 g/l in the DNS reagent, intensifies the color density. It changes the slope of the calibration curve of absorbance versus glucose concentration but does not affect the linearity. The above procedure yields an absorbance of 1 unit for 1 g/l of glucose in the original sample in the absence of phenol in the reagent, as opposed to an absorbance of 2.5 unit for 1 g/l of glucose in 2 g/l of phenol. This property can be exploited to achieve maximum sensitivity for dilute samples.

The above reaction scheme shows that one mole of sugar will react with one mole of 3,5-dinitrosalicylic acid. However, it is suspected that there are many side reactions, and the actual reaction stoichiometry is more complicated than that previously described. The type of side reaction depends on the exact nature of the reducing sugars. Different reducing sugars generally yield different color intensities; thus, it is necessary to calibrate for each sugar. In addition to the oxidation of the carbonyl groups in the sugar, other side reactions such as the decomposition of sugar also competes for the availability of 3,5-dinitrosalicylic acid. Although this is a convenient and relatively inexpensive method, like the ferricyanide reduction method, its specificity is intrinsically low. On the other hand, the color intensity of the DNS method is proportional to the glucose concentration; whereas, it is the *difference* in the absorbance that is proportional to the glucose concentration in the ferricyanide reduction method.

A common recurrent technical problem with the above wet chemical methods is the formation of precipitate in a basic environment from the proteinaceous compounds present in the fermentation broth. Among the various bases tried, sodium carbonate was found to produce the minimum amount of precipitate with the ferricyanide reduction method. In order to prevent the precipitate from interfering with the photometric measurement, two parallel interchangeable filter holders were installed in the line immediately before the spectrophotometer. Clamps were used to direct the sample flow through the unclogged filtration unit, and the dirty filter paper was replaced with a new one. The switching time depended on the sample flow rate, usually every 4-8 hours. The use of a dialysis unit for the elimination of both cells and protein compounds from the reaction mixture may be an attractive alternative that has not been considered in this work. However, the diffusion of glucose across the dialysis membrane and the possible membrane fouling may complicate the measurement.

One of the difficulties encountered was in the construction of an autoclavable continuous flow filter used to generate a stream of clear filtrate from the fermentation broth. The main body of the filtration unit was fabricated out of a 3-inch-dia. rod of Plexiglass. Other machinable materials such as stainless steel or heat-resistant polycarbonate plastic may also be used for this purpose. Rubber O-rings, clamps, and precise lathing techniques were employed to achieve a leak-free, sterile seal. The construction of the cross-flow filtration unit is shown in Figure 3.2.5.

To regenerate the fouled filter membrane after a prolonged period of operation, a backflush capability was built into this glucose analyzer. The excess filtrate was stored in a 5-ml reservoir before being returned to the fermentor. During the filter membrane regeneration, clamps were placed in appropriate positions along the lines, and the direction of the filtrate pump was reversed. By so doing, the filtrate from

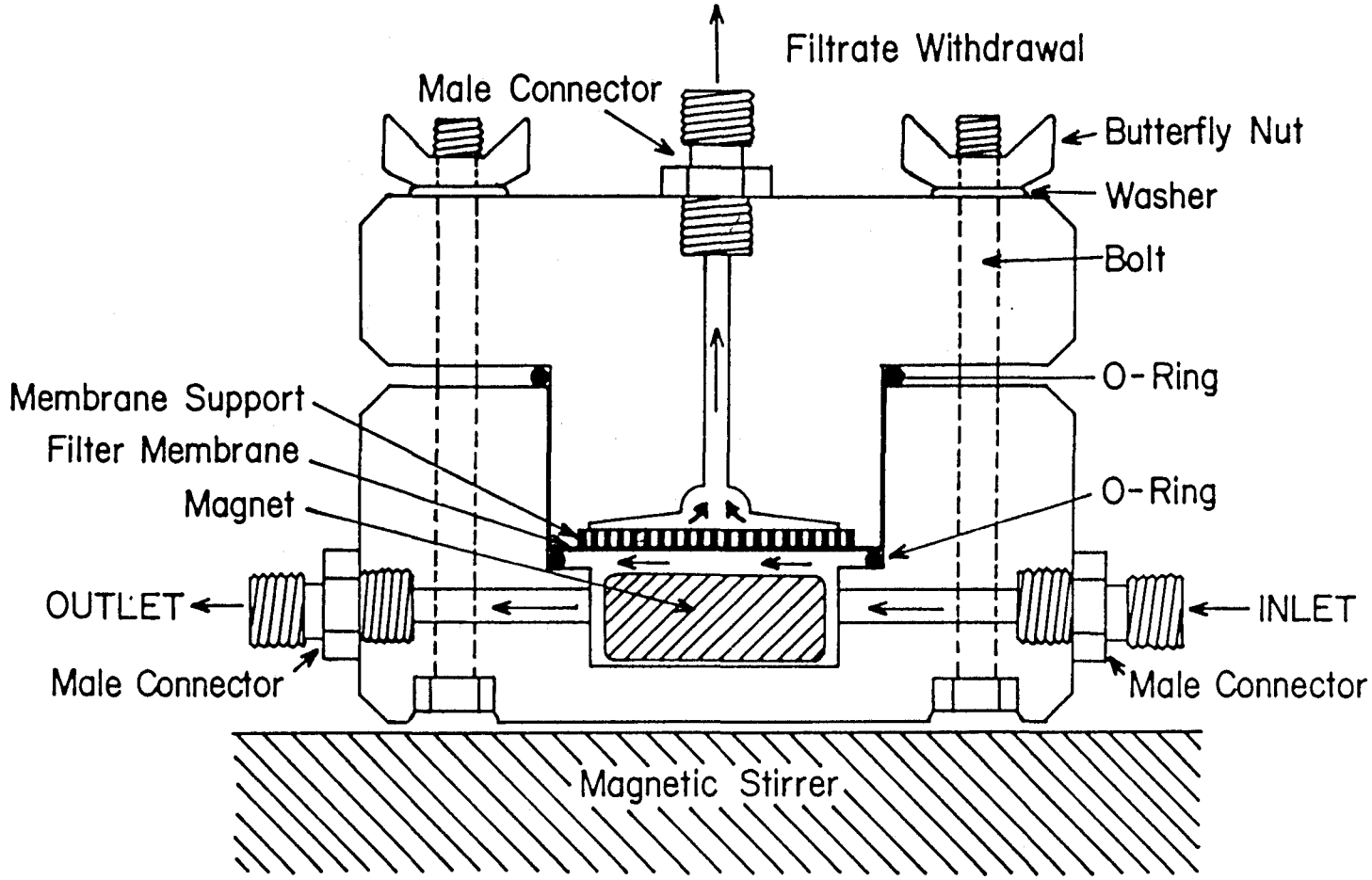


Figure 3.2.5. Construction of the cross-flow filtration unit. The filtration membrane is sandwiched between the male (upper) and the female (lower) part of the filtration unit.

this reservoir was forced back across the membrane to clear the plugged pores. The backflush procedure, which lasted 2 minutes, was carried out manually as often as necessary to restore the efficiency of the continuous filter. The time interval between backflush operations depended on the type of filter membrane used, its pore size and structure, the size and the level of debris and antifoam particles in the fermentor, the cell size and cell density, and the rate of filtrate withdrawal.

A wide range of filter membranes of different materials and pore sizes were tested. A 0.8-μm polyacetate filter membrane with laser-drilled straight-channelled pores (Nucleopore) was among the most efficient of the many filter membranes tested for filtration characteristics. For example, in the absence of contaminants, it was able to function continuously for a stretch of more than 24 hours at a filtrate withdrawal rate of 2 ml/min. However, this filter membrane was quickly plugged when smaller contaminants similar in size to that of the membrane pore (approx. 1–3 μm) were present. In fact, the unexpected plugging of the membrane was often used as an early alarm that warned the presence of contaminants.

The fermentation broth containing cells and debris was pumped into the cross-flow filter's main compartment at a recirculation rate of less than 10 seconds per cycle. This relatively high recirculation rate ensured that the consumption of glucose was minimized during the transport of the broth to the filter membrane surface and that the glucose concentration in the filtrate was close to that in the fermentor. The compartment housed a spinning star-shaped circular magnetic stirrer that prevented the build-up of a filter cake on the surface of the filter membrane. This measure was necessary because the filtration efficiency could have been drastically and fatally reduced if a filter cake was undisturbed and allowed to accumulate.

A suction pressure was applied across the membrane to withdraw a stream of clear filtrate. The clear filtrate was first stripped of air bubbles in a bubble trap so

that the flow rate was not affected. The bubble-free filtrate was then proportionately mixed with a ferric cyanide solution in 40 g/l of sodium carbonate. The relative flow rates of the filtrate stream and the ferric cyanide stream were maintained at a constant ratio of approximately 1 to 3.5, which was the inherent relative flow rate achieved with the combination of a 7013 pump head and a 7014 pump head (Cole-Parmer), respectively, when both heads were installed on the same peristaltic pump. The filtrate consumption rate of this destructive method was 1.3 ml/min. The reaction proceeded for 15 minutes as the mixture passed through a coil submerged in a constant-temperature water bath at 90 °C. The reacted mixture was subsequently cooled to room temperature in a condenser. After the bubbles present in the line were again eliminated in a bubble trap so that the optical properties were not affected by the presence of air bubbles, the absorbance of the reaction mixture was continuously monitored with a spectrophotometer (Spec 21) interfaced to the microcomputer.

3.3 ON-LINE ETHANOL ANALYZER

An on-line ethanol analyzer utilizing a piece of expanded porous PTFE (poly-tetrafluoroethylene or Teflon) tubing was constructed. Other tubing materials such as regular PTFE, Tygon and silicone were tested, and only a minute amount of ethanol could be detected in the carrier gas in each case. A schematic diagram of the final device used in the experiment is shown in Figure 3.3.1.

Nitrogen was used as the carrier gas, and its flow rate was controlled by a mass flow controller (Tylan) set at 30 ml/min. The precise control of the carrier gas flow rate by a state-of-the-art mass flow controller is central to the accurate measurement of the ethanol concentration because the response of the flame ionization detector (FID) is a strong function of the *amount* of combustible materials passing through. The possibility of contamination from this carrier gas line was reduced by installing

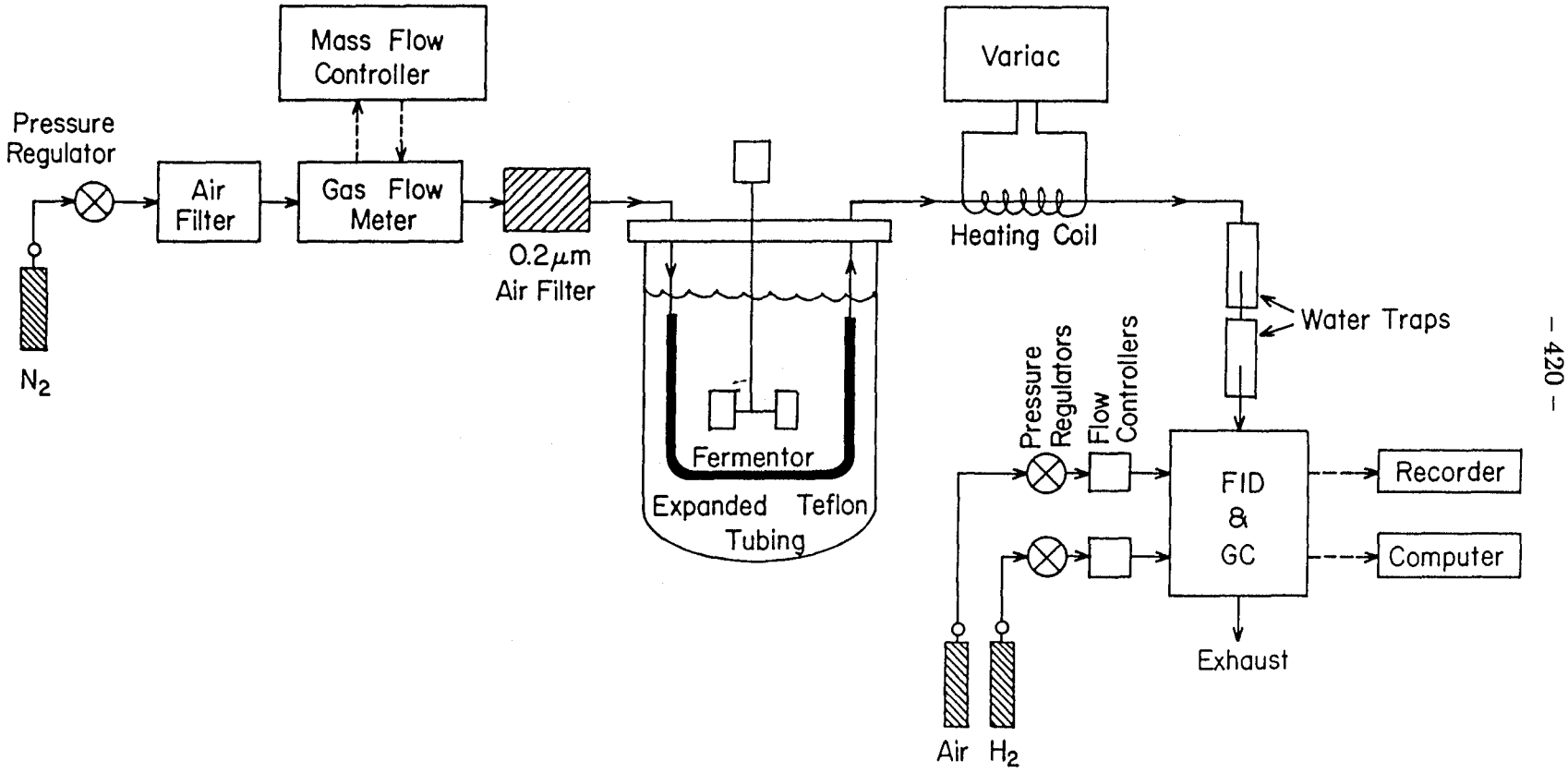


Figure 3.3.1. Schematic of the on-line ethanol analyzer utilizing the expanded (porous) Teflon tubing method.

a 0.2-μm air filter before it was connected to the fermentor. The carrier gas was then passed inside a 1-meter-long expanded Teflon tubing, 3 mm i.d., 4 mm o.d., 0.5 μm pore-size, that was wrapped around the baffles and submerged under the fermentation broth. The flexible expanded Teflon tubing used was marketed under the name of POREFLON and was donated by Sumitomo Electric Industries, Osaka, Japan. Another source of microporous Teflon tubing was a product named Gore-Tex marketed by W. C. Gore (Elkton, Maryland).

The tubing method for detecting ethanol is described by Dairaku and Yamané (1979) and Heinzle *et al.* (1981), and the mathematical description of the diffusion process across the membrane is described by Yamané *et al.* (1981). The tubing method takes advantage of the hydrophobic properties of Teflon in that the expanded porous Teflon is easily penetrable by volatile components such as ethanol but impervious to water below the wetting pressure, or the minimum permeation pressure.

These tubings are available in a wide range of diameter, wall thickness, and porosity. These parameters, in turn, determine the minimum water entry pressure (the pressure drop at which water is forced through the tubing wall), minimum bubble point (the pressure drop at which air can be forced through the wall and sparge on the other side), and air permeability. Typical values of water entry pressure are 40 psi for 0.2μm tubings, 20 psi for 0.45μm tubings, and 10 psi for 1.0μm tubings. Similarly, typical values of bubble point are 13, 7, and 3 psi for the respective tubings. To avoid the external liquid water from entering into the tubing and extinguishing the flame in the FID detector when the pressure inside the tubing was much less than that of the fermentor broth, and to prevent the carrier gas from sparging into the fermentation broth when the reverse condition was true, the pore size of the Teflon tubing as well as the flow rate and the pressure of the carrier gas

flowing inside the tubing were chosen such that the pressure drop across the tubing wall in either direction was below both the minimum water entry pressure and the minimum bubble point.

The carrier gas line was wrapped in a coil of heating wire, and the Variac controlling the energy output from the heating wire was set to prevent the condensation of water vapor in the line. Two water traps were installed immediately before the line entered the gas chromatograph to ensure that water drops would not extinguish the flame in the FID detector. The presence of ethanol in the carrier gas was detected by a flame ionization detector equipped in a gas chromatograph (Varian, Model 3400). It is important to note that although a gas chromatograph is employed, separation of the substance to be detected is not effected with a packed column. Rather, it is accomplished with a Teflon tubing. As a matter of fact, the column is totally bypassed. The GC was operated under the following conditions throughout all the runs: injector temperature = 120 °C, column temperature = 250 °C, detector temperature = 300 °C, air flow rate = 300 ml/min, and hydrogen gas flow rate = 60 ml/min.

The selectivity of the gas chromatograph's FID was reasonably adequate; among those chemical compounds listed in Varian 3400 Gas Chromatograph Operator's Manual as giving minimal or no response with the FID detector are: O_2, CO_2, N_2, air, H_2O, and NH_3. In addition, no volatile component other than ethanol was present at a significant level under the fermentation condition studied. Therefore, the Teflon tubing method of continuous on-line ethanol analysis adapted in this work is highly accurate, sensitive, inexpensive, and simple.

The dead-time due to the transport of the carrier gas through 25 meters of the tubing before reaching the GC was about 9 minutes, and the transient response had a time constant of approximately 5 minutes. The delay in response is caused

both by the plug flow of the carrier gas through the tubing (dead-time) and by the diffusion of the volatile component through the liquid boundary film and the membrane pores (first-order delay). One typical response curve of GC signal to a step change in ethanol concentration is shown in Figure 3.3.2. The stability of the FID's baseline was repeatedly checked overnight with the carrier gas passing through the POREFLON tubing submerged in 2 liters of water in the fermentor, and the drift was found to be negligent, *i.e.,* less than 0.0001 volt out of a normal operating voltage of 3-5 volt. Thus, there was no need for frequent recalibration during a fermentation run. The FID response was found to be linear to the ethanol level in the range tested (0–40 g/l), although only a narrower range is shown in Figure 3.3.3, which was originally generated as one of the calibration curves to be used in converting the electrical signals to ethanol concentrations.

3.4 DATA ACQUISITION SYSTEM

The signals from various instruments were sent to the microcomputer through shielded coaxial cables. On-line data acquisition was automated by a Z-80 based microcomputer with a S-100 data bus, synchronized by a 2-MHz internal clock, and operated under CP/M 80 operating system. The microcomputer consisted of a CPU board, a 64K memory board, a real time clock (SciTronics), a floppy disk controller, two 16-channel A/D boards (Analog Devices), a 16-channel D/A converter (Analog Devices), two serial ports, and two parallel ports. The peripherals included a terminal (Televideo Model 925), two 470K floppy disk drives (Tandon), a printer (μ-science), a plotter (Bausch and Lomb), and a network line for communication with other computers. The general architecture and the working of a microcomputer are described in Appendix G and will not be discussed here.

One of the data acquisition programs is listed in Appendix B. During an experimental run, the oxygen concentration, the carbon dioxide concentration, the

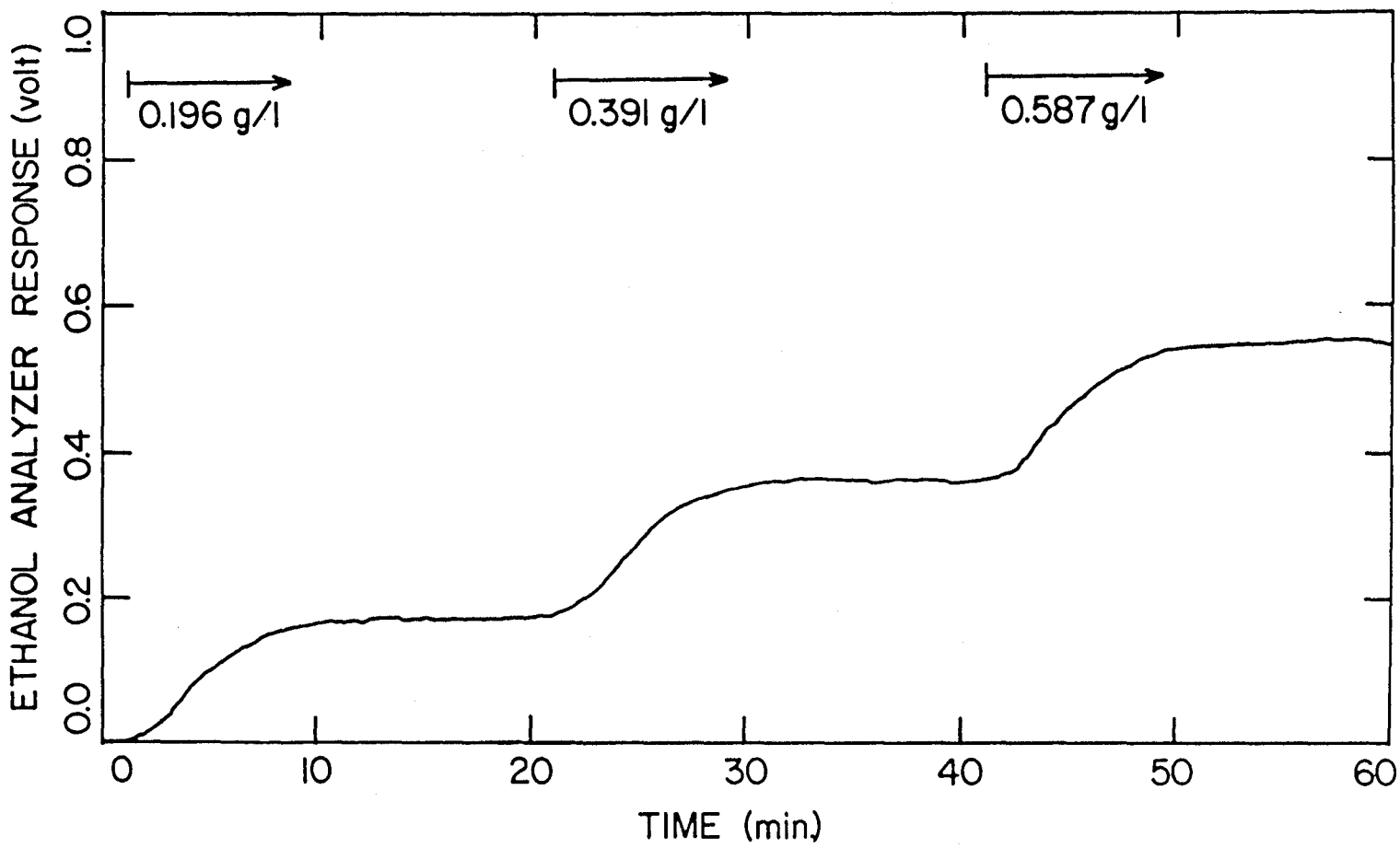


Figure 3.3.2. Response curve of the on-line ethanol analyzer following step changes in ethanol concentration. The analyzer's response time is about 5 minutes.

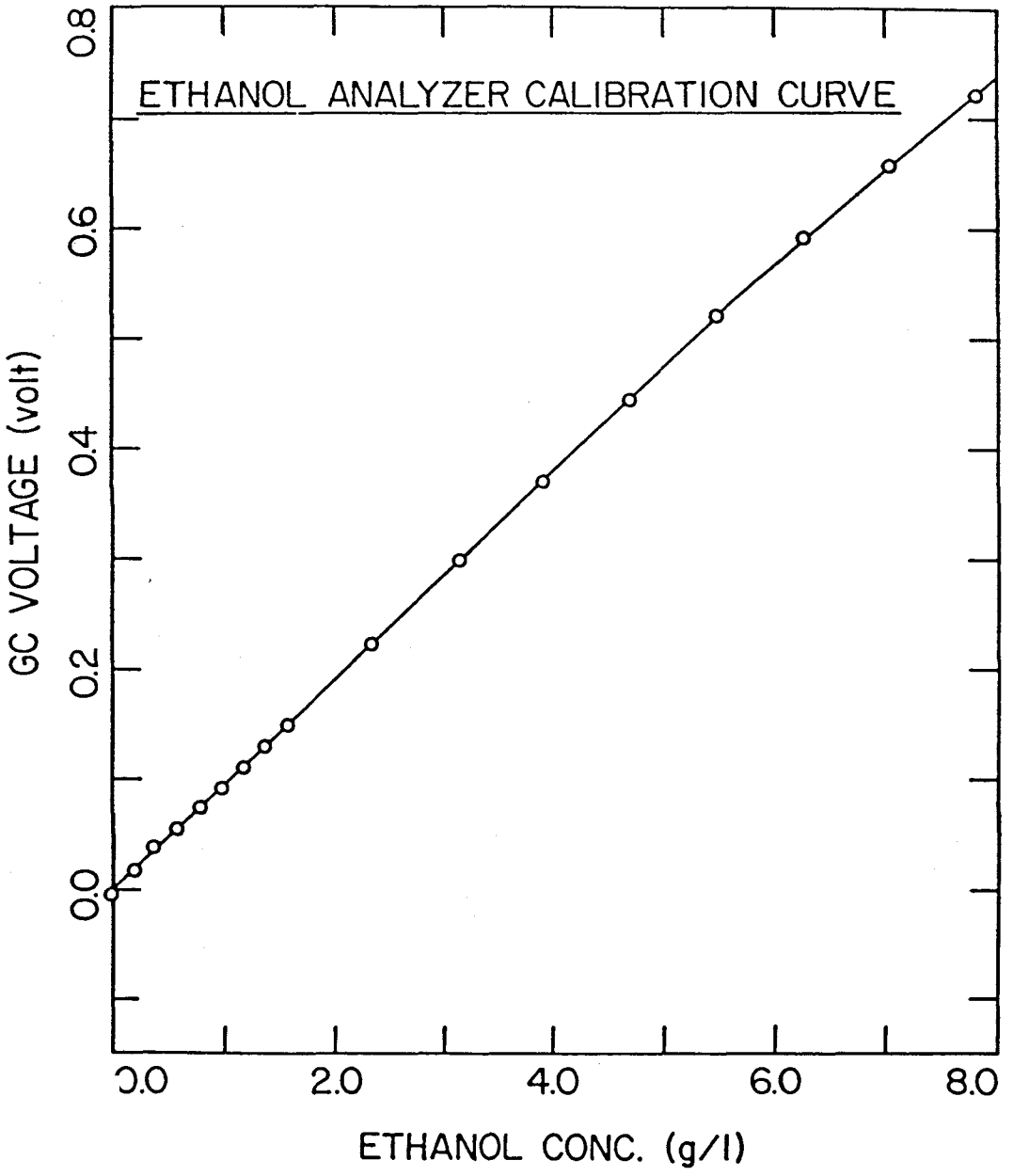


Figure 3.3.3. Ethanol analyzer's response as a function of ethanol concentration.

pressure, the optical density, the glucose concentration, the ethanol concentration, and the nutrient pump speed were continuously sampled, averaged, and recorded at 1-minute intervals. There are significant noise problems with some of the measurements if the sampling rate is too high. There are also practical considerations of data storage. Conversely, alias effects and low levels of confidence may result from the slow response if the sampling rate is too low. Both the acid and base addition pumps' ON/OFF activities were also monitored and recorded.

3.5 EXPERIMENTAL PROCEDURES

Microorganism: Saccharomyces cerevisiae, i.e., the baker's yeast, strain ATCC 4126, was employed as the model system to conduct the microbial dynamic studies. This microorganism was chosen because of its well studied metabolic pathways and the ease of cultivation in a fermentor. This is a facultative anaerobe, 5 to 8 μm in diameter.

Culture Maintenance: A pure stock culture of *Saccharomyces Cerevisiae*, strain ATCC 4126, was maintained by periodic plate transfers, approximately once every month, in a YPG (Yeast extract-Peptone-Glucose) medium (1.5 g/l yeast extract, 10 g/l glucose, 5 g/l peptone, 20 g/l agar). The inoculum was prepared by aseptically transferring a single colony from the pure culture plate with an inoculation loop into a 250-ml flask containing the sterilized growth medium described in Table 3.5.1. The colony was subsequently allowed to propagate in a temperature controlled flask shaker. One ml of the culture near the end of its exponential growth phase was further transferred from the initial flask to another flask so that the culture was acclimated before finally being inoculated into the fermentor.

Nutrient Preparation: A large batch of the nutrient, approximately 40 liters, was prepared according to the recipe presented in Tables 3.5.1–3.5.4, and its pH was first brought close to 5 with KOH pellets. The pH was then adjusted to 5.00 by

Table 3.1
The Composition of the Defined Medium for the Baker's Yeast

Compound	Concentration	
$MgCl_2 \cdot 6H_2O$	0.52	g/l
$(NH_4)_2SO_4$	12.0	g/l
H_3PO_4 (85%)	1.6	ml/l
KCl	0.12	g/l
$CaCl_2 \cdot 2H_2O$	0.2	g/l
NaCl	0.06	g/l
$MnSO_4 \cdot H_2O$	0.024	g/l
$CaSO_4 \cdot 5H_2O$	0.0005	g/l
H_3BO_3	0.0005	g/l
$Na_2MoO_4 \cdot 2H_2O$	0.002	g/l
NiCl	0.0025	mg/l
$ZnSO_4 \cdot 7H_2O$	0.012	g/l
$CoSO_4 \cdot 7H_2O$	0.0023	mg/l
KI	0.0001	g/l
$FeSO_4(NH_4)_2SO_4 \cdot 6H_2O$	0.035	g/l
myo-Inositol	0.125	g/l
Pyridoxine-HCl	0.00625	g/l
Ca-n-Pantothenate	0.00625	g/l
Thiamine-HCl	0.005	g/l
Nicotinic Acid	0.005	g/l
D-Biotin	0.000125	g/l
Carbon Source (*e.g.*, Glucose)	0-50	g/l
EDTA	0.1	g/l

Table 3.2
Mineral Stock Solution (100X)

Compound	Weight-Volume	
H_3PO_4 (85%)	160.	ml
KCl	12.00	g
$CaCl_2 \cdot 2H_2O$	20.00	g
NaCl	6.00	g
$MnSO_4 \cdot H_2O$	2.40	g
$CaSO_4 \cdot 5H_2O$	0.05	g
H_3BO_3	0.05	g
$Na_2MoO_4 \cdot 2H_2O$	0.20	g
NiCl	0.25	mg
$ZnSO_4 \cdot 7H_2O$	1.20	g
$CoSO_4 \cdot 7H_2O$	0.23	mg
KI	0.01	g
Add water to	1 liter	

Table 3.3
Vitamin Stock Solution (100X)

Compound	Weight-Volume	
myo-Inositol	12.5	g
Pyridoxine-HCl	0.625	g
Ca-n-Pantothenate	0.625	g
Thiamine-HCl	0.5	g
Nicotinic Acid	0.5	g
D-Biotin	0.0125	g
Add water to	1 liter	

Table 3.4
Normal Strength Working Nutrient Solution

Compound	Weight-Volume	
Phthalic acid, monopotassium salt	0.20	g
$MgCl_2 \cdot 6H_2O$	0.52	g
EDTA	0.1	g
$(NH_4)_2SO_4$	12.00	g
Mineral Stock Solution	10.	ml
$FeSO_4(NH_4)_2SO_4 \cdot 6H_2O$	0.035	g
Vitamin Stock Solution	10.	ml
Carbon Source (*e.g.* Glucose)	0-50	g
KOH (for pH=5.0)	1.62	g
Add water to	1 liter	

Adjust pH to 5.00 with 1N KOH solution

the drop-wise addition of HCl or KOH solutions. The recipe represents a modified version of that used by Oura (1974). Because of the dissociation constant associated with the phosphate ions, phosphate cannot be used effectively as a buffering agent below a pH of 5.5. Instead, phthalic acid was used. This choice was based on the observation that phthalic acid did not support the growth of *S. cerevisiae* nor affected the growth dynamics (Davison, 1985). This defined medium, which contained various minerals essential for cell growth, as well as the heat-sensitive vitamins, was not autoclaved. Instead, it was forced through an autoclavable 0.2 μm filter (Gelman) under the influence of gravity into two 23-liter autoclaved nutrient jars. The minerals and vitamins were present in the medium in an excess amount so that the cell growth rate was not limited by any of the trace elements.

Fermentor Start-up: The fermentor was thoroughly cleaned with distilled water, and all the necessary attachments to the fermentor were made. Except for some vents that were provided by air filters, the whole fermentation unit, including the accompanying tubings, filters, and the various reservoir jars, was totally sealed from the outside environment. The entire unit was autoclaved at 112 °C for 30 minutes. After cooling to room temperature, the fermentor was filled to the working volume with the same defined growth nutrient as that used to propagate the inoculum in flasks. The fermentor was often left uninoculated for 4-5 days to test for sterility under aeration. When the existence of sterility was confirmed by visually inspecting the clearness of the broth, the fermentor was inoculated aseptically with a sterilized syringe containing approximately 50 ml of the exponentially growing culture.

After autoclaving, only under some unavoidable situations, such as the changing of a waste jar or a filter, was the fermentor ever briefly open to the outside world. Then, the aseptic procedures were strictly followed to reduce the possibility of contamination. Sometimes a 20 % phosphoric acid solution was used in lieu of

ethanol for sterilization when ethanol production/consumption was being studied. A continuous operation free of any contaminants of more than three months was sometimes achieved. The presence/absence of contaminants in the fermentor during a run was determined via visual inspection of the sample under a microscope. It was double-checked by streaking a loopful of the undiluted sample taken from the fermentor on YPG and EMB (Eosin Methylene Blue) agar plates. EMB medium was prepared according to the following recipe: 10.0 g/l peptone, 10 g/l glucose, 5 g/l dipotassium phosphate, 0.4 g/l eosin, 0.065 g/l methylene blue, and 13.5 g/l agar. The presence of contaminants in the nutrient feed reservoir was determined by the turbidity of the solution.

Off-Line Sampling: Off-line samples were aseptically withdrawn from the sampling port periodically. Extra care was taken to ensure that the collected sample was indeed representative of the broth in the fermentor. During each sampling, by squeezing the rubber bulb, the small quantity of the stagnant liquid originally present in the sampling tube was first purged back into the fermentor. The rubber bulb was then released to withdraw approximately 7 ml of the broth from the fermentor. The sample was collected in a glass vial attached to the port. As a Bunsen burner flamed around the sampling port, the original glass sampling vial was removed and quickly replaced with another sterile glass vial. The same sampling procedure was repeated for the second time to withdraw another aliquot of approximately 7 ml. Although two separate vials of samples were taken each time, only the content from the second vial was used for the off-line analysis. The content from the first vial was discarded because when subjected to off-line analysis it yielded concentration readings that were significantly different from those of the subsequent samples. It was probably due to entrainment from the stagnant liquid in the sampling tube despite the initial purging.

From the second vial, 4 ml of the off-line sample was immediately poured into a 5-ml syringe and forced through a 0.45 μm in-line filtration unit (Millipore) that was locked at the tip of the syringe through a Luer-Lok fitting. The filtrate collected in a small, tightly capped glass vial was quickly frozen. The frozen filtrate was to be defrosted later and subjected to a series of analysis for glucose, ethanol, and acetic acid concentrations. Everything was at ready before each sampling so that the time elapsed during the entire sampling procedure, between the initial sample collection and the placement of the filtrate in a freezer, was usually no more than 10 seconds. The lower detection limit of the enzyme kits used in the off-line measurements proved not to be the limiting factor in determining the accuracy of the off-line measurements when the concentrations of glucose or ethanol were low. Rather, it was the ability to stop quickly the consumption of glucose or ethanol. For example, with a specific growth rate of 0.5 hr^{-1}, a cell concentration of 2 g/l, and a substrate to cell yield coefficient of 0.4, the decrease in the substrate concentration in the 10 seconds it took to filter out the cells is $\frac{(0.5)(2)(10)}{(3600)(0.4)} \simeq 7$ mg/l.

Cell Size and Size Distribution: A Coulter counter was routinely used to count the number of cells and the cell size distribution. A 0.100 ml off-line sample from the fermentor was diluted in 10.0 ml of filtered electrolyte solution. The theory of operation of a Coulter particle counter is described in detail in the Operating Manual accompanying the instrument. It is described by Hunt, 1980, and Davison, 1985, as well. The operation of the device is based on the change in the resistivity as particles of different sizes suspended in an electrolyte solution pass through a small orifice that separates the two electrode plates. The sudden jump in the resistivity is detected and counted electronically. Furthermore, the magnitude of the jump in the resistivity signal is approximately linear with the diameter of the particle, although it also depends to a certain extent on the type of microorganisms, growth conditions, and viability. A range of interchangeable orifice sizes are available; the

best results are achieved with an orifice diameter of 2 to 20 times the diameter of the particles to be measured. A large orifice-to-particle diameter ratio decreases the measurable signal intensity, and a small diameter ratio invites chronic plugging problems. Since the yeast cells were approximately 5 μm in diameter, a 20 μm orifice was used consistently in this study. This distribution in the signal fluctuation is sorted with a multichannel analyzer (MCA-ND-555) operated at 1/amplification of 8 and 1/(aperture current) of 8. The channel number is a measure of the volume of the particle passing through the orifice in a Coulter counter; therefore, the cell size is approximately proportional to (channel number)$^{\frac{1}{3}}$, with channel 60 corresponding to 6 μm. The channel number to physical size conversion was determined with latex calibration particles of well-defined uniform diameters (Duke Scientific). Data on frequency and channel number were graphically displayed and sent to a microcomputer through an RS232C interface for further analysis and processing. The volume used for each set of analysis was 50 μl after dilution with the electrolyte solution. This gave a typical total count of 10^4–10^5, *e.g.*, $2 \times 10^7 \sim 2 \times 10^8$ cells/ml, after subtracting the background.

Biomass Concentration Measurement: One ml of the off-line sample was diluted with 5 ml of distilled water, and the optical density was measured at 660 nm with a Spec 21 spectrophotometer (Bausch and Lomb). It was found from Coulter counter measurement that the cell size depended on the growth rate, thus, the dilution rate. Consequently, the optical density to cell dry weight correlation was a function of the operating conditions, such as the dilution rate, because the optical density depended on, among other variables, the cell size. Figure 3.5.1 demonstrates the shift in the cell size distribution as measured with a Coulter counter after the dilution rate was shifted from 0.070 hr^{-1} to 0.229 hr^{-1}. Figure 3.5.2 shows the dependence of the ratio of dry weight to optical density as a function of the dilution rate for yeast growing on glucose. Calibration curves were frequently generated to convert

the optical density measurement to the equivalent dry weight measurement. One such calibration curve is shown in Figure 3.5.3. The dry weight was determined by first passing 100 ml of the withdrawn sample through a pre-weighed 45-μm filter membrane (Millipore) that had been dried and stored in a desiccator, then followed with ample washing. The cells were dried along with the filter paper in a petri dish at 105 °C until the weight of the filter cake remained unchanged, *i.e.*, $\pm$0.001g, over a 6-hour period of time. This generally took 12–24 hours. Prolonged heating was avoided because it caused oxidation and browning of the filter cake and the filter membrane.

Elemental Analysis: Some of the representative dried biomass samples were analyzed for their elemental composition by Larry Henling of Caltech's Analytical Lab. Among the elements analyzed were C, H, N, and the ash content. The missing weight was assumed to be contributed by oxygen, which was not independently analyzed. Organic compounds of known compositions were subjected to the same analytical procedures to verify the accuracy of the externally contracted analysis. Some of the results for those samples taken under various steady-state conditions are presented in Table 3.5.5 for the culture grown on glucose and in Table 3.5.6 for the culture grown on ethanol. Oxygen and hydrogen contents seemed to be uncorrelated to the growth rate. However, a higher nitrogen content, which is related to the protein content in cells, was detected for faster growing cells under either glucose or ethanol limitation. This observation is graphically shown in Figures 3.5.4 and 3.5.5.

Enzymatic Analysis: At the conclusion of each run, the frozen off-line samples were defrosted and analyzed for the glucose and ethanol concentrations with #15-UV and #331-UV enzymatic kits, respectively, from Sigma. Acetic acid concentration was determined with a Boehringer UV enzymatic kit. These enzymatic

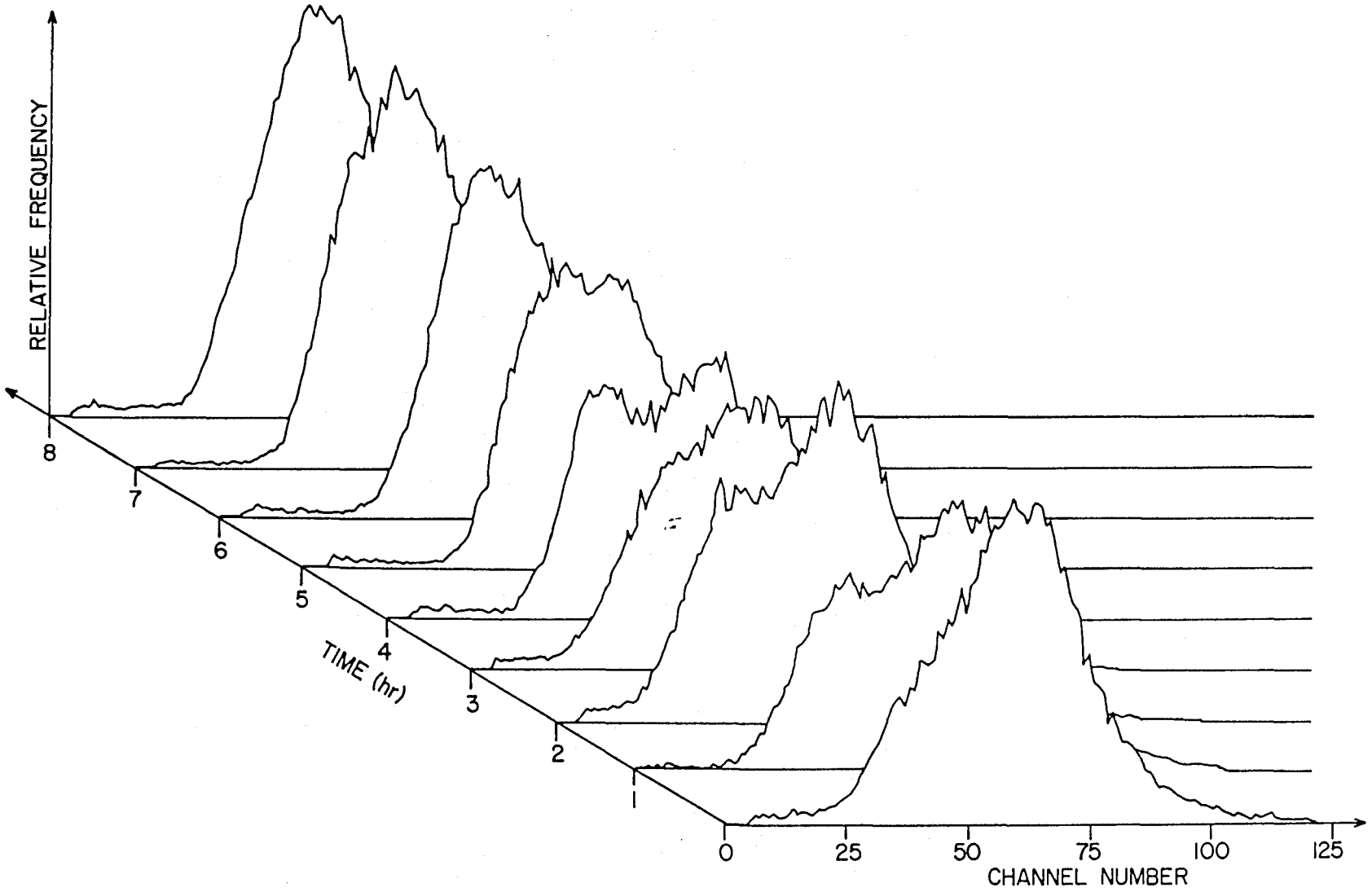


Figure 3.5.1. Changes in the cell size distribution after the dilution rate is shifted up from 0.070 hr^{-1} to 0.229 hr^{-1}.

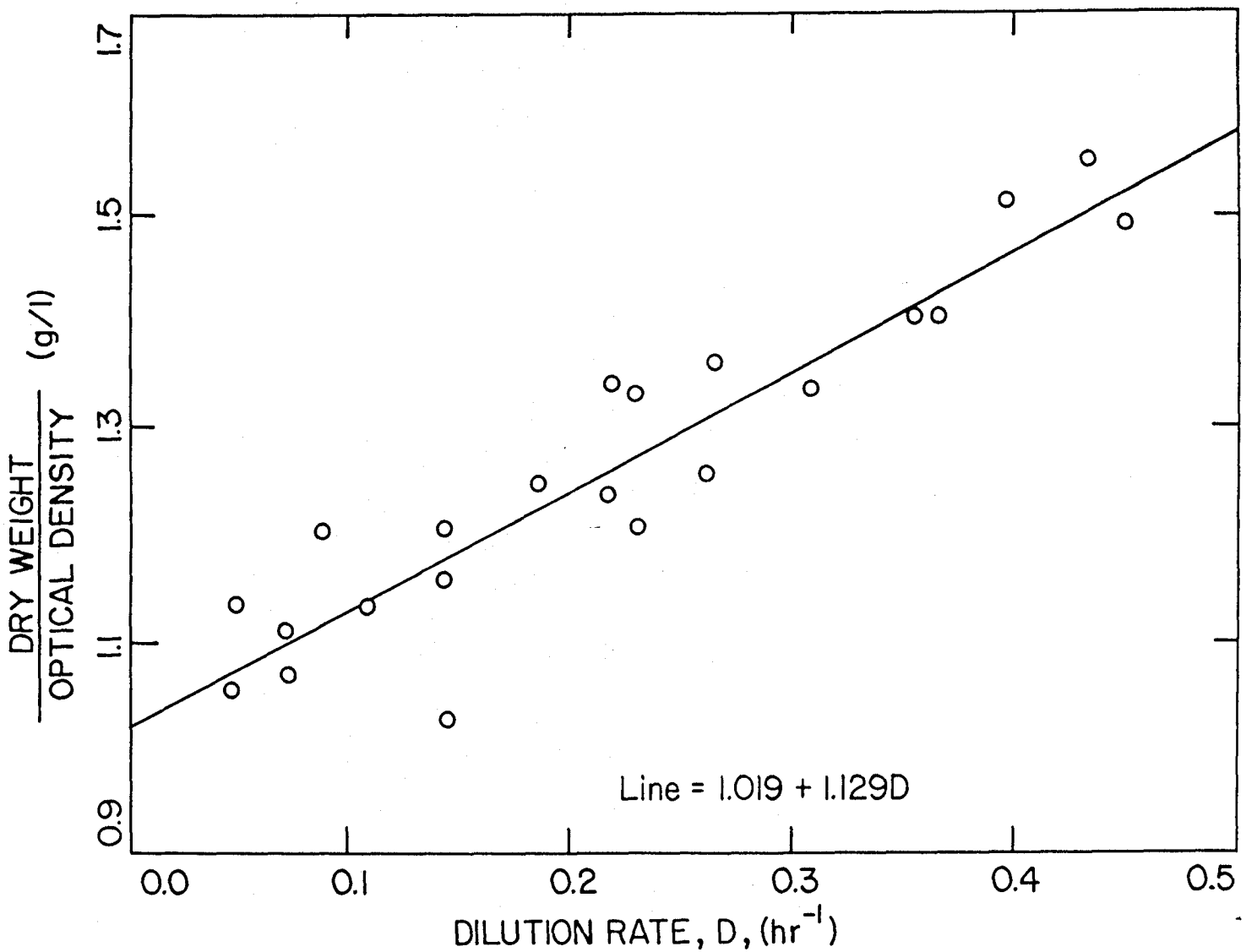


Figure 3.5.2. Ratio of dry weight to optical density vs. dilution rate for cells grown in 5.0 g/l of glucose in the feed.

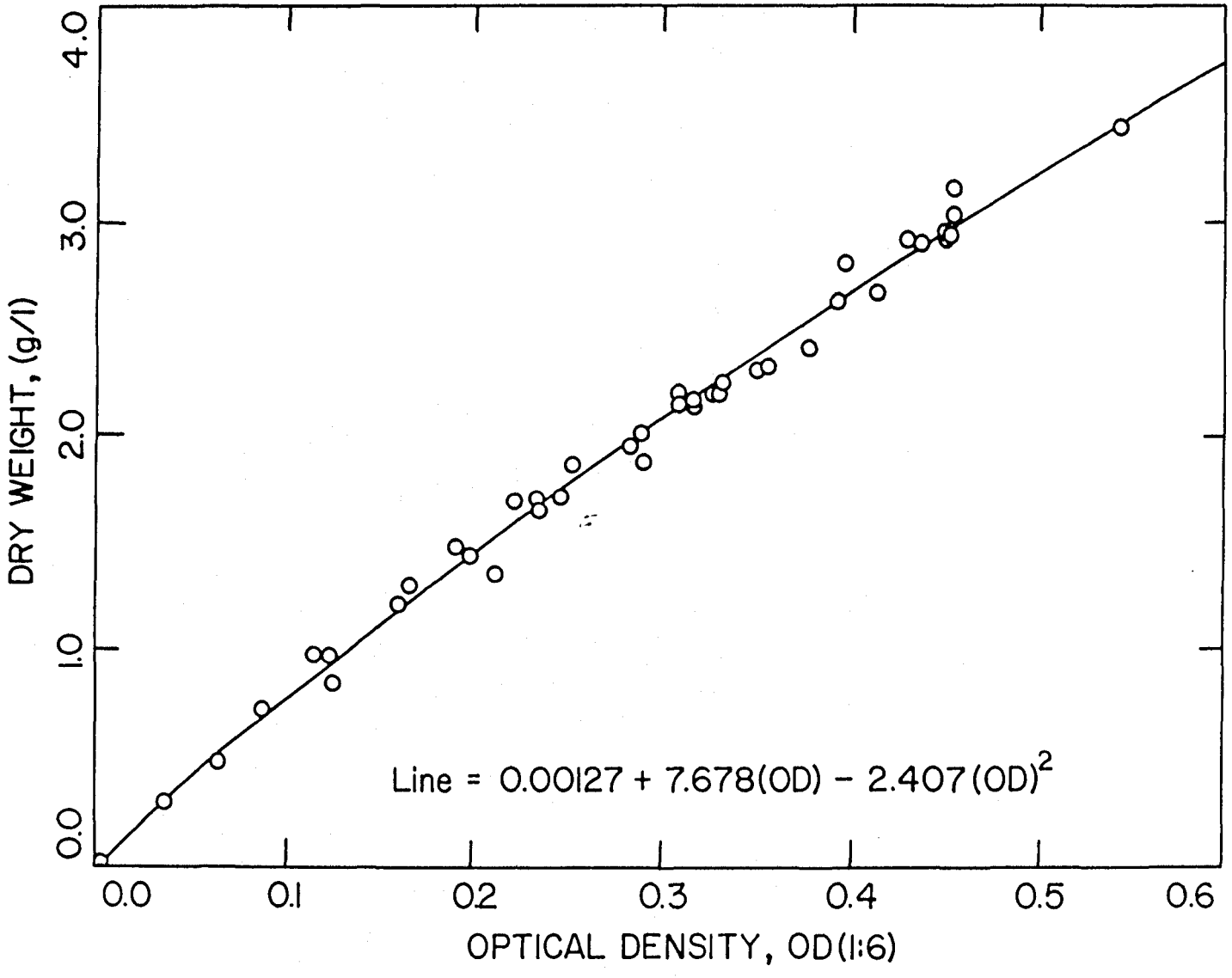


Figure 3.5.3. Calibration curve of dry weight versus optical density at a sixfold dilution.

Table 3.5
Elemental Analysis of C, H, N, & Ash for Steady-State Growth in Glucose.

Dilution Rate	Weight %			Elemental Composition		
(hr^{-1})	C	H	N	β	γ^*	δ
0.0722	43.83	6.35	7.97	1.739	0.545	0.156
0.1419	43.88	6.29	8.41	1.720	0.537	0.164
	43.78	6.44	8.49	1.765	0.536	0.166
0.1566**	43.88	6.27	8.80	1.715	0.531	0.172
	43.51	6.37	8.88	1.757	0.538	0.175
0.2634	44.58	6.48	8.72	1.744	0.508	0.168
0.3542	43.34	6.24	8.90	1.728	0.545	0.176
	43.00	6.22	8.91	1.736	0.556	0.178
0.4317	42.93	6.21	9.23	1.736	0.553	0.184
	42.88	6.21	9.24	1.738	0.554	0.185
Average	43.56	6.31	8.76	1.738	0.540	0.172

* γ was calculated by taking the missing weight with 10 wt% ash.
** The cells were elongated.

Table 3.6
Elemental Analysis of C, H, N, & Ash for Steady-State Growth in Ethanol.

Dilution Rate	Weight %				Elemental Composition		
(hr^{-1})	C	H	N	Ash	β	γ^*	δ
0.0000	45.10	6.35	8.26	9.7	1.690	0.509	0.157
	44.94	6.37	8.26	10.0	1.701	0.508	0.158
0.0640**	47.54	6.82	7.79	6.2	1.721	0.499	0.140
0.0918	45.21	6.49	7.92	9.6	1.723	0.511	0.150
	45.19	6.49	8.01	9.5	1.723	0.511	0.152
0.1040	45.88	6.47	8.14	9.0	1.692	0.499	0.152
	46.11	6.45	8.15	8.9	1.679	0.494	0.152
0.1146	45.71	6.33	8.24	10.0	1.662	0.488	0.155
	45.56	6.38	8.28	9.6	1.680	0.497	0.156
0.1252	45.05	6.20	8.46	12.2	1.651	0.468	0.161
	44.87	6.36	8.46	11.3	1.701	0.485	0.162
0.1325	45.28	6.36	8.47	11.2	1.686	0.475	0.160
	45.33	6.33	8.43	10.9	1.676	0.480	0.159
0.1391	44.56	6.34	8.48	10.2	1.707	0.512	0.163
	44.45	6.38	8.48	9.8	1.722	0.521	0.164
0.1433	44.58	6.24	8.59	10.4	1.680	0.508	0.165
	44.59	6.28	8.59	10.0	1.690	0.514	0.165
Average**	45.15	6.36	8.33	10.1	1.691	0.499	0.158

* γ was calculated by taking the missing weight.
** D=0.0640 hr^{-1} is excluded from averaging.

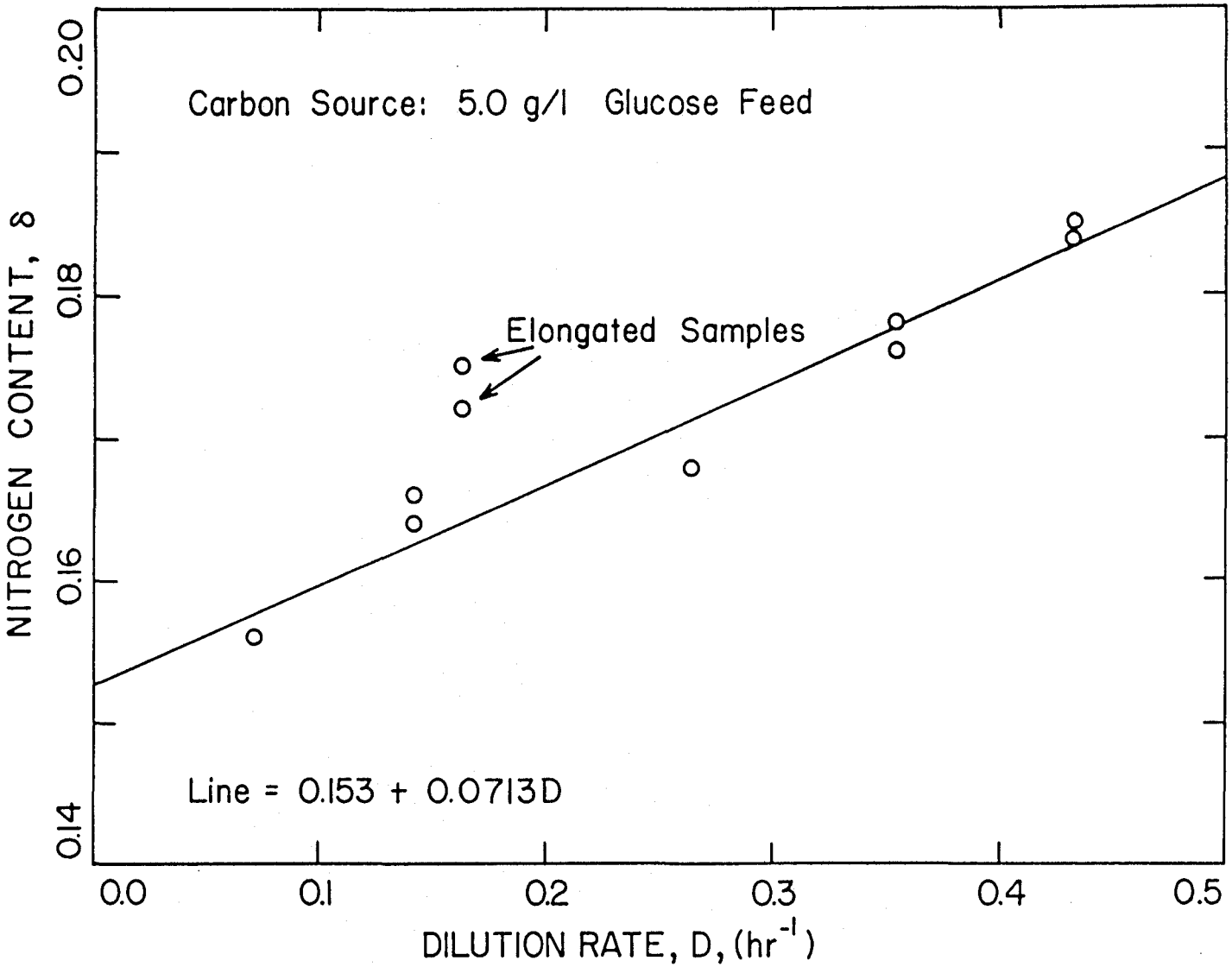


Figure 3.5.4. Changes in the nitrogen content of cell biomass as a function of dilution rate (*i.e.*, specific growth rate) for cells grown in 5.0 g/l of glucose in the feed.

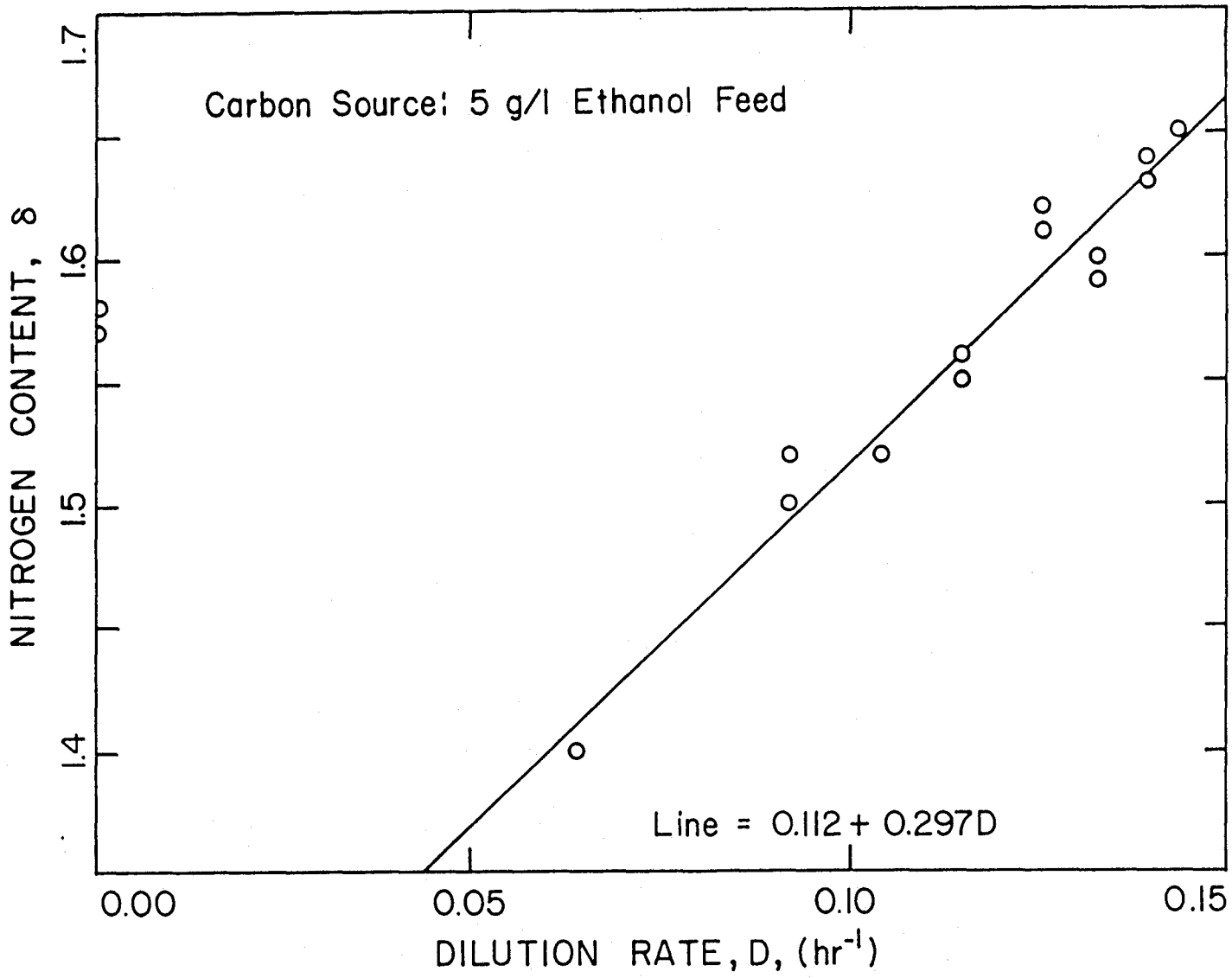


Figure 3.5.5. Changes in the nitrogen content of cell biomass as a function of dilution rate (*i.e.*, specific growth rate) for cells grown in 5.0 g/l of ethanol in the feed.

methods all work quite similarly. The oxidation of the substrate under consideration is specifically catalyzed by an appropriate set of enzymes, and the reactions are coupled such that nicotinamide adenine dinucleotide (NAD) is reduced to NADH, which absorbs light in the UV range (340 nm). For example, in the enzymatic glucose determination, glucose is oxidized by NAD via a two-step process with the participation of hexokinase:

$$\text{Glucose} + \text{ATP} \xrightarrow{\text{Hexokinase}} \text{G-6-P} + \text{ADP} \qquad (3.5.1)$$

$$\text{G-6-P} + \text{NADP} \xrightarrow{\text{G-6-PD}} \text{6-PG} + \text{NADPH}\,, \qquad (3.5.2)$$

where the following abbreviations are used: ADP (adenosine diphosphate), ATP (adenosine triphosphate), G-6-P (glucose-6-phosphate), G-6-PD (glucose-6-phosphate dehydrogenase), NADP (nicotinamide adenine dinucleotide phosphate), NADPH (nicotinamide adenine dinucleotide phosphate, reduced), and 6-PG (6-phosphogluconic acid). The presence of the end product NADPH causes a linear increase in the absorbance at 340 nm. The detailed principle of measurement and analytical procedures are explained in the operation manuals that accompany each product, and will not be reiterated here. The procedure described by the manufacturer, however, was modified slightly, depending on the concentration of the applicable component in the sample.

In the modified procedure, cold distilled water (for glucose kit) or pyrophosphate buffer solution (for ethanol kit) was added to freeze-dried enzyme preparation in the vial provided as prescribed by the original procedure. The reagent vial was gently repeatedly inverted to dissolve the content. A small aliquot of the reconstituted working enzymatic reagent, usually 3.0 ml, was transferred with an adjustable digital pipet to a cuvet. The absorbance of the enzyme solution before the addition of the sample, A_0, was recorded. Immediately, a small, known volume of the sample was added to the enzyme solution and mixed. The enzymatically catalyzed

reaction was allowed to proceed for approximately 10 minutes until the absorbance no longer increased. This final absorbance of the reacted solution, A_1 was recorded. The concentration of glucose or ethanol in the sample is calculated from the following equations:

$$\Delta A = A_1 - A_0 \frac{V_{\text{enzyme}}}{V_{\text{enzyme}} + V_{\text{sample}}} \tag{3.5.3}$$

$$f_{\text{dilution}} = \frac{V_{\text{enzyme}} + V_{\text{sample}}}{V_{\text{sample}}} \tag{3.5.4}$$

$$\text{Conc. (g/l)} = f_{ac} \times \Delta A \times f_{\text{dilution}}, \tag{3.5.5}$$

where V_{enzyme} is the volume of the enzyme reagent used, and V_{sample} is the volume of the sample. The calculation of the increase in the absorbance, ΔA, also takes into consideration the effect of adding sample to the enzyme solution on the blank absorbance. Finally, the dilution factor, f_{dilution}, is combined with the change in the absorbance and the absorbance-to-concentration conversion factor, f_{ac}, to yield the concentration. f_{ac} was 0.028965 and 0.0073955 for glucose and ethanol, respectively, although the actual values varied slightly between runs. The vendors claimed that these highly specific enzymatic methods required no extensive standardization and that the absorbance readings were reliable without considering the interferences from other species. Nevertheless, the absorbance-to-concentration calibration curves were routinely generated by following the same procedures with standard solutions of known glucose and ethanol concentrations in this work to update f_{ac}.

With this modification, the detection limits of these enzymatic methods were widened considerably. For example, by using a higher ratio of sample volume to enzyme reagent volume, the lowest concentrations detectable by the method can be decreased from 2.0 mg/l to 0.1 mg/l for glucose and from 0.1 mg/l to 0.05 mg/l for ethanol.

Gas Chromatography: The saved off-line filtrate samples were acidified to approximately pH=1.52 with 10 μl of 12N HCl. At this pH, more than 99.9% of the

total acetate is present in protonated form, *i.e.*, acetic acid, that can be detected with a gas chromatograph equipped with a Chemsorb 102 column and a flame ionization detector. The ethanol in the filtered sample was also resolved and determined simultaneously from the gas chromatograph. Thus, the off-line ethanol and acetic acid concentrations were determined with two independent methods. The results, as compared with those from the off-line enzymatic method described above, were quite reproducible. The off-line ethanol measurements also agreed with the on-line data.

Other experimental procedures, analytical methods, and microbial cultivation techniques closely follow those described by San (1984) and Davison (1985) and will not be repeated here.

3.6 DATA ANALYSIS PROCEDURES

At the conclusion of each experiment, the original on-line data files stored in a CP/M format on 8.5 inch floppy diskettes during that run were transferred to a VAX 11/780 computer and backed up on a magnetic tape. They were also converted to the PC/DOS (Disk Operating System) format so that the wide range of software developed to run under that operating system could be fully utilized. The subsequent off-line data reduction/analysis was performed mostly on an IBM Personal Computer with 640K operated at 4.7MHz. Except for a few standard subroutines, all the programs needed for data acquisition, reduction, analysis, and presentation were written by the author. A few of the critical ones used in the data analysis of this thesis are listed in Appendix C.

Figure 3.6.1 shows the major steps taken to reduce the raw experimental data. Because of the large amount of information, the files were named consistently, with the first part of the file name indicating the date of the start of the run and the file extension (.XXX) indicating the type of information stored. An interactively

operated program named CORRECT was immediately used to display the logged voltage data on the oxygen concentration, carbon dioxide concentration, pressure, base addition rate, optical density, glucose concentration, and ethanol concentration on two computer monitors, one graphically and the other numerically. The occasional discrete sharp spikes in the raw data were replaced with the last sound data. These extreme noises, if left unaltered, sometimes caused convergence problems in the integration routine used in the subsequent Kalman filter.

The off-line calibration information was processed through regression analysis. Based on the calibration data, individual programs BIOMASS, GLUCOSE, ETHANOL, CEROUR, and PH were run interactively to convert the on-line voltage or absorbance readings to biomass concentration (g/l), glucose concentration (g/l), ethanol concentration (g/l), CER (mole/l), OUR (mole/l), RQ (dimensionless), and ammonia uptake rate (mole/l-hr). The conversions were based on the calibration information supplied to the programs as the computer prompted for them at the beginning of the program execution. Note that the pressure effect was properly considered in the calculation of the gaseous oxygen concentration. Based on the information on the biomass composition, CER, OUR, and ammonia uptake rate, the program BALANCE performed elemental material balance to give the substrate-to-biomass yield coefficient, Y_s, the product-to-biomass yield coefficient, Y_p, and the total growth rate, R. A negative product-to-biomass yield coefficient indicated that the component of interest was being utilized rather than produced by the microorganism. Finally, the results from BALANCE were combined with the concentration measurements in the program KALMAN to obtain better filtered estimates on the concentrations and the observed specific growth rate as functions of time.

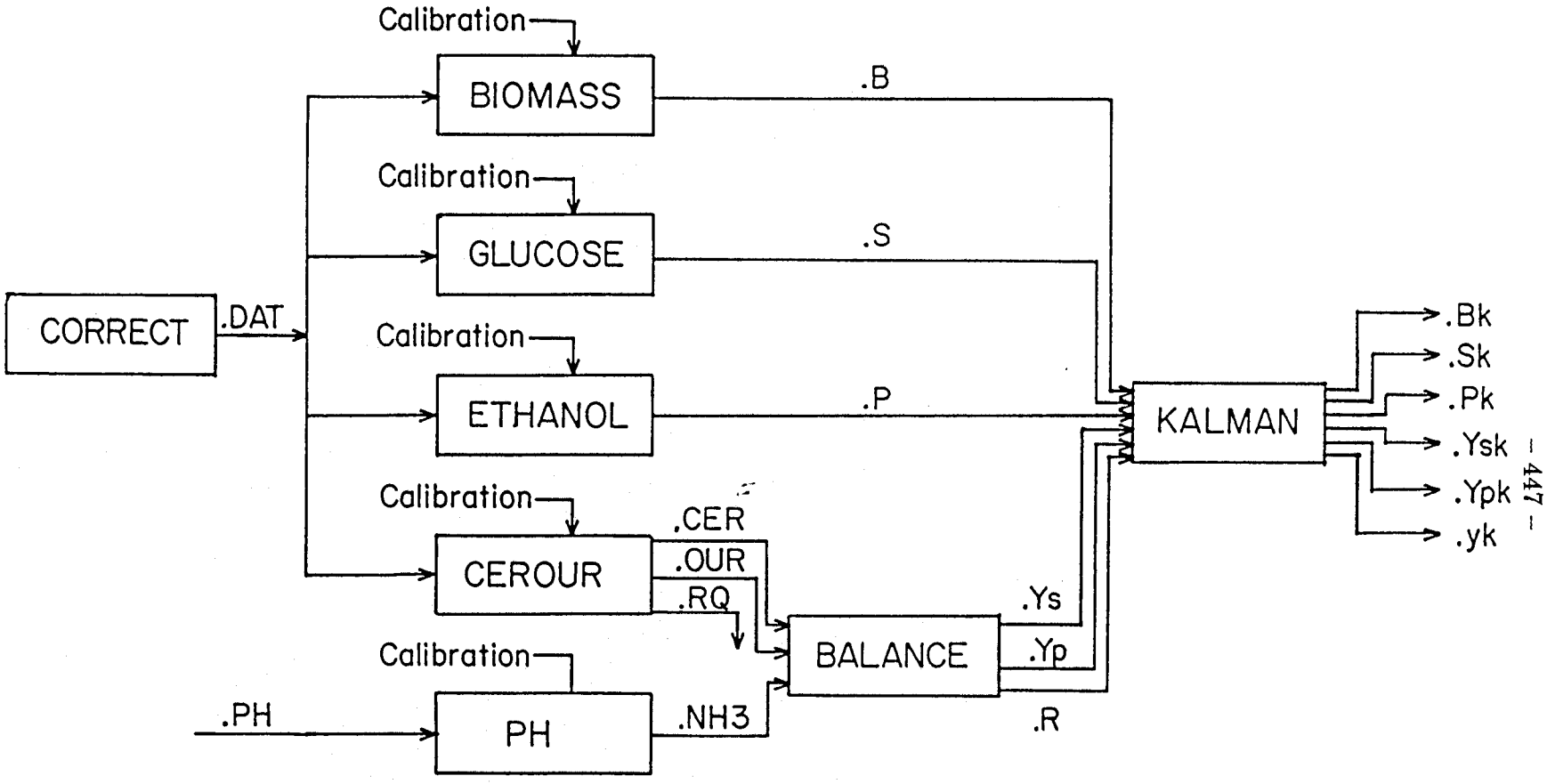


Figure 3.6.1. Data analysis flow diagram of continuous measurements.

The intrinsic specific growth rate as a function of time was calculated from the Kalman estimate of the substrate concentration and the curve on the intrinsic specific growth rate versus the substrate concentration, as discussed in Chapter 2. The kernel inversion programs listed in Appendix D were applied to the intrinsic specific growth rate and the Kalman estimate of the observed specific growth rate to calculate the time-lag kernel.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 INTRODUCTION

To reconstruct a time-lag kernel, the simulation studied in the previous sections suggests that the steps illustrated in Figure 4.1.1 be followed. Although this example is based on a kernel associated with the time-lag effects in the specific growth rate, kernels associated with other variables can also be treated analogously. According to Figure 4.1.1, the first step in the experimental determination of the kernel is to measure the intrinsic specific growth rate. Because the specific growth rate in the absence of any time-lag effect is defined to be the intrinsic specific growth rate, one can measure it when the "quality", or the internal chemical composition of the cell remains unchanged for a long period of time such that any possible time-lag effects cease to dominate the observed behavior. When the intrinsic specific growth $\mu(t)$ stays at a constant value, the observed specific growth rate, $y(t)$, expressed as $y(t) = \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh$, approaches the intrinsic specific growth rate. This concept can be shown mathematically as follows:

$$\lim_{t\to\infty} y(t) = \lim_{t\to\infty} \int_{-\infty}^{t} \mu(h)k(t-h)dh. \tag{4.1.1}$$

With the change of variables $h' = t - h$, the above relationship is transformed to:

$$\lim_{t\to\infty} y(t) = \lim_{t\to\infty} \int_{0}^{-\infty} \mu(t-h')k(h')dh'. \tag{4.1.2}$$

Because the specific growth rate is constant, it is taken outside the integral in the above relationship. This leads to:

$$\begin{aligned} \lim_{t\to\infty} y(t) &= \mu \int_{0}^{-\infty} k(h')dh' \\ &= \mu_0. \end{aligned} \tag{4.1.3}$$

The last step is due to the realization that the 0th moment of the time-lag kernel function is normalized. This point is not coincidental; it is intentionally planned during the original formulation of the theory in order to expedite the physical

PROCEDURE FOR THE GENERATION OF TIME DELAY KERNEL

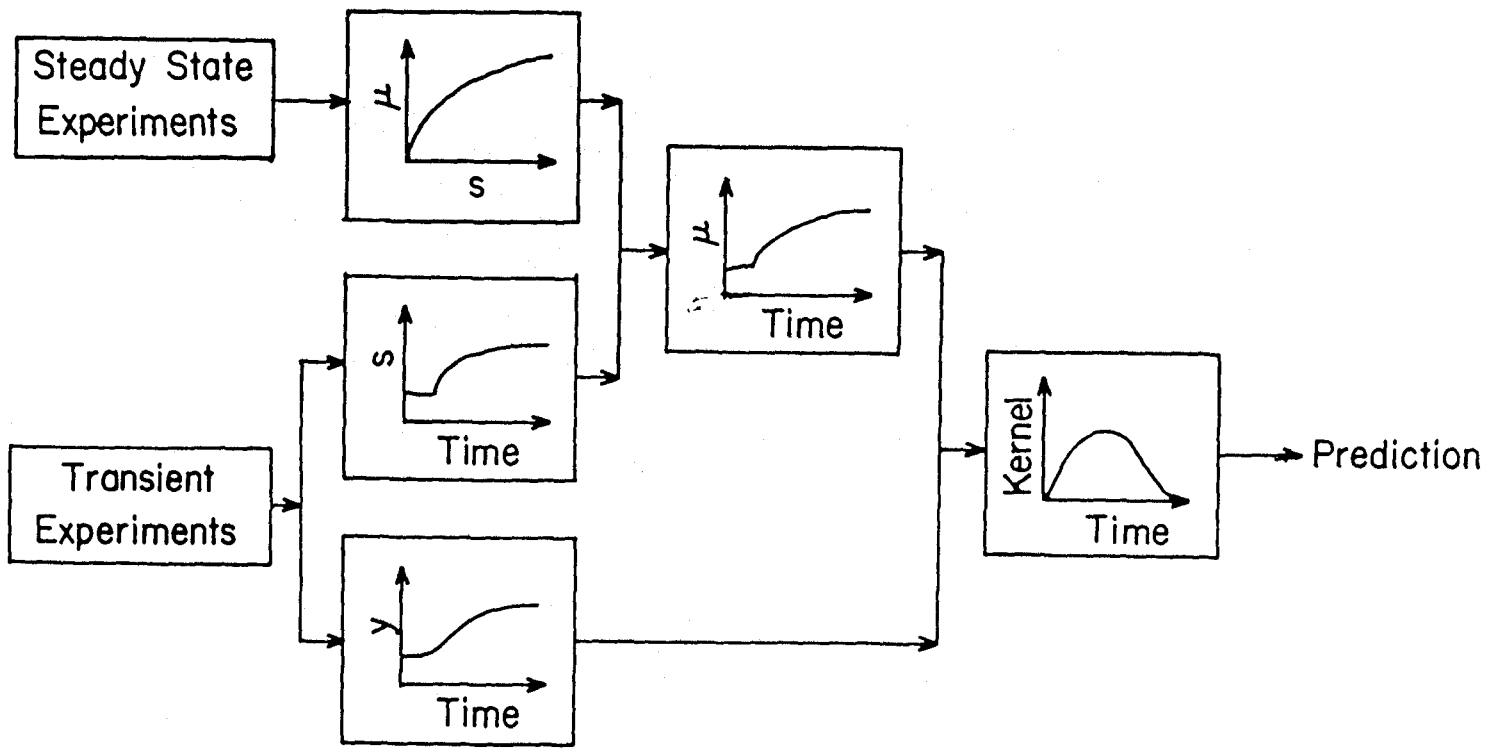


Figure 4.1.1. Experimental procedure for the generation of time-lag kernel.

interpretation of the kernel integral. If the 0th moment of the kernel function were not normalized, the translation from the observed values to the intrinsic values would not have been as straightforward. Thus, when the intrinsic specific growth rate, μ, is a function of limiting substrate concentration, s, in a fermentor, a μ versus s curve can be obtained through a series of steady-state runs with different dilution rates. Note that the specific growth rate as defined in the following set of equations is the same as the dilution rate for that steady-state run:

$$\frac{dx}{dt} = -Dx + \mu(s)x \tag{4.1.4}$$

$$\frac{ds}{dt} = D(s_f - s) - \frac{1}{Y_s}\mu(s)x. \tag{4.1.5}$$

As a result, the required μ versus s curve is constructed by rotating the steady-state s versus the operating D curve. If more than one limiting substrate is utilized during fermentation, more than one such series will be required for each of the limiting substrates present in the fermentor.

During a transient experiment in which the dilution rate or the feed substrate concentration is shifted up or down, the substrate concentration can be continuously measured or estimated as a function of time. By referring to the μ versus s curve established during the earlier steady-state runs, $\mu[s(t)]$ can be generated continuously as well. The substrate concentration needed to generate the $\mu(t)$ curve can either be estimated from a material balance around the fermentor or measured directly with an on-line analyzer. In addition to the substrate concentration, the adaptive Kalman estimation scheme presented earlier can be used to provide a continuous estimate of the observed instantaneous specific growth rate $y(t)$. Finally, from the curves of $\mu(t)$ and $y(t)$, the kernel is generated with the same method as that used to simulate the reconstruction of a kernel.

In the following discussion, the procedure outlined above will be traced to study the time-lag dynamics in a fermentor. First, by using either glucose or ethanol as the

source for energy and carbon, a series of steady-state experiments are performed to determine the intrinsic behavior for the culture of *Saccharomyces cerevisiae*. Then, the shape of the kernel is determined in a variety of experiments in which the dilution rate is shifted up or down. Sinusoidal perturbation experiments of the dilution rate have also been completed for a range of forcing frequencies. With the aid of the parameter and state estimation algorithms, the on-line measurements are utilized to determine the state of the system, including the limiting substrate concentration and the observed instantaneous specific growth rate. From these estimates, the shape of the kernel function is determined. The presently proposed time-lag approach to bioprocess identification and modeling is ultimately tested in terms of the model's capability in predicting the microbial behavior in a variety of situations.

4.2 STEADY-STATE EXPERIMENTS ON ETHANOL

A series of steady-state experiments were performed with 5.0 g/l of ethanol in the feed stream. The biomass concentrations and ethanol concentrations were regularly measured, and it was decided that a steady-state had been reached when the biomass concentration and the ethanol concentration remained unchanged for no less than 5 to 10 residence times. The steady-state experiments were carried out in both directions. More specifically, the dilution rate was first started from a low value and was progressively shifted to higher settings in a step-wise manner. When the culture was nearly totally washed out and failed to reach a steady-state, and when it was evident that the maximum specific growth rate had been exceeded, the dilution rate was brought back down to a value less than the maximum specific growth rate. From then on the dilution rate was shifted down progressively in small steps until it became difficult to assure that a steady-state was reached in

a reasonably acceptable amount of time (about a week). The complete cycle was repeated three times.

The resulting biomass concentration (dry weight) as a function of the dilution rate is plotted in Figure 4.2.1, and the wet weight and the optical density as a function of the dilution rate are plotted in Figures 4.2.2 and 4.2.3, respectively. The variation of the ratio of the dry weight to the optical density as a function of the cell growth rate is shown in Figure 4.2.4. As mentioned earlier, the variation in this ratio is probably due to changes in the cell morphology such as shape and size. In addition to the dry weight and the optical density, the wet weight of the cell is also measured. The wet weight is determined in the process of measuring the dry weight. After about 100 ml of the culture sample was collected and filtered, the wet filter cake was weighed, along with the filter membrane, and the weight of the wet filter membrane was subtracted from the total. Despite the fact that the method employed is a crude one, the variation in the ratio of the wet weight to the dry weight as shown in Figure 4.2.5 can be a rough indication of the water content in the cell. Although the trajectories of the biomass and ethanol concentrations were not exactly retraced between those steady-states reached after a shift-up and those reached after a shift-down, it could not be stated conclusively that a hysteresis existed for the combination of *S. cerevisiae* and ethanol because the deviation was relatively insignificant in view of the experimental errors.

Finally the ethanol concentration as a function of the dilution rate is shown in Figure 4.2.6. Note that the steady-state ethanol concentration remains at a relatively low value for the range of dilution rate below 0.07 hr^{-1}. The ethanol versus dilution rate curve is rotated to generate the more familiar representation of the μ versus s relationship, which is reproduced in Figure 4.2.7. Under the conditions studied, the dilution rate (specific growth rate) versus ethanol concentration curve

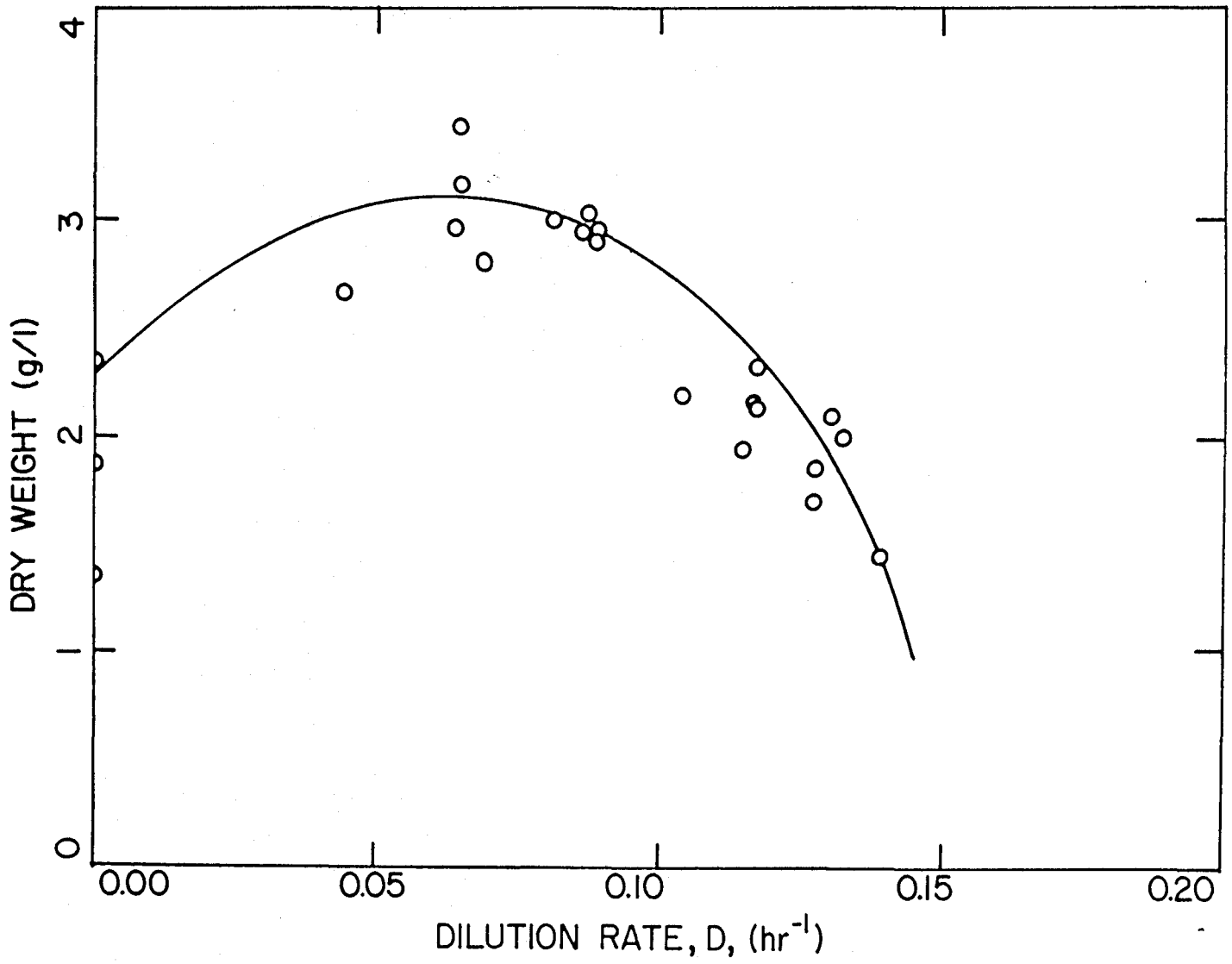


Figure 4.2.1. Changes in the steady-state biomass concentration (dry weight) as a function of dilution rate for cells grown in 5.0 g/l of ethanol in the feed.

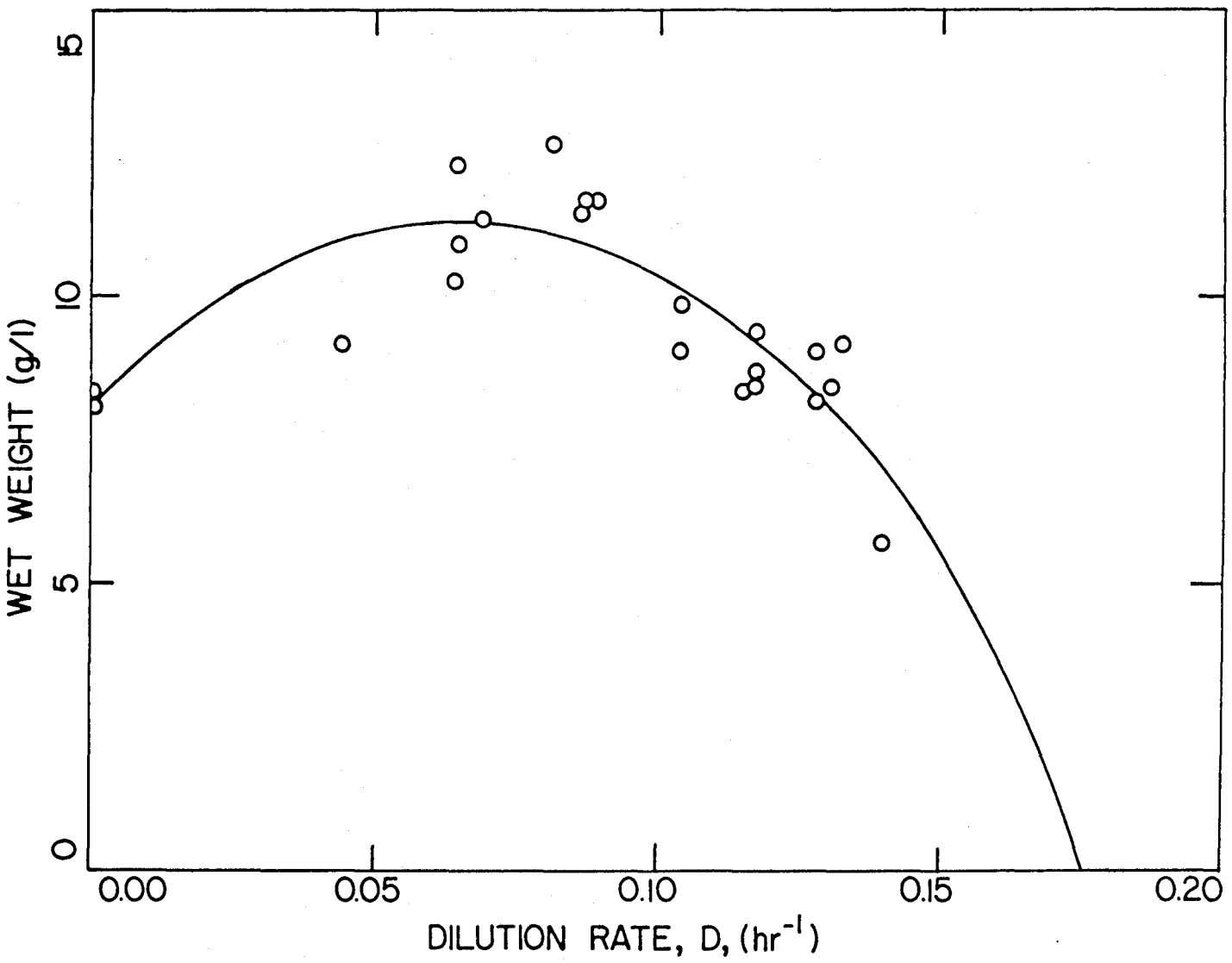


Figure 4.2.2. Changes in the steady-state wet cell biomass weight as a function of dilution rate for cells grown in 5.0 g/l of ethanol in the feed.

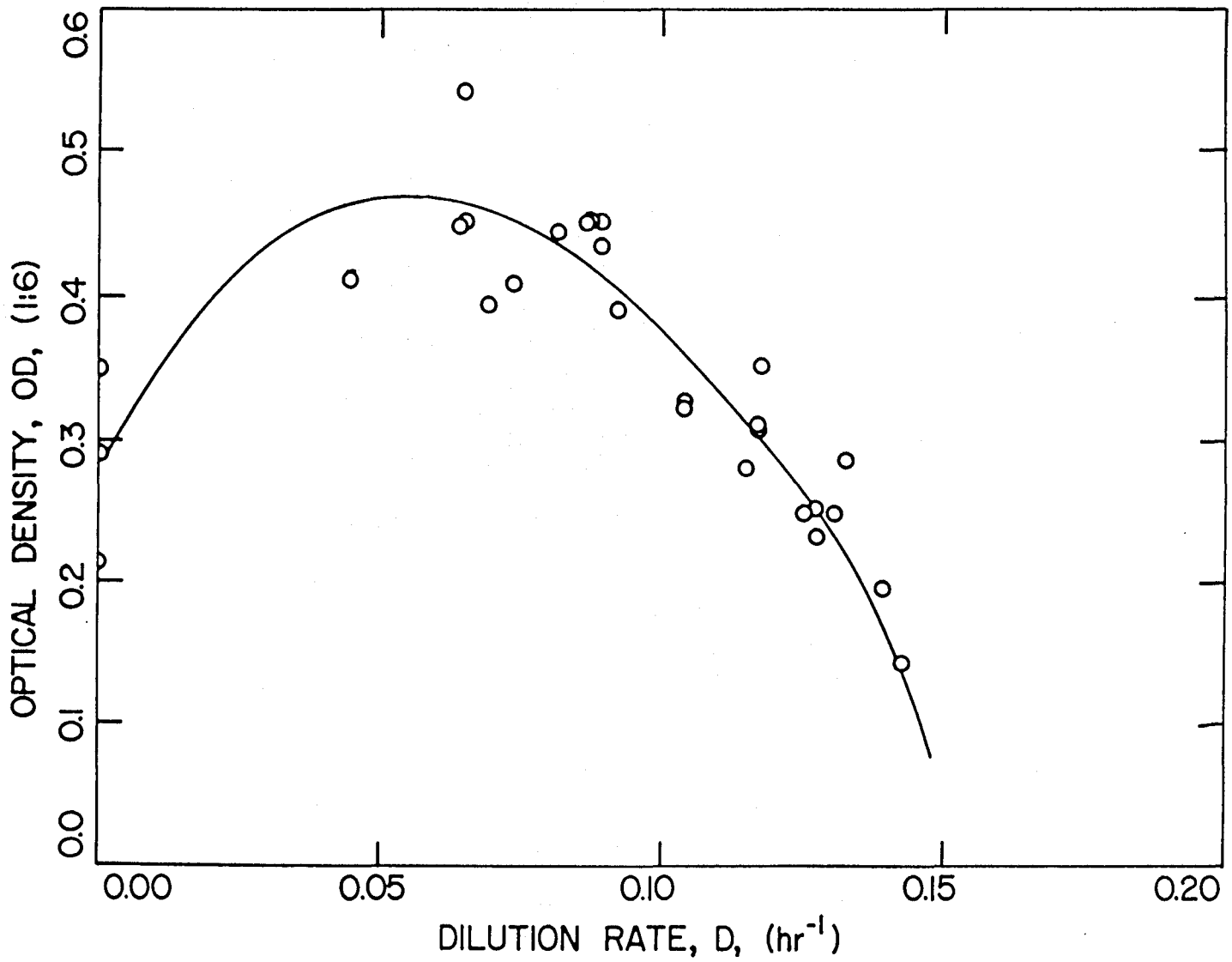


Figure 4.2.3. Changes in the optical density of the steady-state sample diluted 6 times as a function of dilution rate for cells grown in 5.0 g/l of ethanol in the feed.

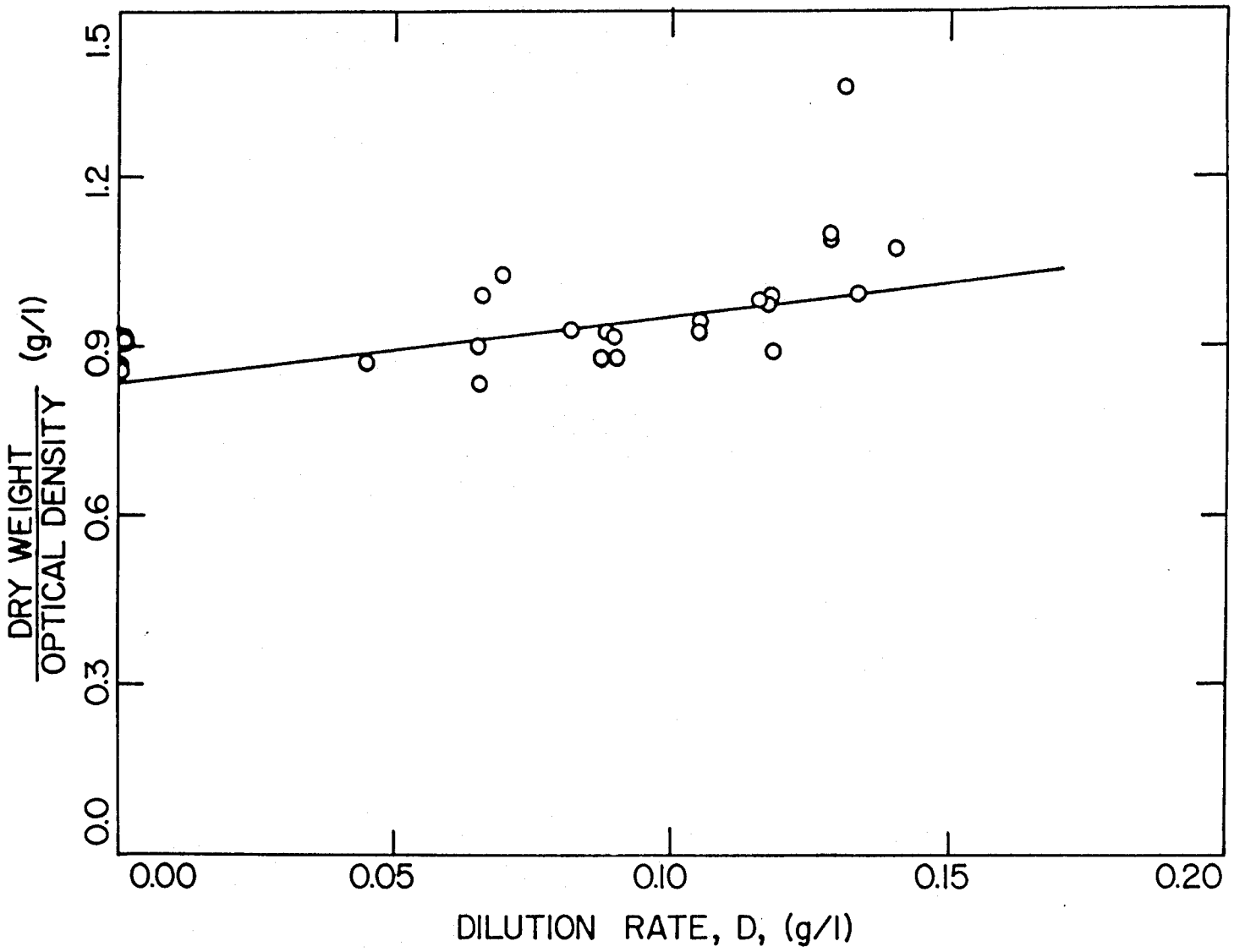


Figure 4.2.4. Ratio of dry weight to optical density vs. dilution rate for cells grown in 5.0 g/l of ethanol in the feed. The variation is probably caused by the changes in the cell morphology.

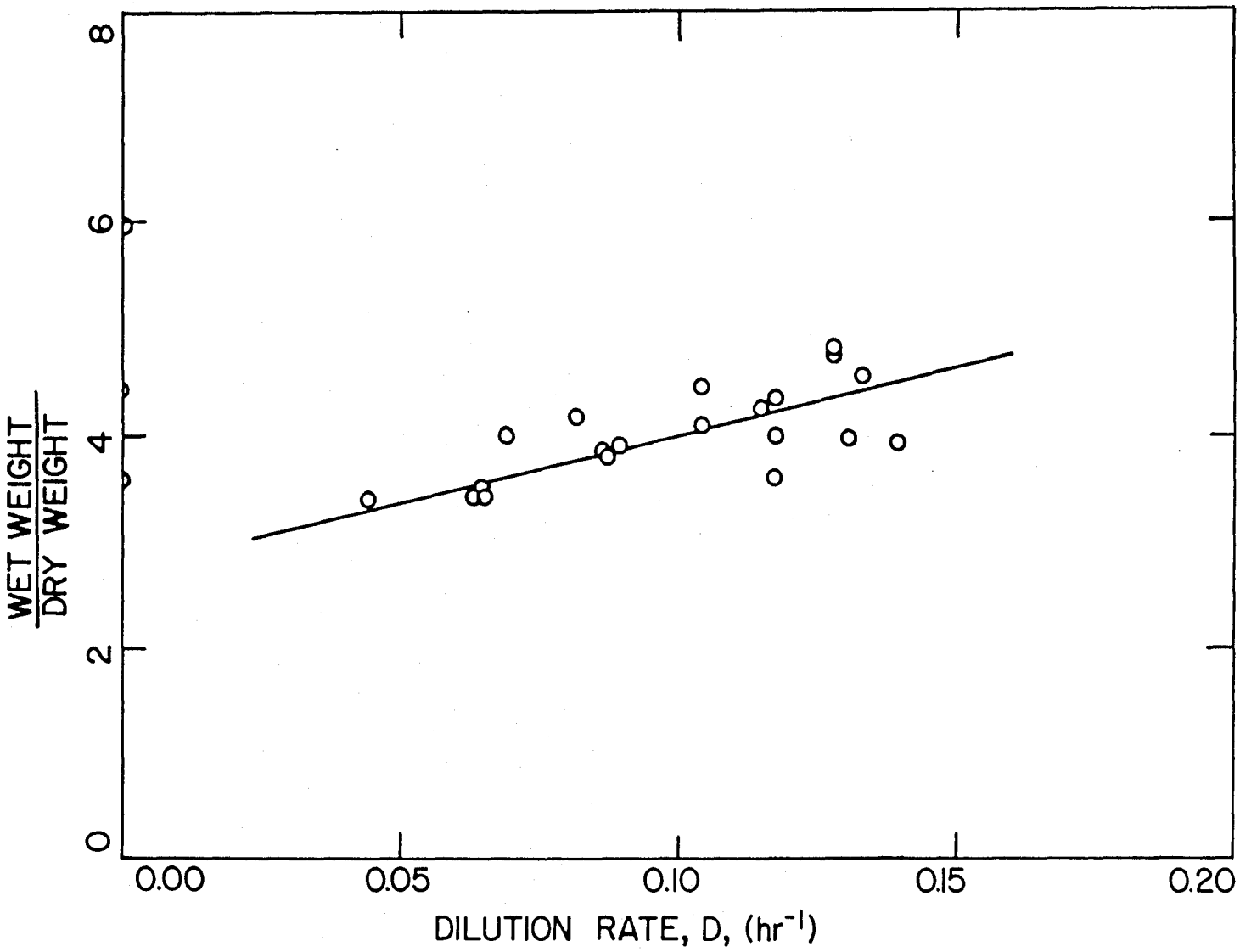


Figure 4.2.5. Ratio of wet weight to dry weight of the biomass material vs. dilution rate for cells grown in 5.0 g/l of ethanol in the feed. The variation is probably due to the water content of the cells.

shows that the function is a monotonically increasing one and that ethanol inhibition is not yet strong enough to invert the sign of the slope of the curve. Although not shown in Figure 4.2.7, when 10.0g/l of ethanol was used in the feed, a steady-state ethanol concentration of 5–6 g/l was achieved at a dilution rate of 0.175 hr^{-1}. Because only a limited number of experiments were performed with 10.0 g/l of ethanol and because most of the experiments of interest were far away from these conditions, these extreme points were not considered in the later analysis. As the ethanol concentration is increased from zero, the specific growth rises very sharply. Thus, a very small value of the Michaelis-Menton half-saturation constant is needed to describe the behavior in the region of low ethanol concentration. The specific growth rate continues to rise at a much slower pace as the ethanol concentration increases beyond 0.05 g/l. The experimental points in Figure 4.2.7 are scattered, and the μ versus s line can only be an approximation.

From the measured biomass and ethanol concentrations, the substrate to biomass yield coefficients as defined in Equations (4.1.4) and (4.1.5) were calculated by setting the derivative in these equations to zero, *i.e.*,

$$Y_s = \frac{x}{s_f - s}. \tag{4.2.1}$$

The result of this calculation for the yeast culture growing in 5.0 g/l of ethanol is shown in Figure 4.2.8.

4.3 STEADY-STATE EXPERIMENTS ON GLUCOSE

By following the same procedure established in Figure 4.1.1, a series of steady-state runs were conducted with 5.0 g/l of glucose in the feed stream. The criterion on the decision of the attainment of a steady-state is similar to that for the steady-state experiments with ethanol as the feed, except that the time during which the biomass, glucose, and ethanol concentrations are required to remain constant is increased from 5–10 residence times to 10–15 residence times.

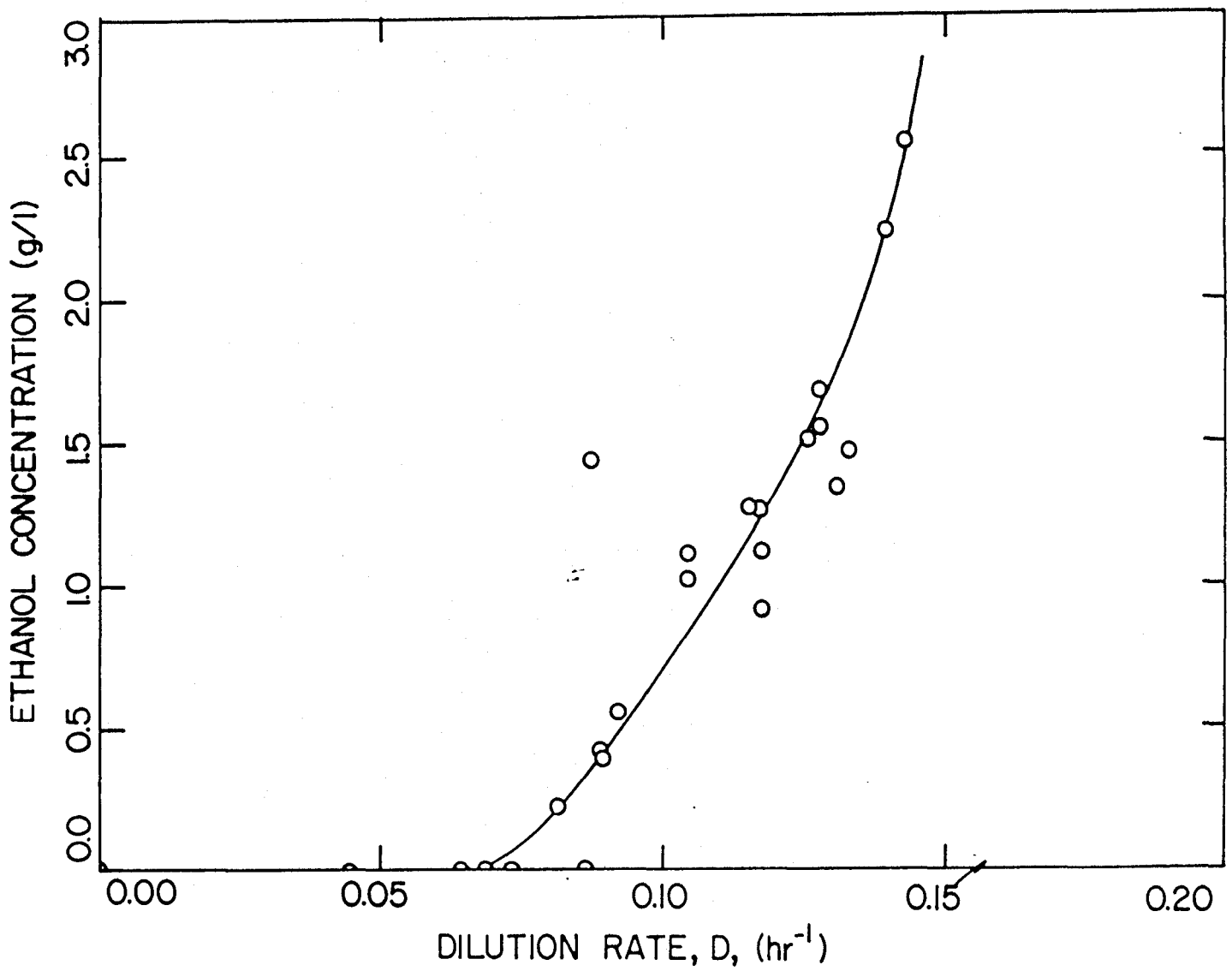


Figure 4.2.6. Changes in the steady-state ethanol (limiting substrate) concentration as a function of dilution rate for cells grown in 5.0 g/l of ethanol in the feed.

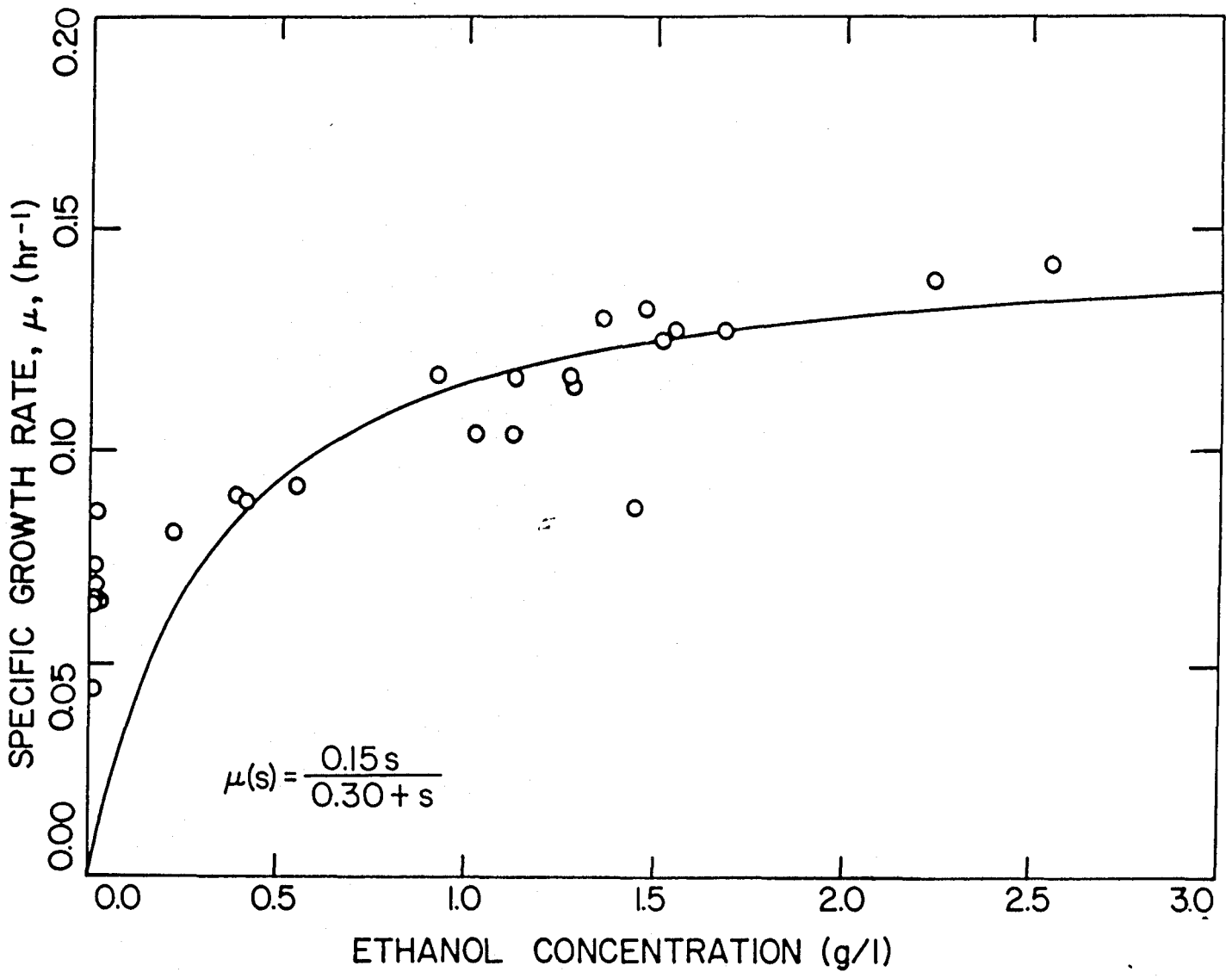


Figure 4.2.7. Intrinsic specific growth rate as a function of the ethanol (limiting substrate) concentration.

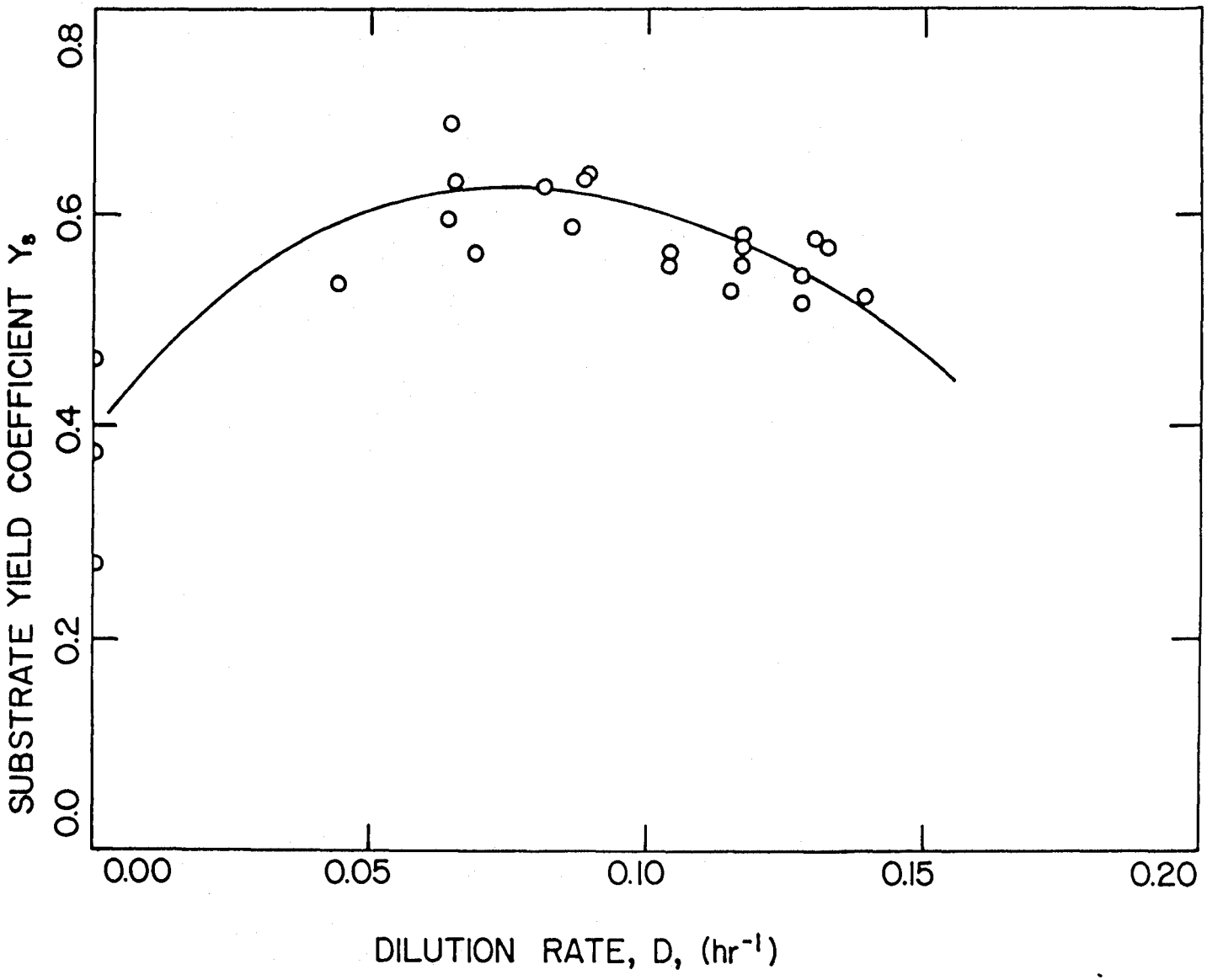


Figure 4.2.8. Dependence of the steady-state substrate to cell biomass yield coefficient on dilution rate for cells grown in 5.0 g/l of ethanol in the feed.

As in the ethanol steady-state experiments, the dilution rate was either progressively increased or decreased and the resulting dry weight as a function of the dilution rate is shown in Figure 4.3.1. The wet weight and optical density are also displayed in Figure 4.3.2 and 4.3.3, respectively. Similar to the findings in the ethanol feed experiments, the ratio of the dry weight to the optical density is a quite strong function of the dilution rate; this fact has been demonstrated in Figure 3.5.2. The ratio of the dry cell weight to the wet cell weight is shown in Figure 4.3.4, which again suggests that the ratio varies as a well correlated function of the specific growth rate. Actually, the correlation may be better if an improvement is made on the method used to measure the wet weight so that the water content between the microbial cells is eliminated in a more regulated manner.

The steady-state glucose concentration is plotted against the operating dilution rate in Figure 4.3.5. It remained low for the entire range of the dilution rates, and the insert in this figure shows the same plot in the expanded scale in the glucose concentration. The curve is rotated and replotted in Figure 4.3.6. The same graph plotted in an expanded scale in Figure 4.3.7 shows that the growth stopped slightly before the glucose concentration was reduced to zero. Because of the low value of the steady-state glucose concentration and because of the scatter in the data points, the parameters for the μ versus s line, especially the half-saturation constant if a Monod model is to be fitted, could not be accurately determined.

The substrate to biomass yield coefficients for this set of steady-state experiments are shown in Figure 4.3.8. Furthermore, ethanol is produced in large quantities at dilution rates above 0.30–0.35 hr^{-1}, although its production is relatively insignificant (about 0.005 g/l out of 5.0 g/l of glucose feed) at lower dilution rates. This fact is shown in Figure 4.3.9, magnified in Figure 4.3.10. This sudden rise in the ethanol production coincides with the sudden rise in the glucose concentration

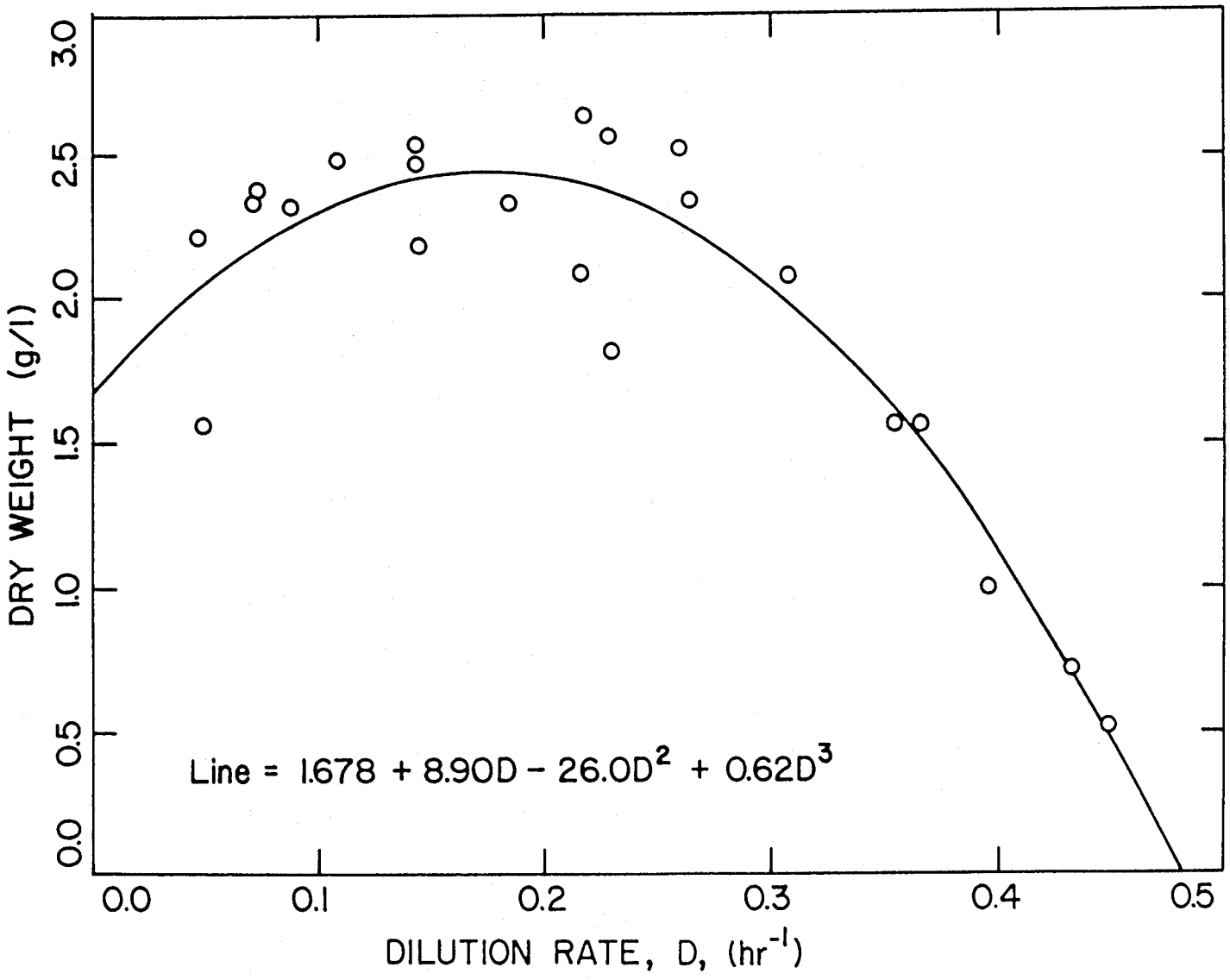


Figure 4.3.1. Changes in the steady-state biomass concentration (dry weight) as a function of dilution rate for cells grown in 5.0 g/l of glucose in the feed.

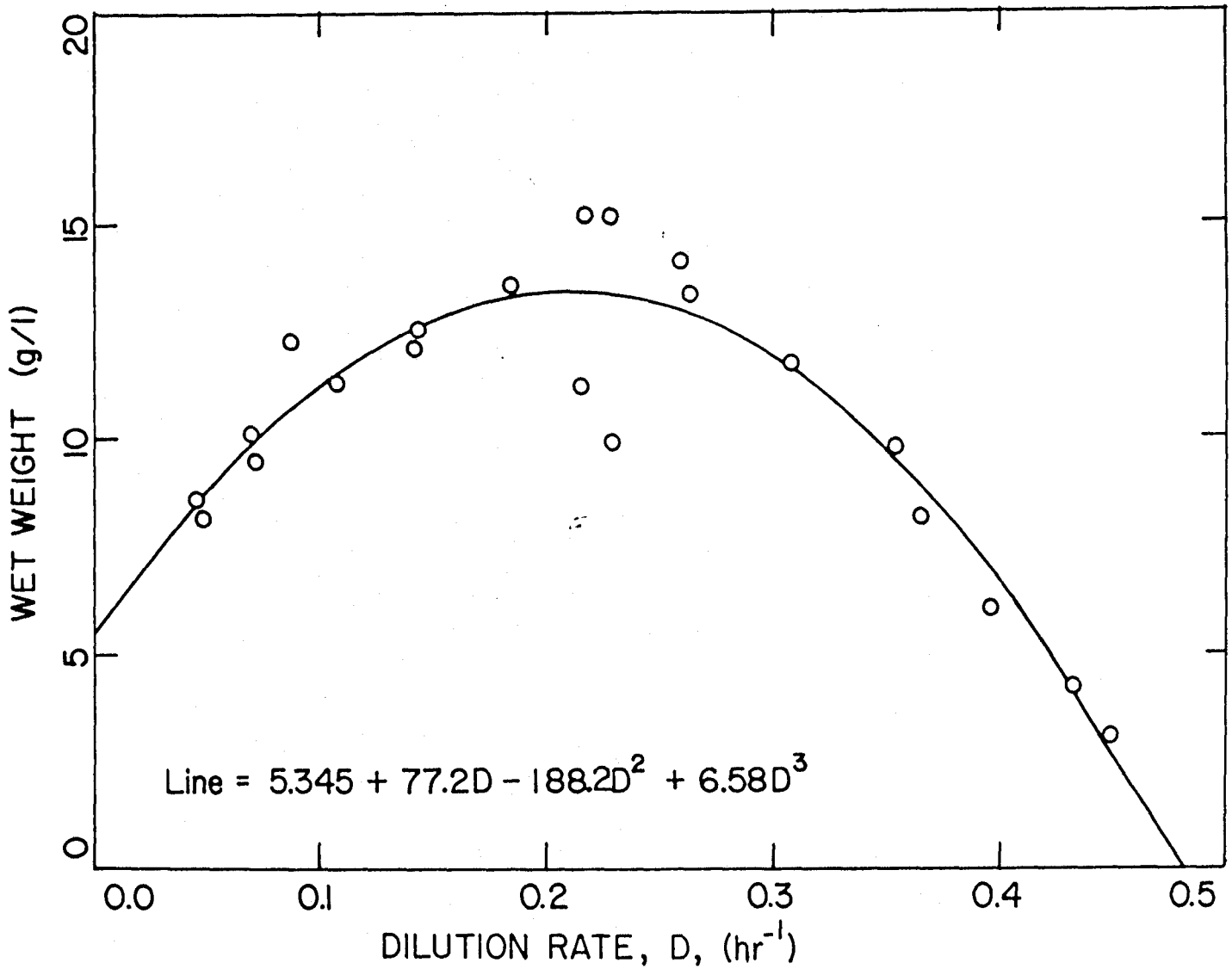


Figure 4.3.2. Changes in the steady-state wet cell biomass weight as a function of dilution rate for cells grown in 5.0 g/l of glucose in the feed.

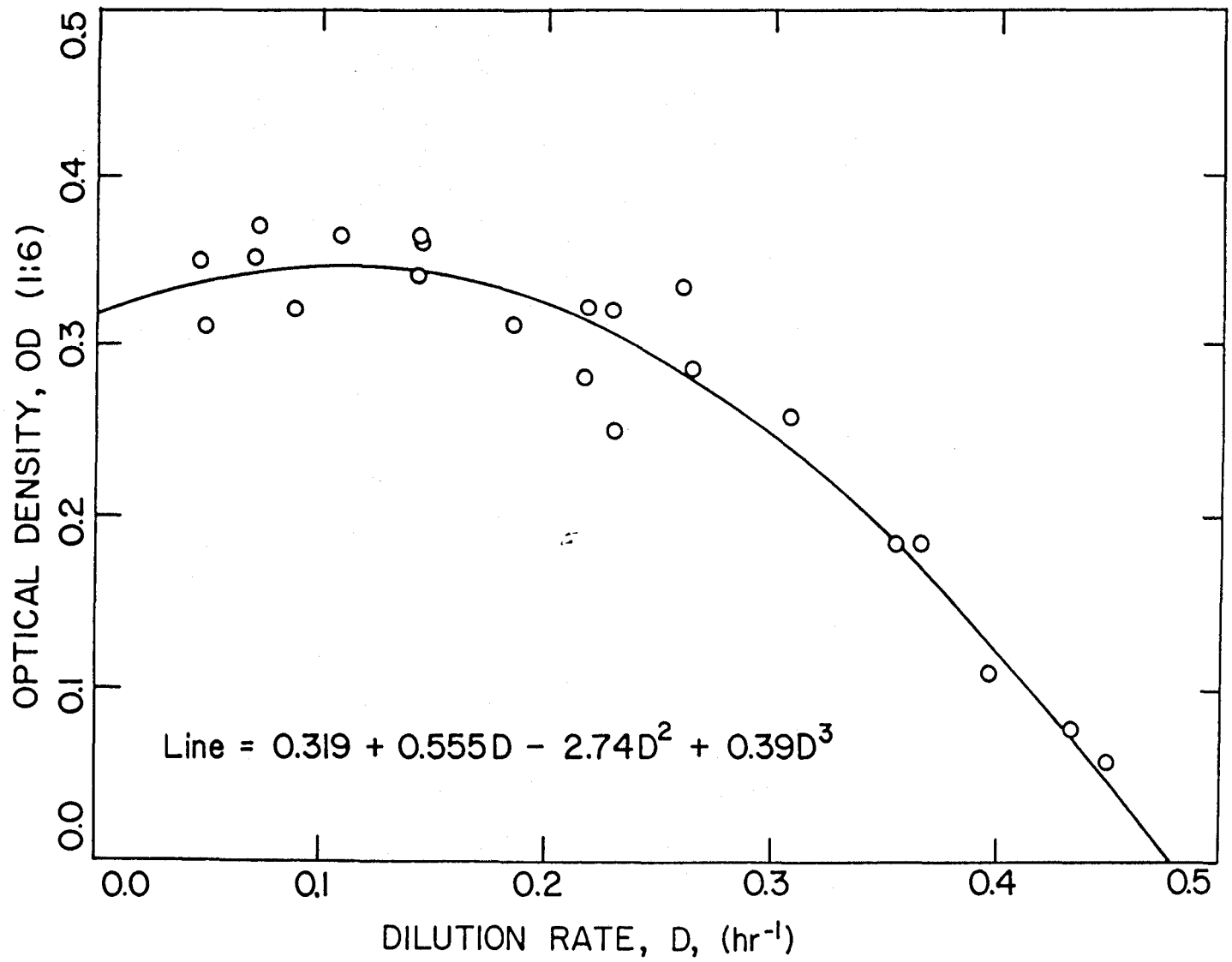


Figure 4.3.3. Changes in the optical density of the steady-state sample diluted 6 times as a function of dilution rate for cells grown in 5.0 g/l of glucose in the feed.

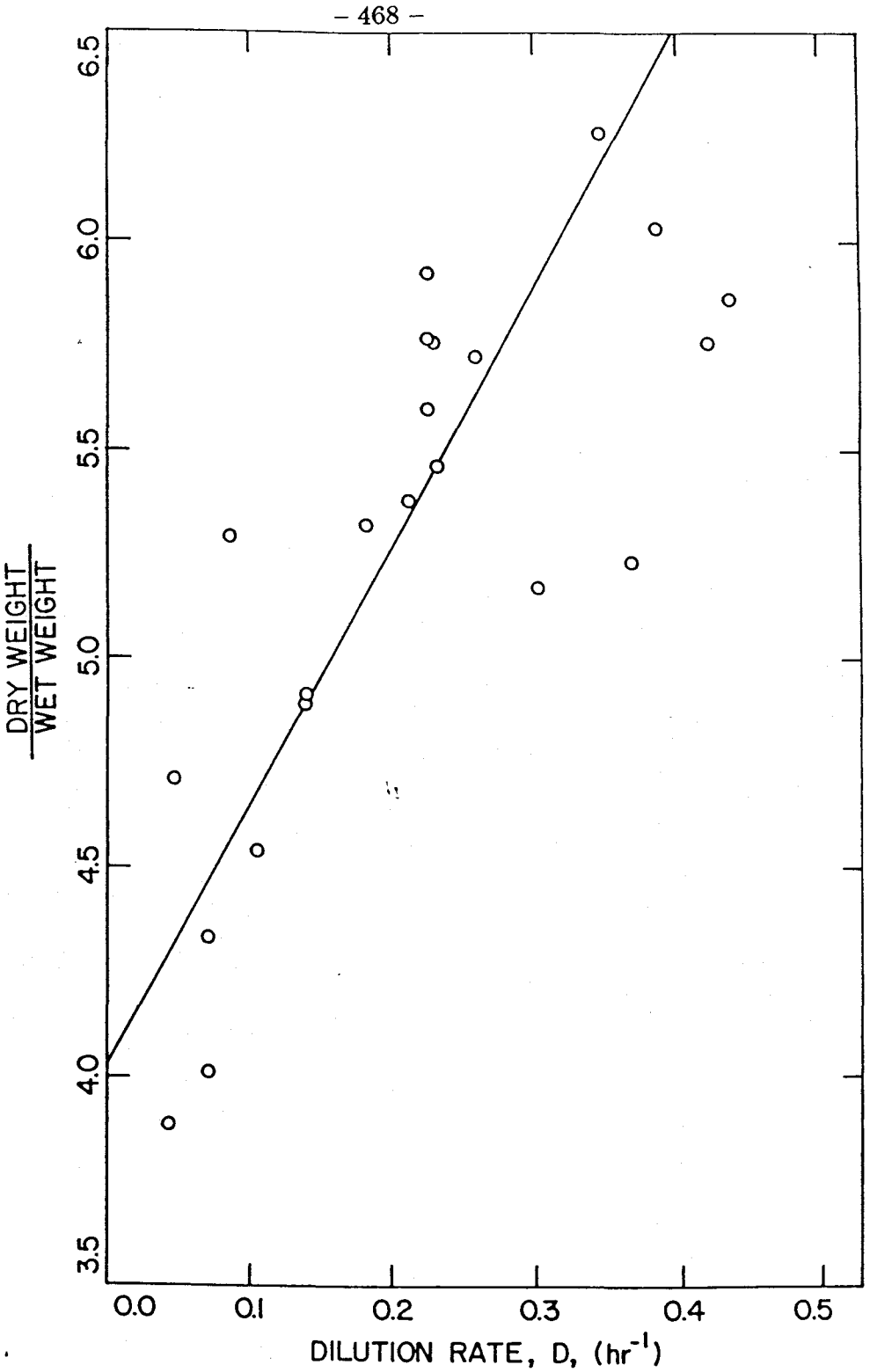


Figure 4.3.4. Ratio of wet weight to dry weight of the biomass material vs. dilution rate for cells grown in 5.0 g/l of glucose in the feed. The variation is probably due to the water content of the cells.

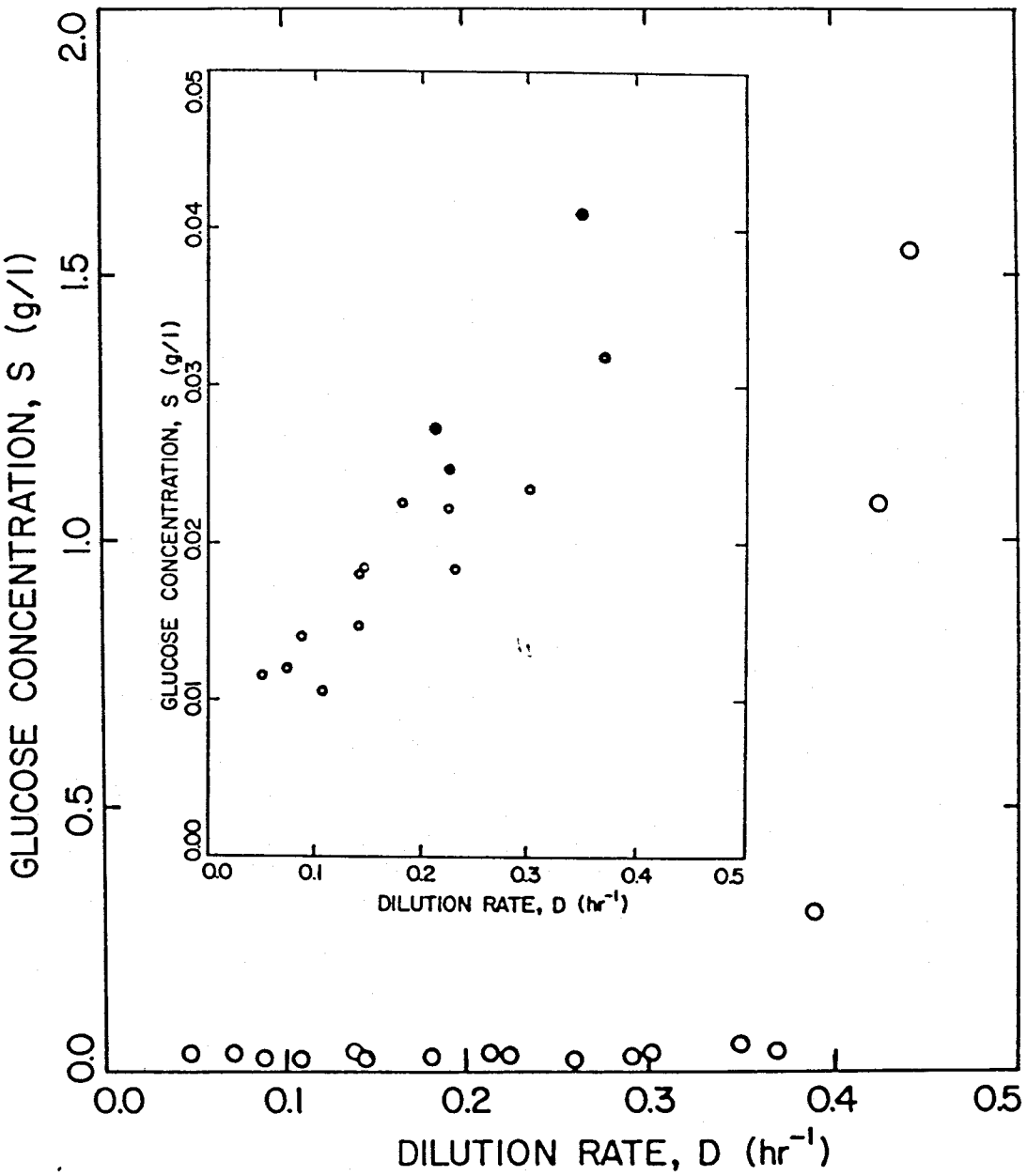


Figure 4.3.5. Changes in the steady-state glucose (limiting substrate) concentration as a function of dilution rate for cells grown in 5.0 g/l of glucose in the feed.

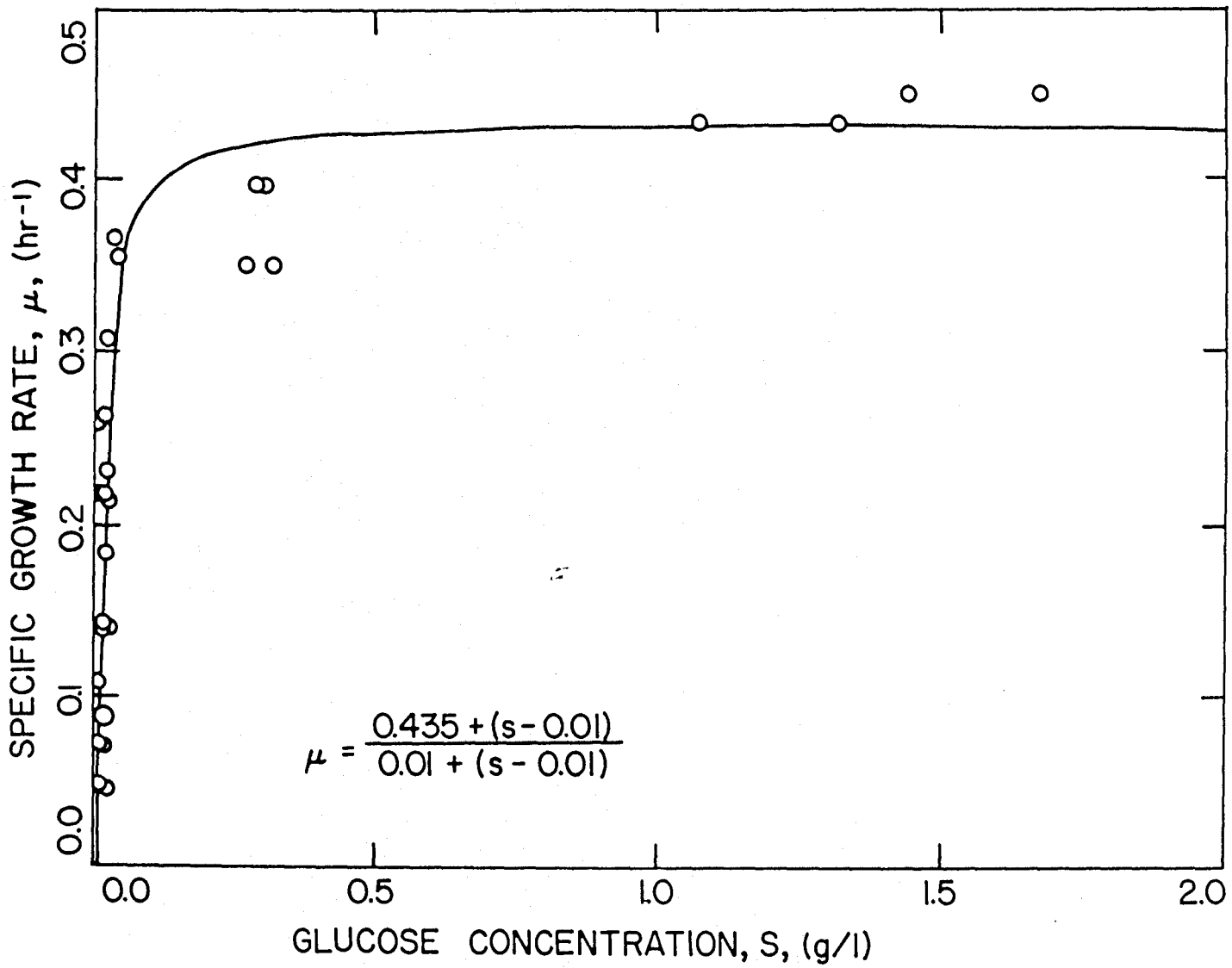


Figure 4.3.6. Intrinsic specific growth rate as a function of the glucose (limiting substrate) concentration. The same graph is plotted in an expanded scale in the glucose concentration in Figure 4.3.7.

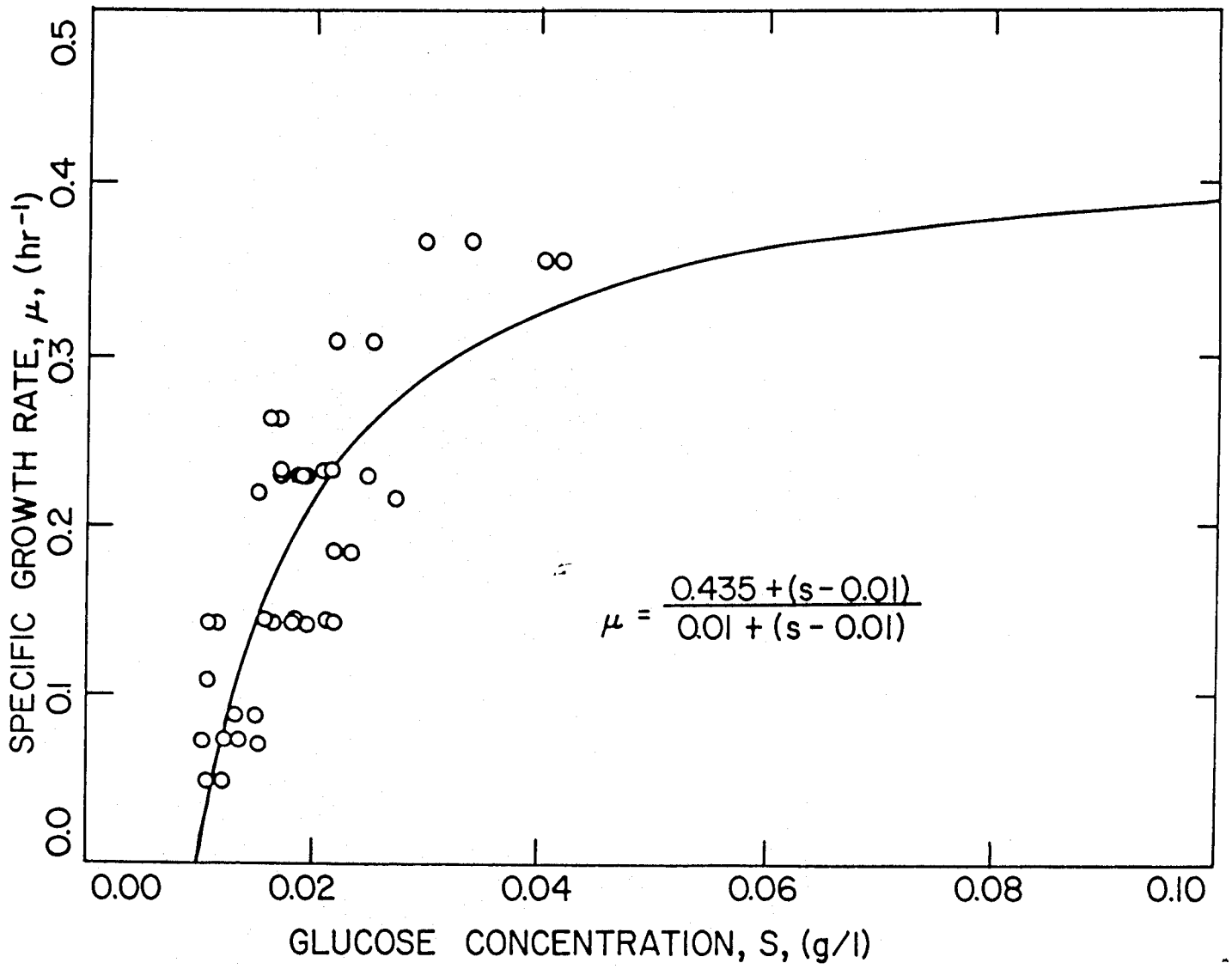


Figure 4.3.7. Intrinsic specific growth rate as a function of the glucose (limiting substrate) concentration.

with an increasing dilution rate. This observation corresponds to the well known Crabtree effect, which states that fermentative growth accompanied by ethanol production is possible even with adequate aeration in an environment with high glucose concentrations. For the baker's yeast under study, the Crabtree effect is dominant at a glucose concentration of approximately 0.04 g/l.

4.4 TRANSIENT EXPERIMENTS – BATCH

The results from one of the batch experiments on 5 g/l of glucose are shown in Figures 2.1.1 to 2.1.6. The data obtained from another batch glucose experiment are shown in Figures 4.4.1 to 4.4.9. Throughout the remainder of this thesis, the noisy solid lines are reserved for the directly measured or derived on-line continuous values. The circles represent the values determined in separate off-line samples by the methods described in the chapter on materials and methods. The yield coefficients based on glucose and ethanol are the direct result of applying macroscopic and elemental material balances on the fermentor.

The results of one of the many batch growth experiments of *S. cerevisiae* conducted on 5 g/l of ethanol are also described in Figures 4.4.10 to 4.4.17.

4.5 TRANSIENT EXPERIMENTS – CONTINUOUS

The determination of the kernel was carried out for an experiment in which the dilution rate was shifted up from 0.100 hr^{-1} to 0.288 hr^{-1}. As before, *S. cerevisiae* was grown and maintained continuously with a constant feed of 5 g/l of glucose. The resulting kernel is shown in Figure 4.5.1. The response of the glucose concentration in this experiment was close to an impulse; consequently, the intrinsic specific growth rate was nearly an impulse as well. Thus, it was able to derive the kernel function without resorting to the base function expansion in this case. The shape of the kernel function suggests that it has both a 0th-order element and a first-order element.

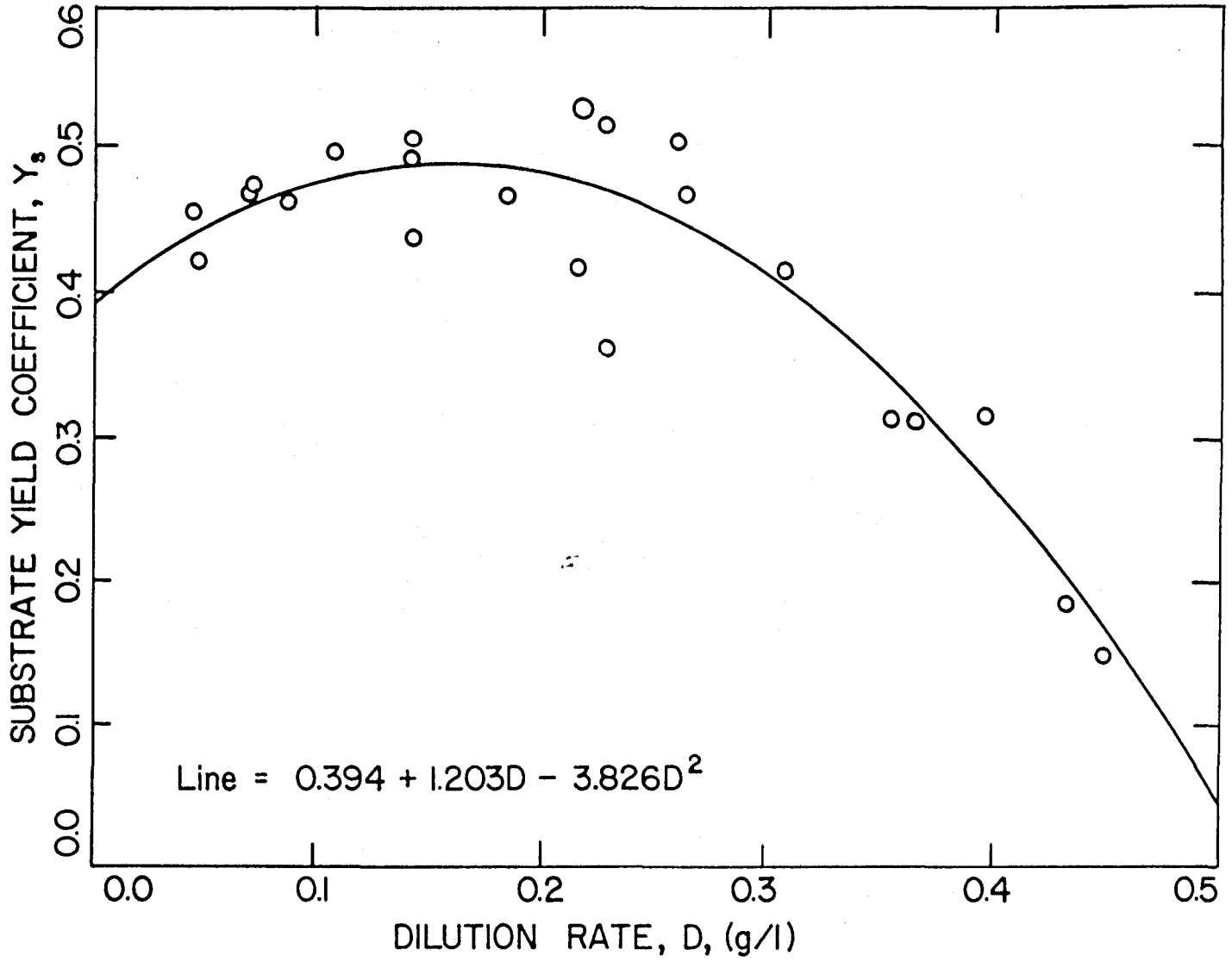


Figure 4.3.8a. Dependence of the steady-state substrate (glucose) to cell biomass yield coefficient on dilution rate for cells grown in 5.0 g/l of glucose in the feed.

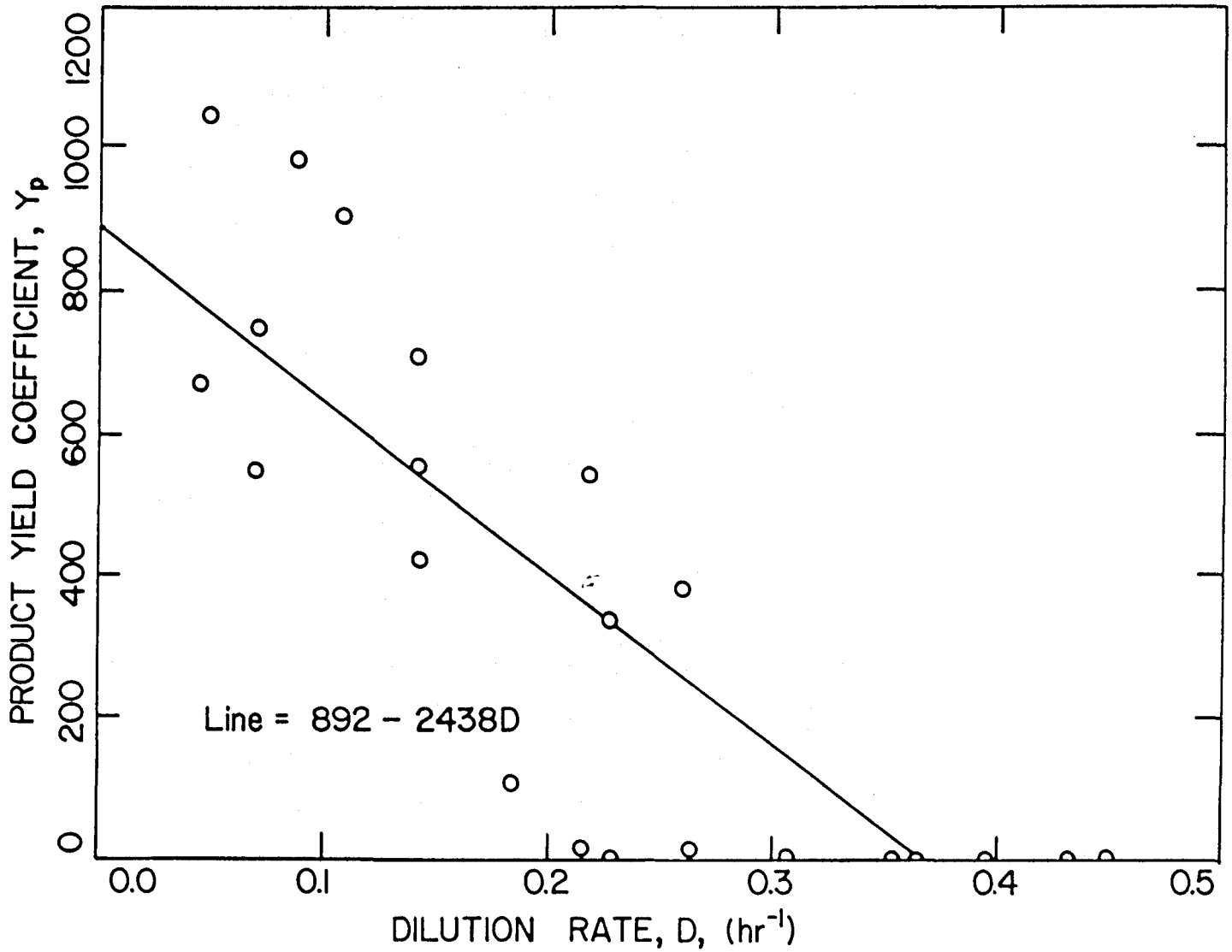


Figure 4.3.8b. Dependence of the steady-state product (ethanol) to cell biomass yield coefficient on dilution rate for cells grown in 5.0 g/l of glucose in the feed.

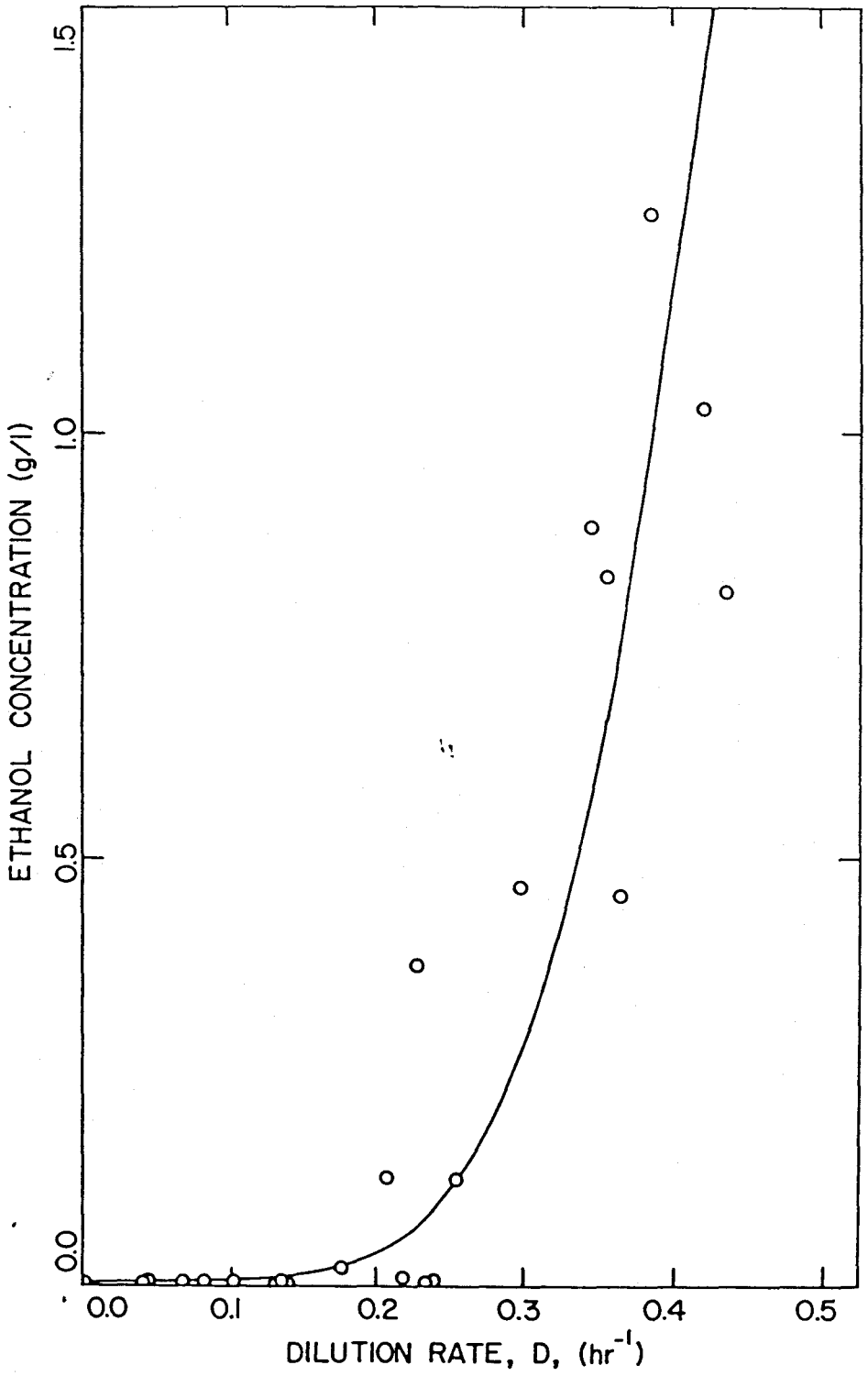


Figure 4.3.9. Changes in the steady-state ethanol concentration as a function of dilution rate for cells grown in 5.0 g/l of glucose in the feed. The same graph is plotted in an expanded scale in the glucose concentration in Figure 4.3.10.

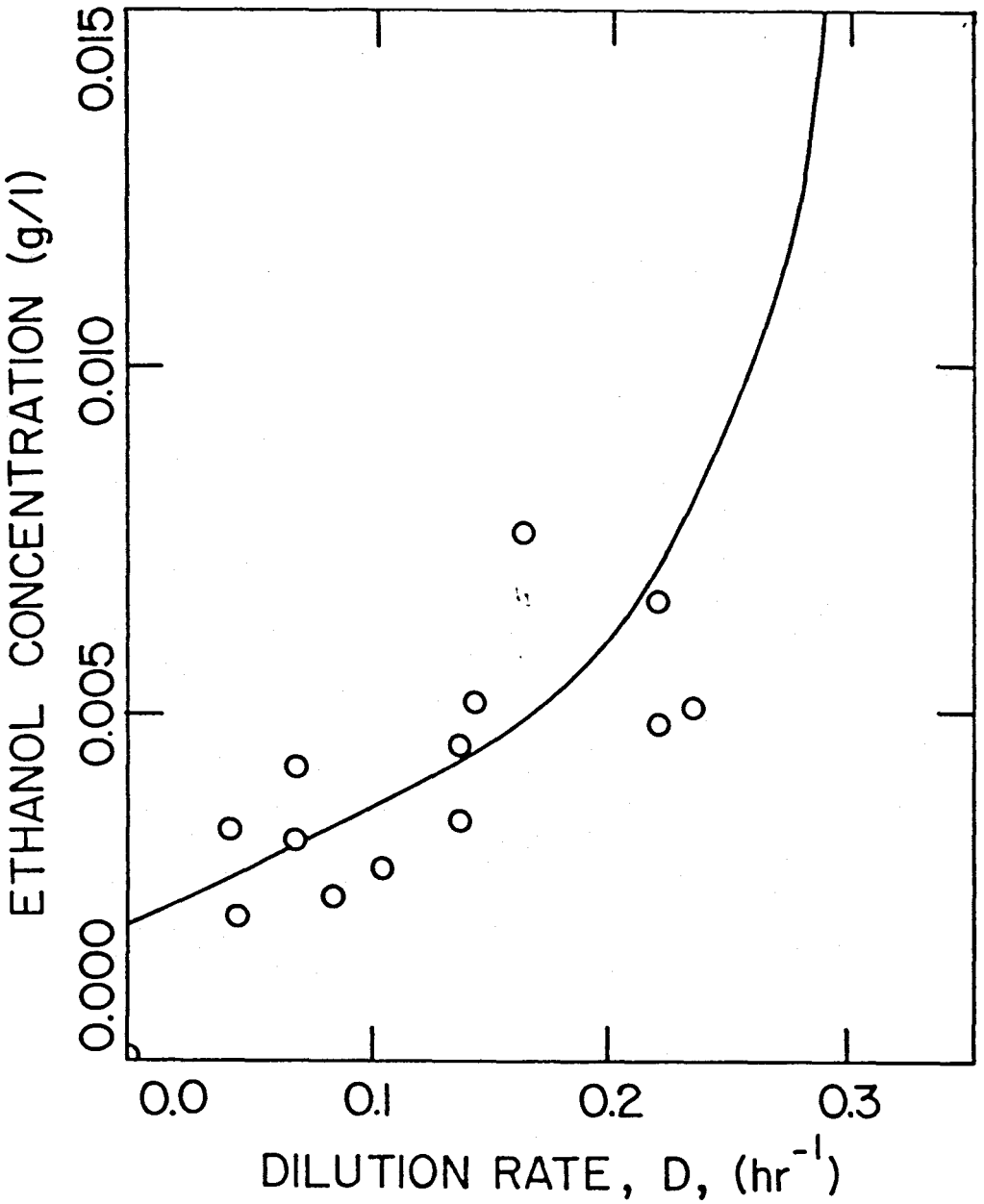


Figure 4.3.10. Changes in the steady-state ethanol concentration as a function of dilution rate for cells grown in 5.0 g/l of glucose in the feed.

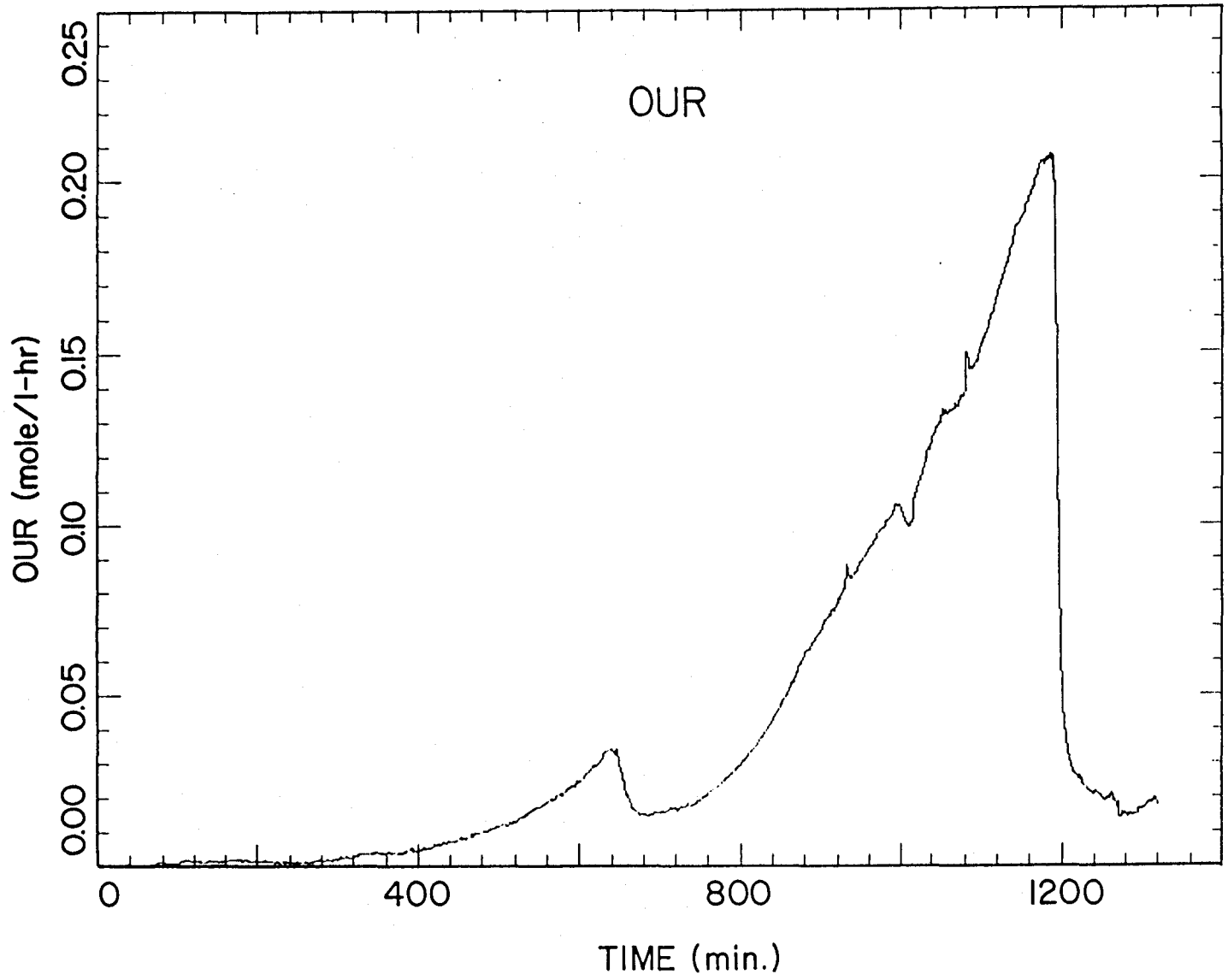


Figure 4.4.1. Oxygen uptake rate (OUR) as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. Note that there are two distinct regions of high growth activities.

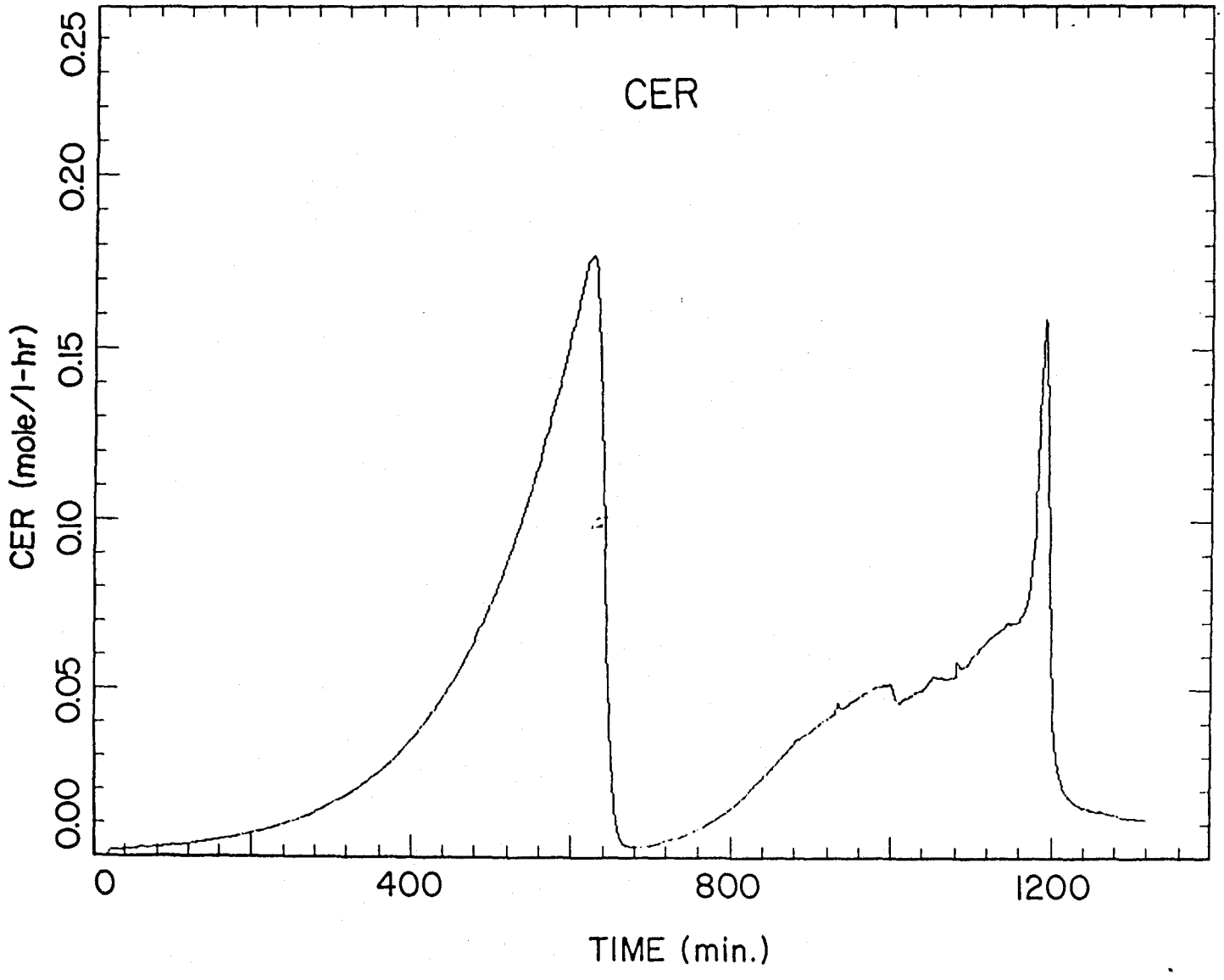


Figure 4.4.2. Carbon dioxide evolution rate (CER) as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose.

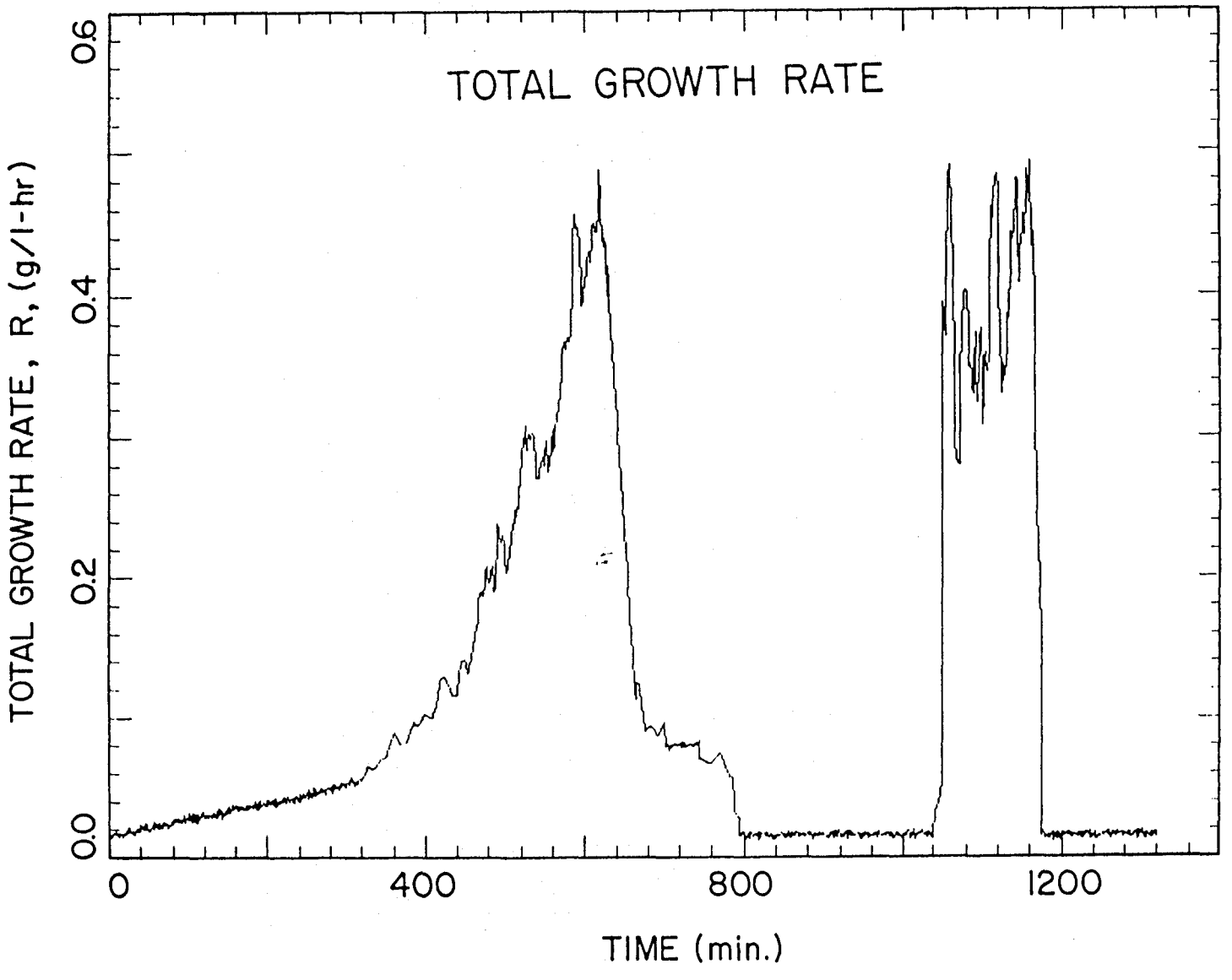


Figure 4.4.3. Total growth rate as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The total growth rate is calculated from the rate of base addition needed to maintain a constant pH in the fermentor.

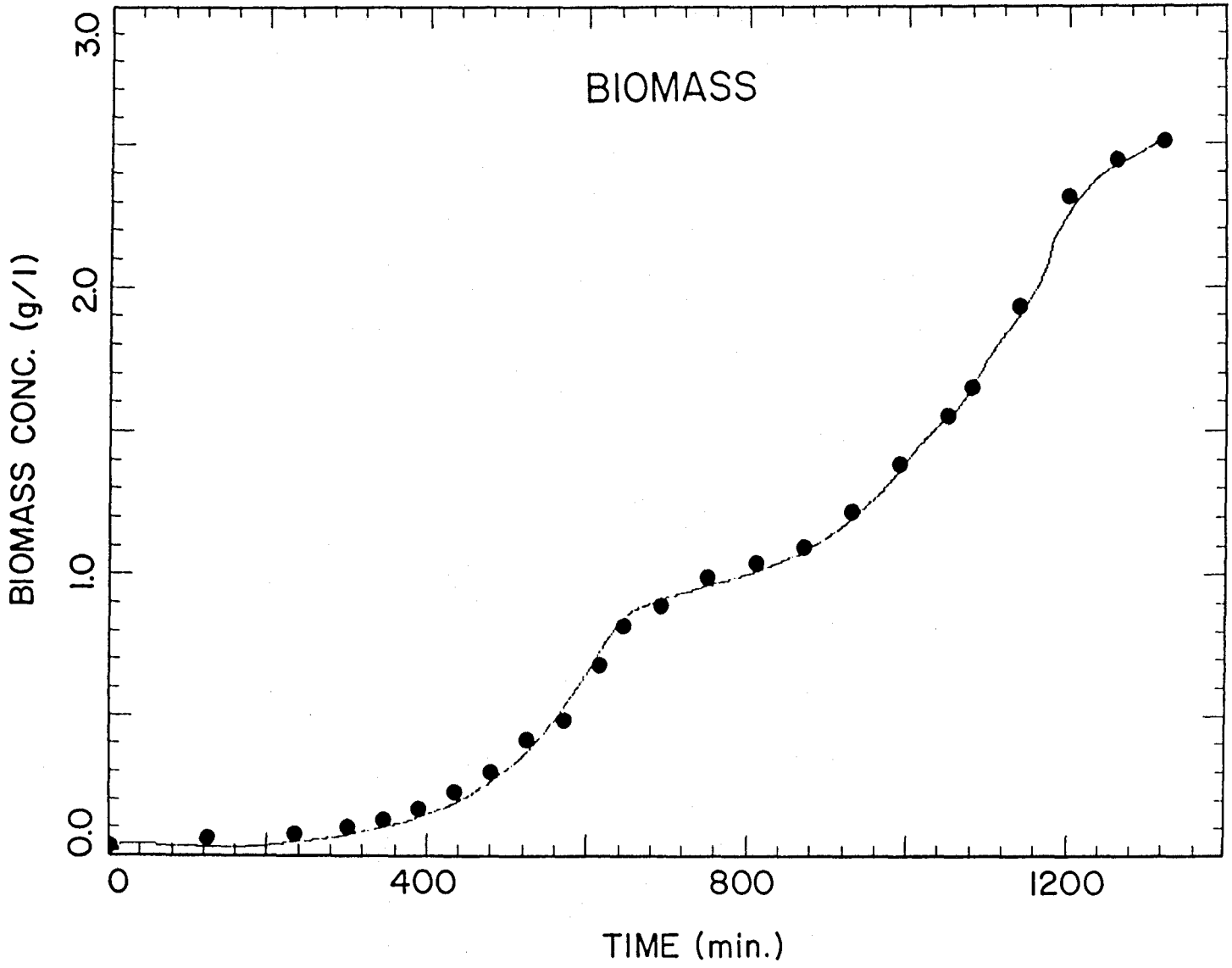


Figure 4.4.4. Cell biomass concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

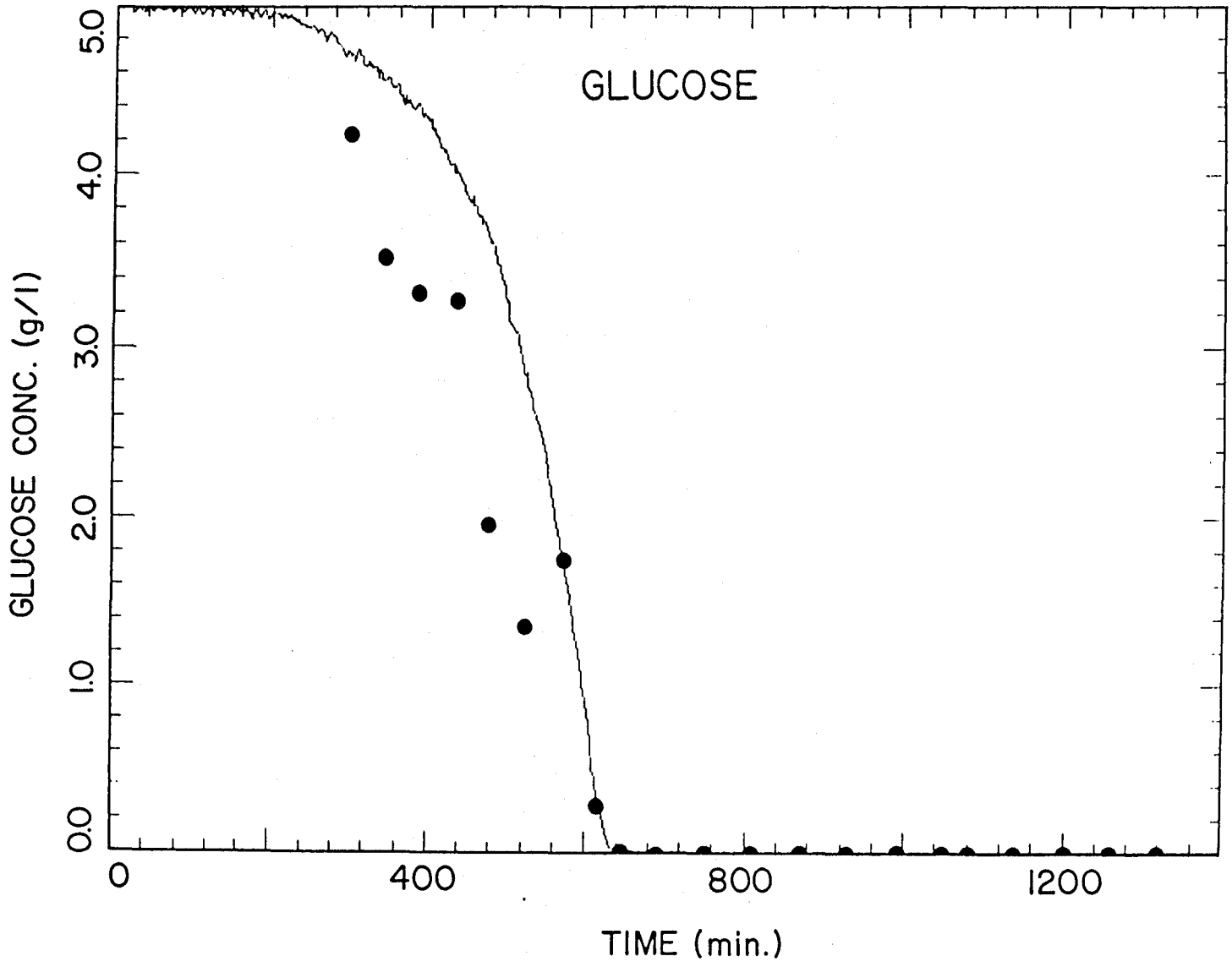


Figure 4.4.5. Glucose concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

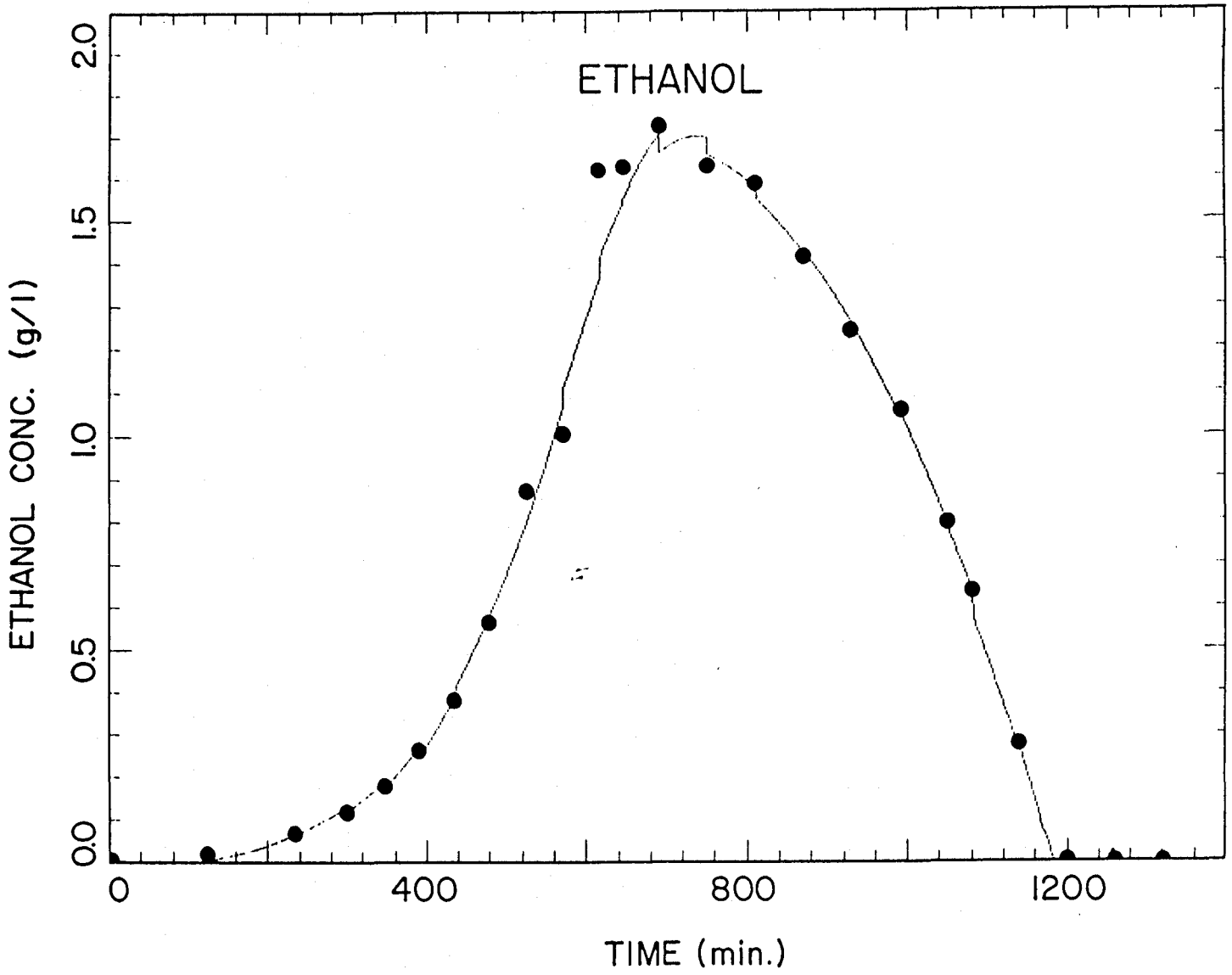


Figure 4.4.6. Ethanol concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

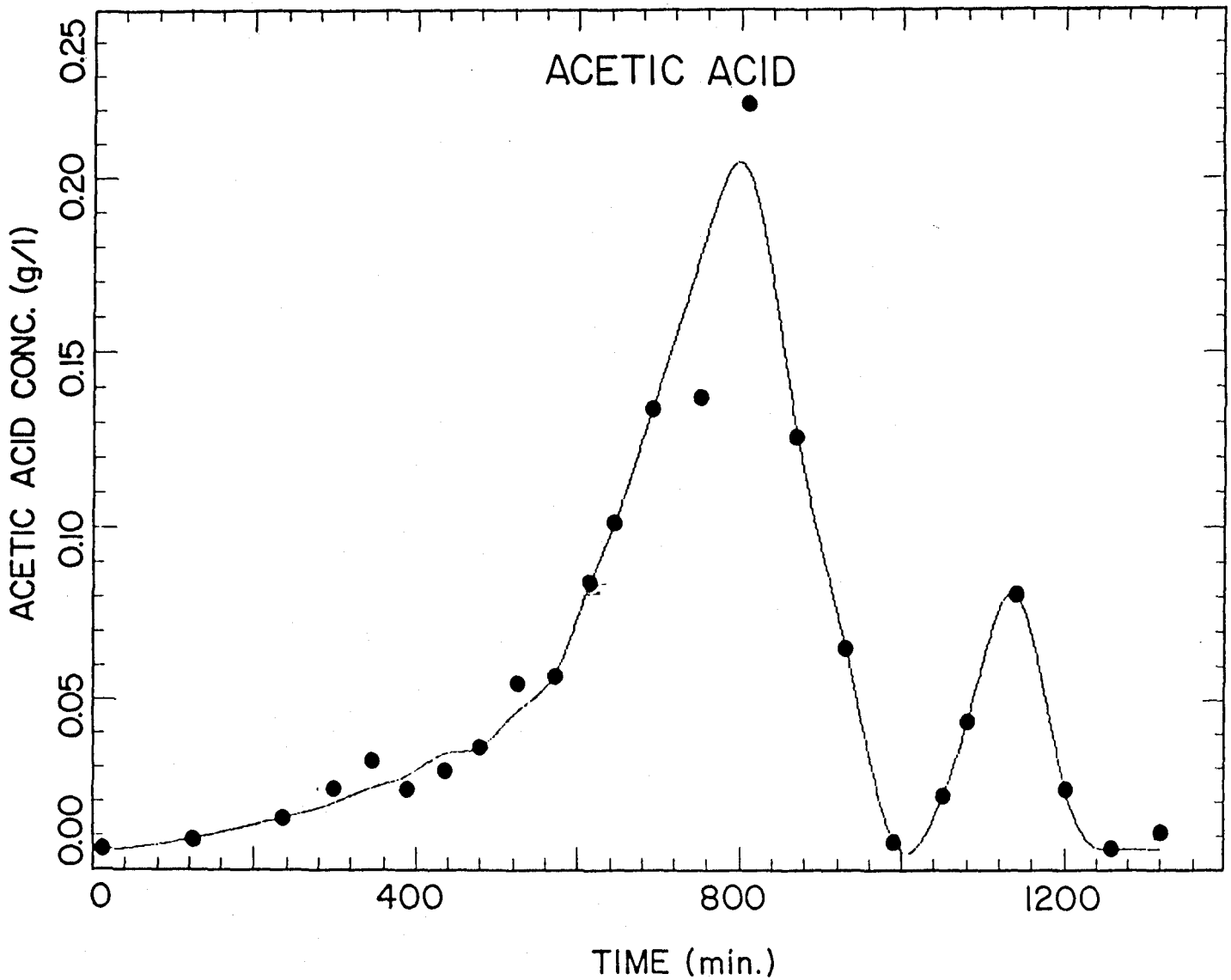


Figure 4.4.7. Acetic acid concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The solid line represents the cubic spline fit, and the open circles indicate the off-line analysis on a gas chromatograph.

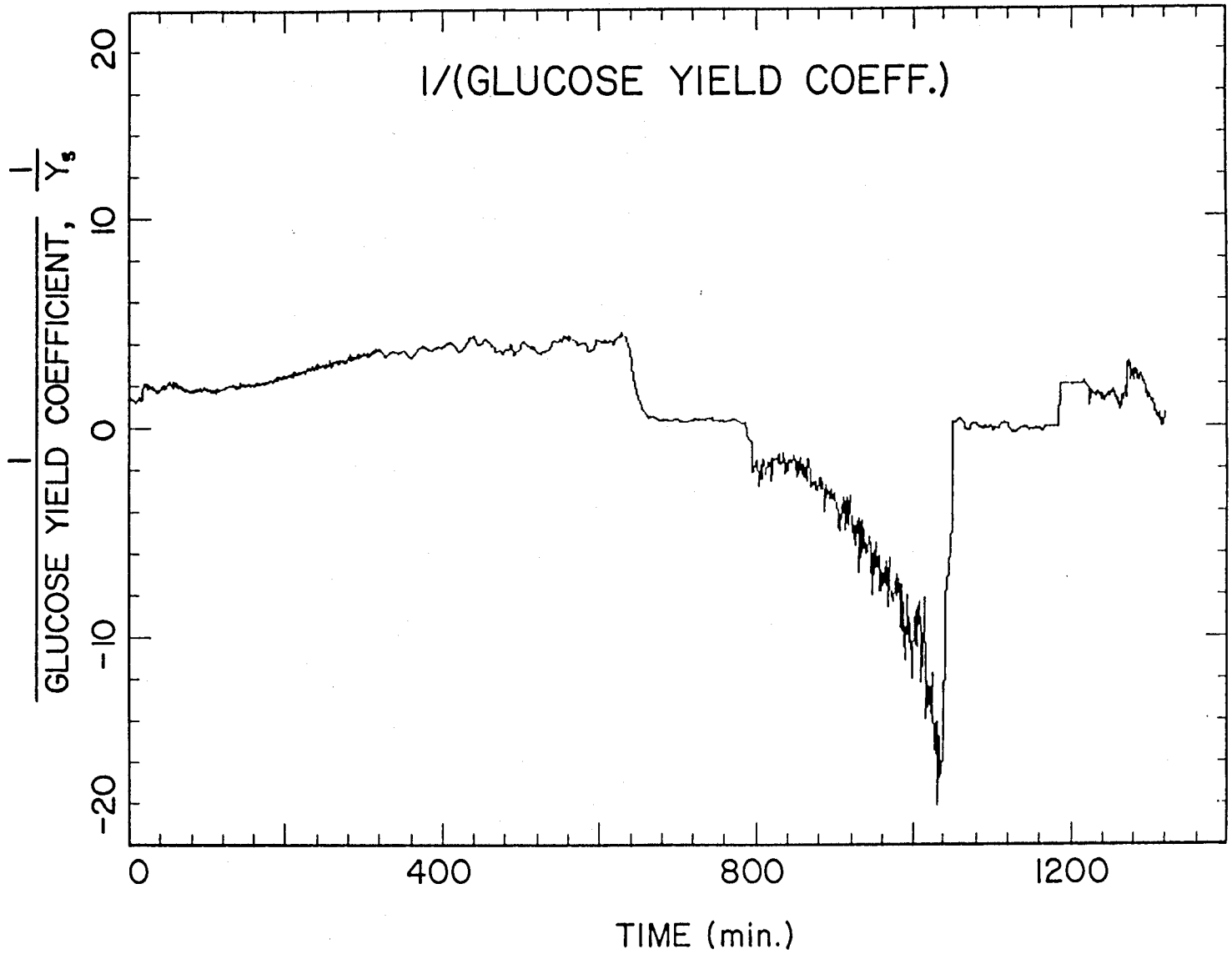


Figure 4.4.8. Substrate (glucose) to cell biomass yield coefficient as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The yield coefficient is calculated with the aid of macroscopic material and elemental balance.

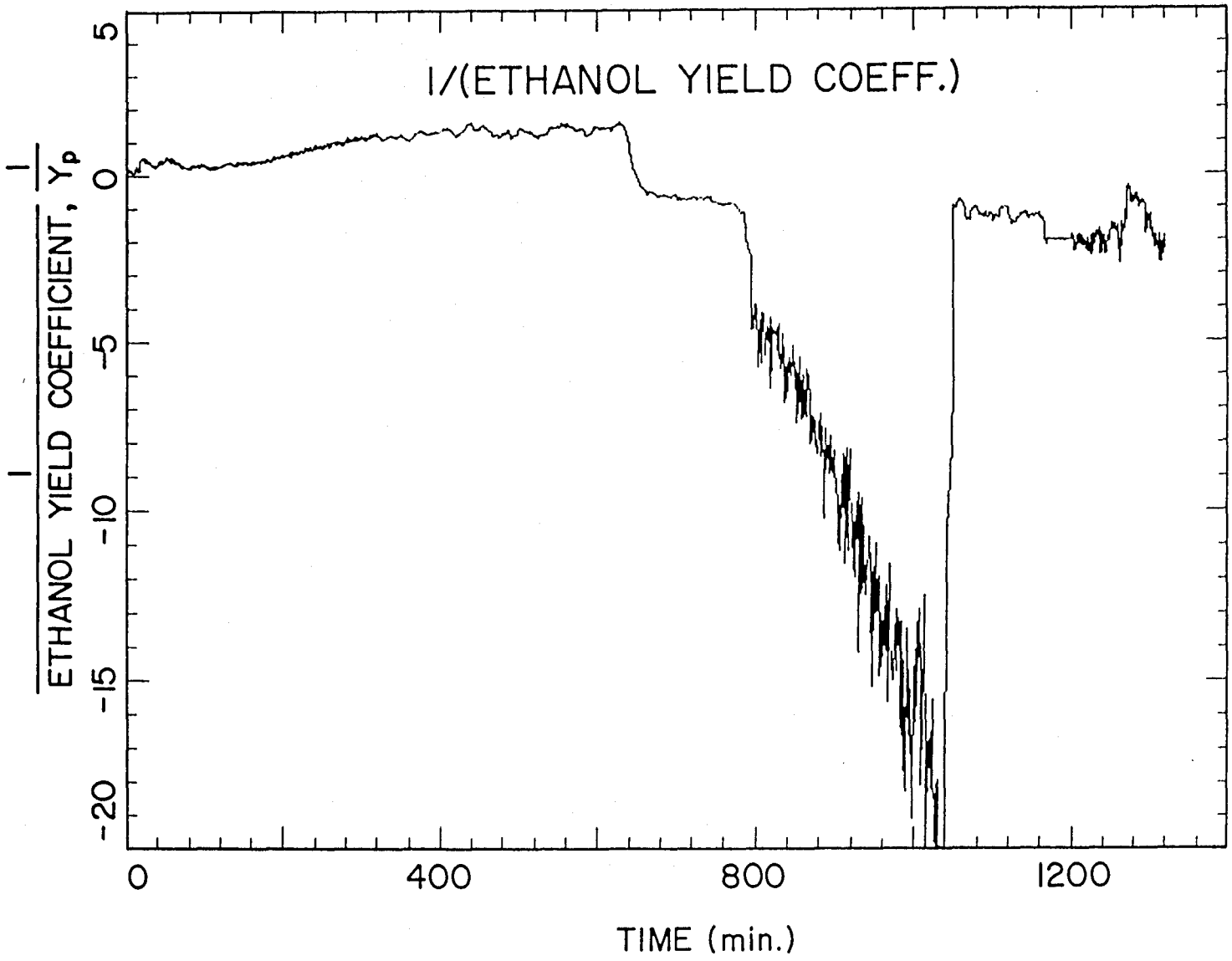


Figure 4.4.9. Product (ethanol) to cell biomass yield coefficient as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The yield coefficient is calculated with the aid of macroscopic material and elemental balance.

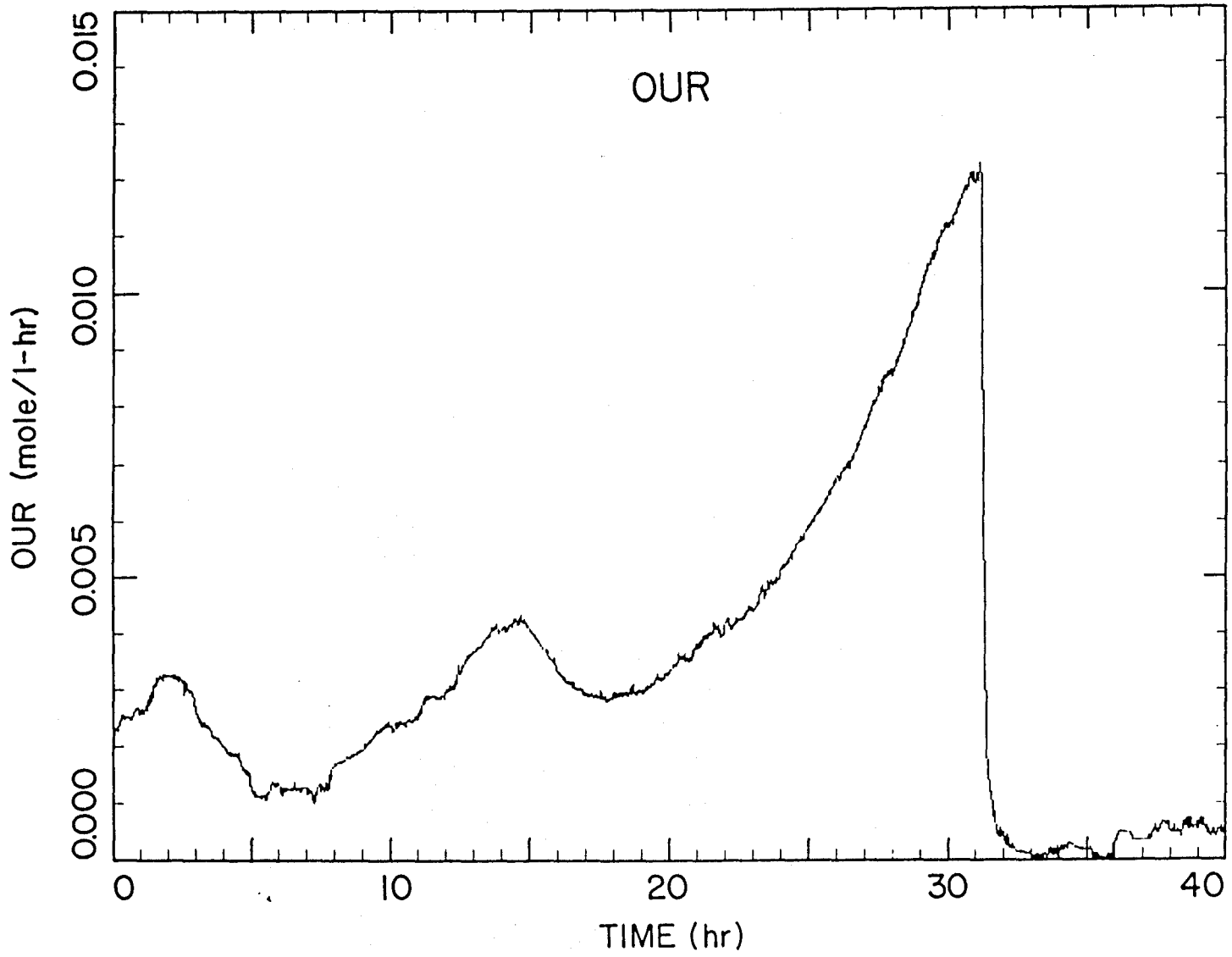


Figure 4.4.10. Oxygen uptake rate (OUR) as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol.

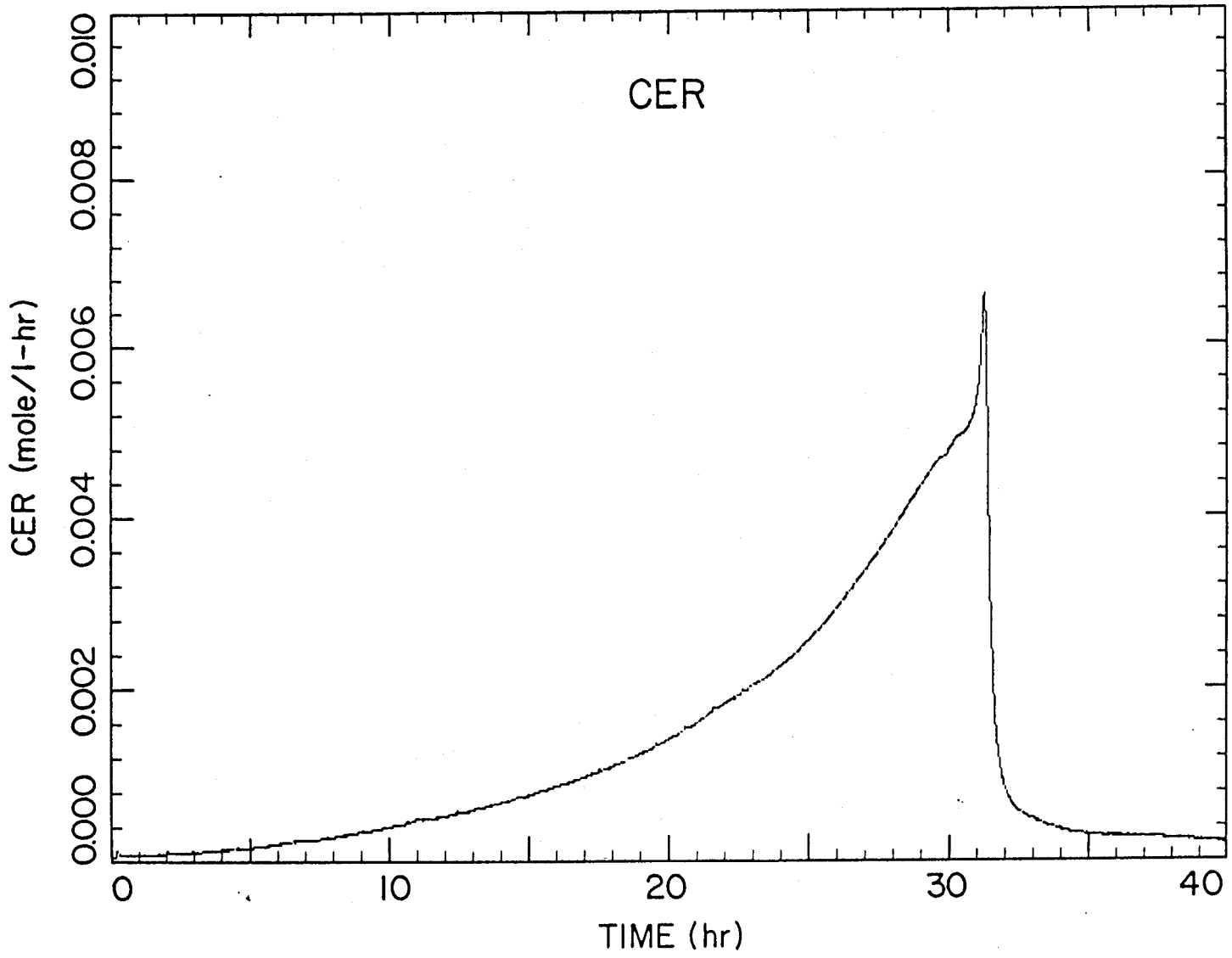


Figure 4.4.11. Carbon dioxide evolution rate (CER) as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol.

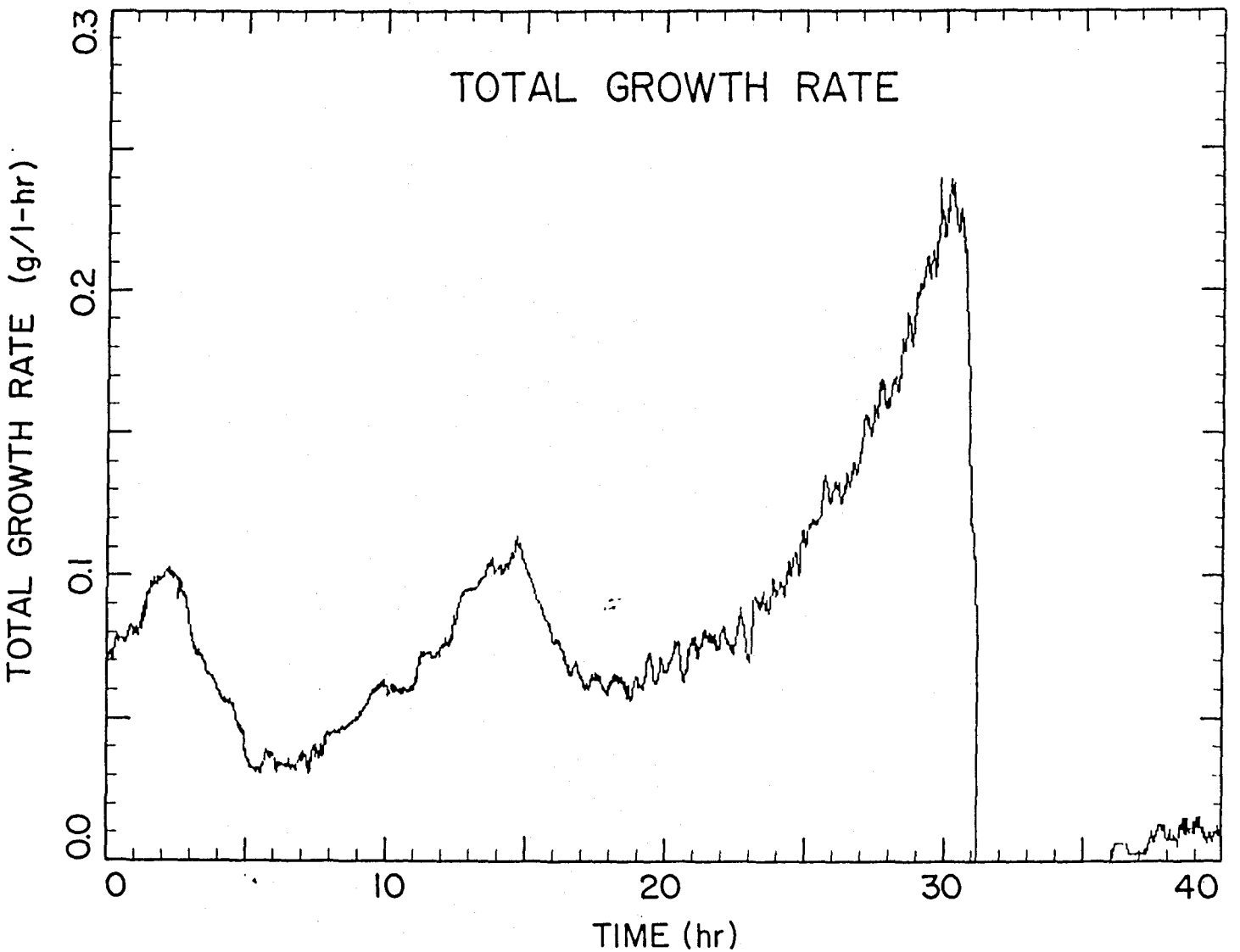


Figure 4.4.12. Total growth rate as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol. The total growth rate is calculated from the rate of base addition needed to maintain a constant pH in the fermentor.

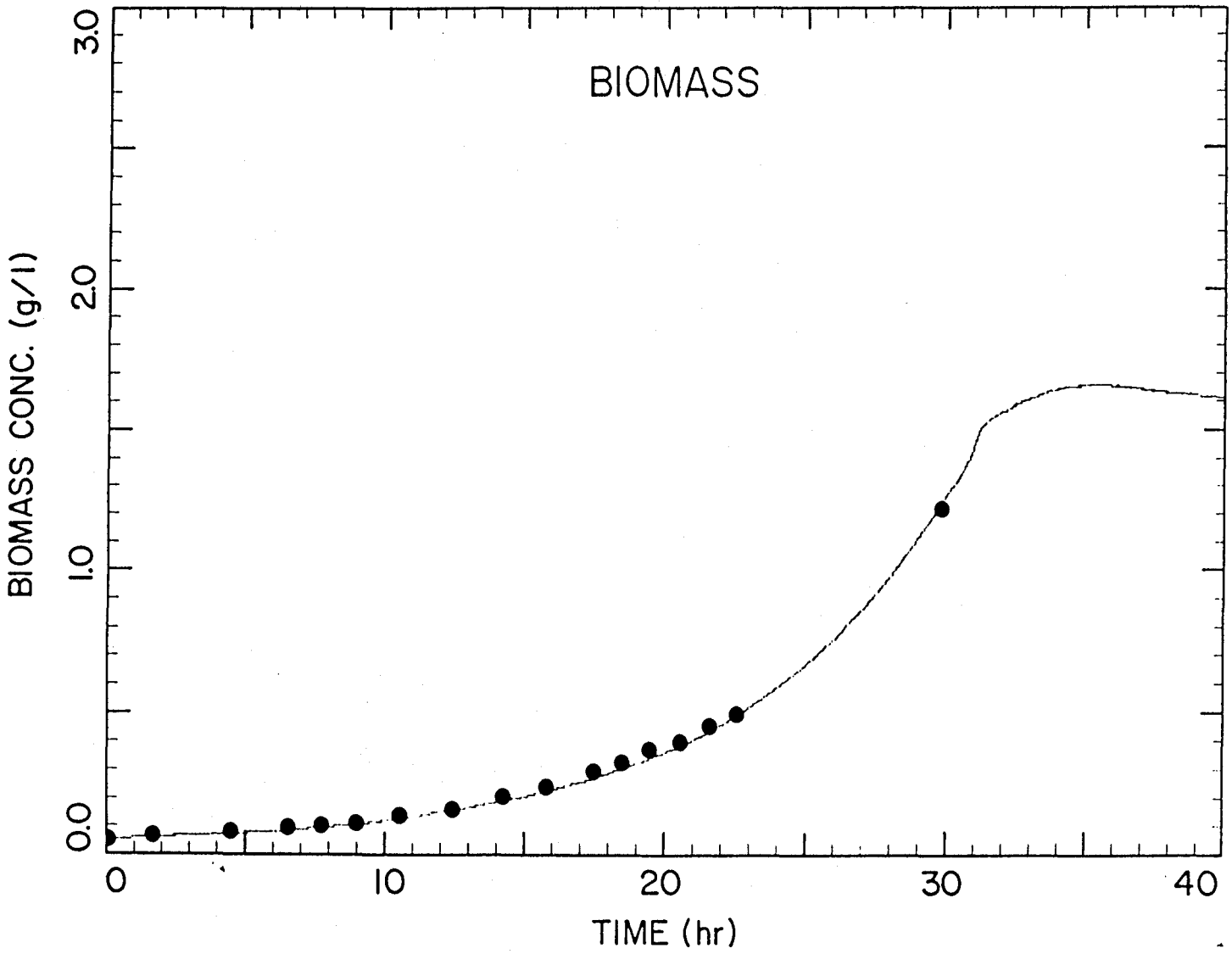


Figure 4.4.13. Cell biomass concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

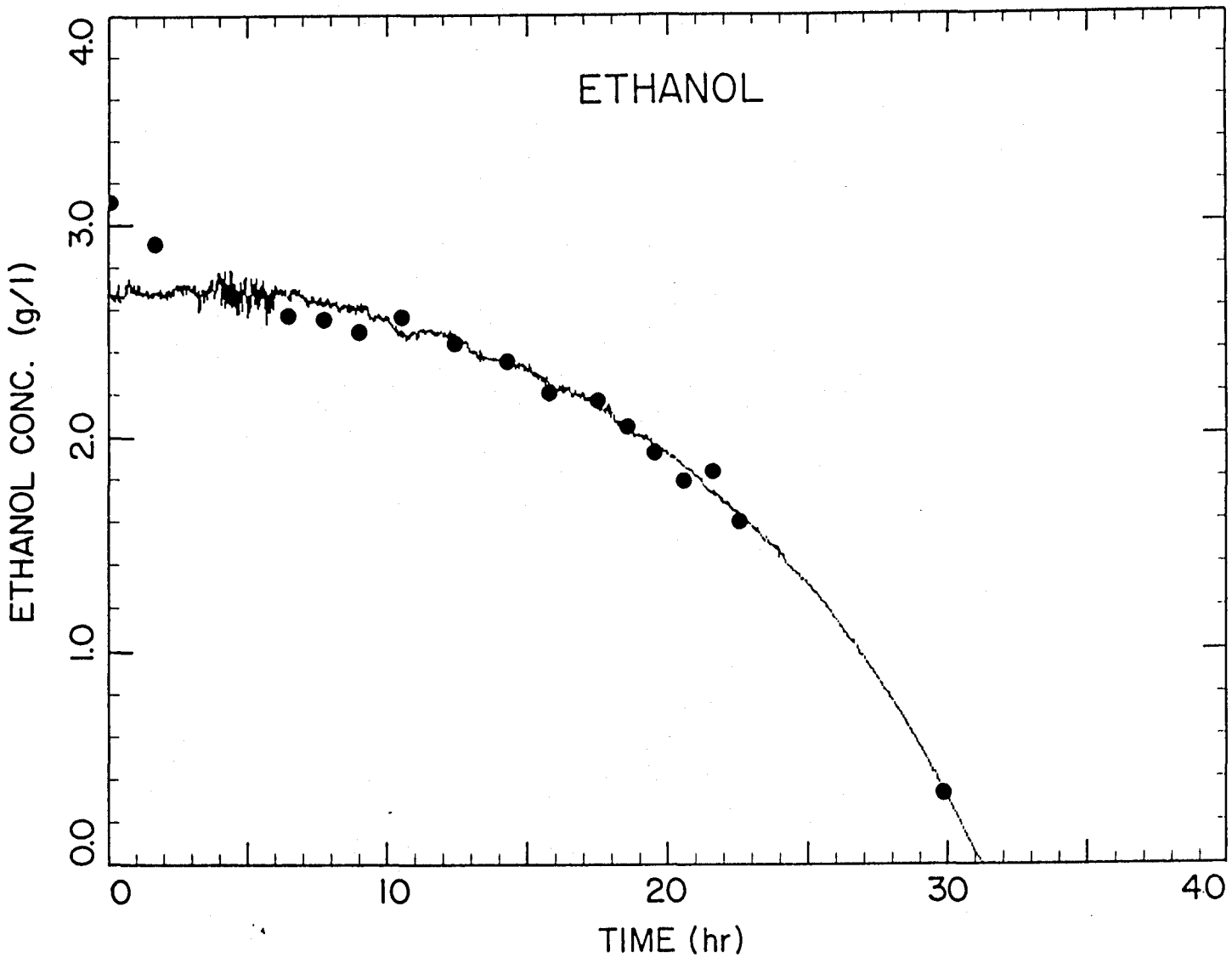


Figure 4.4.14. Ethanol concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol. The solid line represents the on-line measurement, and the open circles indicate the off-line measurements.

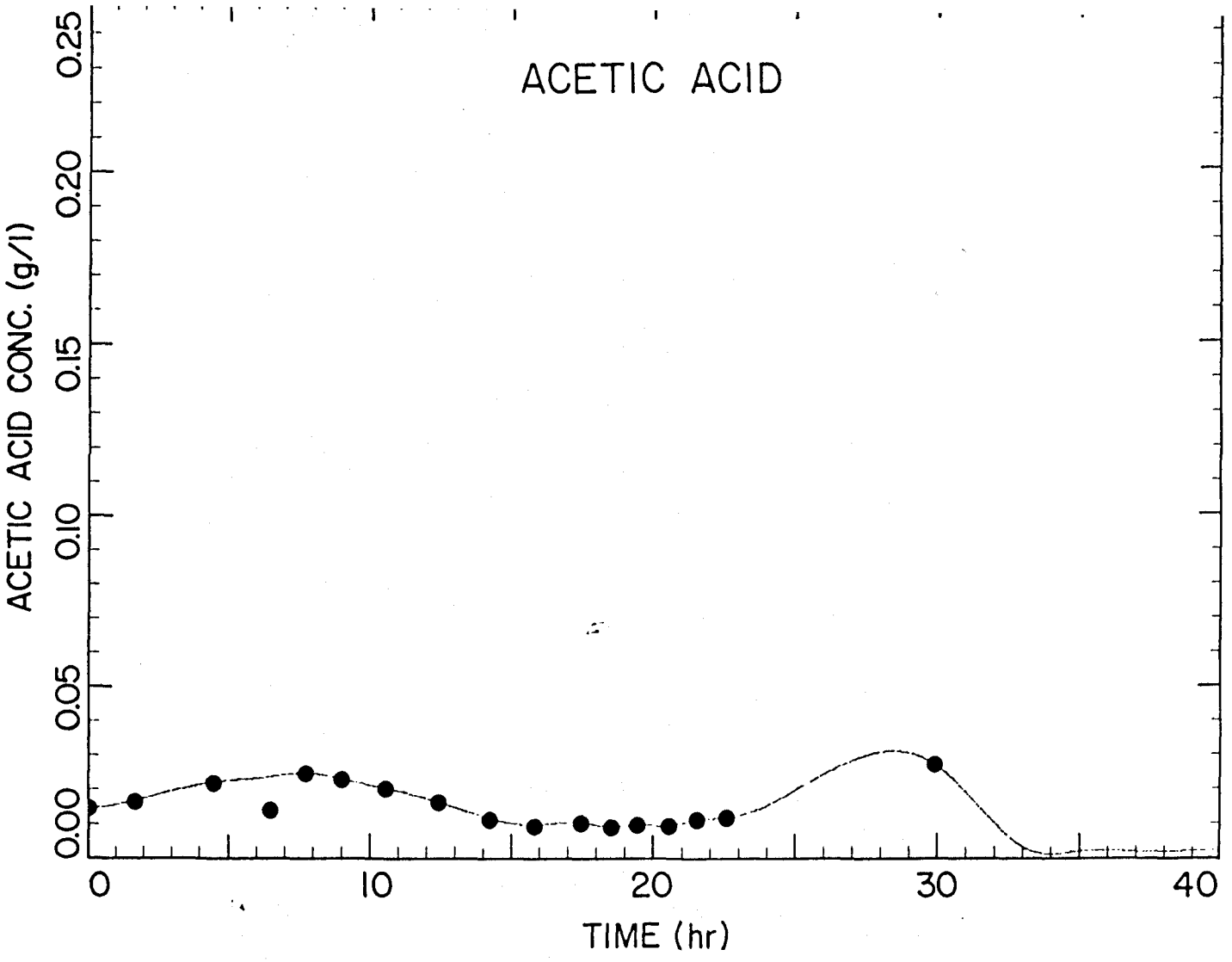


Figure 4.4.15. Acetic acid concentration as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol. The solid line represents the cubic spline fit, and the open circles indicate the off-line analysis on a gas chromatograph. The acetic acid concentration remained low for the entire duration of the fermentation.

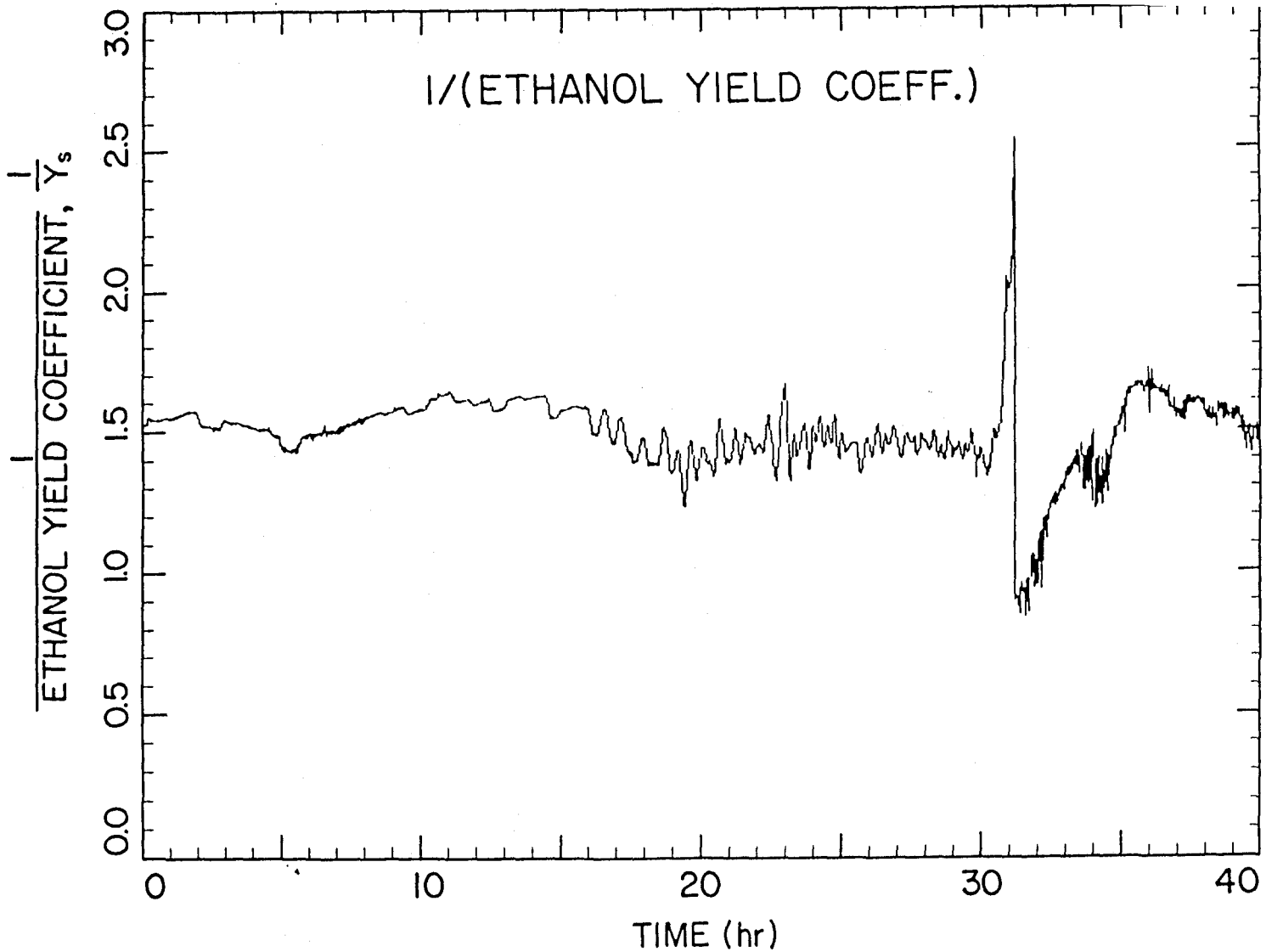


Figure 4.4.16. Substrate (ethanol) to cell biomass yield coefficient as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol. The yield coefficient is calculated with the aid of macroscopic material and elemental balance.

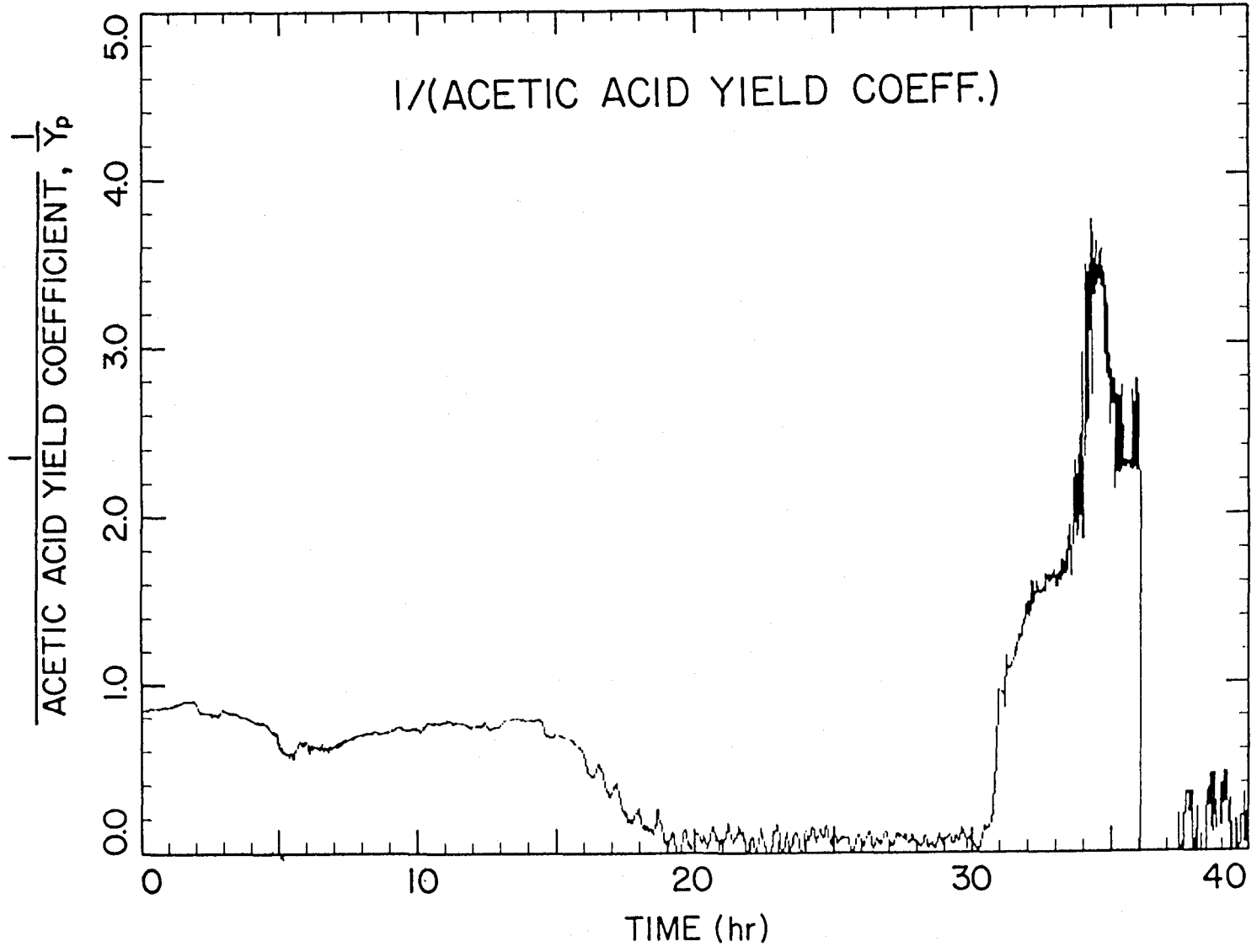


Figure 4.4.17. Product (acetic acid) to cell biomass yield coefficient as a function of time in a batch fermentation of *S. cerevisiae* in 5.0 g/l of ethanol. The yield coefficient is calculated with the aid of macroscopic material and elemental balance.

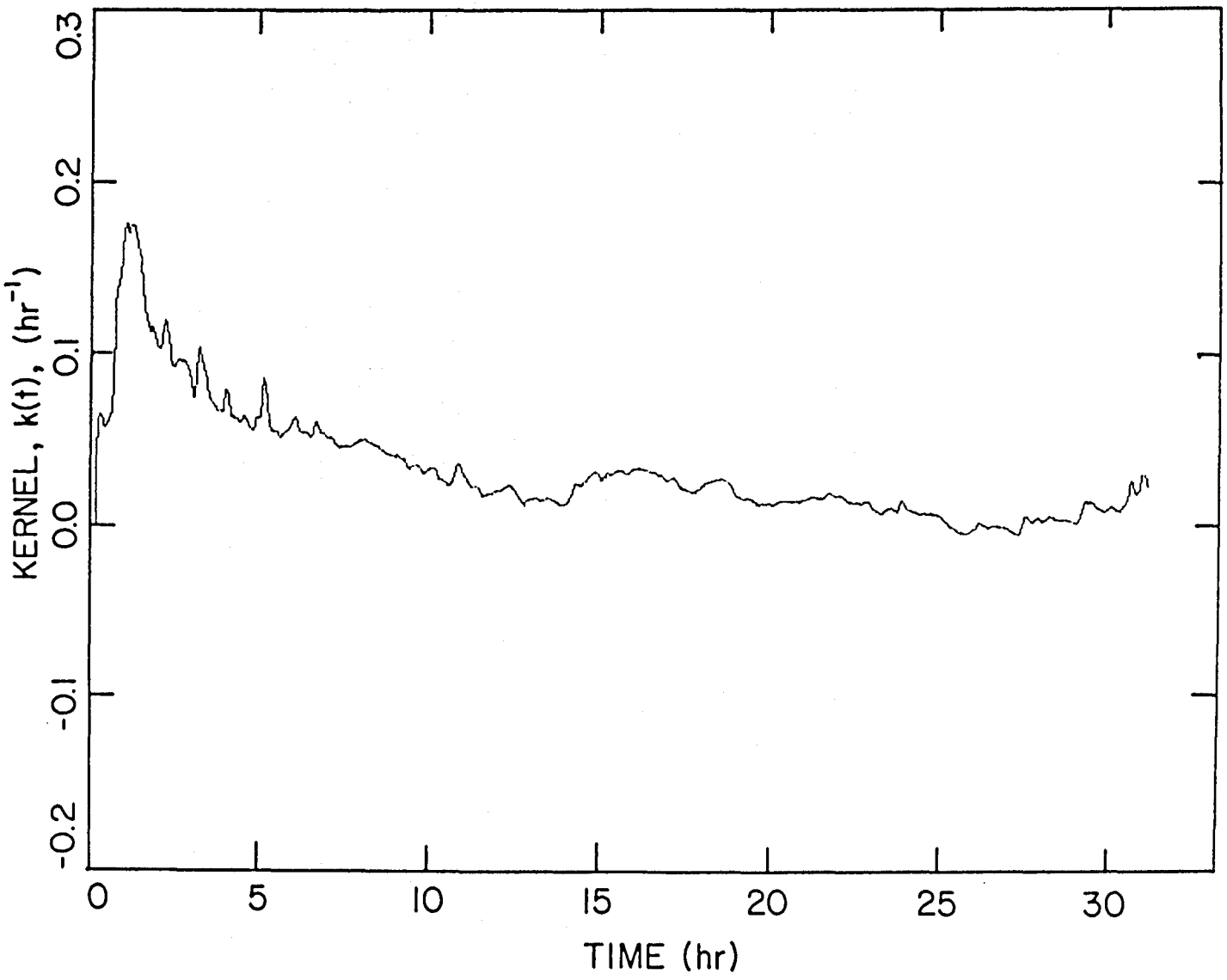


Figure 4.5.1. Time-lag kernel function of a continuous fermentation resulting from the shift up in the dilution rate from 0.100 hr^{-1} to 0.288 hr^{-1}.

4.6 RESULTS

The kernel obtained above is used to predict the batch response of the run presented in Figures 2.1.1 to 2.1.6. The predictions of the time-lag model are presented in Figures 4.6.1 through 4.6.5, along with the off-line measurements represented by the circles. The smooth solid line next to the averaged noisy measurement in Figure 4.6.1 represents the evolution of biomass concentration as a function of time based on the time-lag model of this thesis. For comparison, the time course predicted by the corresponding Monod model is also shown in the same figure. Both of the curves are generated by using the same set of growth parameters obtained in the series of steady-state experiments described earlier; the only difference is that time-lag effects are recognized in the time-lag approach and ignored in the corresponding conventional Monod model. It can be seen that the time-lag approach gives a far superior prediction for the time variation of the biomass concentration than the Monod model does. Similar predictions are also generated for the glucose and ethanol concentrations in Figures 4.6.2 and 4.6.3. Once again, the superiority of the time-lag model is well demonstrated in these two figures.

The total growth rates predicted by both models are shown in Figure 4.6.4, and the specific growth rates are shown in Figure 4.6.5. These growth rate curves indicate the source of discrepancies in the two modeling approaches. According to the Monod model, the specific growth rate of the yeast is instantaneously raised to a high value and the growth starts immediately when the inoculum is placed inside the fermentor. On the other hand, the time-lag model predicts a slow rise in the specific growth rate, and only approximately 5 hours after the inoculation does the growth rate become significant. The time needed for the cells to synthesize the necessary intermediates and enzymes before a full growth can be ensured is accounted for in the time-lag model but neglected in the Monod model. As a

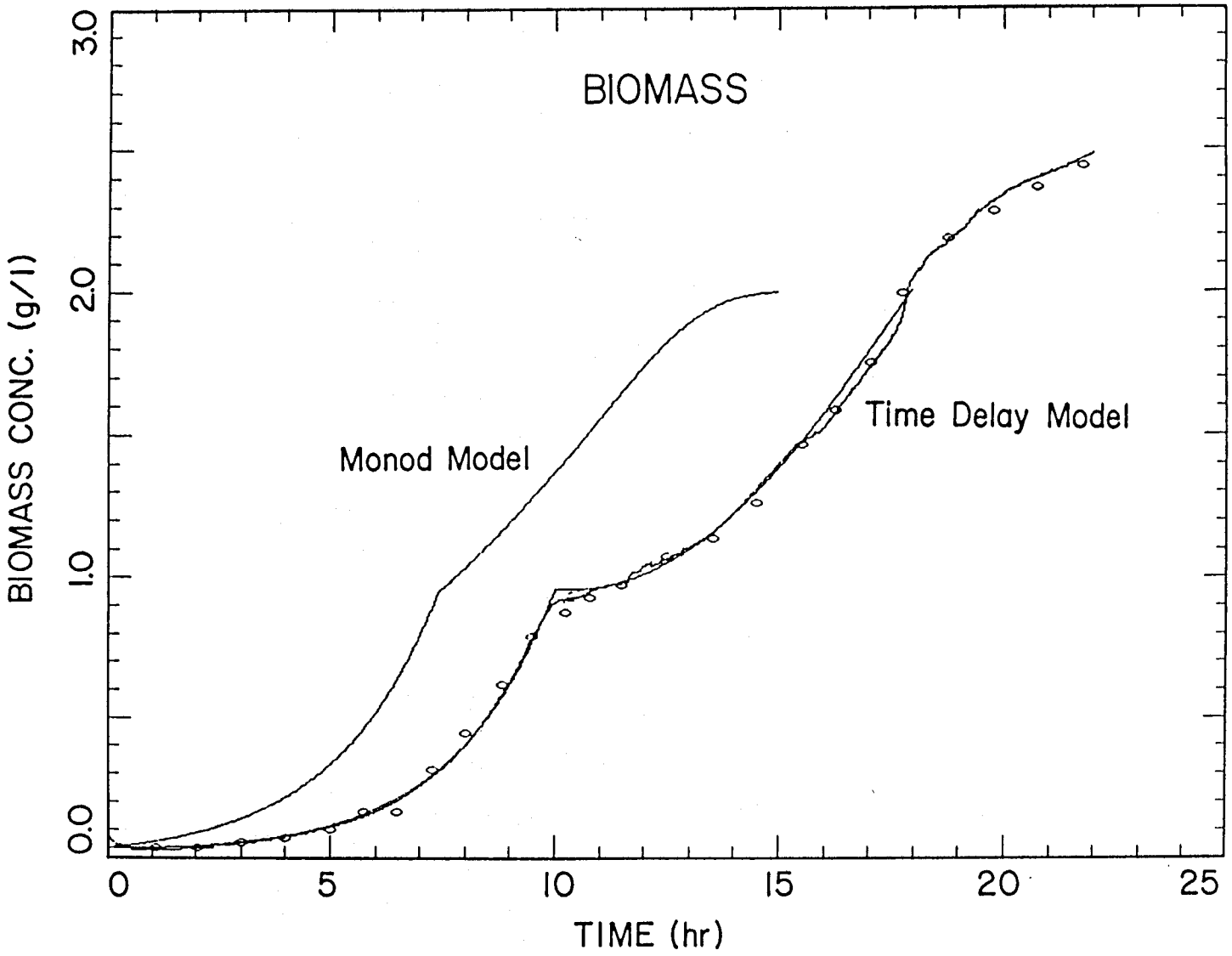


Figure 4.6.1. Comparison of the prediction of the cell biomass concentration as a function of time from the time-lag model and the Monod model in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The parameters used for the computer simulation are listed in Figure 4.6.6.

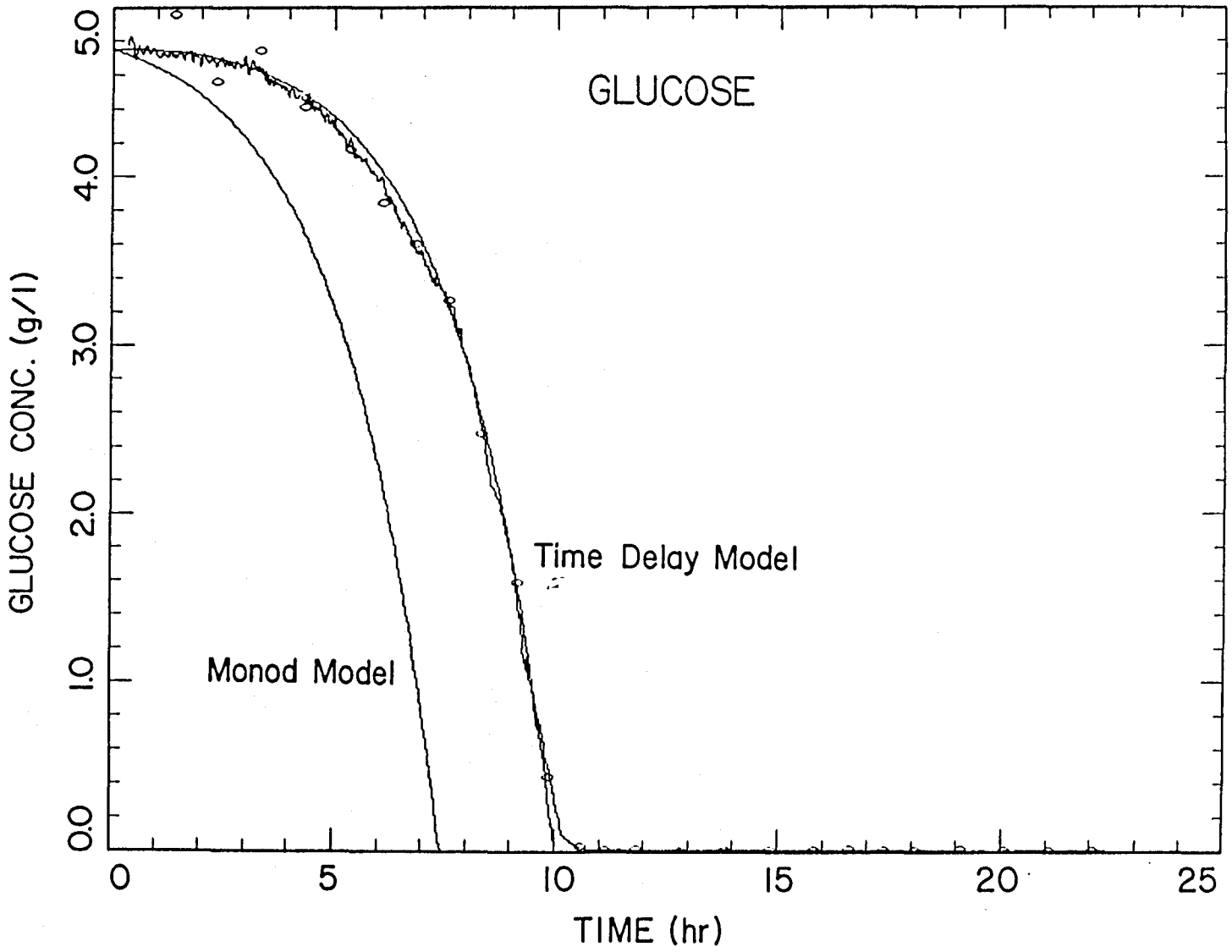


Figure 4.6.2. Comparison of the prediction of the glucose concentration as a function of time from the time-lag model and the Monod model in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The parameters used for the computer simulation are listed in Figure 4.6.6.

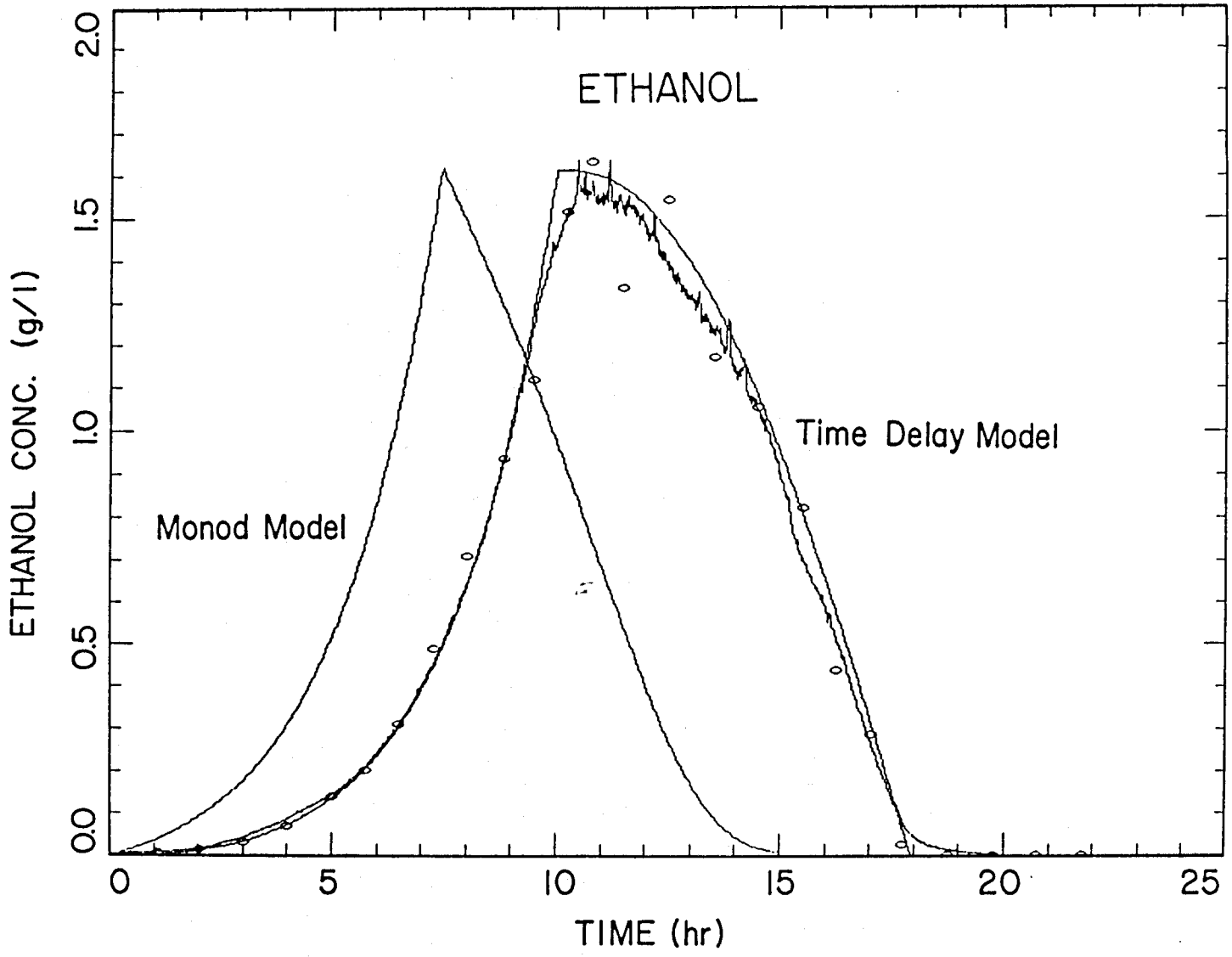


Figure 4.6.3. Comparison of the prediction of the ethanol concentration as a function of time from the time-lag model and the Monod model in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The parameters used for the computer simulation are listed in Figure 4.6.6.

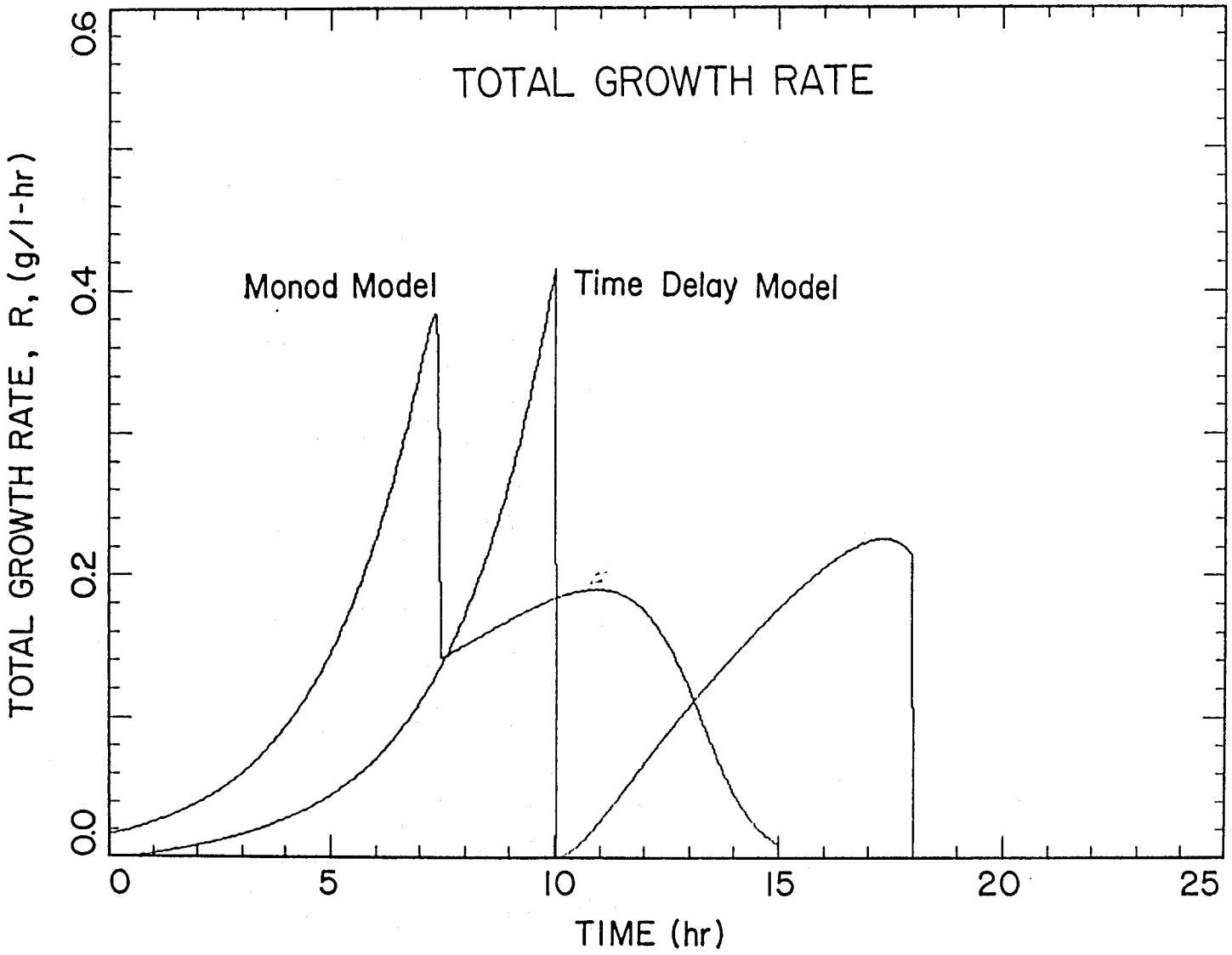


Figure 4.6.4. Comparison of the prediction of the total growth rate as a function of time from the time-lag model and the Monod model in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The parameters used for the computer simulation are listed in Figure 4.6.6.

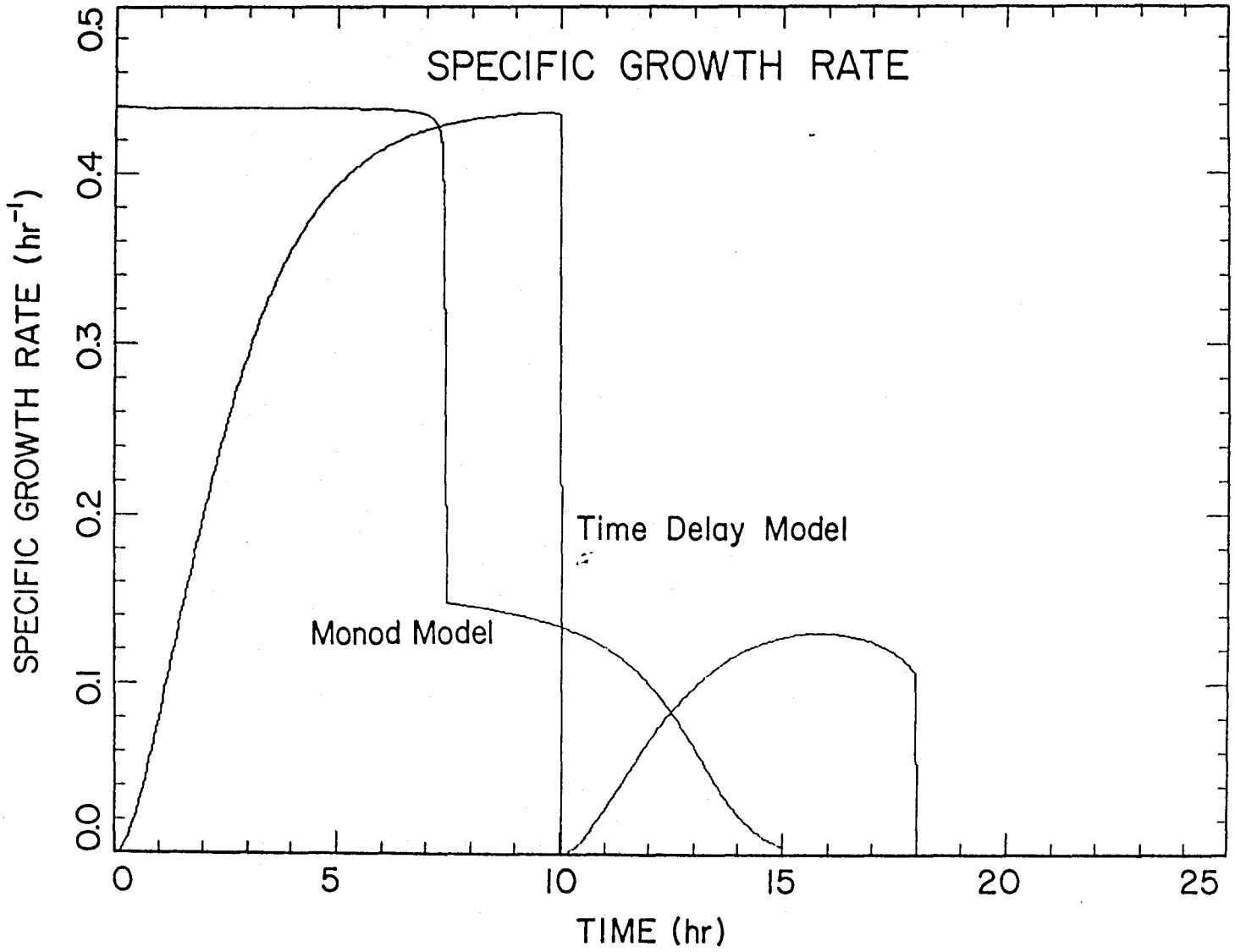


Figure 4.6.5. Comparison of the prediction of the specific growth rate as a function of time from the time-lag model and the Monod model in a batch fermentation of *S. cerevisiae* in 5.0 g/l of glucose. The parameters used for the computer simulation are listed in Figure 4.6.6.

result, the Monod model predicts a much higher growth rate in the beginning of the batch fermentation, which leads to a much sooner exhaustion of the glucose substrate.

The existence of two distinct and completely separated growth regions for the diauxic growth of yeast on glucose and ethanol is experimentally demonstrated in Figure 2.1.3, which shows the time course of the total growth rate calculated from the on-line pH activities. Because the oxygen uptake rate and the carbon dioxide evolution rate are indicative of the level of intracellular metabolic activities, the gas measurements shown in Figures 2.1.1 and 2.1.2 further confirm that the growth regions are separated. These two separated growth regions are predicted by the time-lag model as shown in Figure 4.6.4. Note that although the Monod model also predicts two regions of growth arising from the consecutive consumption of glucose and ethanol, they are not separated and qualitatively incorrect. The drop in the growth rate predicted by the Monod model occurs too soon and is quantitatively incorrect, as well.

The kernel function used to generate the predicted curves for the time-lag model is shown in Figure 4.6.6, along with the values for all the other parameters.

The sensitivity of the time-lag model to the variations in the assumed model parameters is studied. The values of the maximum specific growth rate, μ_m, the half-saturation constant, K_s, the glucose to cell yield coefficient, Y_s, the product (ethanol) yield coefficient, Y_p, the relative order of the kernel, a_0, and the lag time-constant, T, are varied around the original settings. Shown in Figures 4.6.7 through 4.6.13 are the changes in the predicted values of the biomass concentration, the glucose concentration, the ethanol concentration, and the specific growth rate as these model parameters are varied. One sees that the response is affected when μ_m is changed in either direction by 10 %, while even a 10-fold change in the value

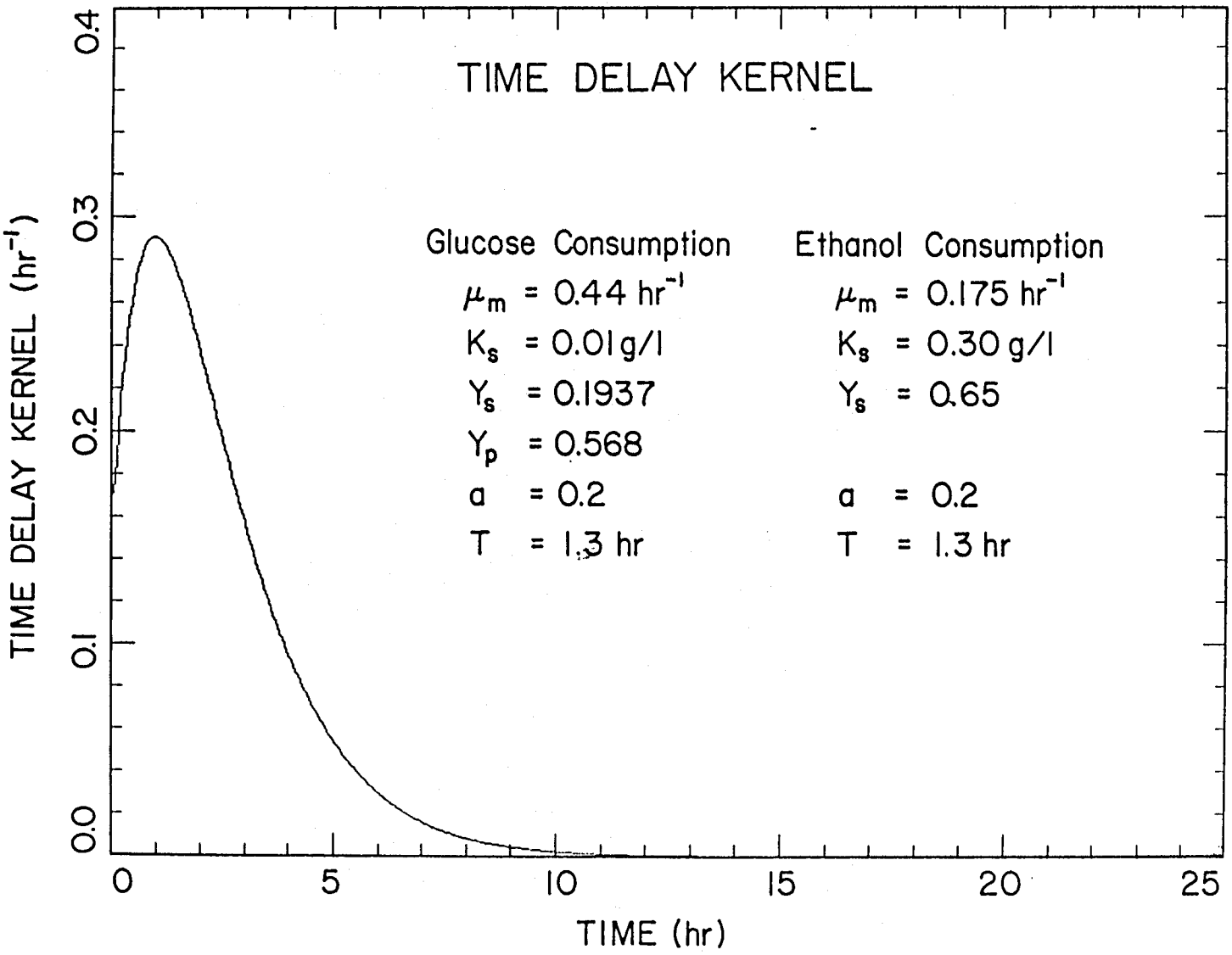


Figure 4.6.6. Time-lag kernel function and various parameters used to generate the curves in Figures 4.6.1 – 4.6.5. The kernel is composed of both a 0th-order exponential distribution function and a 1st-order exponential distribution function with a lag time constant of 1.3 hr.

of K_s does not seriously affect the prediction. The relative insensitivity of the concentration response to K_s can be explained in terms of the much higher glucose concentration prevailing throughout the first phase of the fermentation compared to the value of K_s. Changes in the values of Y_s affect mainly the biomass and ethanol concentrations during the diauxic phase, whereas changes in the values of Y_p affect only the ethanol concentration. One interesting observation is that the relative order of the time-lag kernel shifts the concentration responses only indistinctly. Similar to the insensitivity of K_s, this is only true for a batch fermentation. Finally, the lag time-constant of the kernel function strongly influences the course of the output. Of course, in the limit as T approaches 0, one expects the behavior to approach that predicted by the Monod model shown in Figures 4.6.1 through 4.6.6.

Because a small amount of the cell-free filtrate is destructively used in the analysis of the glucose concentration, the flow rate associated with the biomass dynamic equation is slightly lower than the flow rate associated with the glucose dynamic equation. Physically, this is equivalent to the existence of a cell recycle loop. Mathematically, slightly different dilution rates of D_x and D_s were appropriately used in the analysis of the system dynamics, as described by the following set of equations, when the glucose analyzer was in operation:

$$\frac{dx}{dt} = -D_x x + \mu(s)x \tag{4.6.1}$$

$$\frac{ds}{dt} = D_s(s_f - s) - \frac{1}{Y_s}\mu(s)x \tag{4.6.2}$$

$$\frac{dp}{dt} = -D_s p + \frac{1}{Y_p}\mu(s)x. \tag{4.6.3}$$

For a batch fermentor, the variable D_s is equal to 0, and the variable D_x is equal to $-\frac{F}{V}$, where F is the flow rate of the filtrate stream in the continuous cross-flow filter flowing into the glucose analyzer and V is the instantaneous volume of the broth in the bioreactor. For a continuous fermentor, D_s is the dilution rate based

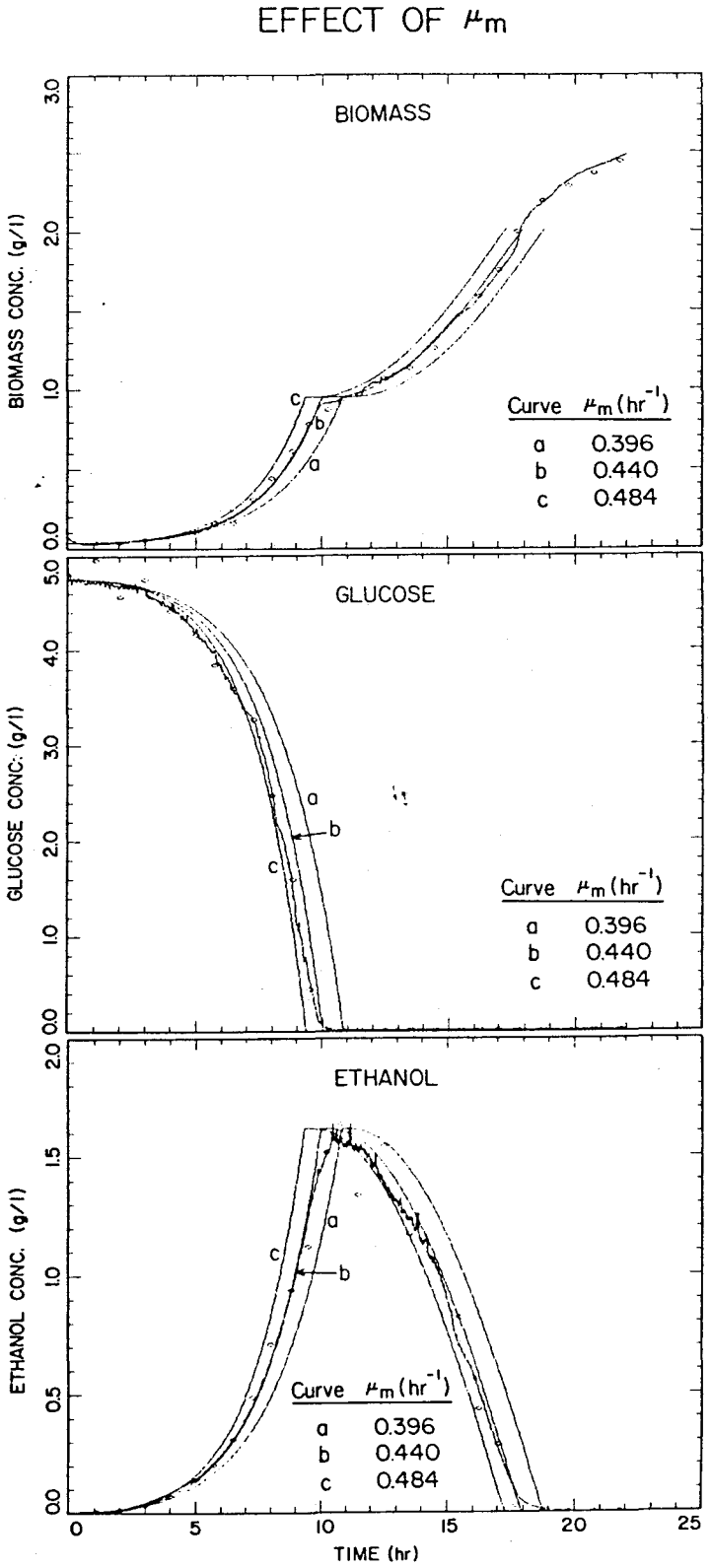


Figure 4.6.7. Effect of the maximum specific growth rate, μ_m, on the predicted trajectories of a) biomass concentration, b) glucose concentration, and c) ethanol concentration.

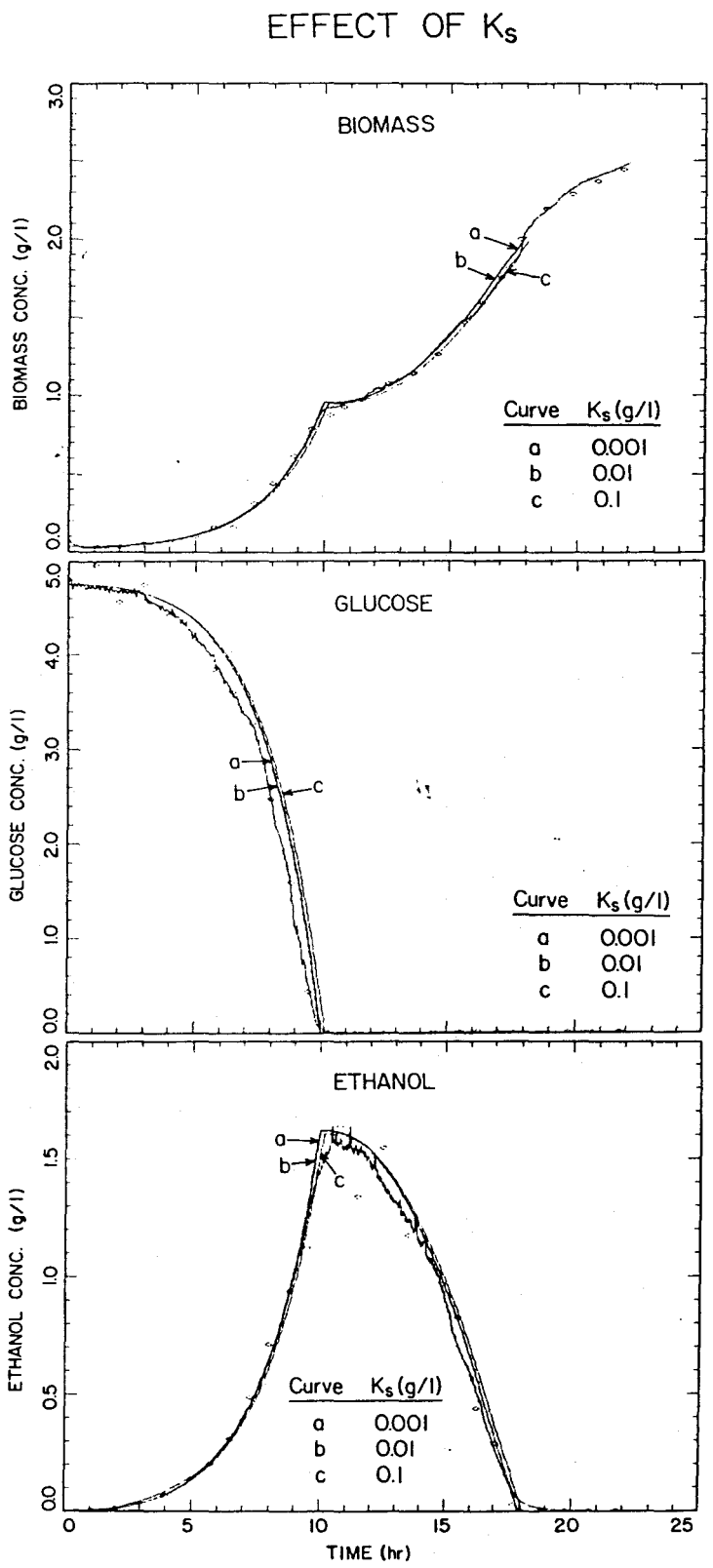


Figure 4.6.8. Effect of the Michaelis-Menton constant, K_s, on the predicted trajectories of a) biomass concentration, b) glucose concentration, and c) ethanol concentration.

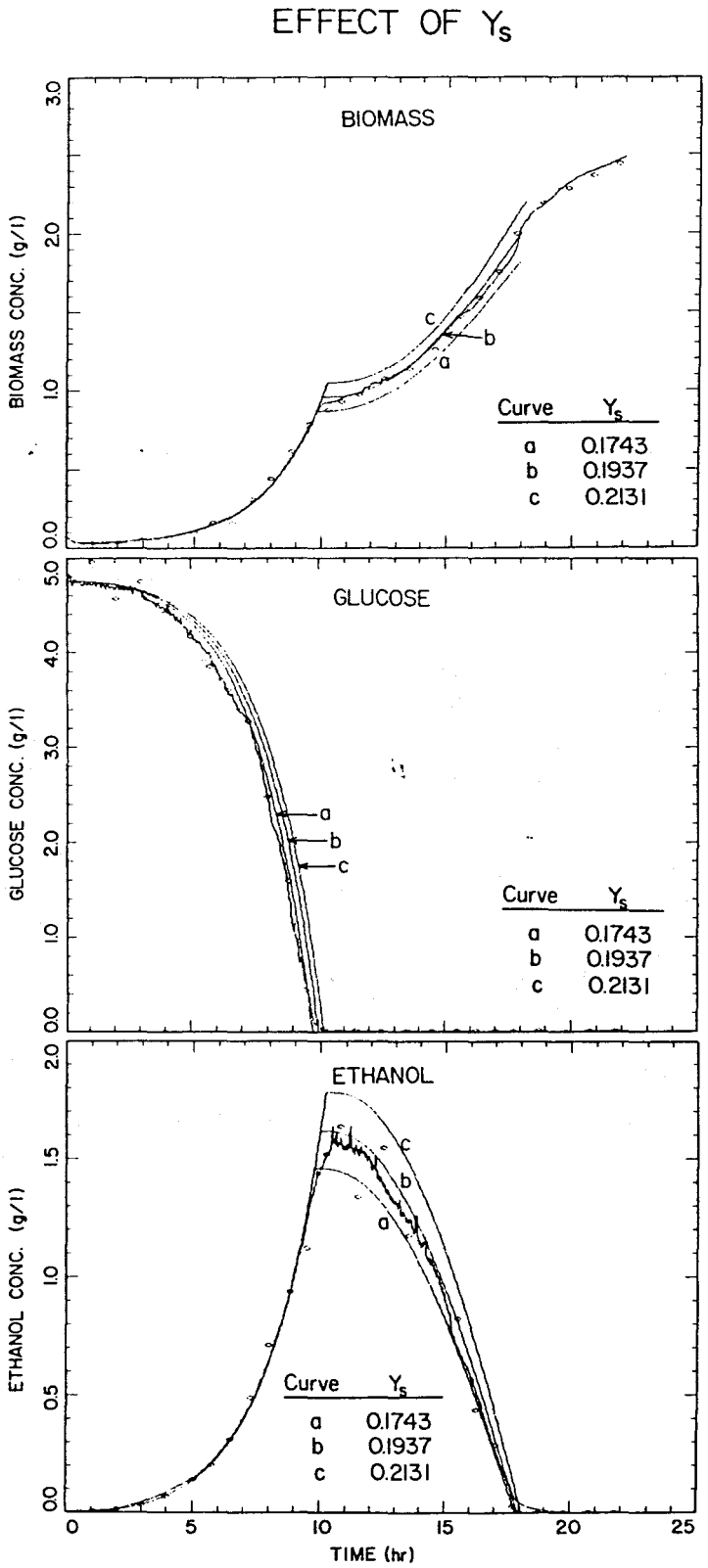


Figure 4.6.9. Effect of the glucose to biomass yield coefficient, Y_s, on the predicted trajectories of a) biomass concentration, b) glucose concentration, and c) ethanol concentration.

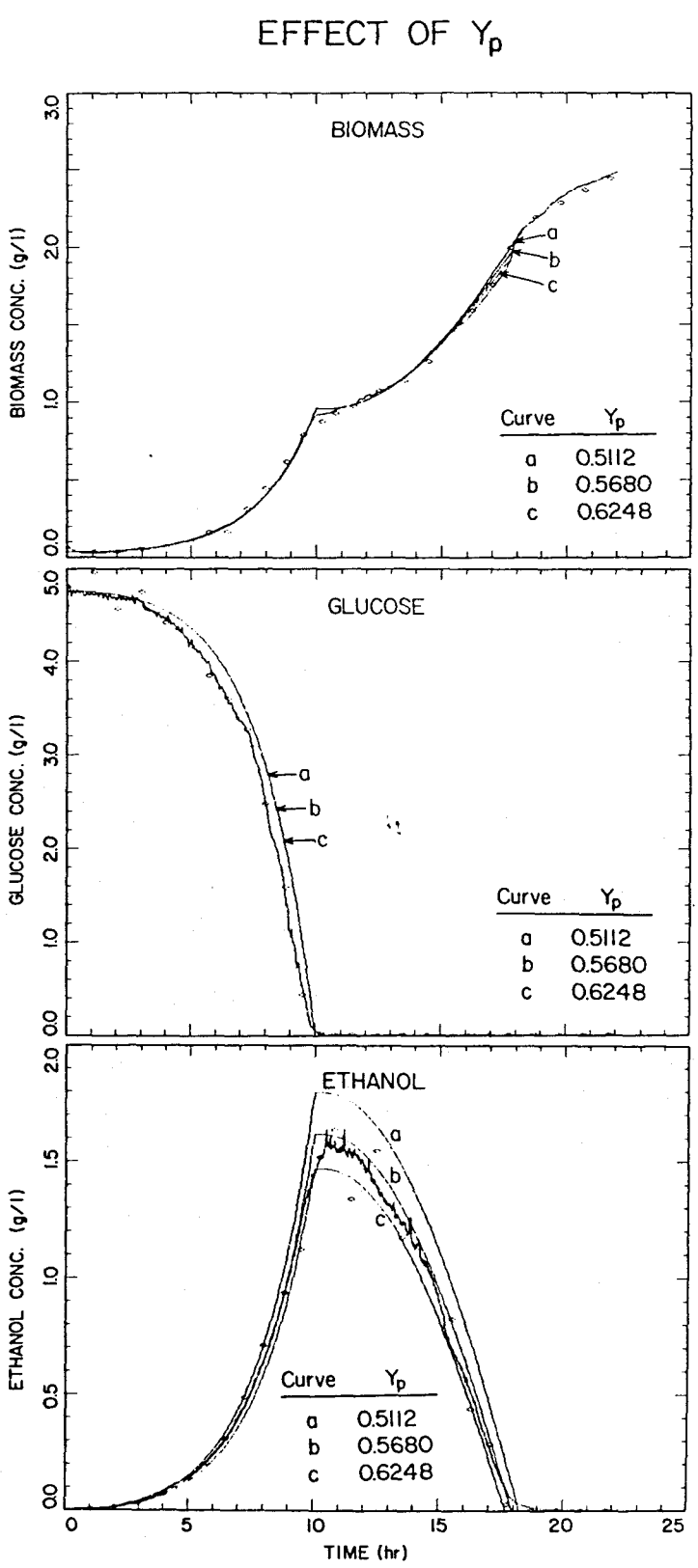


Figure 4.6.10. Effect of the ethanol to biomass yield coefficient, Y_p, on the predicted trajectories of a) biomass concentration, b) glucose concentration, and c) ethanol concentration.

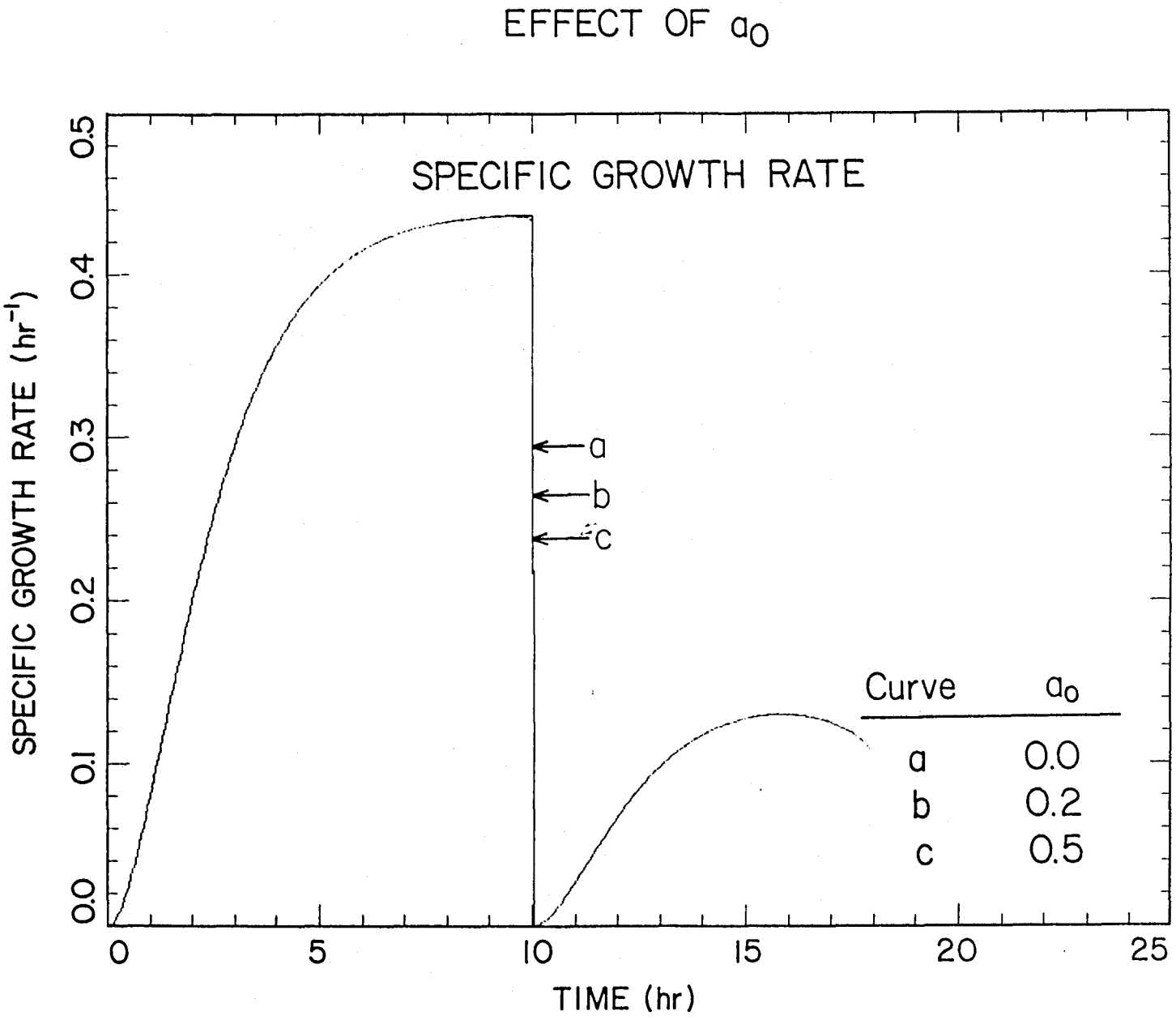


Figure 4.6.11. Effect of a_0 on the predicted values of the specific growth rate. Note that the difference is hardly noticeable.

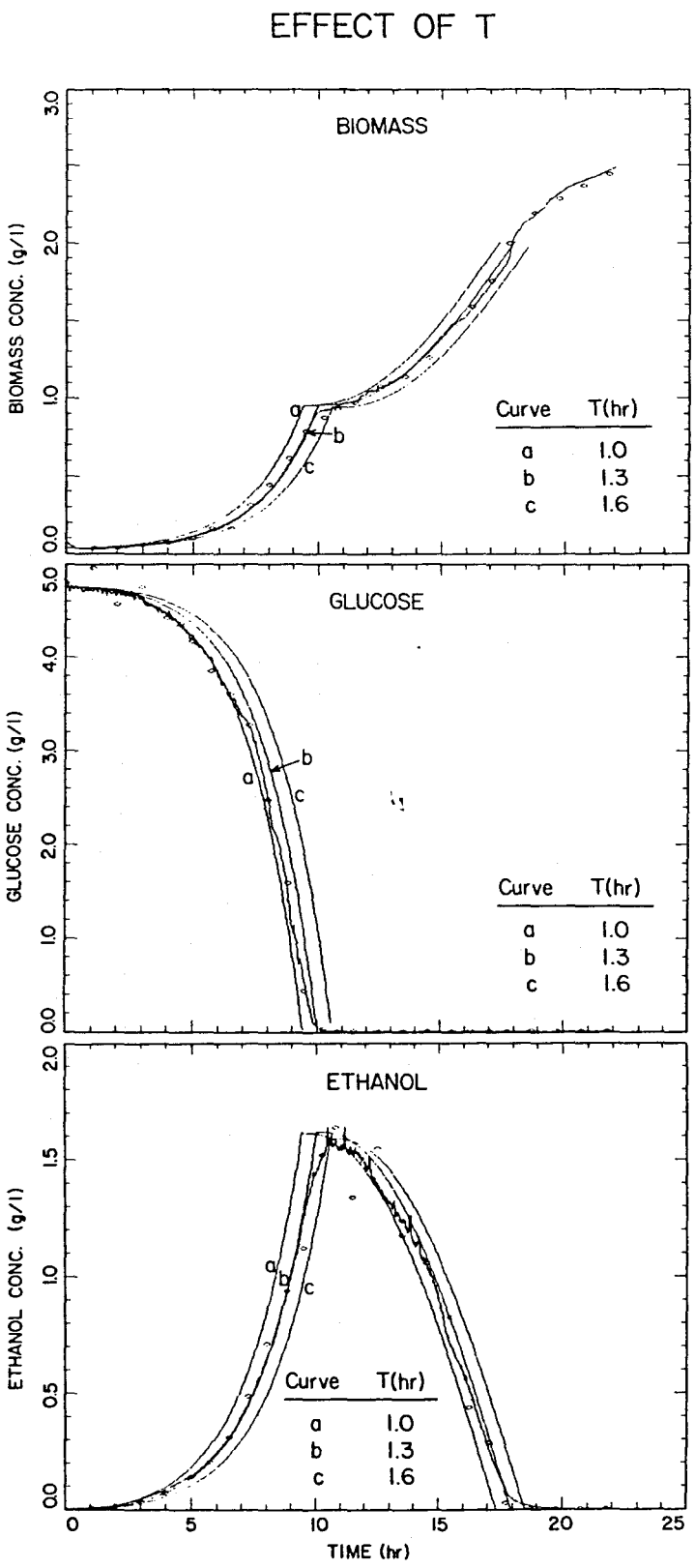


Figure 4.6.12. Effect of the lag time constant, T, on the predicted trajectories of a) biomass concentration, b) glucose concentration, and c) ethanol concentration.

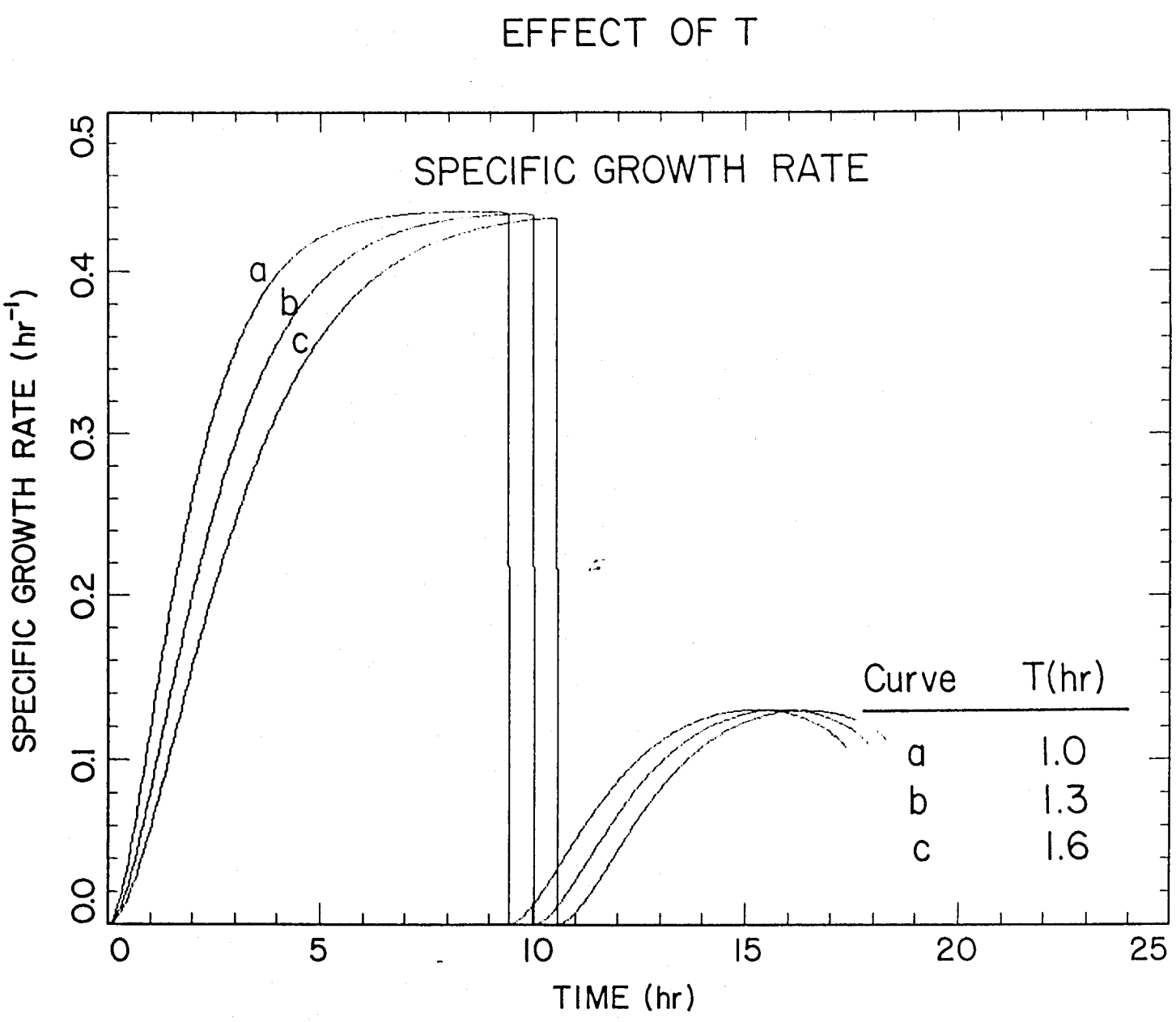


Figure 4.6.13. Effect of the lag time constant, T, on the predicted values of the specific growth rate.

on the inlet nutrient flow rate, *i.e.* $\frac{F_0}{V}$, and D_x is the dilution rate based on the outlet waste flow rate, *i.e.* $\frac{F_0-F}{V}$. The $D_x x$ term does not represent the loss of biomass due to the outflow; it represents the effect of concentrating the biomass material due to loss of fluid in the filter. Consequently, the continued increase in the biomass concentration in Figure 4.6.1 after the exhaustion of all the substrates does not suggest a continued cell growth but reflects the concentration effect of the glucose analyzer. The glucose analyzer continued to be in operation even after the exhaustion of glucose was evident.

4.7 DISCUSSION

An observability test shows that the Kalman filter cannot observe the state of substrate concentration in Equations (2.2.5) and (2.2.6) with only the measurements of the total growth rate and the substrate to biomass yield coefficient. Because of the relatively small value of the glucose concentration observed throughout most of the continuous runs, the Kalman filter technique failed to estimate the glucose concentration correctly in the absence of a continuous glucose measurement. For a very small glucose concentration, s, the term $D(s_f - s)$ is not sensitive to the actual value of s. It is the difference of $s_f - s$, not s alone, that makes up the term. Thus, a 0.05 g/l (*i.e.*, 1 %) of uncertainty in the feed glucose concentration of 5.0 g/l will cause the same absolute magnitude of uncertainty in the estimated values for the residual glucose concentration, which itself is 0.01–0.03 g/l under most of the circumstances. Therefore, a 1% uncertainty in the feed glucose concentration translates into an error of 500 % in the estimate for the residual glucose concentration in a fermentor. This is clearly not acceptable, and an independent means of measuring the glucose concentration continuously on-line is needed. Similarly a small error in the measurement of the dilution rate will also lead to the same conclusion.

Furthermore, the curve of the specific growth rate versus the limiting substrate concentration is extremely steep at low glucose concentrations. Consequently, the estimate for the intrinsic specific growth rate from the substrate concentration measurement is also exceedingly sensitive to the glucose concentration, and idealistically, an experiment should be planned such that the glucose concentration profile as a function of time is as near an impulse as possible to achieve a dependable estimation of the kernel.

Although the above discussion concerns the fermentation of glucose, the same argument over the substrate observability problem of the Kalman filter and the sensitivity of the $\mu(s)$ function also applies to the continuous fermentation in which ethanol is employed as the limiting substrate.

Because of these difficulties, it is suggested that some other variables representing the intrinsic growth rate, perhaps some rate-limiting enzymes that participate in the "master reaction" of a complex reaction network, be identified and their time course be monitored with an advanced analytic technique. Such capability will greatly enhance the observability of the kernel. Perhaps those types of kernel associated with variables other than the specific growth rate can be determined as well. In general, vital information is stored within the composition of the intracellular mixture of bio-molecules. To gain a more fundamental understanding on the workings of a cell, it is necessary that one quantify this composition. Analogous to the use of a microscope that allows an investigator to observe the shape of an object in greater spatial detail, the capability to monitor continuously the changes in the internal biochemical composition will allow one to see events in greater chronological detail. Thus, achieving a continuous measurement capability is equivalent to possessing a time magnifier. This capability poses perhaps the greatest promise in the true understanding of a biological system. Many biologically important molecules

can be detected with a spectrofluorometer because they can be photometrically excited to give fluorescent emission. Among these are a variety of proteins, vitamins, cytochromes, amino acids, and NADH. Many more can be specifically labelled with appropriate fluorescent stains and measured. A fully automated cytoflowmeter or a scanning fluorescence spectrophotometer will greatly contribute toward continuous quantification of the intracellular composition.

We view a microbial cell as a self-controlled and self-sustained chemical reactor. Ultimately, one would like to be able to control directly the reaction conditions. This is theoretically possible when enough information is available on the cell metabolism, regulation, and replication process, but the jump from the basic understanding to the actual implementation of that learning is a major one. This modeling work, aimed at the immediate objective of automating the entire fermentation system, is part of the larger effort toward that jump.

No matter how small a cell is, it is a complete chemical reactor in itself. Currently, one has almost no control over the internal composition of a cell; one is totally at the mercy of the runaway chemical reactor. The runaway chemical reactor's course of action generally benefits the microorganism and ensures its own survival, but it seldom matches the interests of the helpless operator. One can no longer wait and hope to reap profit from the natural action of a cell; he must have at least some control over the events occurring inside the microscopic cell. Even a limited control capability will greatly enhance the performance of a bioreactor. During the past decade, there has been an overwhelming enthusiasm toward the modification of one of the components in this chemical reactor, namely the DNA component, by applying either the genetic manipulation techniques or the hybridoma technology to bestow a cell with novel capabilities. Quite frequently, the desired genes are

already in place in the natural environment and there is no need to encode such capabilities. However, the abundant expression of such genes into the desired product is not guaranteed, whether they are naturally derived or artificially encoded. The level of effort in the bioprocess engineering has not matched that in the field of gene manipulation. Thus, one may build a machine and yet not know how to operate it. One is surrounded by an abundance of highly sophisticated hardware (the cells). And if benefit is to be derived from this hardware, one must generate the software needed to drive the hardware to meet the society's needs. This work comprises a step toward the improved operation of this new machine and the elimination of the disparity between the potential capability of the machine and the actual output.

To operate the complex chemical reactor, one can either scale down the cell handling equipment or scale up the cell to a more manageable size. In this way, one can more closely control the factors affecting the synthesis of the desired product, such as the addition of certain inducers and promoters. For example, one can formulate the chemical composition and reproduce the same condition in the reactor in such a way that the desired product can be selectively synthesized and the waste minimized.

Because of the complex nutrient requirement, until about a decade ago a chemically well-defined synthetic medium without the addition of serum for the cultivation of an animal cell culture was not possible, but today it is. We hope that one day we can formulate the exact conditions existing inside a cell and scale up a cell to a larger biochemical reactor. Indeed, this will be the largest scale up problem one has ever encountered. Undoubtedly, this will be the most complex chemical reactor man has ever designed, which will also demand the most advanced control formulation for its operation. Included in this ultimate process control formulation will not only be the process controller algorithm but also the solutions to the problems in

sensor development, regulator design, the identification of the controlled variables and the manipulated variables, and the process modeling and optimization.

CHAPTER 5

CONCLUSIONS AND PROPOSALS

5.1 CONCLUSIONS

The importance and the presence of time lag have been recognized for many years, and in this thesis we have attempted to offer a simple mathematical means by which the idea of time-lag can be incorporated into the existing models without drastically increasing the complexity of the models. Armed with the time-lag kernel, the proposed model can predict the transient behaviors as well as the steady-state behaviors. The kernel is shown to be the by-product during the reduction of a structured model to an equivalent unstructured model. It contains all the information that is lost during model reduction; all the biological knowledge is compressed into a time-lag kernel. It is powerful, and it has biological significance and a physical interpretation. The representation of a system's dynamics in an input-output format makes it straightforward to identify the cause-effect relationship, a quality indispensable in process design, control, and optimization.

Furthermore, it is shown that despite the presence of time-lag integrals in the dynamic equations, the proposed approach is indeed simpler. By expanding the lag kernel in a series of exponential distribution functions, the integro-differential equations can be easily reduced to a set of first-order ordinary differential equations for which the mathematical theories are well developed and various established analytical techniques are immediately available. It is emphasized that one of the advantages of the proposed methodology is precisely the manner in which the time-lag equations are formulated to allow the direct application of these tools. The stability analysis of a set of equations containing time-lags, for example, is the result of the direct application of such well established mathematical methods.

Finally, it is demonstrated through computer simulated experiments that the time-lag kernels can be feasibly reconstructed from the presently available measurements. The steps required to elucidate the time-lag kernel are outlined and

executed. The time-lag kernels have been determined experimentally for a system of *S. cerevisiae* growing on glucose and ethanol. The performance of the time-lag model in predicting the actual transient behavior is shown to be much superior to that of the unstructured model that does not consider the time-lag effects. As far as the author is aware, this is the first time that steady-state data have been successfully used to describe the transient behavior in a rigorous manner. Among the commonly observed phenomena in a bioreactor correctly predicted by this model are the oscillatory trajectories, the asymmetrical transient behaviors following a shift-up or shift-down in the dilution rate in a continuous fermentor, and the lag phase and diauxic patterns in a batch fermentor.

5.2 PROPOSALS AND FUTURE WORK

As shown in Chapter 2, the time-lag approach to bioprocess modeling can be viewed as a method of lumping a set of reactions. Other previously proposed lumping techniques and dimension reduction methods should further enhance the appeal of the time-lag approach proposed herein. It is well known that the age of the inoculum also affects the length of the lag phase in the beginning of a batch fermentation. Further studies are also desirable to extend the time-lag kernel concept to account for the age or size distribution of a culture. On the other hand, the same transformation technique used here to eliminate the integral in a differential equation can be applied to the similar type of equations frequently encountered in various other fields. For example, the memory effect present in the studies of non-Newtonian polymeric fluids can be treated in a more general manner, and the memory function may be more readily related to the more fundamental underlining causes in terms of inter-molecular interactions, polymeric molecular conformation, or the stretching of chemical bonds. In addition, as demonstrated in Chapter 2, the solution of some of the non-Newtonian flow equations may be greatly simplified if

the memory function is expanded in terms of the exponential distribution functions instead of the more commonly used power series expansions.

The time-lag kernel considered in this thesis is substantially associated with a set of linear time-invariant dynamic equations. The time-lag kernel approach to time dependent dynamic equations may be investigated as the next logical step. Highly nonlinear dynamic equations may be directly studied only as the final resort if quasi-linearization cannot adequately approximate the actual system. However, a visual or mental perception of the kernel for these more complicated situations may prove to be difficult for the uninitiated, and the gain may not be substantial to warrant the use of time-lag kernels because one of the major attractions of the time-lag kernel approach lies in its simplicity. Overreaching should be avoided.

It is important to keep in mind that the proposed time-lag kernel offers a balance between the two conventional approaches. It offers an alternative for those who are caught in a situation where an unstructured model does not meet their needs and at the same time their means does not satisfy a structured model's appetite for all the necessary parameters. Even when a close tracking of all the structured model's states becomes technically feasible in the future, installing all such capabilities in a bioreactor may not be prudent or expedient. It is originally proposed for use in the control and optimization of a fermentation process in which certain mechanisms may be too intricate, irrelevant, insignificant, or simply too impractical to be troubled with. Granted, it is not suitable for the improved fundamental understanding of the dynamics of bioreactors; in this case, one ultimately needs to resort to a fully expanded structured model. Currently, no single modeling method is all powerful, and no such claim is made on this one, either. A model is a tool. As such, it can be extremely useful if it is chosen for the right job. Conversely, an

otherwise perfectly good tool can afflict damage to itself, the work, and even the operator if it is not applied judiciously.

This modeling work contributes to the larger effort toward the ultimate objective of the advanced control of a biological reactor. As repeatedly pointed out in Appendices E and G, the lack of reliable continuous on-line sensors and the inadequacy of the heretofore proposed models constitute the bottleneck restricting the advancement of biochemical engineering. More effort is needed in the area of biosensor development. In the future studies, other recently developed instruments should be added to the fermentor. An ammonia electrode, and a variety of enzyme electrodes and enzyme thermistors will provide one with more valuable information. In the past few years, a gas chromatograph, a high performance liquid chromatograph, and a mass spectrometer have been interfaced to a computer by various independent investigators in the field of biotechnology, and the information obtained from these automated instruments, especially when they are operated simultaneously, should enable one to construct a very reliable model, which in turn can be of great asset in devising the strategy needed to control a fermentor. This includes various modes of fermentor operation, including a transient operation.

One cannot overemphasize the fact that except for the fluorescence measurement of NADH, all the presently available probes capable of being interfaced to a computer measure the level of components outside the microbial cells, while the majority of the enzymatically catalyzed reactions occur inside a cell. In future research, other methods of detecting the internal concentrations should be investigated. For example, various fluorescence measurements, the use of tunable laser, development of on-line staining techniques coupled with the computer interfacing of a flow cytometer should command more attention. The possibility of adapting recent advances in other areas, especially those in the clinical or immunological field,

should be seriously considered. For example the sonar techniques developed in the medical diagnostic field should be quite promising. It may be possible that ultrasonic waves can be used to measure the cell size distribution nondestructively and to eliminate contaminants that have significantly different sizes from the working strain.

REFERENCES

1 Ataai, M. M. and Shuler, M., Simulation of the growth pattern of a single cell of *Escherichia coli* under anaerobic conditions, *Biotech. Bioeng.*, **27**, 1027, 1985.

2 Ataai, M. M. and Shuler, M., Simulation of CFSTR through development of a mathematical model for anaerobic growth of *Escherichia coli* cell population, *Biotech. Bioeng.*, **27**, 1051, 1985.

3 Bailey, J. E. and Ollis, D. F., *Biochemical Engineering Fundamentals*, McGraw-Hill, New York, 1977, p360.

4 Bellman, R. and Cooke, K. L., *Differential-Difference Equations*, Academic Press, New York, 1963.

5 Cooney, C. L., Computer application in fermentaion technology – a perspective, *Biotechnol. Bioeng. Symp.*, **9**, 1, 1979.

6 Cushing, J. M., *Integrodifferential Equations and Delay Models in Population Dynamics*, Springer-Verlag, Berlin, 1977.

7 Crooke, P. S., Wei, C.-J., and Tanner, R. D., The effect of the specific growth rate and yield expression on the existence of oscillatory behavior of a continuous fermentation model, *Chem .Eng. Commun.*, **6**, 333, 1980.

8 Dairaku, K. and Yamané, T., Use of the porous Teflon tubing method to measure gaseous or volatile substances dissolved in fermentation liquids, *Biotech. Bioeng.*, **21**, 1671, 1979.

9 Davison, B. H., *Ph D. Thesis*, California Institute of Technology, 1985.

10 Dinwoodie, R. C. and Mehnert, D. W., A continuous method for monitoring and controlling fermentations using an automated HPLC system. *Biotech. Bioeng.*, **27**, 1060, 1985.

11 Domach, M. M., Leung, S. K., Cahn, R. E., Cocks, G. G., and Shuler, M. L., Computer model for glucose-limited growth of a single cell of *Escherichia coli* B/r-A, *Biotech. Bioeng.*, **26**, 203, 1984.

12 Domach, M. M. and Shuler, M. L., Testing of a potential mechanism for *E. coli* temporal cycle imprecision with a structural model, *J. Theor. Biol.*, **106**, 577, 1984.

13 Essajee, C. K. and Tanner, R. D., The effect of extracellular variables on the stability of the continuous baker's yeast–ethanol fermentation process, *Process Biochemistry*, **14**, No.5, 16, 1979.

14 Fingerhut, B., Ferzola, R., and Marsh, W. H., Application of a ferrocyanide-phosphomolybdate reaction to an automated determination of serum glucose, *Clin. Chim. Acta*, **8**, 953, 1963.

15 Fingerhut, B., Ferzola, R., Marsh, W. H., and Miller, A. B., Automated methods for blood glucose and urea with adaptation for simultaneous determination, *Clinical Chemistry*, **12**, 570, 1966.

16 Fredrickson, A. G., and Tsuchiya, H. M., Microbial kinetics and dynamics, in *Chemical Reactor Theory — A Review*, Lapidus, L. and Amundson, N. R., Eds., Prentice-Hall, Englewood Cliffs, N. J., 1977, Chapt. 7.

17 Grosz, R., Stephanopoulos, G., and San, K.-Y., Studies on on-line bioreactor identification. III. Sensitivity problems with respiratory and heat evolution measurements, *Biotechnol. Bioeng.*, **26**, 1198, 1984.

18 Herbert, D., Continuous culture of microorganisms; some theoretical aspects, in *Continuous Cultivation of Microorganisms. A Symposium*, Mĺek, I. Ed., Czechoslovak Academy of Sciences, Prague, 1958, p45.

19 Heinzle, E., Bolzern, O., Dunn, I. J., and Bourne, J. R., A porous membrane-carrier gas measurement system for dissolved gases and volatiles in fermentation systems, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson., C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, p439.

20 Hunt, J., *Ph D. Thesis*, California Institute of Technology, 1980.

21 Imanaka, T., Kaieda, T., Sato, K., and Taguchi, H., Optimization of α-galactosidase production by mold, *J. Ferment. Technol.*, **50**, 633, 1972.

22 Imanaka, T., Kaieda, T., and Taguchi, H., Unsteady-state analysis of a kinetic model for cell growth and α-galactosidase production in mold, *J. Ferment. Technol.*, **51**, 423, 1973.

23 Imanaka, T., Kaieda, T., and Taguchi, H., Optimization of α-galactosidase production in multi-stage continuous culture of mold, *J. Ferment. Technol.*, **51**, 431, 1973.

24 Klein, B., Morgenstern, S., and Kaufman, H., A modified automated ferricyanide-phosphomolybdic acid procedure for serum or plasma glucose, *Clin. Chim. Acta*, **12**, 816, 1966.

25 Koga, S. and Humphrey, A. E., Study of the dynamic behavior of the chemostat system, *Biotech. Bioeng.*, **9**, 375, 1967.

26 Lee, K. J., Tribe, D. E., and Rogers, P. L., Ethanol production by *Zymomonas mobilis* in continuous culture at high glucose concentrations, *Biotechnology Letters*, **1**, 421, 1979.

27 MacDonald, N., Time delays in chemostat models, in *Population Dynamics*, CRC Press, Boca Raton, FL, 1982, chapt 2.

28 May, R. M., *Stability and Complexity in Model Ecosystems*, Princeton Univ. Press, Princeton, 1973.

29 Miller, G.L., Use of dinitrosalicylic acid reagent for determination of reducing sugar, *Anal. Chem.*, **31**, 426, 1959.

30 Oura, E., Effect of aeration intensity on the biochemical composition of baker's yeast. I. Factors affecting the type of metabolism, *Biotech. Bioeng.*, **16**, 1197, 1974.

31 Pirt, S. J., The maintenance energy of bacteria in growing cultures, *Proc. Roy. Soc. London, Ser. B.*, **163**, 224, 1965.

32 Powell, E. O., Transient changes in the growth rate of microorganisms, in *Continuous Cultivation of Microorganisms*, Málek, I., Beran, K., Fencl, Z., Munk, V., Řičia, J., and Smrčková, H., Eds., Adademia, Prague, 1969, p275.

33 San, K.-Y., *Ph D. Thesis*, California Institute of Technology, 1984.

34 San, K.-Y. and Stephanopoulos, G., Studies on on-line bioreactor identification. II. Numerical and experimental studies, *Biotechnol. Bioeng.*, **26**, 1189, 1984.

35 San, K.-Y. and Stephanopoulos, G., Studies on on-line bioreactor identification. IV. Utilization of pH measurements for product estimation, *Biotechnol. Bioeng.*, **26**, 1209, 1984.

36 Shuler, M. L., Leung, S., and Dick, C. C., A mathematical model for the growth of a single bacterial cell, *Annals of the New Youk Academy of Sciences*, **326**, 35, 1979.

37 Stephanopoulos, G. and San, K.-Y., Studies on on-line bioreactor identification. I. Theory, *Biotechnol. Bioeng.*, **26**, 1176, 1984.

38 Tanner, R. D., An enzyme kinetic model for describing fermentation processes, *Biotech. Bioeng.*, **12**, 831, 1970.

39 Thompson, B. G., Kole, M., and Gerson, D. F., Control of ammonium concentration in *Escherichia coli, Biotech. Bioeng.*, **27**, 818, 1985.

40 Wang, N. S. and Stephanopoulos, G., Computer applications to fermentation processes, *CRC Critical Reviews in Biotechnology*, **2**, No. 1, 1, 1984.

41 Yamané, T., Matsuda, M., and Sada, E., Application of porous Teflon tubing method to automatic fed-batch culture of microorganisms. I. Mass transfer through porous Teflon tubing, *Biotech. Bioeng.*, **23**, 2493, 1981.

42 Yamané, T., Matsuda, M., and Sada, E., Application of porous Teflon tubing method to automatic fed-batch culture of microorganisms. II. Automatic constant-value control of fed substrate (ethanol) concentration in semibatch culture of yeast, *Biotech. Bioeng.*, **23**, 2509, 1981.

APPENDIX A

EXTERNAL CONTROL OF PUMP SPEED

A EXTERNAL CONTROL OF PUMP SPEED

Figure A.1 shows the schematic of the MasterFlex pump speed controller (Cole-Parmer). The control signal flow and the current flow that supplies power to the pump motor are indicated in the diagram.

The voltage of the power supply, V_H, and the voltage of the spent current after passing through the DC pump motor, V_L, are extracted and fed into the feedback sensor. The difference in these two voltages represents the loss in the potential, thus, the speed of the pump motor. The feedback sensor converts this difference originally ranging between 0 and 115 volts into a voltage ranging between 0 and 5 volts with respect to the signal ground, identical to the output from the speed setpoint. Notice that the signal ground in this speed controller circuit is 12 volts above the power ground, which is a floating one compared to the true earth ground due to the method employed in obtaining a rectified current source. The comparator subtracts the pump speed voltage from the setpoint voltage. This relatively weak signal is subsequently fed into a follower and converted into a stronger signal capable of driving the subsequent circuit. Finally, this strengthened signal is used to adjust V_L in the ramp capacitor. Because the resistance in the pump motor has a fixed value, varying V_L eventually has the same effect as controlling the speed of the pump.

Because the MasterFlex pump speed controller has a floating power ground and a similarly floating signal ground that are both quite different from the earth ground, directly replacing the speed set point voltage with the the digital-to-analog signals from a computer will create a path for a large current flow, as shown by the dotted line in Figure A.2. Thus, the direct connection will not work, and an interface circuit must be built to achieve common mode rejection and signal isolation between the speed controller and the computer. A direct connection without a

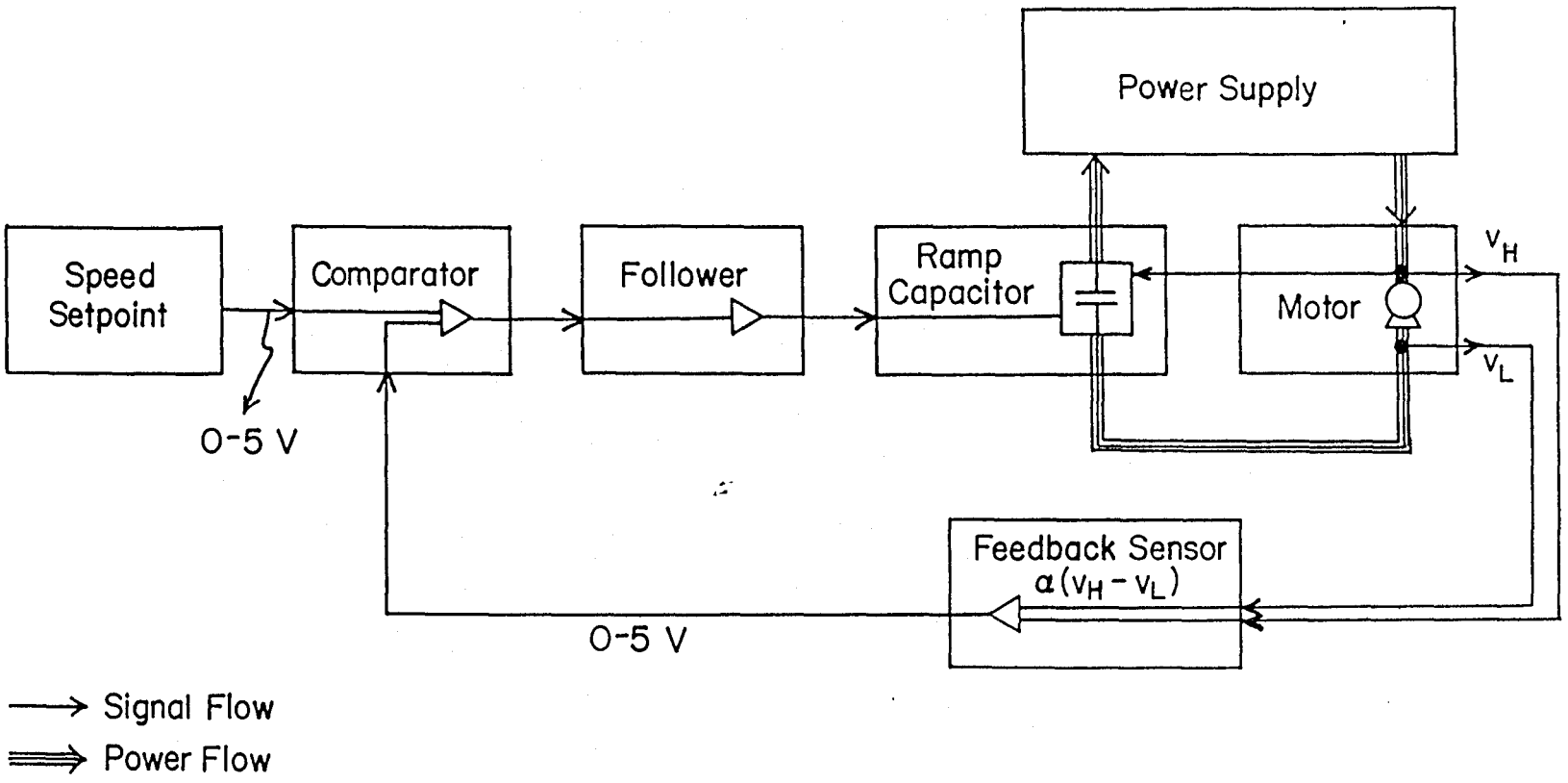


Figure A.1. Block diagram of the MasterFlex pump speed controller.

proper interface circuit will easily overload and burn out the power circuit in the pump speed controller, for the pump speed controller is made to believe that it is receiving a very large setpoint. Similarly, the computer is also in grave danger of being severely damaged, if the D/A board does not have built-in overload protection.

Physical isolation between a computer and an actuating device can be achieved with an isolation amplifier, a transformer, or an opto-isolator, with an opto-isolator being the most compact, convenient, and inexpensive. An opto-isolator is composed of a light emitting diode and a light sensitive transistor pair. The intensity of light emitted from the diode depends on the current passing through it. The emitted light is detected by the light sensitive transistor. The current induced in the transistor in turn depends on the intensity of light it receives. Thus, working in concert, the current in one loop induces current in another loop. The current from one part of the circuit does not physically flow into the second part of the circuit; there is no electrical connection. The signals are connected only by a light transmission, accomplishing the electrical isolation.

Two interface circuits utilizing opto-isolators were built. The first circuit diagram is shown in Figure A.3. In Part I of the circuit, the signal from the computer's D/A, ranging between 0 and 10 volts are converted into 1.01-1.05 volts required by the 4N30 opto-isolator. This conversion is necessary due to the highly nonlinear input-output response of the 4N30 chip. The range of 1.01-1.05 volts represents the narrow linear working limits of the 4N30 chip. This narrow working range also makes the adjustment of the potentiometers somewhat tricky. The first part of this circuit is powered by an external power source of +15V and -15V. Part II of the circuit converts output from the opto-isolator to 0-5 volts required by the speed setpoint. A double-throw switch is installed to change between LOCAL and REMOTE modes of setpoint inputs.

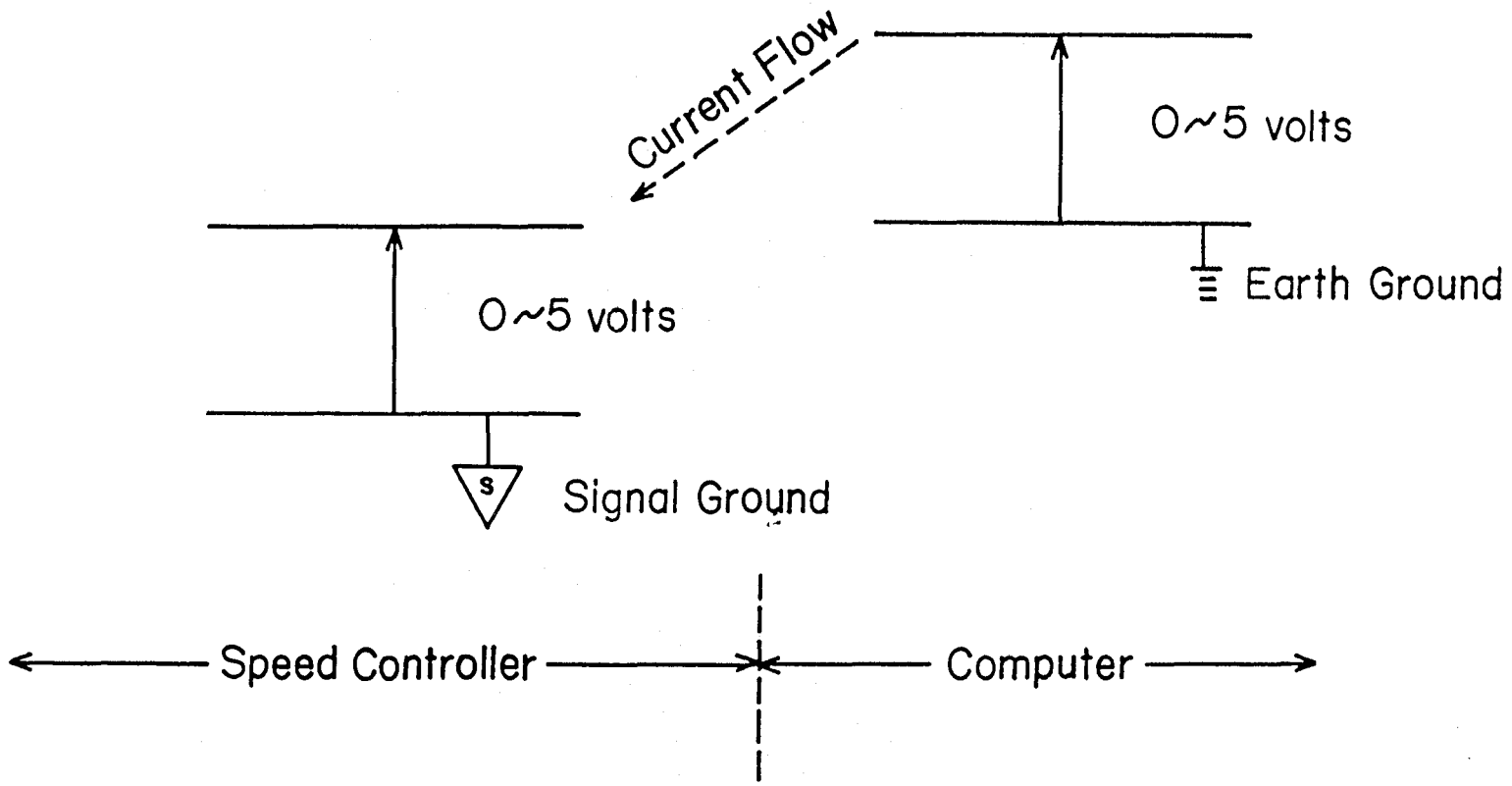


Figure A.2. Potentials are relative quantities. A large current flow can result if two voltages are referenced to different grounds.

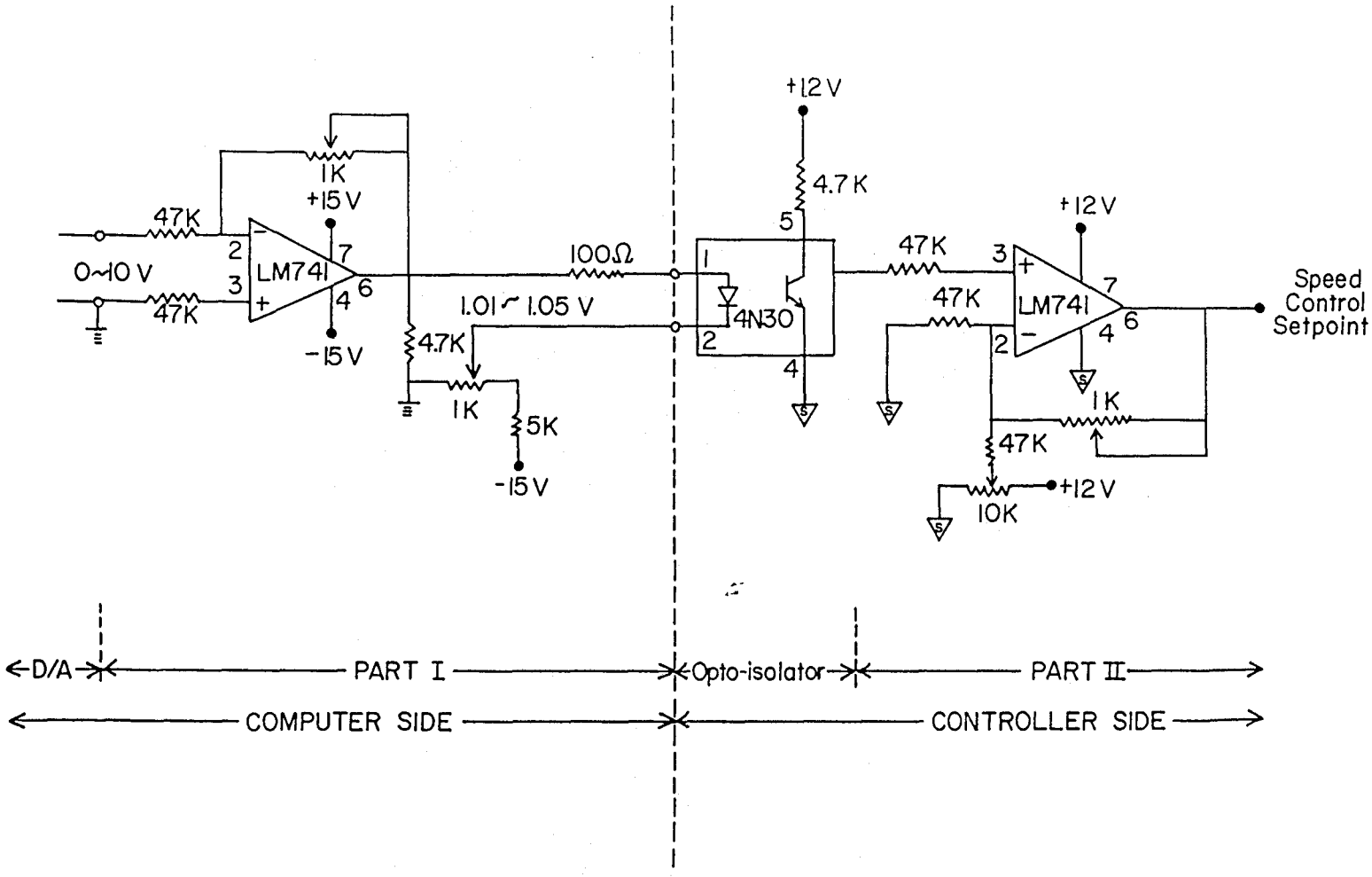


Figure A.3. Interface circuit # 1. Computer and controller are separated by an opto-isolator.

The second interface circuit is shown in Figure A.4. Part a) of the circuit, consisting of two nearly identical parallel parts, is used to duplicate the input current, $i_{1in} = \frac{v_{in}}{R_{in}}$. The current induced between the collector and the base of the light sensitive transistor, i_1, is a function, $f()$ of the input current, *i.e.*,

$$i_1 = f(i_{1in}) = \frac{v_1}{R_1}.$$

Similarly, for the lower half of the circuit:

$$i_2 = f(i_{2in}) = \frac{v_2}{R_2}.$$

The operational amplifiers # 1 and # 2 provide the respective currents i_1 and i_2. The input-output characteristics for the two opto-isolators should almost be the same, and they are also expected to vary hand-in-hand as they are both subjected to the same temperature change or as they age together. The feedback of the # 3 operational amplifier enables one to equate: $v_1 = v_2$. Thus,

$$i_{2in} = f^{-1}\{\frac{R_1}{R_2} f(i_{1in})\}.$$

If R_2 is tuned such that $R_1 = R_2$, then $i_{2in} = i_{1in}$. Finally, the output current, i_{2in}, is provided by the transistor. The overall effect is that the input current is duplicated exactly in a current loop that is electrically isolated from the input, and the entire circuit is made insensitive to the nonlinear characteristics of the opto-isolator.

Part b) of this second circuit simply converts the current into an output signal that can be received by the actuating device. The overall relationship between the input signal, v_{in}, and the output voltage, v_{out}, is expressed by:

$$v_{out} = \frac{R_3 R_4}{R_{in}}(\frac{1}{R_4} + \frac{1}{R_5} + \frac{1}{R_6})v_{in} - \frac{R_4}{R_6} v_4,$$

where v_4 is adjusted by the pot R_7.

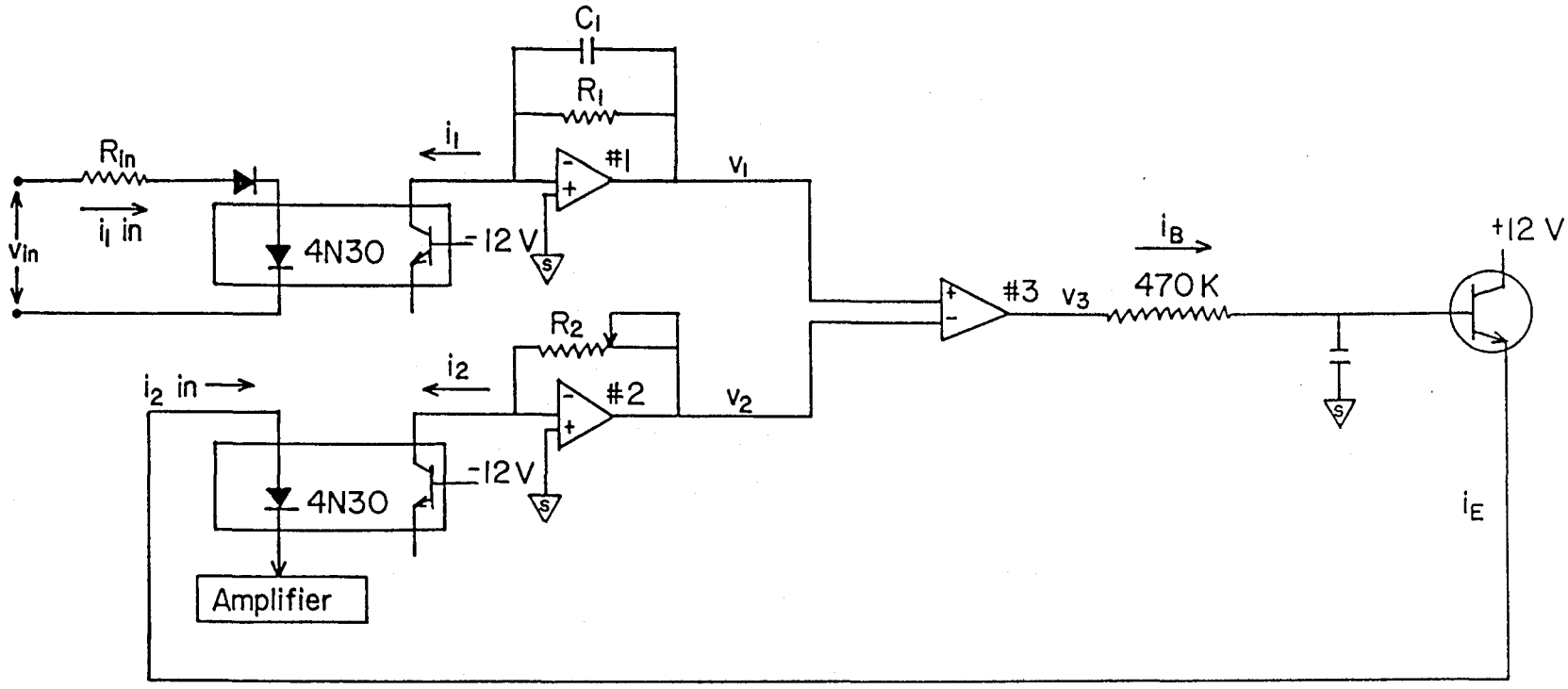


Figure A.4a. Interface circuit # 2. Close duplication of the input current, i_{1in} is achieved in i_{2in} by two nearly identical subcomponents.

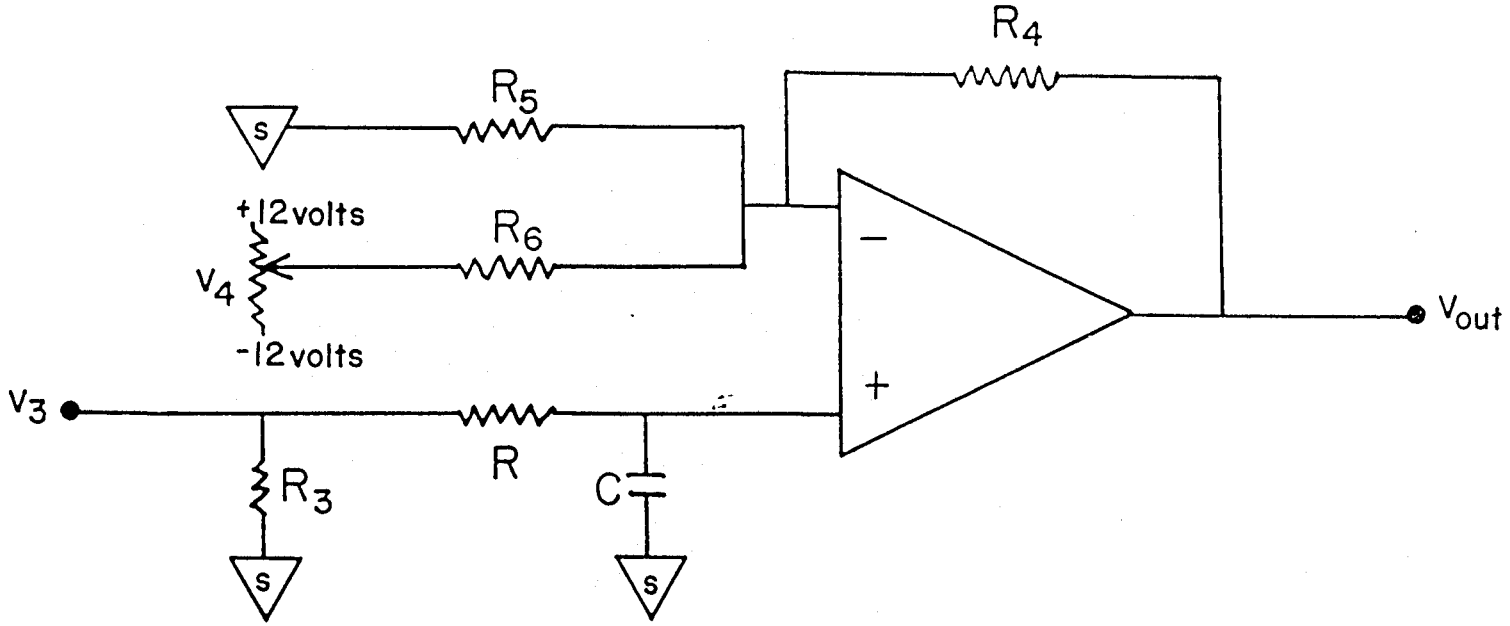


Figure A.4b. A typical signal amplification circuit that replaces the "Amplifier" box in part a).

Although these two isolation interface units were built specifically for the MasterFlex speed pump controllers, the same interface circuit design can be employed repeatedly for many other applications as well. Note that because 4N30 requires 1mA to operate, the critical input voltage is 2.2 volt if R_{in} is 2.2 KΩ. This critical input voltage can be overcome with additional minor modification to the existing circuit. In the final version, a Schmitt trigger and a solid state relay were added to reverse the direction of rotation in a peristaltic pump. The reverse flow was utilized in the programmed periodic backflushing of the continuous cell filtration unit.

APPENDIX B

LISTING

OF THE ON-LINE DATA ACQUISITION PROGRAM

```
(* ************************************************************************ *)
(* ************************************************************************ *)
(* ***** The following program was compiled with MT Plus Pascal        ***** *)
(* ***** compiler (Digital Research) and was one of the programs run   ***** *)
(* ***** during some experiments in which the nutrient flow rate was   ***** *)
(* ***** sinusoidally controlled.                                      ***** *)
(* ************************************************************************ *)
(* ************************************************************************ *)

PROGRAM EXPERIMENT;

CONST PI = 3.1416;

TYPE  STRING9 = ARRAY[1..9] OF CHAR;

VAR   F1,F2,F3  : TEXT;        (* data file identifier *)
      F1_NAME   : STRING;      (* data file name *)
      F2_NAME   : STRING;      (* data file name for base pump activity *)
      F3_NAME   : STRING;      (* data file name for acid pump activity *)
      F_RESULT  : INTEGER;     (* data file *)
      TIME      : ARRAY[0..12] OF BYTE;
      DAYS      : ARRAY[0..6] OF STRING9;
      YEAR      : 1982..1990;
      MONTH     : 1..12;
      DAY       : 1..31;
      WEEKDAY   : 0..6;
      HOUR      : 0..23;
      MINUTE    : 0..59;
      SECOND    : 0..59;
      COMMENT   : STRING;
      PHHIGHA   : BOOLEAN;     (* acid addition ON *)
      PHHIGHB   : BOOLEAN;     (* base addition ON *)
      OLDSEC    : INTEGER;     (* time of last round of sampling *)
      DELTA_T   : INTEGER;     (* data sampling interval *)
      COUNT     : INTEGER;     (* number of sec the pH controller is ON *)
      N_SEC     : INTEGER;     (* the time of pH controller ON/OFF *)
      TAKE      : INTEGER;     (* number of sample to average *)
      I         : INTEGER;     (* ith data point *)
      PHA       : REAL;        (* voltage across the acid ON/OFF relay *)
      PHB       : REAL;        (* voltage across the base ON/OFF relay *)
      VOLTAGE   : REAL;        (* channel voltage *)
      DA_PERIOD : REAL;        (* sine period in # of data points *)
      DA_SHIFT  : REAL;        (* sine phase shift in degree *)
      DA_DEGREE : REAL;        (* sine angle in degree *)
      DA_OMEGA  : REAL;        (* sine angle inradian *)
      DA_VMAX   : REAL;        (* maximum output voltage to channel 0 *)
      DA_VMIN   : REAL;        (* minumum output voltage to channel 0 *)
      DA_VOLTAGE: REAL;        (* voltage to be output to channel 0 *)
      DA_AMP    : REAL;        (* amplitute for the sine wave *)
      DA_BASE   : REAL;        (* base for the sine wave *)
      DA_I      : INTEGER;     (* counter for sine wave *)
      CH0,CH1,CH2,CH3,CH4,CH5,CH6,CH7,CH8  : REAL;(* channel voltage *)
```

```
(* ************************************************************************ *)
(* **** The following function is used to detect the pressing of keyboard *** *)
(* ************************************************************************ *)

EXTERNAL FUNCTION CBDOS(FUN, PARM : INTEGER) : INTEGER;


(* ************************************************************************ *)
(* ***** enter file informations, dates, and comments                 ***** *)
(* ************************************************************************ *)

PROCEDURE  INITIALIZE;

  BEGIN (* begin of procedure INITIALIZE *)
    WRITE('ENTER NAME OF DATAFILE:                    '); READLN(F1_NAME);
    WRITE('ENTER NAME OF PH FILE - BASE:              '); READLN(F2_NAME);
    WRITE('ENTER NAME OF PH FILE - ACID:              '); READLN(F3_NAME);
    WRITELN('ENTER ANY COMMENT ON THIS RUN:           '); READLN(COMMENT);
    WRITE('ENTER NUMBER OF SECONDS BETWEEN SAMPLING: '); READLN(DELTA_T);
    WRITELN('   ANALYSER      CHANNEL  #');
    WRITELN('   OXYGEN           0');
    WRITELN('   CO2              1');
    WRITELN('   PH-HIGH-BASE     2');
    WRITELN('   PRESSURE         3');
    WRITELN('   OD               4');
    WRITELN('   GLUCOSE          5');
    WRITELN('   ETHANOL          6');
    WRITELN('   PH-HIGH-ACID     7')
  END; (* end of procedure INITIALIZE *)


(* ************************************************************************ *)
(* ***** Enter D/A converter information                              ***** *)
(* ************************************************************************ *)

PROCEDURE ASK_DA;

  BEGIN (* begin of procedure ASK_DA *)
    WRITE('ENTER THE MAXIMUM VOLTAGE FOR DA: '); READLN(DA_VMAX);
    WRITE('ENTER THE MINIMUM VOLTAGE FOR DA: '); READLN(DA_VMIN);
    WRITE('ENTER THE PERIOD IN POINTS:       '); READLN(DA_PERIOD);
    WRITE('ENTER THE PHASE SHIFT IN DEGREES: '); READLN(DA_SHIFT);
    DA_AMP  := (DA_VMAX - DA_VMIN)/2.0;
    DA_BASE := (DA_VMAX + DA_VMIN)/2.0;
    DA_I    := 0;
  END; (* end of procedure ASK_DA *)


(* ************************************************************************ *)
(* ***** A/D converter with amplification                             ***** *)
(* ************************************************************************ *)

FUNCTION AD(CHANNEL,AMPLIF : INTEGER) : REAL;
```

```
  CONST     PORT = $A0;                    (* address of the base port *)
            MASK = $20;                    (* completion code for A/D *)

  VAR    LOW_BYTE,HIGH_BYTE : INTEGER;

  BEGIN (* begin of function AD *)
    (* store the amplification factor in the first port *)
      IF AMPLIF=4
      THEN OUT[PORT]:=$02
      ELSE OUT[PORT]:=$00;
    (* store the channel number in the second port *)
    OUT[(PORT+1)]:=CHANNEL;
    (* output 00hex to the third port to trigger A/D conversion *)
    OUT[(PORT+2)]:=$00;
    (* wait for the completion of A/D conversion, as signaled by the first port
       returning a value of MASK *)
    WAIT(PORT,MASK,TRUE);
    (* obtain the result from the third port (low byte) and the fourth port
       (high byte), with a 12-bit resolution, i.e. 0 to 2047 *)
    LOW_BYTE :=INP[(PORT+2)];
    HIGH_BYTE:=INP[(PORT+3)];
    (* The most significant bit contained in the fourth port determines the
       polarity of the voltage *)
    IF HIGH_BYTE > 128
      THEN
        BEGIN
          HIGH_BYTE:=HIGH_BYTE-255;
          LOW_BYTE :=LOW_BYTE -255
        END;
    (* 0 corresponds to 0.0 volt; 2047 corresponds to 5.0 volt *)
    AD:= 5.0*(256.0*HIGH_BYTE+LOW_BYTE)/2047.0
  END; (* end of function AD *)


(* ************************************************************************ *)
(* ***** D/A converter                                                ***** *)
(* ************************************************************************ *)

PROCEDURE DA(VAR VOLTAGE:REAL; CHANNEL:INTEGER);

  CONST     PORT = $C0;                    (* address of the base port *)

  VAR    HIGH_BYTE,LOW_BYTE : INTEGER;
         DA_VOLTAGE         : REAL;

  BEGIN (* begin of procedure DA *)
    (* D/A is the reverse of the A/D listed above *)
    (* The D/A range is set at -10.0 to +10.0 volt *)
    IF VOLTAGE >= 0.0
      THEN
        BEGIN
          DA_VOLTAGE:= VOLTAGE;
          HIGH_BYTE := TRUNC(2047*DA_VOLTAGE/2560);
          LOW_BYTE  := TRUNC(2047*DA_VOLTAGE/10-256*HIGH_BYTE);
```

```
        OUT[(PORT+CHANNEL*2)]   := HIGH_BYTE;
        OUT[(PORT+CHANNEL*2+1)] := LOW_BYTE
      END
    ELSE
      BEGIN
        DA_VOLTAGE:= VOLTAGE + 10.0;
        HIGH_BYTE := TRUNC(2047*DA_VOLTAGE/2560);
        LOW_BYTE  := TRUNC(2047*DA_VOLTAGE/10-256*HIGH_BYTE);
        OUT[(PORT+CHANNEL*2)]   := HIGH_BYTE + 128;
        OUT[(PORT+CHANNEL*2+1)] := LOW_BYTE
      END;
  END; (* end of procedure DA *)


(* ************************************************************************ *)
(* ***** read real time clock RTC-100                                 ***** *)
(* ************************************************************************ *)

PROCEDURE READ_TIME;

  CONST     PORT0 =24;                  (* address of the base port *)
            PORT1 =25;
            PORT2 =26;
            PORT3 =27;

  VAR I : INTEGER;

  BEGIN (* begin of READ REAL-TIME-CLOCK *)
    OUT[PORT1]:=$F0;
    OUT[PORT0]:=$0F;
    OUT[PORT3]:=$FC;
    OUT[PORT1]:=$F4;
    FOR  I:=0 TO 12 DO
      BEGIN
        OUT[PORT0]:=I;
        TIME[I]:=SHR( INP[PORT0], 4)
      END;
    OUT[PORT1]:=$F8;
    OUT[PORT0]:=$0F;
    OUT[PORT3]:=$F8;
    OUT[PORT1]:=$FC;
    OUT[PORT0]:=$0F;
    YEAR:=(TIME[12]*10)+TIME[11];
    MONTH:=((TIME[10] & 3)*10)+TIME[9];
    DAY:=(TIME[8]*10)+TIME[7];
    WEEKDAY:=TIME[6];
    HOUR:=((TIME[5] & 3)*10)+TIME[4];
    MINUTE:=(TIME[3]*10)+TIME[2];
    SECOND:=(TIME[1]*10)+TIME[0]
  END; (* end of READ REAL-TIME-CLOCK *)


(* ************************************************************************ *)
(* ***** read seconds off real time clock RTC-100                     ***** *)
```

```
(* ************************************************************************** *)

PROCEDURE READ_SEC;

  CONST     PORT0 =24;                    (* address of the base port *)
            PORT1 =25;
            PORT2 =26;
            PORT3 =27;

  BEGIN (* begin of READ_SEC *)
    OUT[PORT1]:=$F0;
    OUT[PORT0]:=$0F;
    OUT[PORT3]:=$FC;
    OUT[PORT1]:=$F4;
    OUT[PORT0]:=0;
    TIME[0]:=SHR(INP[PORT0],4);
    OUT[PORT0]:=1;
    TIME[1]:=SHR(INP[PORT0],4);
    OUT[PORT1]:=$F8;
    OUT[PORT0]:=$0F;
    OUT[PORT3]:=$F8;
    OUT[PORT1]:=$FC;
    OUT[PORT0]:=$0F;
    SECOND:=(TIME[1]*10)+TIME[0]
  END; (* end of READ_SEC *)


(* ************************************************************************** *)
(* ************************************************************************** *)
(* ***** take in lab data with channels 0-7 *)                        ***** *)
(* ***** output voltage to channel 0 to control the nutrient pump     ***** *)
(* ************************************************************************** *)
(* ************************************************************************** *)

BEGIN  (* begin of EXPERIMENT *)
  (* read in filename, sampling time, and comments *)
  INITIALIZE;
  ASK_DA;

  (* open data output file and pH output file *)
  ASSIGN(F1,F1_NAME);  REWRITE(F1);
  ASSIGN(F2,F2_NAME);  REWRITE(F2);
  ASSIGN(F3,F3_NAME);  REWRITE(F3);

  (* output comment *)
  WRITELN(F1,COMMENT);

  (* read in the current time and initialize *)
  READ_TIME;
  WRITE(F1,MONTH:2,'/',DAY:2,'/',YEAR:2,'    ');
  WRITELN(F1,HOUR:2,':',MINUTE:2,':',SECOND:2,' ',DELTA_T:3,' seconds/sample');
  PHHIGHB:=FALSE;
  PHHIGHA:=FALSE;
```

```
  (* wait until the initial number is entered *)
  WRITE('WHEN READY, ENTER THE INITIAL NUMBER, (E.G. 0) : '); READLN(I);


(* ************************************************************************ *)
  (* keep taking data until CTRL-C is pressed *)
  REPEAT

    (* ********************************************************************** *)
    (* calculate the sinusoidal voltage with the given amplitude and period *)
    DA_OMEGA   := DA_I/DA_PERIOD + DA_SHIFT/360.0;
    DA_DEGREE  := DA_OMEGA*360.0;
    DA_VOLTAGE := DA_AMP * SIN(2.0*PI*DA_OMEGA) + DA_BASE;

    (* output calculated voltage to D/A converter *)
    DA(DA_VOLTAGE,0);

    (* restart the period when the end of the period is reached *)
    IF DA_I >= DA_PERIOD THEN DA_I:=0;
    DA_I := DA_I+1;

    (* ********************************************************************** *)
    (* initialize at the start of each datum interval *)
    I:=I+1;
    N_SEC:=0;
    TAKE:=0;
    COUNT:=0;

    (* keep reading A/D and averaging throughout the datum interval *)
    WHILE ( DELTA_T > N_SEC ) DO

      BEGIN (* keep taking data and average *)

        (* ******************************************************************* *)
        TAKE:=TAKE+1;
        VOLTAGE:=AD(0,1);  CH0:=(CH0*(TAKE-1)+VOLTAGE)/TAKE;
        VOLTAGE:=AD(1,1);  CH1:=(CH1*(TAKE-1)+VOLTAGE)/TAKE;
        VOLTAGE:=AD(3,1);  CH3:=(CH3*(TAKE-1)+VOLTAGE)/TAKE;
        VOLTAGE:=AD(4,4);  CH4:=(CH4*(TAKE-1)+VOLTAGE)/TAKE;
        VOLTAGE:=AD(5,4);  CH5:=(CH5*(TAKE-1)+VOLTAGE)/TAKE;
        VOLTAGE:=AD(6,4);  CH6:=(CH6*(TAKE-1)+VOLTAGE)/TAKE;

        (* ******************************************************************* *)
        (* read pH controller information once every second *)
        READ_SEC;

        IF SECOND<>OLDSEC THEN

          BEGIN (* if second<>oldsec *)

            (* base addition information *)
            PHB:=AD(2,1);

            (* keep track of the number of second the pH controller is ON *)
```

```
        (* output to a file recording the activity of the base pump *)
        IF PHB > 0.5 THEN
          BEGIN
            COUNT:=COUNT+1;
            IF NOT PHHIGHB THEN
              BEGIN
                WRITE(F2,I:6,N_SEC:3,' -');
                PHHIGHB:=TRUE
              END
          END
        ELSE IF PHHIGHB THEN
              BEGIN
                WRITELN(F2,I:6,N_SEC:3);
                PHHIGHB:=FALSE
              END;

        (* acid addition information *)
        PHA:=AD(7,1);

        (* keep track of the number of second the pH controller is ON *)
        (* output to a file recording the activity of the acid pump *)
        IF PHA > 0.5 THEN
          BEGIN
            COUNT:=COUNT+1;
            IF NOT PHHIGHA THEN
              BEGIN
                WRITE(F3,I:6,N_SEC:3,' -');
                PHHIGHA:=TRUE
              END
          END
        ELSE IF PHHIGHA THEN
              BEGIN
                WRITELN(F3,I:6,N_SEC:3);
                PHHIGHA:=FALSE
              END;

        OLDSEC:=SECOND;
        N_SEC:=N_SEC+1

      END (* if second<>oldsec then *)

  END; (* keep taking data and average *)


(* ************************************************************************ *)
(* convert the OD and glucose voltage to absorbance *)
CH4:=CH4/4;
CH4:=-LN(CH4)/2.303;
CH5:=CH5/4;
CH5:=-LN(CH5)/2.303;

(* output data both to data files and to screen in column format *)
(* 1st column ... oxygen          *)
(* 2nd column ... carbon dioxide *)
```

```
    (* 3rd column ... pH controller  *)
    (* 4th column ... pressure       *)
    (* 5th column ... biomass        *)
    (* 6th column ... glucose        *)
    (* 7th column ... ethanol        *)

WRITELN(F1,I:6,CH0:7:3,CH1:7:3,COUNT:4,CH3:7:3,CH4:7:3,CH5:7:3,CH6:7:3,DA_VOLTAGE:7:3);
    WRITELN(
I:6,CH0:7:3,CH1:7:3,COUNT:4,CH3:7:3,CH4:7:3,CH5:7:3,CH6:7:3,DA_DEGREE:8:3);

    (* CTRL-A resets the values for the D/A converter *)
    IF @BDOS(06,255) = $01 THEN ASK_DA;


  (* keep taking data until CTRL-C is pressed *)
  UNTIL @BDOS(06,255)=$03; (* CTRL-C terminates data taking *)

  CLOSE(F1,F_RESULT);
  CLOSE(F2,F_RESULT);
  CLOSE(F3,F_RESULT);

END. (* end of EXPERIMENT *)

(* ************************************************************************** *)
(* ************************************************************************** *)
```

APPENDIX C

LISTING
OF THE DATA ANALYSIS CODES

```
c ***************************************************************************
c ***************************************************************************
c These programs are used to convert the raw laboratory data into
c   more meaningful information.
c     REGRESS ... regression analysis, used to obtain calibration curves.
c     BIOMASS ... calculate the biomass concentration from lab data.
c     GLUCOSE ... calculate the glucose concentration from lab data.
c     ETHANOL ... calculate the ethanol concentration from lab data.
c     CEROUR  ... calculate CER, OUR, and RQ from lab data.
c     PH      ... calculate the ammonia consumption rate, RNH3
c     BALANCE ... calculate yield coefficients based on CER, OUR, and RNH3
c     KALMAN  ... Kalman filter
c ***************************************************************************
c ***************************************************************************



c ***************************************************************************
c ***************************************************************************
c This program finds the leastsquare fit of a set of points to power expansions.
c   n = max number of independent measurements
c   ipt = number of independent measurements
c   k = max order of fit = kk-3
c   mdp(1) = order of fit = iorder - 1
c   eresp = absolute weight (Y/n)
c   mresp = more points to read (Y/n)
c   wresp = read weight from the same file (y/N)
c   xresp = write x to out.out file (y/N)
c   cresp = center the analysis (y/N)
c The least-squaree cureve at x is (right after the exit from RLFOR) ...
c   y(x) = b(iorder,2) + b(1,2)*x + b(2,2)*x**2 + b(3,2)*x**3
c          + ... + b(iorder-1,2)*x**iorder
c To increase accuracy, perform analysis with an artificial center.
c Link with IMSL's RLFOR in REGRSS.LIB
c ***************************************************************************
c ***************************************************************************
      PROGRAM REGRESS
      parameter (n=500,k=7,kk=10)
      character fname*40,resp*1,eresp*1,mresp*1,wresp*1,cresp*1,xresp*1
      dimension xyw(n,kk),mdp(3),albp(2),anova(13),b(kk,12),pred(1,1)
      real*8 wk(2000)

c ***************************************************************************
c Read in the data of x, y, and weight
c ***************************************************************************
      write(*,600)' assign absolute weight  (Y/n) '
      read(*,500)eresp
      write(*,600)' read data from a file  (Y/n)'
      read(*,500)resp

c ***************************************************************************
c Read data from terminal
c ***************************************************************************
      if(resp .eq. 'n' .or. resp .eq. 'N')then
```

```
        write(*,600)' enter independent & dependent variables & weight:'
        write(*,600)
        do 5 i=1,n
          write(*,651)i
651       format(' x(',i2,')= ')
          read(*,502)xyw(i,1)
          write(*,652)i
652       format(' y(',i2,')= ')
          read(*,502)xyw(i,2)
          if(eresp .eq. 'n' .or. eresp .eq. 'N')then
            write(*,600)' enter the % error: '
            read(*,502)error
            xyw(i,3)=xyw(i,2)*xyw(i,2)*error*error
          else
            xyw(i,3)=1.
          endif
          write(*,600)' more data  (Y/n)'
          read(*,500)mresp

          if(mresp .eq. 'n' .or. mresp .eq. 'N')goto 6
5       continue
6       ipt=i
        write(*,654)ipt
654     format(' Number of points read = ',i5)
        write(*,600)

c ***************************************************************************
c Read data from a file
c ***************************************************************************
      else
        write(*,600)' enter filename: '
        read(*,500)fname
        open(1,file=fname,status='old')
        write(*,600)' read weight from the same file  (y/N)'
        read(*,500)wresp
        if(wresp .eq. 'y' .or. wresp .eq. 'Y')then
          do 10 i=1,n
            read(1,*,end=11)xyw(i,1),xyw(i,2),xyw(i,3)
            xyw(i,3)=xyw(i,3)*xyw(i,3)
10        continue
11        close(1)
        else
          if(eresp .eq. 'n' .or. eresp .eq. 'N')then
            write(*,600)' enter the % error '
            read(*,502)error
          endif
          do 20 i=1,n
            read(1,*,end=21)xyw(i,1),xyw(i,2)
            if(eresp .eq. 'n' .or. eresp .eq. 'N')then
              xyw(i,3)=xyw(i,2)*xyw(i,2)*error*error
            else
              xyw(i,3)=1.
            endif
20          continue
```

```
21        close(1)
        endif
        ipt=i-1
        write(*,654)ipt
        write(*,600)
      endif

c ***************************************************************************
c Center the analysis
c ***************************************************************************
      write(*,600)' want to shift the center for calculation  (y/N)'
      read(*,500)cresp
      if(cresp .eq. 'y' .or. cresp .eq. 'Y')then
        write(*,600)' enter the center for x: '
        read(*,502)xcent
        do 30 i=1,ipt
          xyw(i,1)=xyw(i,1)-xcent
30      continue
      endif

c ***************************************************************************
c Find the least square estimate
c ***************************************************************************
      mdp(3)=0
      albp(1)=0.05
      write(*,600)' enter the order of fit: '
      read(*,501)iorder
      mdp(1)=iorder - 1
      call rlfor(xyw,n,ipt,100.,mdp,albp,anova,b,kk,pred,1,wk,ier)
c ***************************************************************************
c Shift the coefficient given by RLFOR subroutine so that the intercept is the
c   1st element in b(i,2).
c ***************************************************************************
      temp=b(iorder,2)
      do 40 i=iorder,2,-1
40      b(i,2)=b(i-1,2)
      b(1,2)=temp

      write(*,600)
      write(*,600)' the least square coefficients: '
      write(*,600)
      write(*,602)(b(i,2),i=1,iorder)
      write(*,600)

c ***************************************************************************
c Generate a yy vs. xx curve
c ***************************************************************************
      open(2,file='out.out',status='new')
      write(*,600)' enter number of points to be generated in output: '
      read(*,501)npt
      write(*,600)' enter the starting x-value: '
      read(*,502)xstart
      write(*,600)' enter the step size: '
      read(*,502)h
```

```
      write(*,600)' want to write x values  (y/N) '
      read(*,500)xresp

      do 60 j=1,npt
        xx=float(j)*h+xstart
c ***************************************************************************
c Generate a polynomial
c ***************************************************************************
        yy=b(iorder,2)
        do 65 ii=1,iorder-1
65        yy=b(iorder-ii,2)+xx*yy

        if(xresp .eq. 'y' .or. xresp .eq. 'Y')then
          write(2,602)xx,yy
        else
          write(2,602)yy
        endif
60    continue

      close(2)

c ***************************************************************************
c Standard formats
c ***************************************************************************
500   format(a)
501   format(i4)
502   format(e13.5)
600   format(a\)
602   format(1pe13.5,9e13.5)
      stop 'goodbye'
      end



c ***************************************************************************
c ***************************************************************************
c This program reads in on-line OD readings & convert them to
c   biomass concentration (g/l) based on a calibration curve
c ***************************************************************************
c ***************************************************************************
      PROGRAM BIOMASS
      character fname*40,resp*1

c ***************************************************************************
c Calibration curve to convert OD flow cell absorbance to dry wt
c ***************************************************************************
      write(*,600)' enter calibration curve -- a0: '
      read(*,502)a0
      write(*,600)' enter calibration curve -- a1: '
      read(*,502)a1
      write(*,600)' enter calibration curve -- a2: '
      read(*,502)a2

c ***************************************************************************
```

```
c Enter file information
c ***************************************************************************
      open(2,file='out.bio',status='new')
100   write(*,600)' enter filename: '
      read(*,500)fname
      open(1,file=fname,status='old')

      do 10 i=1,9999
        read(1,550,end=50)OD
550   format(31x,f7.3)

c ***************************************************************************
c Convert the on-line OD reading to biomass dry weight
c ***************************************************************************
        b=a0+a1*OD+a2*OD*OD
        write(2,604)b
10    continue

50    close(1)
      write(*,600)' more files  (y/N) '
      read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')goto 100

      close(2)

c ***************************************************************************
c Standard formats
c ***************************************************************************
500   format(a)
501   format(i4)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
      stop 'goodbye'
      end



c ***************************************************************************
c ***************************************************************************
c Read in the lab data on glucose analyzer & calculate glucose conc. (g/l)
c ***************************************************************************
c ***************************************************************************
      PROGRAM GLUCOSE
      character fname*40,resp*1

      write(*,600)' enter the number of points of delay in analyzer: '
      read(*,501)idelay
      write(*,600)' enter the background level of absorbance signal: '
      read(*,502)backgd
      write(*,600)' enter the absorbance per g/l of Fe+3 conc.: '
      read(*,502)slope
      write(*,600)' enter g of Fe+3 to react with 1g of glucose: '
      read(*,502)react
```

```
      write(*,600)' enter the dilution factor of the glucose analyzer: '
      read(*,502)factor
      iend=0
      ibegin=1

c ***************************************************************************
c Enter file information
c ***************************************************************************
      open(2,file='out.out',status='new')
      write(*,600)' enter filename: '
      read(*,500)fname
      open(1,file=fname,status='old')
      do 5 i=1,idelay
        read(1,650,end=50)x
650     format(38x,f7.3)
5     continue

      iend=0
      ibegin=1

200   do 10 i=ibegin,iend
        read(1,650,end=50)x
c ***************************************************************************
c Generate glucose conc.
c ***************************************************************************
        x=(x-backgd)/slope
        y=(Fefeed*(factor-1.)-x*factor)/react
        write(2,604)y
10    continue

c ***************************************************************************
c Enter points of change in the feed Fe+3 conc.
c ***************************************************************************
      write(*,600)' enter the feed Fe+3 conc: '
      read(*,502)Fefeed
      write(*,600)' next point of switch in the Fe+3 feed: '
      read(*,501)iend
      ibegin=i
      goto 200

50    close(1)
      ibegin=i
      write(*,600)' more files  (y/N) '
      read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')then
        write(*,600)' enter filename: '
        read(*,500)fname
        open(1,file=fname,status='old')
        goto 200
      endif

      do 60 i=1,idelay
60    write(2,604)y
      close(2)
```

```
c ***************************************************************************
c Standard formats
c ***************************************************************************
500   format(a)
501   format(i4)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
      stop 'goodbye'
      end


c ***************************************************************************
c ***************************************************************************
c Read in the lab data on GC voltage and gegerate ethanol conc., given the
c   coefficients to a piecewise polynomial calibration curve.
c ***************************************************************************
c ***************************************************************************
      PROGRAM ETHANOL
      character fname*40,resp*1
      dimension a(15)

      iend=0
      ibegin=1
      write(*,600)' enter the number of point of delay: '
      read(*,501)idelay

c ***************************************************************************
c Enter file information
c ***************************************************************************
      open(2,file='out.out',status='new')
100   write(*,600)' enter filename: '
      read(*,500)fname
      open(1,file=fname,status='old')

200   do 10 i=ibegin,iend
        read(1,650,end=50)x
650     format(45x,f7.3)
        if(i .le. idelay)goto 10
c ***************************************************************************
c Generate a polynomial
c ***************************************************************************
        y=a(iorder)
        do 20 ii=1,iorder-1
20        y=a(iorder-ii)+x*y
        write(2,604)y
10    continue

c ***************************************************************************
c Enter coefficients
c ***************************************************************************
      write(*,600)' enter the number of coefficients: '
```

```
      read(*,501)iorder
      do 5 ii=1,iorder
        write(*,654)ii
654     format(' a',i2,': ')
5       read(*,502)a(ii)
      write(*,600)' next point of switch in polynomial: '
      read(*,501)iend
      ibegin=i
      goto 200

50    close(1)
      ibegin=i
      write(*,600)' more files  (y/N) '
      read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')goto 100

      do 60 i=1,idelay
60    write(2,604)y
      close(2)

c *************************************************************************
c Standard formats
c *************************************************************************
500   format(a)
501   format(i4)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
      stop 'goodbye'
      end



c *************************************************************************
c *************************************************************************
c This program calculate CER & OUR & RQ from .dat files in which the volume
c   is either varying due to the continuous withdrawal of samples or
c   piecewise constant.
c The results are output to out.cer and out.our & out.rq files.
c The background effect is fully implemented in O2 calculation.
c The pressrue effect is considered in O2 calculation.
c *************************************************************************
c *************************************************************************
      PROGRAM CEROUR
      dimension CO2cai(3),CO2caf(3)
      character fname*40,resp*1

c *************************************************************************
c Initialization
c *************************************************************************
c     backgound CO2 conc (in %)
      CO2bak=0.033
c     conc. of O2 in air & O2 in the low-range calibration gas (in %)
      O2air =20.946
```

```
      O2calb=19.225

      write(*,600)' enter air flow rate (liter/min): '
      read(*,502)Qin
c     convert into mole/hr
      Qin=Qin*2.39557

c *****************************************************************************
c CO2 calibration
c *****************************************************************************
      write(*,600)' use standard initial cal. curve for CO2  (Y/n) '
      read(*,500)resp
      if(resp .eq. 'n' .or. resp .eq. 'N')then
        write(*,600)' enter the initial calibration curve: '
        write(*,600)
        do 5 i=1,3
          write(*,654)i
654       format(' CO2C',i2,': ')
          read(*,502)CO2cai(i)
5       continue
      else
        CO2cai(1)=0.
        CO2cai(2)=0.29013
        CO2cai(3)=4.67739e-2
      endif
      write(*,600)' enter the final calibration curve: '
      write(*,600)
      do 6 i=1,3
        write(*,654)i
        read(*,502)CO2caf(i)
6     continue

c *****************************************************************************
c O2 calibration
c *****************************************************************************
      EHI=20.6630
      ELO=18.9358
      write(*,600)' enter initial O2 cal. for 19,225% (volt): '
      read(*,502)O2Lcai
      write(*,600)' enter initial O2 cal. for air     (volt): '
      read(*,502)O2Hcai
      write(*,600)' enter initial cal. pressure       (volt): '
      read(*,502)Pcai
      write(*,600)' enter final   O2 cal. for 19,225% (volt): '
      read(*,502)O2Lcaf
      write(*,600)' enter final   O2 cal. for air     (volt): '
      read(*,502)O2Hcaf
      write(*,600)' enter final   cal. pressure       (volt): '
      read(*,502)Pcaf
c     convert the final calibration to the same basis as the initial pressure
      temp=(O2Hcaf-O2Lcaf)/(O2air-O2calb)/Pcaf*(Pcai-Pcaf)
      O2Lcaf=O2Lcaf+temp*O2calb
      O2Hcaf=O2Hcaf+temp*O2air
```

```
c ***************************************************************************
c Enter volume information
c ***************************************************************************
      write(*,600)' enter total pts for this run: '
      read(*,501)ntotal
      write(*,600)' enter rate of cont. volume withdrawal (liter/hr): '
      read(*,502)Rvol
      Rvol=Rvol/60.
      write(*,600)' enter initial volume of fermentor (liter): '
      read(*,502)Vi
      Vsample=0.
      write(*,600)' next point of sampling: '
      read(*,501)iend
      ibegin=1

c ***************************************************************************
c Enter file information
c ***************************************************************************
      open(2,file='out.cer',status='new')
      open(3,file='out.our',status='new')
      open(4,file='out.rq',status='new')

100   write(*,600)' enter filename: '
      read(*,500)fname
      open(1,file=fname,status='old')

200   do 10 i=ibegin,iend

        read(1,550,end=50)O2v,CO2v,Pv
550     format(6x,f7.3,f7.3,4x,f7.3)
        weight = float(i)/float(ntotal)

c ***************************************************************************
c Convert CO2v (in volt) to CO2per (in %)
c ***************************************************************************
        CO2i = CO2cai(1) + CO2cai(2)*CO2v + CO2cai(3)*CO2v*CO2v
        CO2f = CO2caf(1) + CO2caf(2)*CO2v + CO2caf(3)*CO2v*CO2v
        CO2per = CO2i*(1.-weight) + CO2f*weight

c ***************************************************************************
c Convert O2v (in volt) to O2per (in %)
c       pressure correction is included
c ***************************************************************************
        O2v=O2v+13.16*(Pcai-Pv)

c       convert O2v (in volt) to O2per (in %)
        O2i=(O2v-O2Lcai)*(EHI-ELO)/(O2Hcai-O2Lcai)
          O2i=O2i+ELO+0.00265*CO2per+0.358
          O2i=O2i/1.00358
        O2f=(O2v-O2Lcaf)*(EHI-ELO)/(O2Hcaf-O2Lcaf)
          O2f=O2f+ELO+0.00265*CO2per+0.358
          O2f=O2f/1.00358
        O2per = O2i*(1.-weight) + O2f*weight
```

```
c ***************************************************************************
c Calculate Qout
c ***************************************************************************
        Qout = Qin*79.021/(100.-CO2per-O2per)

c ***************************************************************************
c Calculate CER & OUR (in mole/hr-liter)
c ***************************************************************************
        volume=Vi-Rvol*float(i)-Vsample
        CER = (Qout*CO2per-Qin*CO2bak)/100./volume
        OUR = (Qin*O2air-Qout*O2per)/100./volume
        RQ  = CER/OUR

c ***************************************************************************
c Output CER & OUR & RQ
c ***************************************************************************
        write(2,604)CER
        write(3,604)OUR
        write(4,604)RQ
10      continue

        write(*,600)' enter the cumulative volume withdrawn (liter): '
        read(*,502)Vsample
        write(*,600)' next point of sample: '
        read(*,501)iend
        ibegin=i
        goto 200

50      close(1)
        ibegin=i
        write(*,600)' more files  (y/N) '
        read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')goto 100

        close(2)
        close(3)
        close(4)

c ***************************************************************************
c Standard formats
c ***************************************************************************
500     format(a)
501     format(i6)
502     format(e13.5)
600     format(a\)
604     format(1pe13.5)
        stop 'goodbye'
        end



c ***************************************************************************
c ***************************************************************************
c This program calculates the rate of base addition based on ON-OFF information
```

```
c The method is based on San's paper
c
c CONTINUOUS MODE
c   cR = Ra*(1-exp(-D*dt2))/(1-exp(-D*dt))    B&B 26, p1212, eqn (19)
c
c FEDBATCH MODE
c          V0*dt2 + Fpsec*(t2*t2-t1*t1)/2.
c     cR = ------------------------------    B&B 26, p1213, eqn (26)
c          V0*dt  + Fpsec*(t2*t2-t0*t0)/2.
c
c BATCH MODE
c     cR = Ra*dt2/dt                         B&B 26, p1212, eqn (20)
c
c     where dt2 = interval in which pH controller is ON-OFF
c           dt  = interval between ON-ON = dt1+dt2
c     ion, ionold = cumulative sec at which pH controller is turned ON
c     ionmin, ionsec = min & sec at which pH controller is turned ON
c     iof = cumulative sec at which pH controller is turned OFF
c     iofmin, ioffsec = min & sec at which pH controller is turned OFF
c     itotal = the total number of seconds pH controller is on
c ***************************************************************************
c ***************************************************************************
      PROGRAM PH
      dimension dt2(5000),dt(5000)
      character mode*1,fname*40,resp*1

c ***************************************************************************
c Initialization
c ***************************************************************************
      lastpt=0
      dt2ave=0.
      dtave=0.
      nave=0
      ionold=0
      itotal=0

      write(*,600)' mode of operation (c=cont.,b=batch,f=fedbatch: '
      read(*,500)mode
      write(*,600)' enter the total pts for this run: '
      read(*,501)ntotal
      write(*,600)' enter number of pts to average:          '
      read(*,501)na
      write(*,600)' enter rate of base addition (moles/hr): '
      read(*,502)Ra
      if(mode .eq. 'c' .or. mode .eq. 'C')then
        write(*,600)' enter volume of fermentor (liter):       '
        read(*,502)volume
        Ra=Ra/volume
      endif

c ***************************************************************************
c Enter file information
c ***************************************************************************
      open(2,file='out.ph',status='new')
```

```
100   write(*,600)' enter filename: '
      read(*,500)fname
      open(1,file=fname,status='old')

c ***************************************************************************
c ******************** main data analysis loop, PART I **********************
c ***************************************************************************

c ***************************************************************************
c Read in the pH controller ON/OFF data
c ***************************************************************************
200   read(1,551,end=50)ionmin,ionsec,iofmin,iofsec
551   format(2x,i4,1x,i2,4x,i4,1x,i2)

c ***************************************************************************
c Calculate the intervals
c ***************************************************************************
      ion=ionmin*60+ionsec
      iof=iofmin*60+iofsec
      dt2temp=float(iof-ion)
      dttemp=float(ion-ionold)
      itotal=itotal+iof-ion
c     lastpt is the last recorded point
c     ionmin eq lastpt+1=current point means that the last ionmin was also the
c     same as the current ionmin
      if(ionmin .eq. lastpt+1)then
        dt2ave=(dt2ave*float(nave)+dt2temp)/float(nave+1)
        dtave=(dtave*float(nave)+dttemp)/float(nave+1)
        nave=nave+1
      else
c     assign dt and dt2 up to but not including the present ionmin
150     lastpt=lastpt+1
        dt2(lastpt)=dt2ave
        dt(lastpt)=dtave
        dt2ave=dt2temp
        dtave=dttemp
        nave=1
        if(ionmin .gt. lastpt+1)goto 150
      endif
      ionold=ion
      goto 200

c ***************************************************************************
c ***************** End of the main data analysis loop, PART I **************
c ***************************************************************************

c ***************************************************************************
c Reset data file
c ***************************************************************************
50    close(1)
      write(*,600)' more files  (y/N) '
      read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')goto 100
```

```
c ************************************************************************
c Fill up the rest of data
c ************************************************************************
      do 60 i=lastpt+1,ntotal
        dt(i) =dtave
60      dt2(i)=dt2ave
      write(*,553)itotal
553   format(' total number of sec pH controller is ON:        ',i6)
      write(*,*)

      if(mode .eq. 'f' .or. mode .eq. 'F')then
c       enter volume information ***********************************************
        write(*,600)' enter the rate of filtrate removal (liter/hr):'
        read(*,502)Fphr
        Fpsec=Fphr/3600.
        Fpmin=Fphr/60.
        write(*,600)' enter initial volume of fermentor (liter): '
        read(*,502)Vinit
      endif

      ibegin=1
      iend=0

      dt2ave=0.
      dtave=0.

c ************************************************************************
c ******************** main data analysis loop, PART II ******************
c ************************************************************************

300   do 10 i=ibegin,iend

c ************************************************************************
c Take the running average
c ************************************************************************
        if(i .le. na)dt2ave=dt2ave+(dt2(i)-dt2ave)/float(i)
        if(i .le. na)dtave=dtave+(dt(i)-dtave)/float(i)
        if(i .gt. na)dt2ave=dt2ave+(dt2(i)-dt2(i-na))/float(na)
        if(i .gt. na)dtave=dtave+(dt(i)-dt(i-na))/float(na)

c       the output is shifted by na/2 to encounter the average effect
        if(i .le. na/2)goto 10

        if(mode .eq. 'f' .or. mode .eq. 'F')then
c             V0*dt2 + Fpsec*(t2*t2-t1*t1)/2.
c         cR = ------------------------------
c             V0*dt  + Fpsec*(t2*t2-t0*t0)/2.
          volume=Vinit-Fpmin*float(i)-Vsample
          up = volume*dt2ave - Fpsec*dt2ave*(2.*dtave-dt2ave)/2.
          dn = volume*dtave  - Fpsec*(dtave*dtave)/2.
          cR = Ra/volume*up/dn
        endif
        if(mode .eq. 'b' .or. mode .eq. 'B')then
          cR = Ra*dt2ave/dtave/volume
```

```
        endif
        if(mode .eq. 'c' .or. mode .eq. 'C')then
          cR = Ra*(1.-exp(-D*dt2ave))/(1.-exp(-D*dtave))
        endif

        write(2,604)cR
10    continue

c ***************************************************************************
c *************** End of the main data analysis loop, PART II ***************
c ***************************************************************************

c ***************************************************************************
c Re-enter dilution rate, volume of fermentor, and sample withdrawn.
c ***************************************************************************
      if(iend .eq. ntotal)goto 15
      if(mode .eq. 'f' .or. mode .eq. 'F')then
        write(*,600)' enter cumulative vol. taken as sample: (liter)'
        read(*,502)Vsample
        write(*,600)' enter next point of sampling: '
      endif
      if(mode .eq. 'b' .or. mode .eq. 'B')then
        write(*,600)' enter volume of fermentor (liter): '
        read(*,502)volume
        write(*,600)' enter next point of sampling: '
      endif
      if(mode .eq. 'c' .or. mode .eq. 'C')then
        write(*,600)' enter the dilution rate in (hr-1):
        read(*,502)D
        D=D/3600.
        write(*,600)' up to which pt is this dilution rate valid
      endif
      read(*,501)iend
      ibegin=i
      if(iend .gt. ntotal)iend=ntotal
      goto 300

c ***************************************************************************
c Flush out the rest
c ***************************************************************************
15    do 20 i=1,na/2
20      write(2,604)cR
      close(2)

c ***************************************************************************
c Standard formats
c ***************************************************************************
500   format(a)
501   format(i4)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
      stop 'goodbye'
      end
```

```
c ***************************************************************************
c ***************************************************************************
c This read in data from NH3 & CER & OUR files.
c It calculates the total rate of growth (R) and yield coefficients (Ys & Yp).
c The .NH3 file is actually the net rate of OH- addition. (Both base and acid
c addition rates are combined.)
c
c Glucose ---> biomass + ethanol
c     Glucose ... substrate
c     Ethanol ... product
c
c Ethanol ---> biomass + Acetic acid
c     Ethanol ...     substrate
c     Acetic acid ... product
c Rnh3 = (c+FF)R
c
c The INVERSE of the real yield is output to .Ysm and Ypm files
c ***************************************************************************
c ***************************************************************************
      PROGRAM BALANCE
      character carbon*1,fname1*40,fname2*40,fname3*40

c ***************************************************************************
c Initialization
c ***************************************************************************
      write(*,600)' Carbon source (g/E): (g=glucose, e=ethanol): '
      read(*,500)carbon
      write(*,600)' enter biomass formula: '
      write(*,600)
      write(*,600)' H: '
      read(*,502)beta
      write(*,600)' O: '
      read(*,502)gamma
      write(*,600)' N: '
      read(*,502)delta
      write(*,600)' enter the ash content in %: '
      read(*,502)ash
c ***************************************************************************
c Calculate molecular weight of biomass
c ***************************************************************************
      Wbio = 12. + beta + 16.*gamma + 14.*delta
      Wbio = Wbio/(1.-ash/100.)
c ***************************************************************************
c Calculate the "degree of reductance"
c ***************************************************************************
      if(carbon .eq. 'g' .or. carbon .eq. 'G')then
        z = ( beta - 2.*gamma - 3.*delta ) / 4.
      else
        z = ( beta - 2.*gamma - 3.*delta -2.) / 4.
      endif
```

```
c ***************************************************************************
c Enter file information
c ***************************************************************************
      write(*,600)' enter filename containing CER: '
      read(*,500)fname1
      open(1,file=fname1,status='old')
      write(*,600)' enter filename containing OUR: '
      read(*,500)fname2
      open(2,file=fname2,status='old')
      write(*,600)' enter filename containing Rnh3: '
      read(*,500)fname3
      open(3,file=fname3,status='old')
      open(11,file='out.Ysm',status='new')
      open(12,file='out.Ypm',status='new')
      open(13,file='out.Rm',status='new')

      ibegin=1
      iend=0
100   do 10 i=ibegin,iend
        read(1,*,end=50)CER
        read(2,*,end=50)OUR
        read(3,*,end=50)Rnh3

c ***************************************************************************
c Calculate the stoichiometric coefficients & the yield coefficients
c ***************************************************************************
        if(carbon .eq. 'g' .or. carbon .eq. 'G')then
          f = delta*(CER-OUR)/Rnh3 - z
          e = delta*CER/Rnh3
          a = (1.+e+2.*f)/6.
          Ys = a*180./Wbio
          Yp = f*46./Wbio
          R = Rnh3/delta*Wbio
        else
          temp1 = (OUR-1.5*CER)/Rnh3
          temp2 = (delta*temp1 + z)
          f = temp2/(1.-FF*temp1)
          R = Rnh3/(delta+FF*f)
          e = CER/R
          a = (1.+e+2.*f)/2.
          Ys = a*46./Wbio
          Yp = f*60./Wbio
          R = R*Wbio
        endif

c ***************************************************************************
c Output Ys & Yp
c ***************************************************************************
        write(11,604)Ys
        write(12,604)Yp
        write(13,604)R
10    continue
      write(*,600)' enter the correction factor of pH addition: '
      read(*,502)FF
```

```
      write(*,600)' up to which pt is this factor valid: '
      read(*,501)iend
      ibegin=i
      goto 100

50    close(1)
      close(2)
      close(3)
      close(11)
      close(12)
      close(13)

c *************************************************************************
c Standard formats
c *************************************************************************
500   format(a)
501   format(i4)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
      stop 'goodbye'
      end



c *************************************************************************
c *************************************************************************
c KALMAN FILTERING
c Read in .Bm, .Sm, .Pm, .Ysm, .Ypm, .Rm 'measurement' files
c Estimate the biomass, substrate,product concentrations.
c Estimate the specific growth rate, Ys, and Yp.
c Output results as Bk, Sk, Pk, Ysk, Ypk, MUk ... (k stands for kalman)
c Reciprocal of Ys and Yp are used in working
c Assignment of number
c   1 .. biomass conc.  (gm/liter)
c   2 .. substrate conc.(gm/liter)
c   3 .. product conc.  (gm/liter)
c   4 .. Ys             (gm biomass/gm substrate)
c   5 .. Yp             (gm biomass/gm product)
c   6 .. MU             (hr-1) .. read in (hr); work in (min); output in (hr)
c Link this program with DERIV, MATRIX, and DVERK(IMSL).
c Size ...
c     m=dimension of measurement equations
c     n=dimension of state equations
c     nsize = (nstart-1) + n*(n+1)/2          ... the size of integration
c *************************************************************************
c *************************************************************************
      PROGRAM KALMAN
      parameter (m=6,n=6,nstart=7,nsize=27)

      external deriv
      character fname1*40,fname2*40,fname3*40,fname4*40,fname5*40,
     *          fname6*40
      dimension x(nsize),xdot(nsize),c(24),w(nsize,9)
```

```
      dimension p(n,n)
      dimension Ym(m),error2(n),err2in(n)
      common/c1/Db,Ds,sf,Ym,error2,err2in

      call seta(p,0.,n,n)
c ***************************************************************************
c Open input data files
c ***************************************************************************
      write(*,600)' enter input filename (.Bm):  '
      read(*,500)fname1
      open(11,file=fname1,status='old')
      write(*,600)' enter input filename (.Sm):  '
      read(*,500)fname2
      open(12,file=fname2,status='old')
      write(*,600)' enter input filename (.Pm):  '
      read(*,500)fname3
      open(13,file=fname3,status='old')
      write(*,600)' enter input filename (.Ysm): '
      read(*,500)fname4
      open(14,file=fname4,status='old')
      write(*,600)' enter input filename (.Ypm): '
      read(*,500)fname5
      open(15,file=fname5,status='old')
      write(*,600)' enter input filename (.Rm):  '
      read(*,500)fname6
      open(16,file=fname6,status='old')

c ***************************************************************************
c Open output data files
c ***************************************************************************
      open(21,file='out.Bk',status='new')
      open(22,file='out.Sk',status='new')
      open(23,file='out.Pk',status='new')
      open(24,file='out.Ysk',status='new')
      open(25,file='out.Ypk',status='new')
      open(26,file='out.Muk',status='new')

c ***************************************************************************
c Input initial data from keyboard
c ***************************************************************************
      write(*,600)' enter initial biomass conc. (g/liter): '
      read(*,502)x(1)
      write(*,600)' enter initial substrate conc. (g/liter): '
      read(*,502)x(2)
      write(*,600)' enter initial product conc. (g/liter): '
      read(*,502)x(3)
      write(*,600)' enter initial 1/Ys: '
      read(*,502)x(4)
      write(*,600)' enter initial 1/Yp: '
      read(*,502)x(5)
      write(*,600)' enter initial specific growth rate (hr-1): '
      read(*,502)x(6)

      write(*,600)' enter error in biomass conc. (g/liter): '
```

```
      read(*,502)error2(1)
      write(*,600)' enter error in substrate conc. (g/liter): '
      read(*,502)error2(2)
      write(*,600)' enter error in product conc. (g/liter): '
      read(*,502)error2(3)
      write(*,600)' enter error in 1/Ys: '
      read(*,502)error2(4)
      write(*,600)' enter error in 1 1/Yp: '
      read(*,502)error2(5)
      write(*,600)' enter error in specific growth rate (hr-1): '
      read(*,502)error2(6)

c     convert to inverse variance
      do 10 i=1,n
         error2(i)=error2(i)*error2(1)
         err2in(i)=1./error2(1)
         p(i,i)=error2(i)
10    continue
c     These errors are also considered equivalent to the measurement error
c     read in the measurement error in total growth rate
      write(*,600)' enter error in total growth rate meas. (g/l-hr):'
      read(*,502)error2(6)
      error2(6)=error2(6)*error2(6)
      err2in(6)=1./error2(6)

c ***************************************************************************
c Initialize everything
c ***************************************************************************
      write(*,600)' enter tolerance in integration (0.0001-0.0005) '
      read(*,502)tol
      tincrm = 1./60.
      sf     = 5.00
      call sqra2x(p,x,n,nstart,n,nsize)
      index=2
      call setx(c,0.,24)
      c(1)=3.
      c(2)=1.e-6
      time = 0.

      ibegin=1
      iend=0
100   do 200 i=ibegin,iend

         write(*,501)i
c ***************************************************************************
c Input data data files
c ***************************************************************************
         read(11,*,end=999)Ym(1)
         read(12,*,end=999)Ym(2)
         read(13,*,end=999)Ym(3)
         read(14,*,end=999)Ym(4)
         read(15,*,end=999)Ym(5)
         read(16,*,end=999)Ym(6)
```

```
c ***************************************************************************
c Estimate kalman state variables
c ***************************************************************************
        time = float(i-1)*tincrm
        timend = float(i)*tincrm
        call dverk(nsize,deriv,time,x,timend,tol,index,c,nsize,w,ier)
        write(21,604)x(1)
        write(22,604)x(2)
        write(23,604)x(3)
        write(24,604)x(4)
        write(25,604)x(5)
        write(26,604)x(6)

c ***************************************************************************
c Check for integration errors
c ***************************************************************************
      if(index.ge.0 .and. ier.le.0) go to 200
      write(*,97)timend
97    format(' error in deriv at t= ',f8.2)
      write(*,98)index
98    format(' index =              ',i4)
      write(*,99)ier
99    format(' ier =                ',i4)
      tol = tol*1.1
      write(*,182)tol
182   format(' tol = ',f8.5)

200   continue

c ***************************************************************************
c Read in new dilution rate
c ***************************************************************************
      write(*,600)' enter dilution rate for B eqn. (hr-1): '
      read(*,502)Db
      write(*,600)' enter dilution rate for S eqn. (hr-1): '
      read(*,502)Ds
      write(*,600)' up to which pt are these dilution rate valid  '
      read(*,501)iend
      ibegin=i
      goto 100

999   close(11)
      close(12)
      close(13)
      close(14)
      close(15)
      close(16)
      close(21)
      close(22)
      close(23)
      close(24)
      close(25)
      close(26)
```

```
c ***************************************************************************
c Standard formats
c ***************************************************************************
500   format(a)
501   format(i6)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
      stop 'goodbye'
      end


c ***************************************************************************
c ***************************************************************************
c State equations and variance equations to be used with Kalman filter
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE DERIV(nsize,time,x,xdot)

c ***************************************************************************
c Assign sizes
c ***************************************************************************
      parameter (m=6,n=6,nstart=7)
      dimension x(nsize),xdot(nsize),p(n,n),pdot(n,n),q(n,n),rinv(m,m),
     *fx(n,n),hx(m,n),hxt(n,m),err(m),g(n,m),gt(m,n),qq(m,m),
     *temp1(m),temp2(n),temp3(m,n),temp4(n,m),temp5(n,n),temp6(n,n)
      dimension Ym(m),error2(n),err2in(n)
      common/c1/Db,Ds,sf,Ym,error2,err2in

c ***************************************************************************
c Assign state variables
c ***************************************************************************
      bb = x(1)
      ss = x(2)
      pp = x(3)
      yys= x(4)
      yyp= x(5)
      uu = x(6)
c ***************************************************************************
c Assign matrix p
c ***************************************************************************
      call sqrx2a(p,x,n,nstart,n,nsize)
c ***************************************************************************
c Assign the bottom half of p
c ***************************************************************************
      call sysfil(p,n)
c ***************************************************************************
c Assign fx
c ***************************************************************************
      fx(1,1) = uu-Db
      fx(1,6) = bb
      fx(2,1) = -uu*yys
      fx(2,2) = -Ds
      fx(2,4) = -uu*bb
```

```
      fx(2,6) = -bb*yys
      fx(3,1) = uu*yyp
      fx(3,3) = -Ds
      fx(3,5) = uu*bb
      fx(3,6) = bb*yyp
c ***************************************************************************
c Assign hx
c ***************************************************************************
      hx(1,1) = 1.
      hx(2,2) = 1.
      hx(3,3) = 1.
      hx(4,4) = 1.
      hx(5,5) = 1.
      hx(6,1) = uu
      hx(6,6) = bb
      call transp(hxt,hx,m,n)
c ***************************************************************************
c Assign err
c ***************************************************************************
      err(1) = Ym(1)-bb
      err(2) = Ym(2)-ss
      err(3) = Ym(3)-pp
      err(4) = Ym(4)-yys
      err(5) = Ym(5)-yyp
      err(6) = Ym(6)-uu*bb
c ***************************************************************************
c Assign rinv
c ***************************************************************************
      rinv(1,1) = err2in(1)
      rinv(2,2) = err2in(2)
      rinv(3,3) = err2in(3)
      rinv(4,4) = err2in(4)
      rinv(5,5) = err2in(5)
      rinv(6,6) = err2in(6)
c ***************************************************************************
c Assign q
c ***************************************************************************
      call q2abat(qq,hx,p,hxt,temp4,m,n)
      qq(1,1) = 2.*(err(1)*err(1)-error2(1)-qq(1,1))
      qq(2,2) = 2.*(err(2)*err(2)-error2(2)-qq(2,2))
      qq(3,3) = 2.*(err(3)*err(3)-error2(3)-qq(3,3))
      qq(4,4) = 2.*(err(4)*err(4)-error2(4)-qq(4,4))
      qq(5,5) = 2.*(err(5)*err(5)-error2(5)-qq(5,5))
      qq(6,6) = 2.*(err(6)*err(6)-error2(6)-qq(6,6))
      if(qq(1,1) .le. 0.)qq(1,1) = 0.
      if(qq(2,2) .le. 0.)qq(2,2) = 0.
      if(qq(3,3) .le. 0.)qq(3,3) = 0.
      if(qq(4,4) .le. 0.)qq(4,4) = 0.
      if(qq(5,5) .le. 0.)qq(5,5) = 0.
      if(qq(6,6) .le. 0.)qq(6,6) = 0.
      call setoff (qq,0.,m)
      g(1,1) = 1.
      g(2,2) = 1.
      g(3,3) = 1.
```

```
      g(4,4) = 1.
      g(5,5) = 1.
      g(6,6) = 1./bb
      call transp(gt,g,n,m)
      call q2abat(q,g,qq,gt,temp3,n,m)
c ***************************************************************************
c Calculate the state eqn.
c ***************************************************************************
      call mulyax(temp1, rinv,          err,m,m)
      call mulyax(temp2,  hxt,temp1,n,m)
      call mulyax( xdot,    p,temp2,n,n)
        xdot(1) = xdot(1) + (uu-Db)*bb
        xdot(2) = xdot(2) + Ds*(sf-ss)-uu*bb*yys
        xdot(3) = xdot(3) - Ds*pp+uu*bb*yyp
c ***************************************************************************
c Calculate covariance eqn.
c ***************************************************************************
      call q2abat(temp5,  hxt, rinv,   hx,temp3,n,m)
      call q2abat( pdot,    p,temp5,    p,temp6,n,n)
      call subcab( pdot,    q, pdot,n,n)
      call mulcab(temp5,   fx,            p,n,n,n)
      call addcab( pdot, pdot,temp5,n,n)
      call transp(temp6,temp5,n,n)
      call addcab( pdot, pdot,temp6,n,n)
c ***************************************************************************
c Relate pdot to xdot
c ***************************************************************************
      call sqra2x(pdot,xdot,n,nstart,n,nsize)

      return
      end



c ***************************************************************************
c ***************************************************************************
c This subroutine is originally copied from IMSL and modified to work under
c   IBM DOS with MicroSoft FORTRAN compiler.
c See IMSL documentation for the proper usage.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE DVERK (n,fcn,x,y,xend,tol,ind,c,nw,w,ier)
      integer           n,ind,nw,ier
      real              x,y(n),xend,tol,c(1),w(nw,9)
      integer           k
      real              zero,one,two,three,four,five,seven,ten,half,p9
      real              c4d15,c2d3,c5d6,c1d6,c1d15,c2d96,temp
      real              rk(39),reps,rtol
      data              zero/0.0/,one/1.0/,two/2.0/,three/3.0/
      data              four/4.0/,five/5.0/,seven/7.0/
      data              ten/10.0/,half/0.5/,p9/0.9/
      data              c4d15/.2666667/
      data              c2d3/.6666667/
      data              c5d6/.8333333/
```

```
      data                  c1d6/.1666667/
      data                  c1d15/.6666667e-1/
      data                  c2d96/120.4273/
      data                  reps/1.1921e-07/
      data                  rtol/2.4651e-32/
      data                  rk( 1)/.1666667e+00/
      data                  rk( 2)/.5333333e-01/
      data                  rk( 3)/.2133333e+00/
      data                  rk( 4)/.8333333e+00/
      data                  rk( 5)/.2666667e+01/
      data                  rk( 6)/.2500000e+01/
      data                  rk( 7)/.2578125e+01/
      data                  rk( 8)/.9166667e+01/
      data                  rk( 9)/.6640625e+01/
      data                  rk(10)/.8854167e+00/
      data                  rk(11)/.2400000e+01/
      data                  rk(12)/.8000000e+01/
      data                  rk(13)/.6560458e+01/
      data                  rk(14)/.3055556e+00/
      data                  rk(15)/.3450980e+00/
      data                  rk(16)/.5508667e+00/
      data                  rk(17)/.1653333e+01/
      data                  rk(18)/.9455882e+00/
      data                  rk(19)/.3240000e+00/
      data                  rk(20)/.2337882e+00/
      data                  rk(21)/.2035465e+01/
      data                  rk(22)/.6976744e+01/
      data                  rk(23)/.5648180e+01/
      data                  rk(24)/.1373816e+00/
      data                  rk(25)/.2863023e+00/
      data                  rk(26)/.1441786e+00/
      data                  rk(27)/.7500000e-01/
      data                  rk(28)/.3899287e+00/
      data                  rk(29)/.3194444e+00/
      data                  rk(30)/.1350384e+00/
      data                  rk(31)/.1078330e-01/
      data                  rk(32)/.6980519e-01/
      data                  rk(33)/.6250000e-02/
      data                  rk(34)/.6963012e-02/
      data                  rk(35)/.6944444e-02/
      data                  rk(36)/.6138107e-02/
      data                  rk(37)/.6818182e-01/
      data                  rk(38)/.1078330e-01/
      data                  rk(39)/.6980519e-01/
      ier = 0
      if (ind.lt.1.or.ind.gt.6) go to 290
      go to (5,5,40,145,265,265), ind
    5 if (n.gt.nw.or.tol.le.zero) go to 295
      if (ind.eq.2) go to 15
      do 10 k=1,9
         c(k) = zero
   10 continue
      go to 30
   15 continue
```

```
      do 20 k=1,9
         c(k) = abs(c(k))
   20 continue
      if (c(1).ne.four.and.c(1).ne.five) go to 30
      do 25 k=1,n
         c(k+30) = abs(c(k+30))
   25 continue
   30 continue
      c(10) = reps
      c(11) = rtol
      c(20) = x
      do 35 k=21,24
         c(k) = zero
   35 continue
      go to 45
   40 if (c(21).ne.zero.and.(x.ne.c(20).or.xend.eq.c(20))) go to 285
      c(21) = zero
   45 continue
   50 continue
      if (c(7).eq.zero.or.c(24).lt.c(7)) go to 55
      ind = -1
      go to 9005
   55 continue
      if (ind.eq.6) go to 60
      call fcn (n,x,y,w(1,1))
      c(24) = c(24)+one
   60 continue
      c(13) = c(3)
      if (c(3).ne.zero) go to 120
      temp = zero
      if (c(1).ne.one) go to 70
      do 65 k=1,n
         temp = amax1(temp,abs(y(k)))
   65 continue
      c(12) = temp
      go to 115
   70 if (c(1).ne.two) go to 75
      c(12) = one
      go to 115
   75 if (c(1).ne.three) go to 85
      do 80 k=1,n
         temp = amax1(temp,abs(y(k))/c(2))
   80 continue
      c(12) = amin1(temp,one)
      go to 115
   85 if (c(1).ne.four) go to 95
      do 90 k=1,n
         temp = amax1(temp,abs(y(k))/c(k+30))
   90 continue
      c(12) = amin1(temp,one)
      go to 115
   95 if (c(1).ne.five) go to 105
      do 100 k=1,n
         temp = amax1(temp,abs(y(k))/c(k+30))
```

```
100 continue
    c(12) = temp
    go to 115
105 continue
    do 110 k=1,n
       temp = amax1(temp,abs(y(k)))
110 continue
    c(12) = amin1(temp,one)
115 continue
    c(13) = ten*amax1(c(11),c(10)*amax1(c(12)/tol,abs(x)))
120 continue
    c(15) = c(5)
    if (c(5).eq.zero) c(15) = one
    if (c(6).ne.zero.and.c(5).ne.zero) c(16) = amin1(c(6),two/c(5))
    if (c(6).ne.zero.and.c(5).eq.zero) c(16) = c(6)
    if (c(6).eq.zero.and.c(5).ne.zero) c(16) = two/c(5)
    if (c(6).eq.zero.and.c(5).eq.zero) c(16) = two
    if (c(13).le.c(16)) go to 125
    ind = -2
    go to 9005
125 continue
    if (ind.gt.2) go to 130
    c(14) = c(4)
    if (c(4).eq.zero) c(14) = c(16)*tol**c1d6
    go to 140
130 if (c(23).gt.one) go to 135
    temp = two*c(14)
    if (tol.lt.c2d96*c(19)) temp = p9*(tol/c(19))**c1d6*c(14)
    c(14) = amax1(temp,half*c(14))
    go to 140
135 continue
    c(14) = half*c(14)
140 continue
    c(14) = amin1(c(14),c(16))
    c(14) = amax1(c(14),c(13))
    if (c(8).eq.zero) go to 145
    ind = 4
    go to 9005
145 continue
    if (c(14).ge.abs(xend-x)) go to 150
    c(14) = amin1(c(14),half*abs(xend-x))
    c(17) = x+sign(c(14),xend-x)
    go to 155
150 continue
    c(14) = abs(xend-x)
    c(17) = xend
155 continue
    c(18) = c(17)-x
    do 160 k=1,n
       w(k,9) = y(k)+c(18)*w(k,1)*rk(1)
160 continue
    call fcn (n,x+c(18)*c1d6,w(1,9),w(1,2))
    do 165 k=1,n
       w(k,9) = y(k)+c(18)*(w(k,1)*rk(2)+w(k,2)*rk(3))
```

```
165 continue
    call fcn (n,x+c(18)*c4d15,w(1,9),w(1,3))
    do 170 k=1,n
       w(k,9) = y(k)+c(18)*(w(k,1)*rk(4)-w(k,2)*rk(5)+w(k,3)*rk(6))
170 continue
    call fcn (n,x+c(18)*c2d3,w(1,9),w(1,4))
    do 175 k=1,n
       w(k,9) = y(k)+c(18)*(-w(k,1)*rk(7)+w(k,2)*rk(8)-w(k,3)*rk(9)
   1   +w(k,4)*rk(10))
175 continue
    call fcn (n,x+c(18)*c5d6,w(1,9),w(1,5))
    do 180 k=1,n
       w(k,9) = y(k)+c(18)*(w(k,1)*rk(11)-w(k,2)*rk(12)+w(k,3)*rk(13)
   1   -w(k,4)*rk(14)+w(k,5)*rk(15))
180 continue
    call fcn (n,x+c(18),w(1,9),w(1,6))
    do 185 k=1,n
       w(k,9) = y(k)+c(18)*(-w(k,1)*rk(16)+w(k,2)*rk(17)-w(k,3)
   1   *rk(18)-w(k,4)*rk(19)+w(k,5)*rk(20))
185 continue
    call fcn (n,x+c(18)*c1d15,w(1,9),w(1,7))
    do 190 k=1,n
       w(k,9) = y(k)+c(18)*(w(k,1)*rk(21)-w(k,2)*rk(22)+w(k,3)*rk(23)
   1   -w(k,4)*rk(24)+w(k,5)*rk(25)+w(k,7)*rk(26))
190 continue
    call fcn (n,x+c(18),w(1,9),w(1,8))
    do 195 k=1,n
       w(k,9) = y(k)+c(18)*(w(k,1)*rk(27)+w(k,3)*rk(28)+w(k,4)*rk(29)
   1   +w(k,5)*rk(30)+w(k,7)*rk(31)+w(k,8)*rk(32))
195 continue
    c(24) = c(24)+seven
    do 200 k=1,n
       w(k,2) = w(k,1)*rk(33)+w(k,3)*rk(34)-w(k,4)*rk(35)+w(k,5)
   1   *rk(36)+w(k,6)*rk(37)-w(k,7)*rk(38)-w(k,8)*rk(39)
200 continue
    temp = zero
    if (c(1).ne.one) go to 210
    do 205 k=1,n
       temp = amax1(temp,abs(w(k,2)))
205 continue
    go to 260
210 if (c(1).ne.two) go to 220
    do 215 k=1,n
       if (y(k).eq.zero) go to 280
       temp = amax1(temp,abs(w(k,2)/y(k)))
215 continue
    go to 260
220 if (c(1).ne.three) go to 230
    do 225 k=1,n
       temp = amax1(temp,abs(w(k,2))/amax1(c(2),abs(y(k))))
225 continue
    go to 260
230 if (c(1).ne.four) go to 240
    do 235 k=1,n
```

```
         temp = amax1(temp,abs(w(k,2))/amax1(c(k+30),abs(y(k))))
  235 continue
      go to 260
  240 if (c(1).ne.five) go to 250
      do 245 k=1,n
         temp = amax1(temp,abs(w(k,2)/c(k+30)))
  245 continue
      go to 260
  250 continue
      do 255 k=1,n
         temp = amax1(temp,abs(w(k,2))/amax1(one,abs(y(k))))
  255 continue
  260 continue
      c(19) = temp*c(14)*c(15)
      ind = 5
      if (c(19).gt.tol) ind = 6
      if (c(9).ne.zero) go to 9005
  265 continue
      if (ind.eq.6) go to 275
      x = c(17)
      do 270 k=1,n
         y(k) = w(k,9)
  270 continue
      c(22) = c(22)+one
      c(23) = zero
      if (x.ne.xend) go to 50
      ind = 3
      c(20) = xend
      c(21) = one
      go to 9005
  275 continue
      c(23) = c(23)+one
      if (c(14).gt.c(13)) go to 50
      ind = -3
      go to 9005
  280 continue
      ier = 132
      go to 9000
  285 continue
      ier = 131
      go to 9000
  290 continue
      ier = 130
      go to 9000
  295 continue
      ier = 129
 9000 continue
      call uertst (ier,'6hdverk ')
 9005 continue
      return
      end
```

```
c ***********************************************************************
c ***********************************************************************
c Used in DVERK
c ***********************************************************************
c ***********************************************************************
      SUBROUTINE UERTST (ier,name)
      integer            ier
c the following three lines are changed from the original imsl-vax
c     integer            name(1)
      integer            i,ieqdf,iounit,level,levold,nin,nmtb
      character          name(1),ieq,nameq(6),namset(6),namupk(6)
      data               namset/'u','e','r','s','e','t'/
      data               nameq/' ',' ',' ',' ',' ',' ',' ',' ',' ',' '/
      data               level/4/,ieqdf/0/,ieq/'='/
c the following line is changed
c     call uspkd (name,6,namupk,nmtb)
      write(*,*)' error from dverk'
      call ugetio(1,nin,iounit)
      if (ier.gt.999) go to 25
      if (ier.lt.-32) go to 55
      if (ier.le.128) go to 5
      if (level.lt.1) go to 30
      if (ieqdf.eq.1) write(iounit,35) ier,nameq,ieq,namupk
      if (ieqdf.eq.0) write(iounit,35) ier,namupk
      go to 30
    5 if (ier.le.64) go to 10
      if (level.lt.2) go to 30
      if (ieqdf.eq.1) write(iounit,40) ier,nameq,ieq,namupk
      if (ieqdf.eq.0) write(iounit,40) ier,namupk
      go to 30
   10 if (ier.le.32) go to 15
      if (level.lt.3) go to 30
      if (ieqdf.eq.1) write(iounit,45) ier,nameq,ieq,namupk
      if (ieqdf.eq.0) write(iounit,45) ier,namupk
      go to 30
   15 continue
      do 20 i=1,6
         if (namupk(i).ne.namset(i)) go to 25
   20 continue
      levold = level
      level = ier
      ier = levold
      if (level.lt.0) level = 4
      if (level.gt.4) level = 4
      go to 30
   25 continue
      if (level.lt.4) go to 30
      if (ieqdf.eq.1) write(iounit,50) ier,nameq,ieq,namupk
      if (ieqdf.eq.0) write(iounit,50) ier,namupk
   30 ieqdf = 0
      return
   35 format(19h *** terminal error,10x,7h(ier = ,i3,
     1       20h) from imsl routine ,6a1,a1,6a1)
   40 format(27h *** warning with fix error,2x,7h(ier = ,i3,
```

```
     1       20h) from imsl routine ,6a1,a1,6a1)
    45 format(18h *** warning error,11x,7h(ier = ,i3,
     1       20h) from imsl routine ,6a1,a1,6a1)
    50 format(20h *** undefined error,9x,7h(ier = ,i5,
     1       20h) from imsl routine ,6a1,a1,6a1)
    55 ieqdf = 1
       do 60 i=1,6
    60 nameq(i) = namupk(i)
    65 return
       end


c ******************************************************************************
c ******************************************************************************
c Used in DVERK
c ******************************************************************************
c ******************************************************************************
      SUBROUTINE UGETIO(iopt,nin,nout)
      integer           iopt,nin,nout
      integer           nind,noutd
      data              nind/1/,noutd/2/
      if (iopt.eq.3) go to 10
      if (iopt.eq.2) go to 5
      if (iopt.ne.1) go to 9005
      nin = nind
      nout = noutd
      go to 9005
    5 nind = nin
      go to 9005
   10 noutd = nout
 9005 return
      end
```

APPENDIX D

LISTING
OF THE KERNEL INVERSION CODES

```
c ***************************************************************************
c ***************************************************************************
c Calculate the kernel in convolution integral by direct numerical integration
c y(t)=integral from 0 to inf of { x(t-h)*c(h)*dh }
c Link with ASM.OBJ (for ASKSHT)
c ***************************************************************************
c ***************************************************************************
      PROGRAM INTEGRATE
      parameter (nn=500)
      character fname1*40,fname2*40,resp*1
      dimension x(nn),y(nn),c(nn)

      write(*,600)' Step size: '
        read(*,502)step
      write(*,600)' System input filename: '
        read(*,500)fname1
        open(11,file=fname1,status='old')
      write(*,600)' System output filename: '
        read(*,500)fname2
        open(12,file=fname2,status='old')
      write(*,600)' First point to be included: '
        read(*,501)ifirst

      do 3 i=1,ifirst-1
        read(11,*)xjunk
3       read(12,*)xjunk

      do 5 i=1,9999
        read(11,*,end=6)x(i)
5       read(12,*,end=6)y(i)
6     close(11)
      close(12)
      n=i-1
      if(n .gt. nn)stop 'Input file dimension is too big'

c ***************************************************************************
c shift in the vertical axis
c ***************************************************************************
      write(*,600)' Vertical offset in the system input: '
        read(*,502)xshift
      write(*,600)' Vertical offset in the system output: '
        read(*,502)yshift
      do 10 i=1,n
        x(i)=x(i)-xshift
10      y(i)=y(i)-yshift

c ***************************************************************************
c Use x(1) as the initial quess for x0
c ***************************************************************************
      write(*,651)x(1)
      write(*,600)' Use x(1) as the initial quess for x0  (Y/n) '
        read(*,500)resp
      if(resp .eq. 'n' .or. resp .eq. 'N' .or. x(1) .eq. 0)then
        write(*,600)' 1st quess of x0 over the vertical offset: '
```

```
         read(*,502)x0
      else
        x0=x(1)
      endif

c *************************************************************************
c See if the kernel is normalized
c *************************************************************************
100   temp=area(x,y,c,n,step,x0,iflag)
      write(*,650)iflag,temp

      write(*,600)' Enter automatic loop  (exit/y/N) '
        read(*,500)resp
      if(resp .eq. 'e' .or. resp .eq. 'E')goto 200
      if(resp .ne. 'y' .and. resp .ne. 'Y')then
        write(*,600)' Next quess of x0 over the vertical offset: '
          read(*,502)x0
        goto 100
      endif

110   write(*,600)' 1st quess of x0 over the vertical offset: '
        read(*,502)xleft
      write(*,600)' 2nd quess of x0 over the vertical offset: '
        read(*,502)xright
      write(*,600)' Tolerance in x0: '
        read(*,502)xtol
        if(xtol .le. 0.)xtol=1.e-8
      write(*,600)' Tolerance in area: '
        read(*,502)ftol
        if(ftol .le. 0.)ftol=1.e-8

      write(*,500)' Hit both shift keys to stop iteration ...'
      write(*,500)' Be patient!'

      xoold=xleft
      temp=area(x,y,c,n,step,xoold,iflag)
        if(iflag .eq. 1)goto 110
        foold=temp-1.
      xold=xright

c *************************************************************************
c Iterate with the improved chord method to find the correct x0.
c *************************************************************************
      do 150 icount=1, 9999
        temp=area(x,y,c,n,step,xoold,iflag)
          if(iflag .eq. 1)goto 110
          foold=temp-1.
        write(*,652)icount,xold,fold+1.
        dx=xold-xoold
        df=fold-foold
        if(abs(dx) .le. xtol)goto 200
        if(abs(df) .le. ftol)goto 200
        xnew=xold-fold*dx/df
        if(xnew .lt.  xleft)xnew=xleft
```

```
c       if(xnew .gt. xright)xnew=xright
        xoold=xold
        foold=fold
        xold=xnew
        call asksht(ishift)
        if(ishift .eq. 3)goto 200
150   continue

200   write(*,600)' Kernel function filename: '
        read(*,500)fname1
        open(11,file=fname1,status='new')
c Do not write out c(1),
c   but add another c(n) at the end so that the total number is still n
        write(11,603)(c(i),i=2,n)
        write(11,603)c(n)
        close(11)

      write(*,600)' Continue  (y/N) '
        read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')then
        write(*,600)' Next quess of x0: '
          read(*,502)x0
        goto 100
      endif

c *************************************************************************
c Standard formats
c *************************************************************************
500   format(a)
501   format(i6)
502   format(e13.5)
600   format(a\)
603   format(1pe13.5)
650   format(' iflag= ',i1,';  Kernel area= ',1pe13.5)
651   format(' x0= ',1pe13.5)
652   format(i5,'th iteration:  x0= ',1pe13.5,'  Kernel area= ',e13.5)
      stop
      end



c *************************************************************************
c *************************************************************************
c Numerical integration of the given array x over 1 to n:
c step=step size,
c Used by PROGRAM INTEGRATE
c *************************************************************************
c *************************************************************************
      REAL FUNCTION XINT(X,N,STEP)
      dimension x(1)
      xint=0.
      do 10 i=1, n
        ia=4-mod(i,2)*2
        a=float(ia)
```

```
        xint=xint+a*x(i)
10    continue
      xint=xint*step/3.
      return
      end



c ***************************************************************************
c ***************************************************************************
c Solve the convolution integral by direct numerical integration
c iflag = 1 ... x0 is too small; calculation failed.
c Used by PROGRAM INTEGRATE
c ***************************************************************************
c ***************************************************************************
      REAL FUNCTION AREA(X,Y,C,N,STEP,X0,IFLAG)
      dimension x(1),y(1),c(1)
      iflag = 0
      do 10 j=1,n
        sum=0.
        do 11 i=1,j-1
11        sum=sum+x(j+1-i)*c(i)
        c(j)=(y(j)/step-sum)/x0
c stop and get out before the program issues math-overflow
        if(abs(c(j)) .ge. 1.e10)then
          iflag=1
          area=0.
          return
        endif
10    continue

c Calculate the area under the kernel
      area=xint(c,n,step)

      return
      end



c ***************************************************************************
c ***************************************************************************
c Direct kernel determination from the polynomial smoothed input and output
c   curves of y and x.
c Given the coefficients to a polynomial y, x, and the order, find the
c   coefficient of least square fit.
c y(t)=ay(1)+ay(2)*t+ay(3)*t**2+ay(4)*t**3+...+ay(iydeg)*t**(iydeg-1)
c x(t)=ax(1)+ax(2)*t+ax(3)*t**2+ax(4)*t**3+...+ax(ixdeg)*t**(ixdeg-1)
c Iterate, until the coefficient of x is unity.
c Maximum orders:
c    ixdeg = 15
c    iydeg = 15
c    iorder  =  k
c Note: the actual starting x-value is xstart-h
c wk ... work space of at least k*(2*n + 3*k + 3)
```

```
c ******************************************************************************
c ******************************************************************************
      PROGRAM POLY-XY
      parameter (n=500,k=6)
      character resp*1
      dimension t(n),yy(n),x(n,k),y(n,k),a(15),ax(15),ay(15)
      dimension b(k),wk(6126)

      write(*,600)' step size: '
        read(*,502)h
      write(*,600)' last flat point for x: '
        read(*,501)ixlast
        xoffset=float(ixlast)*h
      write(*,600)' last flat point for y: '
        read(*,501)iylast
        yoffset=float(iylast)*h
      write(*,600)' x-value of flat points: '
        read(*,502)xflat
      write(*,600)' y-value of flat points: '
        read(*,502)yflat
      write(*,600)' number of coefficients of input polynomial: '
        read(*,501)ixdeg
      do 5 i=1,ixdeg
        write(*,650)i
5       read(*,502)ax(i)
      write(*,600)' number of coefficients of output polynomial: '
        read(*,501)iydeg
      do 10 i=1,iydeg
        write(*,651)i
10      read(*,502)ay(i)

      do 20 i=1,n
20      t(i)=float(i)*h
c ******************************************************************************
c Generate the polynomial function x and its derivatives
c   ideriv=1 ... original x    x(*,1)
c   ideriv=2 ... x'            x(*,2)
c   ideriv=3 ... x''           x(*,3)
c     ...
c   ideriv=k ... x(k-1)        x(*,k)
c ******************************************************************************

      do 40 i=1,ixdeg
40      a(i)=ax(i)
      do 100 ideriv=1,k
c     do 100 ideriv=1,ixdeg-1      !use this to get all the non-zero derivatives
        write(*,652)ideriv-1
652       format(' Processing the ',i2,'th derivative of input')
        do 50 i=1,ixlast
50        x(i,ideriv)=xflat
        if(ideriv .eq. 1)xflat=0.
        imax=ixdeg-ideriv+1
        do 55 i=ixlast+1,n
          x(i,ideriv)=a(imax)
```

```
          do 56 ii=1,imax-1
56          x(i,ideriv)=a(imax-ii)+(t(i)-xoffset)*x(i,ideriv)
55      continue
c       Find the coefficient of the next polynomial
        do 60 i=1,ixdeg-ideriv
60        a(i)=a(i+1)*float(i)
c       write(11,15)(a(i),i=1,ixdeg-ideriv)
100   continue

c *************************************************************************
c Generate the polynomial function y and its derivatives
c *************************************************************************
      do 140 i=1,iydeg
140     a(i)=ay(i)
      do 200 ideriv=1,k
        write(*,653)ideriv-1
653       format(' Processing the ',i2,'th derivative of output')
        do 150 i=1,iylast
150       y(i,ideriv)=yflat
        if(ideriv .eq. 1)yflat=0.
        imax=iydeg-ideriv+1
        do 155 i=iylast+1,n
          y(i,ideriv)=a(imax)
          do 156 ii=1,imax-1
156         y(i,ideriv)=a(imax-ii)+(t(i)-yoffset)*y(i,ideriv)
155     continue
c       Find the coefficient of the next polynomial
        do 160 i=1,iydeg-ideriv
160       a(i)=a(i+1)*float(i)
200   continue

300    write(*,600)' Order of fit (lowest =1): '
       read(*,501)iorder
      if(iorder.le.0 .or. iorder.gt.k)goto 300

350   write(*,600)' Enter Tau: '
        read(*,502)tau

c *************************************************************************
c Form the left hand side of the kernel differential equation:
c     N+1                    d^i y      N          d^i x
c     SUM   bin(N+1,i) T**i ------- =  SUM   a_i  -------
c     i=0                    dt^i       i=0        dt^i
c Change of variables: i=j-1;iorder=N+1 =======>
c   iorder+1                          d^(j-1) y   iorder          d^(j-1) x
c     SUM   bin(iorder,j-1) T**(j-1) ---------- =  SUM   a_(j-1)  ----------
c     j=1                            dt^(j-1)     j=1             dt^(j-1)
c *************************************************************************
      do 240 i=1,n
240     yy(i)=y(i,1)
      do 250 j=2,iorder+1
        tfactor=bin(iorder,j-1)*tau**(j-1)
        do 260 i=1,n
260       yy(i)=tfactor*y(i,j)+yy(i)
```

```
250   continue

c Find the least square estimate
      call leasts(yy,x,b,error,n,iorder,wk,n)
      write(*,500)' Least-square coefficients: '
      write(*,656)(b(i),i=1,iorder)
      write(*,657)error
      write(*,600)' Guess another Tau (Y/n)   '
      read(*,500)resp
      if(resp .eq. 'n' .or. resp .eq. 'N')goto 300
      goto 350

c *************************************************************************
c standard formats
c *************************************************************************
500   format(a)
501   format(i4)
502   format(f15.5)
600   format(a\)
650   format(' ax',i2,': ',\)
651   format(' ay',i2,': ',\)
656   format(1pe15.5,10e15.5)
657   format(' Error = ',1pe15.5)
      stop
      end


c *************************************************************************
c *************************************************************************
c Calculate the real binomial coefficient
c Used by PROGRAM POLY-XY
c *************************************************************************
c *************************************************************************
      FUNCTION NBIN(N,K)
      top=float(n)
      do 10 i=1,k-1
10      top=top*float(n-i)/float(k+1-i)
      nbin=top+0.5
      if(k .eq. 0)nbin=1
      return
      end


c *************************************************************************
c *************************************************************************
c Calculate the integer binomial coefficient
c Used by PROGRAM POLY-XY
c *************************************************************************
c *************************************************************************
      FUNCTION BIN(N,K)
      bin=float(n)
      do 10 i=1,k-1
```

```
10      bin=bin*float(n-i)/float(k+1-i)
      if(k .eq. 0)bin=1.
      return
      end


c ***************************************************************************
c ***************************************************************************
c Find the linear least square fit per p265 of Box & Jenkins without weighing
c   y(i) = a(1)*x(i,1)+a(2)*x(i,2)+...+a(k)*x(i,k)      i=1,2,...,n
c   y = X*a
c where y = n measurements
c       x = independent variables (e.g. x(1,1)=1, x(1,2)=t(1), x(1,3)=t(1)**2,
c                                  etc., where t = true independent variable)
c       a = k estimate of parameters
c       n = number of measurements
c       k = number of degree
c The least squares estimate is given by: a=inverse(Xt*X) * Xt * y
c error = sum of square error = yt*y - bt*Xt*X*b
c wk ... work space of at least k*(2*n+3*k+3)
c nn ... row dimension of X exactly as declared in the calling statement
c Used by PROGRAM POLY-XY
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE LEASTS(Y,X,B,ERROR,N,K,WK,NN)
      dimension wk(1)
      dimension y(1),X(nn,1),b(1)
c Allocation of work space
c    +-n--+-k--+-k--+-n--+-k+3---+  Total work space needed = k*(2*n+3*k+3)
c    k Xt | XX | XXi| C  | wkinv |
c    +----+----+----+----+-------+
      iXt   = 1
      iXX   = k*n+iXt
      iXXi  = k*k+iXX
      iC    = k*k+iXXi
      iwkinv= k*n+iC

c ***************************************************************************
      call TRANSP(wk(iXt),X,n,k,nn,k)
      call MULCAB(wk(iXX),wk(iXt),X,k,n,k,k,nn)
      idgt=0
      call LINV2F(wk(iXX),k,k,wk(iXXi),idgt,wk(iwkinv),ier)
      call MULCAB(wk(iC),wk(iXXi),wk(iXt),k,k,n,k,k)
      call MULYAX(b,wk(iC),y,k,n,k)

      error=XY(y,y,n)-XAY(b,wk(iXX),b,wk(iwkinv),k,k)
      if(ier .eq. 129)write(*,500)' Singular Matrix Inversion'

500   format(a)
      return
      end
```

```
c ***************************************************************************
c ***************************************************************************
c Find the impulse transfer function by the Cyclic Fourier transform method.
c The effect of cyclic convolution is corrected so that difference operation
c   is not necessary.
c x  = system input
c gx = Fourier transformed system input
c y  = system output
c cy = cyclic system output
c gy = Fourier transformed cyclic system output
c c  = kernel
c gk = Fourier transformed kernel
c ***************************************************************************
c ***************************************************************************
      PROGRAM CYCLIC
      parameter (nn=1000)
c     dimension iwk(75),wk(75),x(nn),y(nn),cy(nn),c(nn)      !for nn=500
      dimension iwk(nn),wk(nn),x(nn),y(nn),cy(nn),c(nn)
      complex gx(nn),gy(nn),gk(nn)
      character fname1*40,fname2*40,resp*1

      write(*,600)' Step size: '
        read(*,502)step
      write(*,600)' System input filename: '
        read(*,500)fname1
        open(11,file=fname1,status='old')
      write(*,600)' System output filename: '
        read(*,500)fname2
        open(12,file=fname2,status='old')

      do 5 i=1,9999
        read(11,*,end=6)x(i)
5       read(12,*,end=6)y(i)
6     close(11)
      close(12)
      n=i-1
      if(n .gt. nn)stop 'Input file dimension is too big'

c ***************************************************************************
c Shift in the vertical axis
c ***************************************************************************
      write(*,600)' Original steady-state offset in the system input: '
        read(*,502)xslow
      write(*,600)' Original steady-state offset in the system output: '
        read(*,502)yslow
      do 10 i=1,n
        x(i)=x(i)-xslow
10      y(i)=y(i)-yslow

c ***************************************************************************
c Transform input x
c ***************************************************************************
      call FFTRC(x,n,gx,iwk,wk)
```

```
      do 15 i=2,n/2+1
15      gx(n+2-i)=conjg(gx(i))

c *************************************************************************
c Calculate the initial adjustment of y due to cyclic effects
c *************************************************************************
      write(*,600)' Last point before perturbation: '
        read(*,501)ilast
      write(*,600)' New steady-state value of x: '
        read(*,502)xshigh
        xshigh=xshigh-xslow
      write(*,600)' New steady-state value of y: '
        read(*,502)yshigh
        yshigh=yshigh-yslow

      iter=0

100   iter=iter+1
      write(*,650)iter

c *************************************************************************
c Add the mirror image of y(t) to cy(t)
c *************************************************************************
      do 20 i=1,n-ilast
20      cy(i)=cy(i)+yshigh-y(ilast+i)

c *************************************************************************
c Adjustment of y due to cyclic effects
c *************************************************************************
      do 25 i=1,n
25      cy(i)=cy(i)+y(i)

      write(*,600)' Write cyclic system output  (y/N) '
        read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')then
        write(*,600)' Cyclic system output filename: '
          read(*,500)fname1
          open(11,file=fname1,status='new')
          write(11,603)(cy(i),i=1,n)
          close(unit=11)
      endif

c *************************************************************************
c Transform output cy, which is adjusted for cyclic effects
c *************************************************************************
      call FFTRC(cy,n,gy,iwk,wk)
      do 30 i=2,n/2+1
30      gy(n+2-i)=conjg(gy(i))

      write(*,600)' Cutoff pt (0 if no freq windowing to cyclic y): '
        read(*,501)ncut
      call WINDOW(n,gy,ncut)

      if(ncut .gt. 0)then
```

```
        write(*,600)' Write F-transformed cyclic system output  (y/N) '
          read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')then
          write(*,600)' F-transformed cyclic system output filename: '
            read(*,500)fname1
            open(11,file=fname1,status='new')
            write(11,604)(gy(i),i=1,n)
            close(11)
          endif
        endif

        do 40 i=1,n
40        gk(i)=gy(i)/gx(i)

c ***************************************************************************
c Invert frequency transfer function to get impulse transfer function
c Remember to take the conjgate before calling FFTCC
c ***************************************************************************
        do 50 i=1,n
50        gk(i)=conjg(gk(i))
        call FFTCC(gk,n,iwk,wk)
        do 60 i=1,n
60        gk(i)=conjg(gk(i))/float(n)
c ***************************************************************************
c Take the real part of gk, because the impulse response function is real
c ***************************************************************************
        do 70 i=1,n
70        c(i)=real(gk(i))/step

        write(*,600)' Write kernel function  (y/N) '
          read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')then
          write(*,600)' Kernel function filename: '
            read(*,500)fname1
            open(11,file=fname1,status='new')
            write(11,603)(c(i),i=1,n)
            close(11)
        endif

        write(*,600)' Continue  (y/N) '
          read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')then
c       Calculate the additional adjustment of cy due to near-step input
          do 80 i=1,n-ilast
            cy(i)=0.
c           Little impulse near t=0
              do 81 j=1,i
81              cy(i)=cy(i)-c(j)*(xshigh-x(ilast+i+1-j))
c           step-down long ago in the negative t
              do 82 j=i+1,n-ilast
82              cy(i)=cy(i)-c(j)*(xshigh-x(n+i+1-j))
            cy(i)=cy(i)*step
80        continue
          do 85 i=n-ilast+1,n
```

```
85          cy(i)=0.
          goto 100
        endif

c ***************************************************************************
c Standard formats
c ***************************************************************************
500     format(a)
501     format(i6)
502     format(e13.5)
600     format(a\)
603     format(1pe13.5)
604     format(1pe13.5,e13.5)
650     format(' iteration = ',i4)
        end



c ***************************************************************************
c ***************************************************************************
c Find the impulse transfer function by minimizing the square deviation
c   between the predicted value of y and the measured value of y.
c The order of the equation and the parameters of k are quessed.
c ***************************************************************************
c ***************************************************************************
        PROGRAM MINIMUM
        dimension x(500),y(500),yp(500),h(500),t(500)
        n=500
        step=0.02
        base=0.3
        open(unit=11,file='x',status='old')
        read(11,15)(x(i),i=1,n)
15      format(10e13.5)
        close(unit=11)
        open(unit=11,file='y',status='old')
        read(11,15)(y(i),i=1,n)
        close(unit=11)
        open(unit=11,file='t',status='old')
        read(11,15)(t(i),i=1,n)
        close(unit=11)

c ***************************************************************************
c Generate the impulse response function
c ***************************************************************************
100     type *,'enter a'
        accept *,a
        type *,'enter tau'
        accept *,tau
        do 5 i=1,n
          temp=t(i)/tau
5         h(i)=(a+(1.-a)*temp)*exp(-temp)/tau

c ***************************************************************************
c Calculate the predicted output y
```

```
c ************************************************************************
      do 10 i=1,n
        yp(i)=0.
        do 20 j=1,i
20        yp(i)=yp(i)+h(j)*x(i+1-j)
        do 30 j=i+1,n
30        yp(i)=yp(i)+h(j)*base
        yp(i)=yp(i)*step
10    continue

c ************************************************************************
c Calculate the square of deviation of predicted y from the measured y
c ************************************************************************
      sum=0.
      do 40 i=1,n
        delta=yp(i)-y(i)
40      sum=sum+delta*delta
      type *,'J=',sum
      go to 100
      end



c ************************************************************************
c ************************************************************************
c Given a function (f), find the Fourier transform (g).
c Tapering and windowing are optional
c See the IMSL documentation on the length of the work vector.
c ************************************************************************
c ************************************************************************
      PROGRAM F-TRANS
c     Maxmium length of work vectors = 3*n+150
      parameter (nn=2000)
      character fname*40
      dimension iwk(1000),wk(1000),f(nn)
      complex*8 g(nn)

      write(*,600)' Enter filename to be converted: '
      read(*,500)fname
      open(11,file=fname,status='old')

      do 5 i=1,9999
5       read(11,*,end=6)f(i)
6     close(11)
      n=i-1
      if(n .gt. nn)stop 'Input file dimension is too big'

      write(*,600)' End point of taper ((0=default) if no tapering): '
        read(*,501)ntaper
        call TAPER(n,f,ntaper)

      call FFTRC(f,n,g,iwk,wk)
      do 10 i=2,n/2+1
10      g(n+2-i)=conjg(g(i))
```

```
      write(*,600)' Cutoff pt (0 if no freq windowing): '
        read(*,501)ncut
        call WINDOW(n,g,ncut)

      write(*,600)' Fourier transformed filename: '
        read(*,500)fname
        open(11,file=fname,status='new')
        write(11,604)(g(i),i=1,n)
        close(11)

c standard formats
500   format(a)
501   format(i5)
600   format(a\)
604   format(1pe13.5,e13.5)
      stop
      end



c ***************************************************************************
c ***************************************************************************
c Given the Fourier transform (g), find the inverse transform.
c Windowing is optional
c See the IMSL documentation on the length of the work vector.
c ***************************************************************************
c ***************************************************************************
      PROGRAM FI-TRANS
c     Maxmium length of work vectors ≅ 3*n+150
      parameter (nn=2000)
      character fname*40
      dimension iwk(1000),wk(1000)
      complex*8 g(nn),gwk(nn)

      write(*,600)' Filename to be converted: '
      read(*,500)fname
      open(11,file=fname,status='old')

      do 5 i=1,9999
        read(11,*,end=6)gr,gi
5       g(i)=cmplx(gr,-gi)
6     close(11)
      n=i-1
      if(n .gt. nn)stop 'Input file dimension is too big'

c Work in gwk
100   do 10 i=1,n
10      gwk(i)=g(i)
      write(*,600)' Cutoff pt (0 if no freq windowing): '
        read(*,501)ncut
        call WINDOW(n,gwk,ncut)

      call fftcc(gwk,n,iwk,wk)
```

```
      do 20 i=1,n
20      gwk(i)=conjg(gwk(i))/float(n)

      write(*,600)' Filename to store real results: '
        read(*,500)fname
        open(11,file=fname,status='new')
        write(11,603)(real(gwk(i)),i=1,n)
        close(11)

      write(*,600)' Save complex results (y/N)  '
        read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')then
        write(*,600)' Filename to store complex results: '
          read(*,500)fname
          open(11,file=fname,status='new')
          write(11,604)(gwk(i),i=1,n)
          close(11)
       endif

      write(*,600)' Try different windows (y/N)  '
        read(*,500)resp
      if(resp .eq. 'y' .or. resp .eq. 'Y')goto 100

c standard formats
500   format(a)
501   format(i5)
600   format(a\)
603   format(1pe13.5)
604   format(1pe13.5,e13.5)
      stop 'goodbye'
      end



c ***************************************************************************
c ***************************************************************************
C Find the impulse transfer function by the Fourier transform method.
C Difference operation is optional, and pts at either end can be neglected.
C Tapering and windowing are optional.
c ***************************************************************************
c ***************************************************************************
      PROGRAM FOURIER
      parameter (nn=1000)
      character fname1*40,fname2*40,resp*1
      dimension iwk(nn),wk(nn),x(nn),y(nn)
      complex*8 gx(nn),gy(nn),gk(nn)

      write(*,600)' Step size: '
        read(*,502)step
      write(*,600)' System input filename: '
        read(*,500)fname1
        open(11,file=fname1,status='old')
      write(*,600)' System output filename: '
        read(*,500)fname2
```

```
        open(12,file=fname2,status='old')

      do 5 i=1,9999
        read(11,*,end=6)x(i)
5       read(12,*,end=6)y(i)
6     close(11)
      close(12)
      n=i-1
      if(n .gt. nn)stop 'Input file dimension is too big'

c *************************************************************************
c Shift in the vertical axis
c *************************************************************************
      write(*,600)' Vertical shift for the system input: '
        read(*,502)xshift
      write(*,600)' Vertical shift for the system output: '
        read(*,502)yshift
      do 9 i=1,n
        x(i)=x(i)-xshift
9       y(i)=y(i)-yshift

c *************************************************************************
c Difference operations for x and y arrays
c *************************************************************************
      write(*,600)' enter number of difference operations: '
        read(*,501)idelta
      do 20, j=1,idelta
        do 10 i=1,n-idelta
          x(i)=x(i+1)-x(i)
10        y(i)=y(i+1)-y(i)
        do 11 i=n+1-idelta,n
          x(i)=0.
11        y(i)=0.
20    continue

      if(idelta .ge. 1)then
        write(*,600)' Record the differenced quantities  (y/N) '
        read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')then
          write(*,600)' Differenced system input filename: '
            read(*,500)fname1
            open(11,file=fname1,status='new')
            write(11,603)(x(i),i=1,n)
            close(11)
          write(*,600)' Differenced system output filename: '
            read(*,500)fname1
            open(11,file=fname1,status='new')
            write(11,603)(y(i),i=1,n)
            close(11)
        endif
      endif

c *************************************************************************
c Set the pts [1,ifirst) and (ilast,n] to be 0
```

```
c ***************************************************************************
      write(*,600)' First point to be included: '
        read(*,501)ifirst
      write(*,600)' Last point to be included: '
        read(*,501)ilast
      do 25 i=1,ifirst-1
        x(i)=0.
25      y(i)=0.
      do 26 i=ilast+1,n
        x(i)=0.
26      y(i)=0.

c ***************************************************************************
c Take Fourier transform of x and y
c ***************************************************************************
      write(*,600)' End point of taper ((0=default) if no tapering): '
        read(*,501)ntaper

      call TAPER(n,x,ntaper)
      call FFTRC(x,n,gx,iwk,wk)
      do 30 i=2,n/2+1
30      gx(n+2-i)=conjg(gx(i))

      call TAPER(n,y,ntaper)
      call FFTRC(y,n,gy,iwk,wk)
      do 40 i=2,n/2+1
40      gy(n+2-i)=conjg(gy(i))

      write(*,600)' Want to record the transformed quantities  (y/N) '
        read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')then
          write(*,600)' System input (F-transformed) filename: '
            read(*,500)fname1
            open(11,file=fname1,status='new')
            write(11,604)(gx(i),i=1,n)
            close(11)
          write(*,600)' System output (F-transformed) filename: '
            read(*,500)fname1
            open(11,file=fname1,status='new')
            write(11,604)(gy(i),i=1,n)
            close(11)
        endif

c ***************************************************************************
c Take the convolution division to get the transformed impulse response
c   function
c ***************************************************************************
      do 50 i=1,n
50      gk(i)=gy(i)/gx(i)/step

      write(*,600)' Record the F-transformed kernel function  (y/N) '
        read(*,500)resp
        if(resp .eq. 'y' .or. resp .eq. 'Y')then
        write(*,600)' Impulse transfer (F-transformed) filename: '
```

```
          read(*,500)fname1
          open(11,file=fname1,status='new')
          write(11,604)(gk(i),i=1,n)
          close(11)
        endif

c ***************************************************************************
c Take inverse Fourier transform to get the impulse response function
c ***************************************************************************
      write(*,600)' Cutoff pt (0(default) if no freq windowing): '
        read(*,501)ncut
      call WINDOW(n,gk,ncut)

c Remember to take the conjgate before calling FFTCC
      do 60 i=1,n
60      gk(i)=conjg(gk(i))
      call FFTCC(gk,n,iwk,wk)
      do 70 i=1,n
70      gk(i)=conjg(gk(i))/float(n)

c ***************************************************************************
c Take the real part of gk, because the impulse response function is real
c ***************************************************************************
      do 80 i=1,n
80      x(i)=real(gk(i))

      write(*,600)' Kernel function filename: '
        read(*,500)fname1
        open(11,file=fname1,status='new')
        write(11,603)(x(i),i=1,n)
        close(11)

c ***************************************************************************
c Standard formats
c ***************************************************************************
500   format(a)
501   format(i6)
502   format(e13.5)
600   format(a\)
603   format(1pe13.5)
604   format(1pe13.5,e13.5)
      stop 'goodbye'
      end



c ***************************************************************************
c ***************************************************************************
c Symmetrically taper a given function f(n), starting at ntaper.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE TAPER(n,f,ntaper)
      dimension f(n)
      pi=3.1415
```

```
      do 10 i=1,ntaper
      factor=(1.-cos(pi*float(i-1)/float(ntaper)))/2.
        f(i)=f(i)*factor
10      f(n-i+1)=f(n-i+1)*factor
      return
      end



c ***************************************************************************
c ***************************************************************************
c Look at a function g(n) through a 'cosine window'
c ncut = the point at which the function is cut off
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE WINDOW(n,g,ncut)
      complex*8 g(n)
      if(ncut .le. 0)return

      pi=3.1415
      do 10 i=1,ncut
        factor=(1.+cos(pi*float(i-1)/float(ncut)))/2.
10      g(i)=g(i)*factor
      do 20 i=ncut+1,n/2+1
20      g(i)=0.

c Fourier transform is symmetric
      do 30 i=2,n/2+1
30      g(n+2-i)=conjg(g(i))

      return
      end



c ***************************************************************************
c ***************************************************************************
c IMSL SUBROUTINE (See IMSL Manual for detail)
c Fast Fourier transform of a real valued sequence
c   a(n) ... input vector to be transformed
c   x(n/2+1) ... output complex vector
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE FFTRC(a,n,x,iwk,wk)
      integer             n,iwk(1)
      real                a(n),wk(1)
      complex             x(1)
      integer             nd2p1,nd2,i,mtwo,m,imax,nd4,np2,k,nmk,j
      real                rpi,zero,one,half,theta,tp,g(2),b(2),z(2),ai,
     1                    ar
      complex             ximag,alph,beta,gam,s1,zd
      equivalence         (gam,g(1)),(alph,b(1)),(z(1),ar),(z(2),ai),
     1                    (zd,z(1))
      data                zero/0.0/,half/0.5/,one/1.0/,imax/24/
```

```
      data               rpi/3.141593/

      if (n .ne. 2) go to 5
c n equal to 2
      zd = cmplx(a(1),a(2))
      theta = ar
      tp = ai
      x(2) = cmplx(theta-tp,zero)
      x(1) = cmplx(theta+tp,zero)
      go to 9005
    5 continue
c n greater than 2
      nd2 = n/2
      nd2p1 = nd2+1
c Move a to x
      j = 1
      do 6 i=1,nd2
         x(i) = cmplx(a(j),a(j+1))
         j = j+2
    6 continue
c Compute the center coefficient
      gam = cmplx(zero,zero)
      do 10 i=1,nd2
         gam = gam + x(i)
   10 continue
      tp = g(1)-g(2)
      gam = cmplx(tp,zero)
c Determine the smallest m such that n is less than or equal to 2**m
      mtwo = 2
      m = 1
      do 15 i=1,imax
         if (nd2 .le. mtwo) go to 20
         mtwo = mtwo+mtwo
         m = m+1
   15 continue
   20 if (nd2 .eq. mtwo) go to 25
c n is not a power of two, call fftcc
      call fftcc (x,nd2,iwk,wk)
      go to 30
c n is a power of two, call fft2c
   25 call fft2c (x,m,iwk)
   30 alph = x(1)
      x(1) = b(1) + b(2)
      nd4 = (nd2+1)/2
      if (nd4 .lt. 2) go to 40
      np2 = nd2 + 2
      theta = rpi/nd2
      tp = theta
      ximag = cmplx(zero,one)
c Decompose the complex vector xc into the components of the transform
c   of the input data.
      do 35 k = 2,nd4
         nmk = np2 - k
         s1 = conjg(x(nmk))
```

```
         alph = x(k) + s1
         beta = ximag*(s1-x(k))
         s1 = cmplx(cos(theta),sin(theta))
         x(k) = (alph+beta*s1)*half
         x(nmk) = conjg(alph-beta*s1)*half
         theta = theta + tp
   35 continue
   40 continue
      x(nd2p1) = gam
 9005 return
      end



c ***************************************************************************
c ***************************************************************************
c IMSL SUBROUTINE (See IMSL Manual for detail)
c Fast Fourier transform of a complex valued sequence
c   a(n) ... input/output vector
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE FFTCC (a,n,iwk,wk)
      integer             n,iwk(1)
      real                wk(1)
      complex             a(n)
      integer             i,iam,iap,ibm,ibp,ic,icc,icf,ick,id,idm1,ii,
     1                    ija,ikb,ikt,ill,im,ird,isf,isk,isp,iss,ita,itb,
     2                    j,ja,jf,jj,jk,k,k0,k1,k2,k3,ka,kb,kd2,kf,kh,kn,
     3                    kt,ktp,l,l1,m,mm,mm1,mp
      real                cm,sm,c1,c2,c3,s1,s2,s3,c30,rad,a0,a1,a4,b4,
     1                    a2,a3,b0,b1,b2,b3,zero,half,one,two,z0(2),
     2                    z1(2),z2(2),z3(2),z4(2)
      complex             za0,za1,za2,za3,za4,ak2
      equivalence         (za0,z0(1)),(za1,z1(1)),(za2,z2(1)),
     1                    (za3,z3(1)),(a0,z0(1)),(b0,z0(2)),(a1,z1(1)),
     2                    (b1,z1(2)),(a2,z2(1)),(b2,z2(2)),(a3,z3(1)),
     3                    (b3,z3(2)),(za4,z4(1)),(z4(1),a4),(z4(2),b4)
      data                rad/6.283185/,
     1                    c30/.8660254/
      data                zero,half,one,two/0.0,0.5,1.0,2.0/

      if (n .eq. 1) go to 9005
      k = n
      m = 0
      j = 2
      jj = 4
      jf = 0
c Determine the square factors of n
      iwk(1) = 1
    5 i = k/jj
      if (i*jj .ne. k) go to 10
      m = m+1
      iwk(m+1) = j
      k = i
```

```
      go to 5
   10 j = j + 2
      if (j .eq. 4) j = 3
      jj = j * j
      if (jj .le. k) go to 5
      kt = m
c Determine the remaining factors of n
      j = 2
   15 i = k / j
      if (i*j .ne. k) go to 20
      m = m + 1
      iwk(m+1) = j
      k = i
      go to 15
   20 j = j + 1
      if (j .eq. 3) go to 15
      j = j + 1
      if(j.le.k) go to 15
      k = iwk(m+1)
      if (iwk(kt+1) .gt. iwk(m+1)) k = iwk(kt+1)
      if(kt.le.0) go to 30
      ktp = kt + 2
      do 25  i = 1,kt
         j = ktp - i
         m = m+1
         iwk(m+1) = iwk(j)
   25 continue
   30 mp = m+1
      ic = mp+1
      id = ic+mp
      ill = id+mp
      ird = ill+mp+1
      icc = ird+mp
      iss = icc+mp
      ick = iss+mp
      isk = ick+k
      icf = isk+k
      isf = icf+k
      iap = isf+k
      kd2 = (k-1) / 2 + 1
      ibp = iap + kd2
      iam = ibp + kd2
      ibm = iam + kd2
      mm1 = m-1
      i = 1
   35 l = mp - i
      j = ic - i
      iwk(ill+l) = 0
      if ((iwk(j-1) + iwk(j)) .eq. 4) iwk(ill+l) = 1
      if (iwk(ill+l) .eq. 0) go to 40
      i = i + 1
      l = l - 1
      iwk(ill+l) = 0
   40 i = i + 1
```

```
      if(i.le.mm1) go to 35
      iwk(ill+1) = 0
      iwk(ill+mp) = 0
      iwk(ic) = 1
      iwk(id) = n
      do 45  j = 1,m
         k = iwk(j+1)
         iwk(ic+j) = iwk(ic+j-1) * k
         iwk(id+j) = iwk(id+j-1) / k
         wk(ird+j) = rad/iwk(ic+j)
         c1 = rad/k
         if (k .le. 2) go to 45
         wk(icc+j) = cos(c1)
         wk(iss+j) = sin(c1)
   45 continue
      mm = m
      if (iwk(ill+m) .eq. 1) mm = m - 1
      if (mm .le. 1) go to 50
      sm = iwk(ic+mm-2) * wk(ird+m)
      cm = cos(sm)
      sm = sin(sm)
   50 kb = 0
      kn = n
      jj = 0
      i = 1
      c1 = one
      s1 = zero
      l1 = 1
   55 if (iwk(ill+i+1) .eq. 1) go to 60
      kf = iwk(i+1)
      go to 65
   60 kf = 4
      i = i+1
   65 isp = iwk(id+i)
      if (l1 .eq. 1) go to 70
      s1 = jj * wk(ird+i)
      c1 = cos(s1)
      s1 = sin(s1)
c Factors of 2, 3, and 4 are handled separately.
   70 if (kf .gt. 4) go to 140
      go to (75,75,90,115), kf
   75 k0 = kb + isp
      k2 = k0 + isp
      if (l1 .eq. 1) go to 85
   80 k0 = k0 - 1
      if (k0 .lt. kb) go to 190
      k2 = k2 - 1
      za4 = a(k2+1)
      a0 = a4*c1-b4*s1
      b0 = a4*s1+b4*c1
      a(k2+1) = a(k0+1)-za0
      a(k0+1) = a(k0+1)+za0
      go to 80
   85 k0 = k0 - 1
```

```
      if (k0 .lt. kb) go to 190
      k2 = k2 - 1
      ak2 = a(k2+1)
      a(k2+1) = a(k0+1)-ak2
      a(k0+1) = a(k0+1)+ak2
      go to 85
   90 if (l1 .eq. 1) go to 95
      c2 = c1 * c1 - s1 * s1
      s2 = two * c1 * s1
   95 ja = kb + isp - 1
      ka = ja + kb
      ikb = kb+1
      ija = ja+1
      do 110 ii = ikb,ija
         k0 = ka - ii + 1
         k1 = k0 + isp
         k2 = k1 + isp
         za0 = a(k0+1)
         if (l1 .eq. 1) go to 100
         za4 = a(k1+1)
         a1 = a4*c1-b4*s1
         b1 = a4*s1+b4*c1
         za4 = a(k2+1)
         a2 = a4*c2-b4*s2
         b2 = a4*s2+b4*c2
         go to 105
  100    za1 = a(k1+1)
         za2 = a(k2+1)
  105    a(k0+1) = cmplx(a0+a1+a2,b0+b1+b2)
         a0 = -half * (a1+a2) + a0
         a1 = (a1-a2) * c30
         b0 = -half * (b1+b2) + b0
         b1 = (b1-b2) * c30
         a(k1+1) = cmplx(a0-b1,b0+a1)
         a(k2+1) = cmplx(a0+b1,b0-a1)
  110 continue
      go to 190
  115 if (l1 .eq. 1) go to 120
      c2 = c1 * c1 - s1 * s1
      s2 = two * c1 * s1
      c3 = c1 * c2 - s1 * s2
      s3 = s1 * c2 + c1 * s2
  120 ja = kb + isp - 1
      ka = ja + kb
      ikb = kb+1
      ija = ja+1
      do 135 ii = ikb,ija
         k0 = ka - ii + 1
         k1 = k0 + isp
         k2 = k1 + isp
         k3 = k2 + isp
         za0 = a(k0+1)
         if (l1 .eq. 1) go to 125
         za4 = a(k1+1)
```

```
      a1 = a4*c1-b4*s1
      b1 = a4*s1+b4*c1
      za4 = a(k2+1)
      a2 = a4*c2-b4*s2
      b2 = a4*s2+b4*c2
      za4 = a(k3+1)
      a3 = a4*c3-b4*s3
      b3 = a4*s3+b4*c3
      go to 130
125   za1 = a(k1+1)
      za2 = a(k2+1)
      za3 = a(k3+1)
130   a(k0+1) = cmplx(a0+a2+a1+a3,b0+b2+b1+b3)
      a(k1+1) = cmplx(a0+a2-a1-a3,b0+b2-b1-b3)
      a(k2+1) = cmplx(a0-a2-b1+b3,b0-b2+a1-a3)
      a(k3+1) = cmplx(a0-a2+b1-b3,b0-b2-a1+a3)
135 continue
    go to 190
140 jk = kf - 1
    kh = jk/2
    k3 = iwk(id+i-1)
    k0 = kb + isp
    if (l1 .eq. 1) go to 150
    k = jk - 1
    wk(icf+1) = c1
    wk(isf+1) = s1
    do 145 j = 1,k
       wk(icf+j+1) = wk(icf+j) * c1 - wk(isf+j) * s1
       wk(isf+j+1) = wk(icf+j) * s1 + wk(isf+j) * c1
145 continue
150 if (kf .eq. jf) go to 160
    c2 = wk(icc+i)
    wk(ick+1) = c2
    wk(ick+jk) = c2
    s2 = wk(iss+i)
    wk(isk+1) = s2
    wk(isk+jk) = -s2
    do 155 j = 1,kh
       k = jk - j
       wk(ick+k) = wk(ick+j) * c2 - wk(isk+j) * s2
       wk(ick+j+1) = wk(ick+k)
       wk(isk+j+1) = wk(ick+j) * s2 + wk(isk+j) * c2
       wk(isk+k) = -wk(isk+j+1)
155 continue
160 k0 = k0 - 1
    k1 = k0
    k2 = k0 + k3
    za0 = a(k0+1)
    a3 = a0
    b3 = b0
    do 175 j = 1,kh
       k1 = k1 + isp
       k2 = k2 - isp
       if (l1 .eq. 1) go to 165
```

```
      k = kf - j
      za4 = a(k1+1)
      a1 = a4*wk(icf+j)-b4*wk(isf+j)
      b1 = a4*wk(isf+j)+b4*wk(icf+j)
      za4 = a(k2+1)
      a2 = a4*wk(icf+k)-b4*wk(isf+k)
      b2 = a4*wk(isf+k)+b4*wk(icf+k)
      go to 170
165   za1 = a(k1+1)
      za2 = a(k2+1)
170   wk(iap+j) = a1 + a2
      wk(iam+j) = a1 - a2
      wk(ibp+j) = b1 + b2
      wk(ibm+j) = b1 - b2
      a3 = a1 + a2 + a3
      b3 = b1 + b2 + b3
175 continue
    a(k0+1) = cmplx(a3,b3)
    k1 = k0
    k2 = k0 + k3
    do 185 j = 1,kh
       k1 = k1 + isp
       k2 = k2 - isp
       jk = j
       a1 = a0
       b1 = b0
       a2 = zero
       b2 = zero
       do 180  k = 1,kh
          a1 = a1 + wk(iap+k) * wk(ick+jk)
          a2 = a2 + wk(iam+k) * wk(isk+jk)
          b1 = b1 + wk(ibp+k) * wk(ick+jk)
          b2 = b2 + wk(ibm+k) * wk(isk+jk)
          jk = jk + j
          if (jk .ge. kf) jk = jk - kf
180    continue
       a(k1+1) = cmplx(a1-b2,b1+a2)
       a(k2+1) = cmplx(a1+b2,b1-a2)
185 continue
    if (k0 .gt. kb) go to 160
    jf = kf
190 if ( i .ge. mm ) go to 195
    i = i + 1
    go to 55
195 i = mm
    l1 = 0
    kb = iwk(id+i-1) + kb
    if (kb .ge. kn) go to 215
200 jj = iwk(ic+i-2) + jj
    if (jj .lt. iwk(ic+i-1)) go to 205
    i = i - 1
    jj = jj - iwk(ic+i)
    go to 200
205 if (i .ne. mm) go to 210
```

```
      c2 = c1
      c1 = cm * c1 - sm * s1
      s1 = sm * c2 + cm * s1
      go to 70
  210 if (iwk(ill+i) .eq. 1) i = i + 1
      go to 55
  215 i = 1
      ja = kt - 1
      ka = ja + 1
      if(ja.lt.1) go to 225
      do 220  ii = 1,ja
         j = ka - ii
         iwk(j+1) = iwk(j+1) - 1
         i = iwk(j+1) + i
  220 continue
c The result is now permuted to normal order.
  225 if (kt .le. 0) go to 270
      j = 1
      i = 0
      kb = 0
  230 k2 = iwk(id+j) + kb
      k3 = k2
      jj = iwk(ic+j-1)
      jk = jj
      k0 = kb + jj
      isp = iwk(ic+j) - jj
  235 k = k0 + jj
  240 za4 = a(k0+1)
      a(k0+1) = a(k2+1)
      a(k2+1) = za4
      k0 = k0 + 1
      k2 = k2 + 1
      if (k0 .lt. k) go to 240
      k0 = k0 + isp
      k2 = k2 + isp
      if (k0 .lt. k3) go to 235
      if (k0 .ge. k3 + isp) go to 245
      k0 = k0 - iwk(id+j) + jj
      go to 235
  245 k3 = iwk(id+j) + k3
      if (k3 - kb .ge. iwk(id+j-1)) go to 250
      k2 = k3 + jk
      jk = jk + jj
      k0 = k3 - iwk(id+j) + jk
      go to 235
  250 if (j .ge. kt) go to 260
      k = iwk(j+1) + i
      j = j + 1
  255 i = i + 1
      iwk(ill+i) = j
      if (i .lt. k) go to 255
      go to 230
  260 kb = k3
      if (i .le. 0) go to 265
```

```
      j = iwk(ill+i)
      i = i - 1
      go to 230
  265 if (kb .ge. n) go to 270
      j = 1
      go to 230
  270 jk = iwk(ic+kt)
      isp = iwk(id+kt)
      m = m - kt
      kb = isp/jk-2
      if (kt .ge. m-1 ) go to 9005
      ita = ill+kb+1
      itb = ita+jk
      idm1 = id-1
      ikt = kt+1
      im = m+1
      do 275 j = ikt,im
         iwk(idm1+j) = iwk(idm1+j)/jk
  275 continue
      jj = 0
      do 290 j = 1,kb
         k = kt
  280    jj = iwk(id+k+1) + jj
         if (jj .lt. iwk(id+k)) go to 285
         jj = jj - iwk(id+k)
         k = k + 1
         go to 280
  285    iwk(ill+j) = jj
         if (jj .eq. j) iwk(ill+j) = -j
  290 continue
c Determine the permutation cycles of length greater than or equal to two.
      do 300  j = 1,kb
         if (iwk(ill+j) .le. 0) go to 300
         k2 = j
  295    k2 = iabs(iwk(ill+k2))
         if (k2 .eq. j) go to 300
         iwk(ill+k2) = -iwk(ill+k2)
         go to 295
  300 continue
c Reorder a following the permutation cycles
      i = 0
      j = 0
      kb = 0
      kn = n
  305 j = j + 1
      if (iwk(ill+j) .lt. 0) go to 305
      k = iwk(ill+j)
      k0 = jk * k + kb
  310 za4 = a(k0+i+1)
      wk(ita+i) = a4
      wk(itb+i) = b4
      i = i + 1
      if (i .lt. jk) go to 310
      i = 0
```

```
  315 k = -iwk(ill+k)
      jj = k0
      k0 = jk * k + kb
  320 a(jj+i+1) = a(k0+i+1)
      i = i + 1
      if (i .lt. jk) go to 320
      i = 0
      if (k .ne. j) go to 315
  325 a(k0+i+1) = cmplx(wk(ita+i),wk(itb+i))
      i = i + 1
      if (i .lt. jk) go to 325
      i = 0
      if (j .lt. k2) go to 305
      j = 0
      kb = kb + isp
      if (kb .lt. kn) go to 305
 9005 return
      end


c ***********************************************************************
c ***********************************************************************
c IMSL SUBROUTINE (See IMSL Manual for detail)
c Fast Fourier transform of a complex valued sequence of power two
c   a(n) ... input/output vector
c ***********************************************************************
c ***********************************************************************
      SUBROUTINE  FFT2C (a,m,iwk)
      integer            m,iwk(1)
      complex            a(1)
      integer            i,isp,j,jj,jsp,k,k0,k1,k2,k3,kb,kn,mk,mm,mp,n,
     1                   n4,n8,n2,lm,nn,jk
      real               rad,c1,c2,c3,s1,s2,s3,ck,sk,sq,a0,a1,a2,a3,
     1                   b0,b1,b2,b3,twopi,temp,
     2                   zero,one,z0(2),z1(2),z2(2),z3(2)
      complex            za0,za1,za2,za3,ak2
      equivalence        (za0,z0(1)),(za1,z1(1)),(za2,z2(1)),
     1                   (za3,z3(1)),(a0,z0(1)),(b0,z0(2)),(a1,z1(1)),
     2                   (b1,z1(2)),(a2,z2(1)),(b2,z2(2)),(a3,z3(1)),
     3                   (b3,z3(2))
      data               sq/.7071068/,
     1                   sk/.3826834/,
     2                   ck/.9238795/,
     3                   twopi/6.283185/
      data               zero/0.0/,one/1.0/

      mp = m+1
      n = 2**m
      iwk(1) = 1
      mm = (m/2)*2
      kn = n+1
c Initialize work vector
      do 5  i=2,mp
```

```
         iwk(i) = iwk(i-1)+iwk(i-1)
    5 continue
      rad = twopi/n
      mk = m - 4
      kb = 1
      if (mm .eq. m) go to 15
      k2 = kn
      k0 = iwk(mm+1) + kb
   10 k2 = k2 - 1
      k0 = k0 - 1
      ak2 = a(k2)
      a(k2) = a(k0) - ak2
      a(k0) = a(k0) + ak2
      if (k0 .gt. kb) go to 10
   15 c1 = one
      s1 = zero
      jj = 0
      k = mm - 1
      j = 4
      if (k .ge. 1) go to 30
      go to 70
   20 if (iwk(j) .gt. jj) go to 25
      jj = jj - iwk(j)
      j = j-1
      if (iwk(j) .gt. jj) go to 25
      jj = jj - iwk(j)
      j = j - 1
      k = k + 2
      go to 20
   25 jj = iwk(j) + jj
      j = 4
   30 isp = iwk(k)
      if (jj .eq. 0) go to 40
c Reset trigonometric parameters
      c2 = jj * isp * rad
      c1 = cos(c2)
      s1 = sin(c2)
   35 c2 = c1 * c1 - s1 * s1
      s2 = c1 * (s1 + s1)
      c3 = c2 * c1 - s2 * s1
      s3 = c2 * s1 + s2 * c1
   40 jsp = isp + kb
c Determine fourier coefficients in groups of 4
      do 50 i=1,isp
         k0 = jsp - i
         k1 = k0 + isp
         k2 = k1 + isp
         k3 = k2 + isp
         za0 = a(k0)
         za1 = a(k1)
         za2 = a(k2)
         za3 = a(k3)
         if (s1 .eq. zero) go to 45
         temp = a1
```

```
         a1 = a1 * c1 - b1 * s1
         b1 = temp * s1 + b1 * c1
         temp = a2
         a2 = a2 * c2 - b2 * s2
         b2 = temp * s2 + b2 * c2
         temp = a3
         a3 = a3 * c3 - b3 * s3
         b3 = temp * s3 + b3 * c3
   45    temp = a0 + a2
         a2 = a0 - a2
         a0 = temp
         temp = a1 + a3
         a3 = a1 - a3
         a1 = temp
         temp = b0 + b2
         b2 = b0 - b2
         b0 = temp
         temp = b1 + b3
         b3 = b1 - b3
         b1 = temp
         a(k0) = cmplx(a0+a1,b0+b1)
         a(k1) = cmplx(a0-a1,b0-b1)
         a(k2) = cmplx(a2-b3,b2+a3)
         a(k3) = cmplx(a2+b3,b2-a3)
   50 continue
      if (k .le. 1) go to 55
      k = k - 2
      go to 30
   55 kb = k3 + isp
c Check for completion of final iteration
      if (kn .le. kb) go to 70
      if (j .ne. 1) go to 60
      k = 3
      j = mk
      go to 20
   60 j = j - 1
      c2 = c1
      if (j .ne. 2) go to 65
      c1 = c1 * ck + s1 * sk
      s1 = s1 * ck - c2 * sk
      go to 35
   65 c1 = (c1 - s1) * sq
      s1 = (c2 + s1) * sq
      go to 35
   70 continue
c Permute the complex vector in reverse binary order to normal order
      if(m .le. 1) go to 9005
      mp = m+1
      jj = 1
c Initialize work vector
      iwk(1) = 1
      do 75  i = 2,mp
         iwk(i) = iwk(i-1) * 2
   75 continue
```

```
      n4 = iwk(mp-2)
      if (m .gt. 2) n8 = iwk(mp-3)
      n2 = iwk(mp-1)
      lm = n2
      nn = iwk(mp)+1
      mp = mp-4
c Determine indices and switch a
      j = 2
   80 jk = jj + n2
      ak2 = a(j)
      a(j) = a(jk)
      a(jk) = ak2
      j = j+1
      if (jj .gt. n4) go to 85
      jj = jj + n4
      go to 105
   85 jj = jj - n4
      if (jj .gt. n8) go to 90
      jj = jj + n8
      go to 105
   90 jj = jj - n8
      k = mp
   95 if (iwk(k) .ge. jj) go to 100
      jj = jj - iwk(k)
      k = k - 1
      go to 95
  100 jj = iwk(k) + jj
  105 if (jj .le. j) go to 110
      k = nn - j
      jk = nn - jj
      ak2 = a(j)
      a(j) = a(jj)
      a(jj) = ak2
      ak2 = a(k)
      a(k) = a(jk)
      a(jk) = ak2
  110 j = j + 1
c Cycle repeated until limiting number of changes is achieved
      if (j .le. lm) go to 80

 9005 return
      end



c *************************************************************************
c *************************************************************************
c Simulate the biomass and substrate conc. in a time delay kernel model
c  with constant or sinusoidal dilution rate.
c By setting amp=0., constant dilution rate is also simulated.
c By setting tau=0., the Monod model is also simulated.
c By setting ub=0., the Monod model is also simulated.
c By setting us=0., the Monod model is also simulated.
c By setting D=0., batch fermentation is also simulated.
```

```
c By setting Yb=0., constant yield coefficient is also simulated.
c The program stops if "9999" is the answer to the question:
c  'up to which pt are these dilution rate valid  '
c For the sinusoidal case, report the last high and low values of B(biomass),
c  S(substrate), MU(intrinsic specific growth rate), Y(observed specific
c  growth rate), and D(dilution rate).
c ***************************************************************************
c ***************************************************************************
      PROGRAM SIMULATE
      parameter (n=4)
      character resps*1,respw*1
      external deriv1,deriv2
      real Ks,Ki
      dimension x(n),xdot(n),c(24),w(n,9)
      dimension tmin(5),xmin(5),tmax(5),xmax(5),xold(5),xup(5)
      common/c/um,Ks,Ki,Ya,Yb,Yc,Yd,Ye,ub,us,a,tau,D,sf

c ***************************************************************************
c constant dilution rate or sinusoidal dilution rate
c ***************************************************************************
      write(*,600)' Input simusoidal dilution  (y/N) '
        read(*,500)resps
        if(resps .eq. 'y' .or. resps .eq. 'Y')then
          write(*,600)' Write individual files  (y/N) '
          read(*,500)respw
        else
          respw='Y'
        endif

c ***************************************************************************
c enter model parameters
c ***************************************************************************
      write(*,500)' u(s)=um*s/(Ks+s+Ki*s*s) '
        write(*,600)' um:  '
          read(*,502)um
        write(*,600)' Ks:  '
          read(*,502)Ks
        write(*,600)' Ki:  '
          read(*,502)Ki
      write(*,500)' Y(s)=Ya + (Yb+Yc*s)/(Yd+Ye*s) '
        write(*,500)' Monod  model: Ya=   Yb=0. Yc=0. Yd=1. Ye=0. '
        write(*,500)' Linear model: Ya=   Yb=0. Yc=   Yd=1. Ye=0. '
        write(*,500)' ub     model: Ya=   Yb=-  Yc=0. Yd=0. Ye=1. '
        write(*,500)' us     model: Ya=0. Yb=0. Yc=   Yd=   Ye=1. '
        write(*,600)' Constant part of Ys (Ya):  '
          read(*,502)Ya
        write(*,600)' Variable part of Ys (Yb):  '
          read(*,502)Yb
        write(*,600)' Variable part of Ys (Yc):  '
          read(*,502)Yc
        write(*,600)' Variable part of Ys (Yd):  '
          read(*,502)Yd
        write(*,600)' Variable part of Ys (Ye):  '
          read(*,502)Ye
```

```
      write(*,600)' Mintenence term in biomass equation (ub):  '
        read(*,502)ub
      write(*,600)' Mintenence term in substrate equation (us):  '
        read(*,502)us
      write(*,600)' Fraction of 0th order kernel (a):   '
        read(*,502)a
      write(*,600)' tau: '
        read(*,502)tau
      write(*,600)' Sf:  '
        read(*,502)sf
      write(*,600)' Step size:  '
        read(*,502)step
      write(*,600)' Constant Dilution rate (hr-1): '
        read(*,502)D0
        D=D0
        Damp=0.
        Dfreq=0.
      write(*,600)' up to which pt are these dilution rate valid  '
        read(*,501)iend
      ibegin=1

c *****************************************************************************
c open files
c *****************************************************************************
      if(respw .eq. 'y' .or. respw .eq. 'Y')then
        open(1,file='out.B',status='new')
        open(2,file='out.S',status='new')
        open(3,file='out.mu',status='new')
        open(4,file='out.y',status='new')
        open(5,file='out.D',status='new')
      endif

c *****************************************************************************
c calculate initial steady state values
c *****************************************************************************
      alpha=um/D
      beta=Ks/sf
      betai=Ki*sf
      delta=1.+ub/D
      gamma=us/D
      if(Ki .eq. 0.)then
        s0=sf*beta*delta/(alpha-delta)
      else
        temp=(alpha-delta)*(alpha-delta)-4.*delta*delta*beta*betai
        s0=sf*((alpha-delta)-sqrt(temp))/2./delta/betai
      endif
      yy=Ya + (Yb+Yc*s0)/(Yd+Ye*s0)
      b0=yy*(sf-s0)/(1.+yy*gamma)
      write(*,651)b0,s0

      x(1) = b0
      x(2) = s0
      x(3) = D
      x(4) = 0.
```

```
      index=1
      tol=1.e-4

100   do 200 i=ibegin,iend
      timend=float(i)*step
      time  =float(i-1)*step
      if(resps .eq. 'y' .or. resps .eq. 'Y')D=DO+Damp*sin(Dfreq*time)
      if(tau .gt. 0.)then
        call dverk(n,deriv1,time,x,timend,tol,index,c,n,w,ier)
      else
        call dverk(2,deriv2,time,x,timend,tol,index,c,n,w,ier)
      endif
      xmu=um*x(2)/(ks+x(2))
c *****************************************************************************
c     find the last maximum and minimum values
c     xup(*)=0 ... the curve was moving down the last time
c     xup(*)=1 ... the curve was moving up the last time
      if(resps .eq. 'y' .or. resps .eq. 'Y')then
c       switch in slope from negative to positive (i.e. detect MIN)
        if(xold(1) .lt. x(1) .and. xup(1) .eq. 0)then
          xup(1)=1
          xmin(1)=xold(1)
          tmin(1)=time-step
        endif
        if(xold(2) .lt. x(2) .and. xup(2) .eq. 0)then
          xup(2)=1
          xmin(2)=xold(2)
          tmin(2)=time-step
        endif
        if(xold(3) .lt. xmu .and. xup(3) .eq. 0)then
          xup(3)=1
          xmin(3)=xold(3)
          tmin(3)=time-step
        endif
        if(xold(4) .lt. x(3) .and. xup(4) .eq. 0)then
          xup(4)=1
          xmin(4)=xold(4)
          tmin(4)=time-step
        endif
        if(xold(5) .lt. D .and. xup(5) .eq. 0)then
          xup(5)=1
          xmin(5)=xold(5)
          tmin(5)=time-step-step
        endif
c *****************************************************************************
c       switch in slope from positive to negative (i.e. detect MAX)
        if(xold(1) .gt. x(1) .and. xup(1) .eq. 1)then
          xup(1)=0
          xmax(1)=xold(1)
          tmax(1)=time-step
        endif
        if(xold(2) .gt. x(2) .and. xup(2) .eq. 1)then
          xup(2)=0
```

```
          xmax(2)=xold(2)
          tmax(2)=time-step
        endif
        if(xold(3) .gt. xmu .and. xup(3) .eq. 1)then
          xup(3)=0
          xmax(3)=xold(3)
          tmax(3)=time-step
        endif
        if(xold(4) .gt. x(3) .and. xup(4) .eq. 1)then
          xup(4)=0
          xmax(4)=xold(4)
          tmax(4)=time-step
        endif
        if(xold(5) .gt. D .and. xup(5) .eq. 1)then
          xup(5)=0
          xmax(5)=xold(5)
          tmax(5)=time-step-step
        endif
c ***************************************************************************
c       save the current value for next comparision
        xold(1)=x(1)
        xold(2)=x(2)
        xold(3)=xmu
        xold(4)=x(3)
        xold(5)=D
      endif
      if(respw .eq. 'y' .or. respw .eq. 'Y')then
        write(1,604)x(1)
        write(2,604)x(2)
        write(3,604)xmu
        write(4,604)x(3)
        write(5,604)D
      endif

c ***************************************************************************
c check for integration errors
      if(index.ge.0 .and. ier.le.0) go to 200
      write(*,97)timend
97    format(' error in deriv at t= ',f8.2)
      write(*,98)index
98    format(' index =              ',i4)
      write(*,99)ier
99    format(' ier =                ',i4)
      tol = tol*1.1
      write(*,182)tol
182   format(' tol = ',f8.5)

200   continue

c ***************************************************************************
c read in new dilution rate
c ***************************************************************************
      if(resps .eq. 'y' .or. resps .eq. 'Y')then
        write(*,600)' enter average Dilution rate (hr-1): '
```

```
        read(*,502)D0
        write(*,600)' enter amplitute in hr-1: '
        read(*,502)Damp
        write(*,600)' enter frequency (=2*pi/period) in hr-1: '
        read(*,502)Dfreq
      else
        write(*,600)' enter constant Dilution rate (hr-1): '
        read(*,502)D
      endif
      write(*,600)' up to which pt are these dilution rate valid  '
      read(*,501)iend
      ibegin=i
      if((iend .lt. ibegin) .or. (iend .eq. 9999))goto 999
      goto 100

999   if(respw .eq. 'y' .or. respw .eq. 'Y')then
        close(1)
        close(2)
        close(3)
        close(4)
      endif
      if(resps .eq. 'y' .or. resps .eq. 'Y')then
        write(*,650)
650     format('     tmin          xmin          tmax          xmax')
        write(*,605)tmin(1),xmin(1),tmax(1),xmax(1)
        write(*,605)tmin(2),xmin(2),tmax(2),xmax(2)
        write(*,605)tmin(3),xmin(3),tmax(3),xmax(3)
        write(*,605)tmin(4),xmin(4),tmax(4),xmax(4)
        write(*,605)tmin(5),xmin(5),tmax(5),xmax(5)
      endif

c ***************************************************************************
c some standard formats
c ***************************************************************************
500   format(a)
501   format(i6)
502   format(e13.5)
600   format(a\)
604   format(1pe13.5)
605   format(1pe13.5,5e13.5)
651   format(' b0= ',1pe13.5,' s0= ',e13.5)
      stop 'goodbye'
      end



c ***************************************************************************
c ***************************************************************************
c 0th and 1st order time delay kernel model for biomass and substrate
c Variable yield coefficient and maintenance terms can be included
c Used by PROGRAM SIMULATE
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE DERIV1(N,TIME,X,XDOT)
```

```
      real Ks,Ki
      dimension x(n),xdot(n)
      common/c/um,Ks,Ki,Ya,Yb,Yc,Yd,Ye,ub,us,a,tau,D,sf

      KKs=Ks/1000.
      factor=x(2)/(KKs+x(2))

      bottom=Ks + x(2) + Ki*x(2)*x(2)
      uu    =um*x(2)/bottom
      duu=um*(Ks-Ki*x(2)*x(2))/bottom/bottom
      yy=Ya + (Yb+Yc*x(2))/(Yd+Ye*x(2))

      xdot(1) = (x(3)*factor-D-ub)*x(1)
      xdot(2) = D*(sf-x(2))-x(3)*factor*x(1)/yy-us*x(1)
      xdot(3) = x(4)
      xdot(4) = -2.*x(4)/tau-x(3)/tau/tau+uu/tau/tau+a*duu*xdot(2)/tau
      return
      end



c ***************************************************************************
c ***************************************************************************
c Plain Monod model for biomass and substrate
c Variable yield coefficient and maintenance terms can be included
c Used by PROGRAM SIMULATE
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE DERIV2(N,TIME,X,XDOT)
      real Ks,Ki
      dimension x(n),xdot(n)
      common/c/um,Ks,Ki,Ya,Yb,Yc,Yd,Ye,ub,us,a,tau,D,sf

      KKs=Ks/1000.
      factor=   x(2)/(KKs+x(2))

      uu    =um*x(2)/(Ks +x(2)+Ki*x(2)*x(2))
      yy=Ya + (Yb+Yc*x(2))/(Yd+Ye*x(2))

      xdot(1) = (uu*factor-D-ub)*x(1)
      xdot(2) = D*(sf-x(2))-uu*factor*x(1)/yy-us*x(1)
      return
      end
```

APPENDIX E

APPLICATION OF MACROSCOPIC BALANCES TO THE IDENTIFICATION OF GROSS MEASUREMENT ERRORS

(The text of Appendix E consists of an article coauthored with G. N. Stephanopoulos which has appeared in *Biotechnology & Bioengineering*, **25**, 2177-2208, 1983.)

Application of Macroscopic Balances to the Identification of Gross Measurement Errors

NAM SUN WANG and GREGORY STEPHANOPOULOS, *Chemical Engineering Department, California Institute of Technology, Pasadena, California 91125*

Summary

A systematic method is presented which is capable of both detecting the presence of grossly biased measurement errors and locating the source of these errors in a bioreactor through statistical hypothesis testing. Equality constraints derived from material and energy balances are employed for the detection of data inconsistencies and for the subsequent identification of the suspect measurements by a process of data analysis and rectification. Maximum likelihood techniques are applied to the estimation of the states and parameters of the bioreactor after the suspect measurements have been eliminated. The level of significance is specified by the experimenter while the measurements are assumed to be randomly, normally distributed with zero mean and known variances. Two different approaches of data analysis, batchwise and sequential, that lead to a consistent set of adjustments on the experimental values, are discussed. Several examples based on the fermentation data taken from literature sources are presented to demonstrate the utility of the proposed method, and one set of data is solved numerically to illustrate the computational aspect of the algorithm.

INTRODUCTION

At present, highly accurate and reliable measurements on a fermentation process are frequently difficult to obtain. Moreover, due to various types of errors in measurements, these raw measurements rarely form a consistent set of data which satisfies exactly the energy and material balance equations. These errors may originate in random or systematic sources and may be significantly large. Consequently, the control and optimization of a process with state estimates that are based on grossly erroneous measurements may result in an unstable or nonoptimal operation. Furthermore, experimental studies of microbial growth kinetics based on biased data will certainly lead to false conclusions and unreliable models. The purpose of this article is to demonstrate how a relatively small amount of extra effort in data analysis can allow an experimenter to detect the presence of gross errors, to locate their sources if gross errors are indeed found to exist, and to rectify the data in an optimal manner.

The proposed method makes use of the macroscopic and elemental balances applicable to a biological process. These balances have been employed

Biotechnology and Bioengineering, Vol. XXV, Pp. 2177–2208 (1983)
 CCC 0006-3592/83/092177-32$04.20

in the past for the on-line reactor identification by off-gas analysis.[1] However, they can also be used to test the consistency of data in an overdetermined system. A system overdetermination results when the number of variables that are directly measured is larger than what is needed for the full identification of the system by the application of the available constraints. Erickson and co-workers[2-4] have tested the consistency of various sets of data with regard to the above balances. In this article, we carry these ideas one step further in that we provide a method for identifying the measurements that most likely contain gross errors and cause the violation of the macroscopic balance constraints.

In chemical engineering literature, numerous articles have dealt with the problem of statistical adjustment of raw measurements to fit material balance equations arising from simultaneous chemical reactions. Kuehn and Davidson[5] have studied the basic problem of data adjustment when the measurements contain only small random errors. Ripps[6] has introduced a serial elimination algorithm in which one measurement at a time is totally deleted and the effect of the deletion on an objective function evaluated. Nogita[7] has expanded the usefulness of the serial elimination algorithm, and Madron and co-workers[8] have introduced a powerful, yet simple, multidimensional chi-square (χ^2) test, which we believe to be superior and will be used in this article. Finally, Romagnoli and Stephanopoulos[9] have proposed a highly efficient algorithm that permits sequential processing of the measured data and minimizes the effort in matrix computation, especially when the dimension of the problem is large.

Despite the fact that errors are present in virtually all the measurements taken in a fermentation process, in the past only a small number of articles have appeared in the literature which recognize this fact and incorporate statistics in the analysis of fermentation data. De Kwaadsteniet and co-workers[10] have considered the presence of random errors in their calculation of Y_{ATP}^{max}, m_{ATP}, and the 95% confidence intervals. De Kok and Roels,[11] Geurts and co-workers,[12] and Dekkers and co-workers[13] have adjusted their raw data by using the maximum likelihood principle so that the results conform to the elemental balance equations; the adjusted data are then used to estimate such variables as the RQ ratio, Y_{ATP}^{max}, and P/O. Solomon and co-workers[14] have checked for the level of consistency before parameters are estimated from different sets of measurements. Previously, Erickson and co-workers[2-4] have already applied consistency checks to a number of fermentation processes to assess the reliability of the measured parameters. However, in the above studies, no criterion was given to allow a systematic detection of the presence of gross measurement errors, and it is not clear exactly what magnitude of deviation in the consistency test was considered normal and tolerable. Furthermore, when one does indeed conclude the inconsistency to be significant, the number of measurements suspected of having gross errors and the source of errors cannot be determined unambiguously. As pointed out by Solomon and associates,[14] their procedure of simple linear

regression also has the disadvantage that it yields multiple, significantly different estimates for the same parameters.

In the next section, a set of equality constraints is derived from the material and energy balances around a biochemical reactor. The assumptions on the process characteristics and on the measurements are then stated, and the widely recognized solution to the least-squares minimization problem subject to a set of equality constraints is presented without proof. The use of a multidimensional chi-square (χ^2) test is proposed for assessing the validity of the original assumptions on the process characteristics and measurements and to detect the presence of gross errors. Next, a strategy to establish the source of gross errors by means of serial elimination of suspect measurements is discussed. Some useful results that simplify matrix computation and permit sequential analysis are also presented. One literature example is worked out in detail in Appendix A to demonstrate the simplicity and the usefulness of the proposed procedure. A few additional results obtained from the application of the same procedure to literature data are also shown, followed by a brief discussion on the likelihood of instrument malfunction and measurement accuracy during the various phases of a fermentation process.

FORMULATION OF MACROSCOPIC BALANCES

Assuming that the processes of growth and maintenance of microorganisms can be represented by an overall chemical reaction in which substrate is biologically converted to cell mass and extracellular metabolic products, one can write:

$$-aC_xH_yO_z - bO_2 - cNH_3 + C_\alpha H_\beta O_\gamma N_\delta + dH_2O + eCO_2 + fC_{\alpha'}H_{\beta'}O_{\gamma'}N_{\delta'} = 0 \quad (1)$$

where $C_xH_yO_z$, $C_\alpha H_\beta O_\gamma N_\delta$, and $C_{\alpha'}H_{\beta'}O_{\gamma'}N_{\delta'}$ represent the chemical formulae for the substrate, cell biomass, and extracellular metabolic product, respectively. Since it is only the relative amounts of the cell biomass elements that can be determined, α in the cell biomass formula can be set equal to unity without any loss of generality. Also, the stoichiometric coefficient of cell biomass was set equal to unity in the above equation, but any multiple of these coefficients would be an equally valid representation of the process. It should be pointed out that the above overall reaction can be expanded into a set of simultaneous reactions such as biomass formation from substrate, direct product formation from substrate, ATP generation by direct substrate oxidation and oxidative phosphorylation, etc. In the latter formulation some information on reaction stoichiometry is available, but the extents at which each of these parallel reactions participate in the overall process appear as additional unknowns. The final result is that both formulations contain the same amount of information. The advantage of eq. (1) is that it is simpler and

more directly observable, while the expanded scheme is more suggestive of the underlying mechanisms.

If one further assumes that all the important chemical species involved in biological process (1) and their chemical formulae, including that of biomass, are known and constant, it is then possible to write the following elemental balances for C, H, O, and N:

for C:

$$xa = \alpha + e + \alpha' f \tag{2}$$

for H:

$$ya + 3c = \beta + 2d + \beta' f \tag{3}$$

for O:

$$za + 2b = \gamma + d + 2e + \gamma' f \tag{4}$$

for N:

$$c = \delta + \delta' f \tag{5}$$

An energy balance can also be written if the heat of fermentation, Q, is measured:

$$-aH_c - bH_{O_2} - cH_{NH_3} + H_b + dH_{H_2O} + eH_{CO_2} + fH_p + Q = 0 \tag{6}$$

The stoichiometric coefficients can be calculated by considering the macroscopic balances over the bioreactor:

$$\frac{d(C_j V)}{dt} = \nu_j R V + \Phi_j \tag{7}$$

where C_j is the concentration of component j in the system, Φ_j is the net rate of input of component j to the system by transport, R is the total rate of biomass production, ν_j is the stoichiometric coefficient of eq. (1) for component j, and V is the appropriate system volume.

For a steady-state chemostat operation, $(dC_j V)/dt = 0$, and eq. (7) reduces to:

$$\nu_j R = -\frac{\Phi_j}{V} \tag{8}$$

The above equation is also customarily used for O_2 and CO_2 even under transient conditions by applying the quasi-steady-state principle; however, one should be cautioned against the effect of varying pH on the dissolved CO_2 level. Equations (7) or (8) can be used with off-line measurements of the concentrations in the streams entering and leaving the system to yield all the stoi-

chiometric coefficients except that of water. The latter can be calculated by employing one of the elemental-energy balances (2)–(6). The remaining balances must also be satisfied, and they impose four (or three, if the heat is not measured) equality constraints on the measurements. One can subsequently utilize these equality constraints to test the consistency of the obtained measurements (see refs. 2–4) and, furthermore, to locate the suspect measurement if grossly biased errors are detected in the data. The development of a method for this purpose is the objective of this article. It should also be pointed out that these balances can be utilized for the on-line evaluation of the stoichiometric coefficients from the measurements of O_2 and CO_2 concentrations whenever an on-line automated monitoring of the system is desired for control, or other, purposes (see ref. 1).

In a batch operation there is no liquid flow to the system; so, except for O_2 and CO_2, $\Phi_j = 0$ and, as eq. (7) indicates, the stoichiometric coefficients are calculated by taking the derivatives of the various concentrations measured as functions of time. The only exceptions are oxygen and carbon dioxide which can still be calculated from eq. (8) because the bioreactor is a continuous system with respect to the gases even during a batch operation.

If $\mathbf{x}''$ is a vector of the stoichiometric coefficients as its components, then eqs. (2)–(6) can be written as:

$$\mathbf{A}'\mathbf{x}'' = \mathbf{b}' \tag{9}$$

where

$$\mathbf{A}' = \begin{bmatrix} -x & 0 & 0 & \alpha & 0 & 1 & \alpha' \\ -y & 0 & -3 & \beta & 2 & 0 & \beta' \\ -z & -2 & 0 & \gamma & 1 & 2 & \gamma' \\ 0 & 0 & -1 & \delta & 0 & 0 & \delta' \\ -H_c & -H_{O_2} & -H_{NH_3} & H_b & H_{H_2O} & H_{CO_2} & H_p \end{bmatrix} \begin{matrix} \cdots \text{C} \\ \cdots \text{H} \\ \cdots \text{O} \\ \cdots \text{N} \\ \cdots \text{heat} \end{matrix}$$

$$\mathbf{x}'' = \begin{bmatrix} a \\ b \\ c \\ 1 \\ d \\ e \\ f \end{bmatrix} \qquad \mathbf{b}' = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -Q \end{bmatrix}$$

Since not all of the stoichiometric coefficients are measurable by applying eq. (7) or (8), the unmeasured ones are then eliminated from eq. (9) to yield:

$$\mathbf{A}\mathbf{x}' = \mathbf{b} \tag{10}$$

The vector $\mathbf{x}'$ now has as its elements all the *measured* coefficients. If, for example, the stoichiometric coefficient of water is unmeasured then $\mathbf{A}$, $\mathbf{x}'$, and $\mathbf{b}$ will be:

$$\mathbf{A} = \begin{bmatrix} -x & 0 & 0 & \alpha & 1 & \alpha' \\ -y + 2z & 4 & -3 & \beta - 2\gamma & -4 & \beta' - 2\gamma' \\ 0 & 0 & -1 & \delta & 0 & \delta' \\ \left(-z + \frac{H_c}{H_{H_2O}}\right) & \left(-2 + \frac{H_{O_2}}{H_{H_2O}}\right) & \left(\frac{H_{NH_3}}{H_{H_2O}}\right) & \left(\gamma - \frac{H_b}{H_{H_2O}}\right) & \left(2 - \frac{H_{CO_2}}{H_{H_2O}}\right) & \left(\gamma' - \frac{H_p}{H_{H_2O}}\right) \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} a \\ b \\ c \\ 1 \\ e \\ f \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{Q}{H_{H_2O}} \end{bmatrix} \tag{11}$$

It is more convenient to use eq. (10) in its homogeneous form, which can be obtained through a parallel translation. Thus, setting

$$\mathbf{x} = \mathbf{x}' + \mathbf{h} \tag{12}$$

Equation (10) reduces to:

$$\mathbf{A}\mathbf{x} = \mathbf{0} \tag{13}$$

where the constant vector $\mathbf{h}$ is any solution satisfying:

$$\mathbf{A}\mathbf{h} = -\mathbf{b} \tag{14}$$

Equation (13) now summarizes in a compact form all the equality constraints that must be satisfied by the vector of *measured* variables $\mathbf{x}$. With regards to the consistency of these measurements, one can ask the following questions:

1) What is a proper criterion to test whether the measurement vector $\mathbf{x}$ contains only random errors or grossly biased errors?

2) If only random errors are present, what is the best estimate of the measured quantities?

3) If gross errors are detected, what measurement is likely to be biased, and if the source of inconsistency is identified, how can the remaining measurements be rectified to yield a set of consistent estimates?

Answering the above questions is the subject of the work reported here. The starting point is eq. (13) representing the equality constraints that must be satisfied by a vector of measured variables **x**. This general representation is equally applicable to a multitude of cases involving a variety of carbon and nitrogen sources, products, cell biomass, and a full spectrum of possible measurements. Each individual case will have to be reduced to the form of eq. (13) before the algorithms of the following section can be applied.

STATISTICAL HYPOTHESIS TESTING FOR GROSS ERROR DETECTION

Hereafter, let **A** be a $m \times n$ matrix and **x** be a vector of measured variables of dimension n. Hence, $\mathbf{Ax} = \mathbf{0}$ is composed of m linearly independent equations, with $1 \leq m < n$. Since all actual measurements contain random errors, Equation (13) generally is not satisfied. Let δ be the vector of measurement errors, and **x** and $\bar{\mathbf{x}}$ be the vectors of *true values* and *measured values*, respectively. Then,

$$\mathbf{x} = \bar{\mathbf{x}} + \delta \tag{15}$$

We assume that the error vector, δ, is normally distributed with a zero mean and with a variance-covariance matrix ψ. Mathematically, this can be expressed as:

$$E[\delta] = \mathbf{0} \tag{16}$$

$$\psi \equiv E[(\bar{\mathbf{x}} - \mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})^T] = E[\delta\delta^T] \tag{17}$$

where E is the expected value operator. In the balance of any additional information that may indicate otherwise, each measurement error is assumed to be independent of one another and uncorrelated. Thus, under these assumptions, ψ can be considered diagonal. Since the actual values for the components of ψ are not known in reality, any best estimates such as simple variances will suffice.

The presence of the random errors δ in the measurements will produce a vector of residuals ϵ in the balance equations. It is easy to show that ϵ is related to δ by:

$$\epsilon \equiv \mathbf{A}\delta = -\mathbf{A}\bar{\mathbf{x}} \tag{18}$$

$$E[\epsilon] = \mathbf{A}E[\delta] = \mathbf{0} \tag{19}$$

$$\phi \equiv E[\epsilon\epsilon^T] = \mathbf{A}E[\delta\delta^T]\mathbf{A}^T = \mathbf{A}\psi\mathbf{A}^T \tag{20}$$

An estimate of the error δ can be obtained by solving the problem:

$$\underset{\delta}{\text{Minimize}}\ J = \delta^T\psi^{-1}\delta \text{ subject to } \mathbf{A}\delta = \epsilon$$

According to the above formulation, the best estimate of δ is obtained by minimizing the sum of the error squares scaled according to the level of confi-

dence one places on each measurement. The solution to the above constrained minimization problem is given by[15]:

$$\hat{\delta} = \psi \mathbf{A}^T \phi^{-1} \epsilon \tag{21}$$

which also coincides with the maximum likelihood estimation.[8] This is because when δ is normally distributed, the function to be minimized is the same for the least-square minimization problem as for the maximum likelihood minimization problem. When δ is not normally distributed, the result presented in eq. (21) remains a valid solution to the weighted least squares minimization problem posed above, although it will no longer be the maximum likelihood estimate. Hereafter, the diacritical mark $\hat{}$ signifies that the variable in concern is the estimated quantity rather than the true value or the measured value. The corresponding variance-covariance structures for $\hat{\delta}$ and $(\hat{\mathbf{x}} - \mathbf{x})$ are respectively:

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \hat{\delta} \tag{22}$$

$$\hat{\psi} \equiv E[\hat{\delta}\hat{\delta}^T] = \psi \mathbf{A}^T \phi^{-1} E[\epsilon\epsilon^T] \phi^{-1} \mathbf{A}\psi = \psi \mathbf{A}^T \phi^{-1} \mathbf{A}\psi \tag{23}$$

$$E[(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T] = E[(\hat{\delta} - \delta)(\hat{\delta} - \delta)^T] = \psi - \hat{\psi} \tag{24}$$

The last equality in Equation (24) is due to the fact that

$$\begin{aligned} E[(\hat{\delta} - \delta)(\hat{\delta} - \delta)^T] &= E[\hat{\delta}\hat{\delta}^T] - E[\delta\hat{\delta}^T] - E[\hat{\delta}\delta^T] + E[\delta\delta^T] \\ &= \hat{\psi} - E[\delta\hat{\delta}^T] - E[\hat{\delta}\delta^T] + \psi \end{aligned}$$

and that

$$\begin{aligned} E[\delta\hat{\delta}^T] = E[\hat{\delta}\delta^T] &= E[(\psi \mathbf{A}^T \phi^{-1} \epsilon)\delta^T] = E[\psi \mathbf{A}^T \phi^{-1} \mathbf{A}\delta\delta^T] \\ &= \psi \mathbf{A}^T \phi^{-1} \mathbf{A}\psi = \hat{\psi} \end{aligned}$$

Because of the negative term in eq. (24), the resulting estimate $\hat{\mathbf{x}}$ has a smaller standard deviation than the raw measurement $\bar{\mathbf{x}}$, and, therefore, $\hat{\mathbf{x}}$ is likely to be more accurate and reliable than $\bar{\mathbf{x}}$ after corrections are made. It is also certainly internally consistent. Note that although we have started with the assumption that $\bar{x}_j$'s are uncorrelated, the resulting estimates, $\hat{x}_j$'s, are inherently correlated; this can also be concluded from the fact that the off-diagonal elements of the matrix $\hat{\psi}$, consequently of $E[(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T]$, contain nonzero entries.

We can now formulate a test function to judge whether the residuals in the balance equations deviate significantly away from their expected distribution of zero means. The most reasonable and natural choice for the test function is the weighted square of the residuals of the balance equations:

$$h = \epsilon^T \phi^{-1} \epsilon \tag{25}$$

The distribution that applies to the above test function needs to be established at this point. If the residuals ϵ are assumed to have identical and independent normal distributions, i.e., if $\epsilon \sim N(\mathbf{0}, \phi)$, where ϕ is diagonal, then

the test function h is equal to the sum of squares of m elements which are independently and normally distributed. In this case h follows a chi-square (χ^2) distribution with m degrees of freedom.[14] However, it is not the residuals but the measurements which are independent and uncorrelated with covariance ψ assigned by the experimenter according to the level of confidence he has on each measurement. Therefore, ψ is diagonal, but ϕ [which is related to ψ by eq. (20)] in general is not diagonal because ϵ is correlated. Thus, we cannot yet conclude that h has a χ^2 distribution or that the degree of freedom is m because ϵ is composed of m elements. Using the fact that

$$\hat{\delta}^T\psi^{-1}\hat{\delta} = (\psi\mathbf{A}^T\phi^{-1}\epsilon)^T\psi^{-1}(\psi\mathbf{A}^T\phi^{-1}\epsilon) = \epsilon^T\phi^{-1}(\mathbf{A}\psi\mathbf{A}^T)\phi^{-1}\epsilon$$
$$= \epsilon^T\phi^{-1}(\phi)\phi^{-1}\epsilon = \epsilon^T\phi^{-1}\epsilon$$

we can express the test function h in eq. (25) as:

$$h = \hat{\delta}^T\psi^{-1}\hat{\delta} = \sum_{j=1}^{n} \frac{\hat{\delta}_j^2}{\sigma_j^2} \tag{26}$$

where n is the number of measurements, and $\sigma_j = \psi_{jj}^{1/2}$ is the standard deviation of the jth measurement. Note that the test function now has the same form as the performance index.

Since each of the n elements $\hat{\delta}_j/\sigma_j$ is approximately distributed according to the standardized Gaussian distribution, we conclude that the random variable h, being the sum of squares, is best characterized by a χ^2 distribution. Some comments of the degree of freedom of this χ^2 distribution is in order. Although the term $\Sigma_{j=1}^{n} \delta_j^2/\sigma_j^2$, which is unfortunately unknown, has n degrees of freedom, the actually computed term $\Sigma_{j=1}^{n} \hat{\delta}_j^2/\sigma_j^2$ has only m degrees of freedom, which is the same as the number of constraint equations. This is so because $\hat{\delta}_j$'s are not all independent of one another. This may be reasoned in the following manner. When only $n - m$ measurements are taken, the system is just determined. In this case, all the measurements are necessarily interpreted at their full values. No adjustment on them is possible because there is no equation to constrain the $n - m$ measurements; thus, there is no degree of freedom. However, each measurement in addition to the original $n - m$ measurements introduces one constraint equation, which makes some adjustments possible, and gives the term $\Sigma_{j=1}^{n} \hat{\delta}_j^2/\sigma_j^2$ one degree of freedom. Conceptually, we can treat the first $n - m$ measurements as the bare essentials that are needed to determine the system and the remaining m measurements as the luxuries that are used to provide adjustments and increase the accuracy of the estimates. Thus, the whole set of n measurements only supports m degrees of freedom.

We want to emphasize that the data adjustment algorithm presented in eq. (21) does not use the assumption of uncorrelated measurement errors. Consequently, the results in eqs. (22)–(24) remain valid even when the covariance matrix ψ is not diagonal. The assumption of the error vector δ being independent and uncorrelated is used in reaching the conclusion as to what distribu-

tion function best describes the random variable h, namely the χ^2 distribution with m degrees of freedom. Thus, if the error vector δ is not totally uncorrelated, then, strictly speaking, the test function h does not follow a χ^2 distribution. In actual situations, some correlations in measurements are unavoidable. For example, an error in gas flow rate measurement results in correlated errors in b (i.e., the O_2 consumption rate) and e (i.e., the CO_2 evolution rate). The effect of correlated nondiagonal covariance matrix ψ on the evaluation of h is briefly discussed in Appendix C. In fermentation systems, we have found the effect of off-diagonal terms in ψ to be relatively small (approximately 10% for the example in the Appendices). Therefore, despite the problems arising from correlation, the χ^2 criteria still proves to be a superbly valuable tool in detecting and identifying large systematic errors in measurements.

Our judgment on error detection is strongly dependent on the outcome of the hypothesis testing performed on the test function. We reject the hypothesis that the amount of errors present in the measurements is not significant with a confidence level of $1 - \theta$ if $h \geq \chi^2_{1-\theta}(m)$, i.e. if the test is failed. On the other hand, if the test is passed, we cannot prove that the assumptions that measurements contain only random errors are satisfied. (See Fig. 1 for a graphical interpretation of this fact. For the readers' convenience, some of the most frequently used numbers of the χ^2 distribution function are provided in Table I.) A confidence level of 90 or 95%, or equivalently 0.1 or 0.05 for θ, is usually adequate. We will use a confidence level of 90% in the numerical examples presented later.

By using the residual values and the value of ϕ as determined from eqs. (18) and (20), one can apply this test before the adjustments are determined. If the test shows that measurement errors are consistent with their variance-covariance structure, the amount of correction needed to satisfy balance equation (13) is calculated from eq. (21), and the optimal constrained estimate of x is given by eq. (22). On the contrary, if the above test fails, we conclude that one or more of the measurements have grossly biased errors that cannot be statistically accounted for by the assumption that they are all randomly and normally distributed. In this case we need to proceed one step further to identify the source of the abnormally large errors.

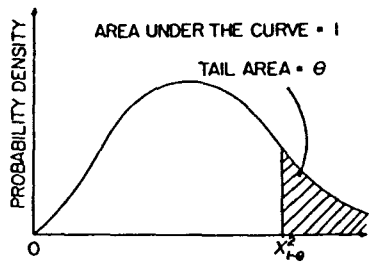


Fig. 1. $\chi^2_{1-\theta}$ probability distribution with m degrees of freedom.

TABLE I
Probability Points of the χ^2 Distribution with m Degrees of Freedom[a]

	θ (tail area probability)					
Degrees of freedom	0.50	0.25	0.10	0.05	0.025	0.01
$m = 1$	0.455	1.32	2.71	3.84	5.02	6.63
$m = 2$	1.39	2.77	4.61	5.99	7.38	9.21
$m = 3$	2.37	4.11	6.25	7.81	9.35	11.3
$m = 4$	3.36	5.39	7.78	9.49	11.1	13.3

[a]Source is ref. 18.

LOCATION OF THE SOURCES OF GROSS ERRORS

If the source of a gross error remains unidentified, a single large correction, which is actually needed only on the erroneous measurement, is distributed among all other statistically correct measurements which need little adjustment. To locate the measurements with gross errors when the original complete set of data fail the hypothesis test, we delete one measurement at a time, and calculate the performance index after the deletion of each measurement. Due to the numerical identity of the performance index J and the test function h, the calculation of performance indices can be carried out *before* the actual adjustment is made. These performance indices obtained from each deletion of one measurement are then compared with one another and also with the χ^2 criteria, with the degree of freedom appropriately decreased by one to account for the loss of one constraint equation due to the deletion of one measurement.

If the deletion of one measurement is all that is needed to pass the test, we shall conclude that there is only one erroneous measurement. The one whose deletion yields the lowest values for the performance index is usually judged to be erroneous because it is most likely to be the correct choice. To be sure, the source of error may actually lie in other measurements whose deletion result in slightly higher performance indices. Nonetheless, without any further information, this is the best decision in a statistical sense that one can make. Those measurements whose deletion also give rise to performance indices that are, though not the minimum but, low enough to pass the hypothesis testing should also be scrutinized for possible sources of errors.

If the deletion of a single measurement fails to produce an acceptable value for the test function, we delete one more measurement from the remaining $n - 1$ measurements. All the possible combinations of the n measurements in groups of two must be tried, for discarding indiscriminately the two measurements that individually yielded the two lowest performance indices does not necessarily result in the lowest value of the test function. If the deletion of two measurements is not successful, the process is repeated until the χ^2 test is

passed or until $m - 1$ measurements have been deleted. Although inconsistency is usually due to the presence of gross errors, it may also be caused by erroneous modeling and process disorder. For example, one may have neglected some significant species that are actually involved in the reaction in a fermenter; the biomass elemental composition used in eq. (1) may have been erroneous. Since the occurrence of gross errors is usually infrequent, the presence of numerous gross errors in a small set of data is statistically very improbable. Therefore, such encounters should be approached with extreme skepticism with regards to the validity of the constraint equations, the original estimation of the magnitude of random errors an experimenter assigns to his data, or any one of the other assumptions made in deriving eq. (13).

The above discussion described the basic idea underlying the algorithm for the identification of the suspect measurement. In carrying out the deletion process, one can return to eq. (9), treat the questionable measurements as unmeasured, form a new balance eq. (13), compute the performance index, and finally calculate the maximum likelihood estimate for the remaining measurements. Another way is described by Romagnoli and Stephanopoulos.[9] Although it is conceptually slightly more involved, it is much simpler to program and to implement in actual calculations. Because this method successfully avoids inverting a square matrix repeatedly, it is indispensable especially when the size of the problem is large. Of course, both methods yield exactly the same result.

The ensuing discussion will follow that of Romagnoli and Stephanopoulos,[9] and concerned readers are referred to their recent article for a detailed discussion. The original measurement vector $\mathbf{x}$ is partitioned into two parts; c measurements are suspected of having gross errors and $u = n - c$ measurements are assumed to possess only small random errors with zero mean values:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_u \\ \mathbf{x}_c \end{bmatrix} \qquad \begin{aligned} &\text{dimension } [\mathbf{x}_u] = u \\ &\text{dimension } [\mathbf{x}_c] = c \\ &\text{dimension } [\mathbf{x}] = n = u + c \end{aligned} \tag{27}$$

Similarly, matrices $\mathbf{A}$ and ψ can be partitioned as:

$$\mathbf{A} = [\mathbf{A}_u \mathbf{A}_c] \tag{28}$$

$$\psi = \begin{bmatrix} \psi_u & \mathbf{0} \\ \mathbf{0} & \psi_c \end{bmatrix} \tag{29}$$

When there are gross errors, the variance-covariance matrix for the measurements is:

$$\psi_n = \begin{bmatrix} \psi_u & \mathbf{0} \\ \mathbf{0} & \psi_c + \Delta\psi \end{bmatrix} \tag{30}$$

where $\Delta\psi$ is the bias in the covariance of the c suspect measurements. The distinction between ψ and ψ_n is that ψ corresponds to the situation in where there are no gross errors; whereas, ψ_n includes a bias term, $\Delta\psi$. It can be shown that the new maximum likelihood estimate of δ when gross errors are present is still given by eq. (21), whose applicable form is now:

$$\hat{\delta}_n = \psi_n \mathbf{A}^T \phi_n^{-1} \epsilon \tag{31}$$

where

$$\phi \equiv \mathbf{A}\psi\mathbf{A}^T$$

$$\phi_n \equiv \mathbf{A}\psi_n\mathbf{A}^T = \phi + \mathbf{A}_c(\Delta\psi)\mathbf{A}_c^T \tag{32}$$

Again, subscript n signifies that the variable carrying it involves the bias term $\Delta\psi$. Using a matrix inversion lemma, we can express ϕ_n^{-1} in eq. (32) as:

$$\phi_n^{-1} = \phi^{-1}\{\mathbf{I} - \mathbf{A}_c[(\Delta\psi)^{-1} + \mathbf{A}_c^T\phi^{-1}\mathbf{A}_c]^{-1}\mathbf{A}_c^T\phi^{-1}\} \tag{33}$$

When one has enough information about the process to produce a rough estimate of the bias term, $\Delta\psi$, he should use it. Otherwise, a very drastic measure, a total deletion of these c measurements, is taken. Since setting the bias $(\Delta\psi)_i$ in each of the c measurements to ∞ is equivalent to deleting the c measurements, the estimate of δ of the *remaining* measurements is:

$$\hat{\delta}_u = \psi_u \mathbf{A}_u^T \phi_u^{-1} \epsilon \tag{34}$$

where

$$\phi_u^{-1} = \phi^{-1}\{\mathbf{I} - \mathbf{A}_c[\mathbf{A}_c^T\phi^{-1}\mathbf{A}_c]^{-1}\mathbf{A}_c^T\phi^{-1}\} \tag{35}$$

is the variance-covariance matrix for ϵ after c measurements are deleted. Note that $\phi_u^{-1} \equiv \lim_{(\Delta\psi)_i \to \infty} \phi_n^{-1} \neq (\mathbf{A}_u\psi_u\mathbf{A}_u^T)^{-1}$, is still a $m \times m$ matrix even after the deletion of c measurements. Since ϕ_u^{-1} is whatever left of ϕ_n^{-1} after the deletion of c measurements, we expect ϕ_u^{-1} to be necessarily singular due to the reduction in the number of constraint equations. Since one always works with ϕ_u^{-1} rather than ϕ_u, there is no need to be concerned with inverting a singular matrix ϕ_u^{-1} back to ϕ_u. Note also that $\hat{\delta}_u$ in eq. (34), being the result after deletion, has the dimension of $u = n - c$; whereas, $\hat{\delta}_n$ in eq. (31), being the estimate before deletion, has the full dimension of n. The term $\mathbf{A}_c[\mathbf{A}_c^T\phi^{-1}\mathbf{A}_c]^{-1}\mathbf{A}_c^T\phi^{-1}$ in eq. (35) represents the fraction of correction applied to the original $\phi^{-1} = (\mathbf{A}\psi\mathbf{A}^T)^{-1}$ due to the deletion of c measurements. If deletion is made for one measurement at a time, we need to invert only a scalar $\mathbf{A}_c^T\phi^{-1}\mathbf{A}_c$ to calculate $\hat{\delta}_u$.

The new performance index (or test function) is now:

$$h = \epsilon^T\phi_u^{-1}\epsilon = \hat{\delta}_u^T\psi_u^{-1}\hat{\delta}_u \tag{36}$$

The first equality in the above equation is used to evaluate h *before* adjustments are computed, and ϵ is readily obtained from eq. (18).

An algorithm implementing the above results in the deletion process can be summarized as follows: First, the inverse of ϕ is calculated for the repeated

use with eq. (35) in the sequel. After the measurements to be deleted have been selected, matrices, A and ψ are partitioned as indicated by eqs. (28) and (29), and ϕ_u^{-1} is calculated from eq. (35). With ϕ_u^{-1} known, the test function h can be evaluated from eq. (36) and compared to the criterion. If the test fails, a new set of measurements is selected and the process is repeated. If the test passes, a suspect measurement has been located, and the corrected values of the measurements are calculated by applying eq. (34).

An example that fully illustrates the method has been worked out in detail in Appendix A. Also another approach that analyzes the residual errors by considering one equation at a time is discussed in Appendix B. This method allows a sequential processing and is particularly useful when the dimension of the matrix A is large.

APPLICATION TO LITERATURE EXAMPLES

The methodology outlined in the previous sections was applied to the full set of data of de Kok and Roels.[11] (See Appendix A for a detailed demonstrative calculation involving one of the sets of data in this example.) The results of the analysis are displayed in Table IIA. The corrected values (Table IIB)

TABLE IIA
Performance Indices for a Glucose-Limited Steady-State Continuous Culture of *S. cerevisiae* CBS 426[a]

	h				
		After deletion of			
D (h^{-1})	No deletion	*a*	*b*	*B*	*e*
0.008	3.91	1.53	3.90	1.69	0.67[c]
0.008	35.07[b]	27.06	2.12[d]	26.43	34.96
0.017	2.07	0.04	1.70	0.06	1.19
0.033	1.65	0.00	1.31	0.01	1.18
0.047	1.99	0.07	1.85	0.10	1.18
0.052	0.23	0.01	0.15	0.00	0.21
0.072	2.42	1.86	0.01	1.73	1.27
0.076	2.20	0.01	1.78	0.00	1.98
0.092	0.53	0.20	0.50	0.23	0.12
0.092	1.01	0.17	0.43	0.13	1.00
0.102	2.50	2.29	0.11	2.20	0.86
0.112	2.71	0.43	1.12	0.33	2.70
0.113	1.73	1.07	1.51	1.14	0.11
0.118	0.54	0.25	0.09	0.22	0.46

[a] Sources of data is ref. 11.
[b] The test function lies outside the 90% confidence level.
[c] The CO_2 measurement is questionable.
[d] The O_2 measurement is to be deleted.

TABLE IIB
The Measured and the Estimated Values of Stoichiometric Coefficients for a Glucose-Limited Steady-State Continuous Culture of *S. cerevisiae* CBS 426[a]

	Stoichiometric coefficients[b]				
D (h^{-1})	a	b	e	c	d
	(mol/C mol biomass)				
0.008	0.333	1.1	1.4[c]		[d]
	0.364	1.136	1.186	0.170	1.526[e]
0.008	0.35	3.8	1.4		
	0.375	1.202[f]	1.252	0.170	1.592
0.017	0.383	1.5	1.6		
	0.416	1.437	1.487	0.170	1.827
0.033	0.317	1.04	1.1		
	0.342	1.005	1.054	0.170	1.395
0.047	0.413	1.2	1.2		
	0.378	1.220	1.270	0.170	1.610
0.052	0.300	0.82	0.86		
	0.309	0.807	0.857	0.170	1.197
0.072	0.29	0.54	0.73		
	0.275	0.597	0.647	0.170	0.987
0.076	0.322	0.67	0.73		
	0.290	0.689	0.739	0.170	1.079
0.092	0.3	0.7	0.7		
	0.288	0.680	0.730	0.170	1.070
0.092	0.317	0.7	0.8		
	0.297	0.734	0.784	0.170	1.124
0.102	0.3	0.62	0.85		
	0.290	0.691	0.741	0.170	1.081
0.112	0.35	0.78	0.92		
	0.316	0.843	0.893	0.170	1.233
0.113	0.3	0.8	1.0		
	0.316	0.845	0.895	0.170	1.235
0.118	0.285	0.57	0.67		
	0.274	0.596	0.646	0.170	0.986

[a] Source of data is ref. 11.
[b] c and d are calculated from elemental balance equations using the corrected values of a, b, and e.
[c] The CO_2 measurement is questionable.
[d] Raw experimental data.
[e] The maximum likelihood estimates with deletion carried out at a 90% confidence level.
[f] Maximum likelihood estimate after O_2 measurement is deleted.

differ slightly from those obtained by the original investigators mainly because the uncertainty we assign to each measurement is different from that assigned by them. We have consistently used errors of 6, 11.7, 5 and 11.1% for the glucose, oxygen, biomass, and carbon dioxide measurements, respectively. Besides the oxygen measurement in the second set of data at $D = 0.008\ h^{-1}$, which has already been determined to be in error in Appendix A,

the carbon dioxide measurement in the first set of data at $D = 0.008\ h^{-1}$ is also suspected of being erroneous. The errors of the rest of the measurements appear to lie within the expected limits.

Table III contains the results of data analysis performed on the reported data of Geurts and co-workers.[12] In their study, *S. cerevisiae* CBS 426 was grown continuously in a medium containing a mixed substrate of glucose and ethanol in various ratios. The measured variables are the net fluxes of glucose, ethanol, oxygen, biomass, and carbon dioxide. The suggested formula for the biomass, $CH_{1.83}O_{0.56}N_{0.17}$, is used to derive the constraint equations. An error of 10% is used for all the measurements except for that of ethanol, for which an error of 20% is used to obtain the corrected values. The fact that the test functions for each run are all below the value of $\chi^2_{0.9}(m = 2) = 4.61$ indicates that the original assumptions made on the process characteristics and measurement error structures are not violated significantly. We suspect that, although not a 90% confidence level, the oxygen measurement of run No. 2 may be erroneous because its deletion greatly slashes the performance index from 4.06 down to a mere 0.31. The corrected values of the measurements for this and the following examples are not given because of space limitation. They can be calculated by applying the proper formulae of the text.

The raw data of Dekkers and associates[13] of the growth of *S. cerevisiae* CBS 426 under both aerobic and anaerobic conditions in a continuous fermentor were also analyzed. Glucose was used as the carbon source and ammonium sulfate as the nitrogen source. Ethanol formation was detected under aerobic conditions, and both ethanol and glycerol productions were detected during anaerobic fermentation. In the aerobic studies, the reported average value of $CH_{1.83}O_{0.56}N_{0.17}$ is used as the biomass composition, and in anaerobic studies the average value of $CH_{1.86}O_{0.60}N_{0.16}$ is used. An error of 10% is used for all the measurements in deriving the test function and the corrected values.

TABLE III
Performance Indices for a Steady-State Continuous Culture of *S. cerevisiae* CBS 426 on a Mixed Substrate of Glucose and Ethanol[a]

Run	Carbon source in the feed glucose-ethanol (wt %)	h					
		No deletion	After deletion of the measurement of				
			Glucose	Ethanol	O_2	Biomass	CO_2
1	8.75-1.25	0.43	0.33	0.02	0.35	0.30	0.43
2	7.50-2.50	4.06	3.36	2.99	0.31[b]	3.56	1.26
3	5.00-2.50	0.61	0.60	0.23	0.16	0.58	0.42
4	5.00-5.00	0.04	0.00	0.03	0.04	0.00	0.02
5	2.50-7.50	1.83	1.43	0.30	0.18	1.29	1.04
6	1.25-8.75	0.23	0.02	0.16	0.19	0.03	0.08

[a] Source of data is ref. 12.
[b] O_2 measurement is questionable.

IDENTIFICATION OF GROSS MEASUREMENT ERRORS

The results of data analysis for the aerobic growth experiments of Dekkers and co-workers[13] are shown in Table IV and those for the anaerobic case are shown in Table V. The measured values for the aerobic case, as can be seen from the test functions presented in Table IV, are reasonably internally consistent. The hypothesis test is passed for every single set of data except, possibly, the run at $D = 0.205\ h^{-1}$ for which a process disorder could have been caused by a different sterilization method.

As shown in Table V, the consistencies for the anaerobic data are not as good as for the aerobic data; three out of fourteen sets of data are judged to be inconsistent at a 90% confidence level. The inconsistencies of the runs at $D = 0.014\ h^{-1}$ and $D = 0.051\ h^{-1}$ are believed to be caused by erroneous readings of carbon dioxide, and that at $D = 0.058\ h^{-1}$ is probably due to a high reading of ethanol. Since for the purposes of analysis an error of 10% is consistently applied to all the measurements, this estimate of uncertainty may have been somewhat unrealistic. The inconsistency at $D = 0.014\ h^{-1}$ may not be as serious as the test function indicates if a more realistic set of variances is used for runs at low dilution rate where the absolute values of the readings are

TABLE IV
Performance Indices for a Glucose-Limited Steady-State Continuous Culture of *S. cerevisiae* CBS 426 with Ethanol Production under Aerobic Conditions[a]

	h					
		After deletion of the measurement of				
D (h^{-1})	No deletion	Glucose	O_2	Biomass	CO_2	Ethanol
0.008	1.74[b]	1.65	0.45	1.70	0.02	1.58
0.017	0.21	0.01	0.21	0.02	0.14	0.09
0.033	0.36	0.19	0.28	0.22	0.07	0.34
0.047	0.72	0.32	0.59	0.38	0.18	0.67
0.052	0.05	0.00	0.05	0.00	0.04	0.03
0.073	2.92	2.25	0.17	1.97	1.17	0.36
0.094	0.08	0.04	0.02	0.03	0.06	0.00
0.103	3.17	2.99	0.01	2.82	0.57	1.27
0.129	0.77	0.03	0.77	0.07	0.64	0.51
0.145	0.24	0.21	0.08	0.22	0.00	0.22
0.174	0.80	0.04	0.66	0.01	0.80	0.21
0.185	0.70	0.03	0.67	0.08	0.50	0.57
0.202	0.62	0.25	0.48	0.17	0.60	0.08
0.205	3.38	2.40	0.30	2.74	0.01	2.67
0.220	0.73	0.22	0.68	0.14	0.73	0.20
0.230	2.27	0.75	1.44	0.48	2.09	0.19
0.256	1.92	0.07	1.63	0.21	1.11	1.78
0.277	0.76	0.15	0.27	0.23	0.09	0.75
0.287	0.94	0.05	0.63	0.12	0.36	0.92

[a]Source of data is ref. 13.
[b]When the measured value of net ethanol flow is 0, a variance of 0.01 is used.

TABLE V
Performance Indices for a Glucose-Limited Steady-State Continuous Culture of *S. cerevisiae* CBS 426 with Ethanol and Glycerol Production under Anaerobic Conditions[a]

		h				
		After deletion of the measurement of				
D (h^{-1})	No deletion	Glucose	Biomass	CO_2	Ethanol	Glycerol
0.012	2.02	0.04	0.00	1.90	1.37	0.28
0.014	3.32	1.27	1.64	0.63	3.31	2.56
0.014	10.99[b]	9.53	10.19	0.00[c]	9.05	10.99
0.017	0.27	0.13	0.16	0.04	0.26	0.24
0.027	1.52	n.a.	n.a.	n.a.	n.a.	n.a.
0.030	0.33	0.18	0.13	0.28	0.02	0.04
0.034	1.52	0.32	0.52	0.56	1.52	1.09
0.051	7.17[b]	3.37	4.50	0.74[c]	5.98	6.79
0.053	0.57	0.26	0.18	0.51	0.06	0.03
0.056	0.88	0.61	0.72	0.01	0.64	0.88
0.058	1.66	1.14	1.34	0.02	1.18	1.65
0.058	10.20[b]	8.69	7.81	8.24	0.00[d]	4.56
0.062	2.61	2.46	2.58	0.23	0.94	2.43
0.067	0.79	0.78	0.74	0.20	0.15	0.57
0.094	1.70	0.44	0.67	0.47	1.68	1.29

[a]Source of data is ref. 13.
[b]The test function lies outside the 90% confidence level.
[c]The CO_2 measurement is to be deleted.
[d]Ethanol measurement is to be deleted.

generally small in magnitude and the percentage of error is large. However, the error in ethanol reading at $D = 0.058\ h^{-1}$ is more definite. A visual inspection of the plot of ethanol production rate versus D shown in Dekkers' article reveals that this measurement is indeed an outlier. Thus, the systematic application of the proposed procedure is able to single out a statistically erroneous measurement without reference to other sets of data or human experience.

The next set of analysis is based on the data originally reported by Selga and co-workers[16] and later extensively analyzed by Erickson and co-workers,[2] and the results are given in Table VI. *Brevibacterium* was grown in an undefined complex medium that contains, among other nutrients, molasses and corn extract. The consumption rates of reducing sugars and oxygen and the productivities of biomass, carbon dioxide, and lysine in a batch fermentor were recorded for each 12-h interval for the first 48 h of the fermentation. Erickson's practice of relating the heat evolution rate to the oxygen consumption rate is followed in the statistical analysis here. The heat evolution rate is taken as another independent oxygen measurement which is related to the heat evolution rate by 108 kcal = 1 mol O_2 (i.e., 27 kcal/equivalent available electrons transferred to oxygen). As before, an error of 10% is assigned to all the

TABLE VI
Performance Indices for the Batch Growth of *Brevibacterium* in a Complex Medium with Lysine Production under Aerobic Conditions[a]

			h					
			After deletion of the measurement of					
Run	Period (h)	No deletion	Sugar	O_2	Heat	Biomass	CO_2	Lysine·HCl
1	0-12	72.79[b,c]	7.59	57.30	70.51	8.45	62.80	11.09
	12-24	28.95[b]	0.37[d]	21.63	28.95	0.54	22.33	1.11
	24-36	0.75	0.07	0.51	0.71	0.04	0.75	0.02
	36-48	2.33	0.03	1.81	2.33	0.01	2.19	0.03
	0-48	1.44	0.08	1.21	1.41	0.10	1.15	0.16
2	0-12	62.25[b]	2.05[d]	53.30	65.00	2.61	55.75	4.60
	12-24	2.51	0.19	1.73	2.45	0.09	2.51	0.01
	24-36	4.24	0.10	3.97	4.15	0.20	3.38	0.43
	36-48	0.35	0.17	0.03	0.34	0.20	0.08	0.23
	0-48	2.85	0.10	2.66	2.84	0.19	2.16	0.37
3	0-12	10.23[b]	0.07[d]	9.46	10.07	0.12	9.64	0.59
	12-24	5.77	0.20	4.70	5.75	0.09	7.53	0.20
	24-36	1.57	0.13	1.48	1.47	0.18	1.44	0.32
	36-48	24.89[b]	0.71[d]	21.97	23.77	0.81	21.16	1.41
	0-48	0.14	0.14	0.12	0.01	0.14	0.14	0.14

[a]Source of data is ref. 2.
[b]The test function lies outside the 90% confidence level.
[c]The following three sets of simultaneous deletion of two measurements results in passing performance indices: h (heat and sugar) = 0.27, h (heat and biomass) = 0.01, and h (heat and lysine) = 0.99.
[d]Sugar measurement is to be deleted.

measurements except heat measurements for which an error of 40% is used. The empirical formulae of $C_8H_{13}O_4N$ and CH_2O suggested by Erickson are used for the biomass and reducing sugars. Table VI shows that 5 out of 12 sets of data in 12-h intervals fail the χ^2 test. These inconsistencies all occur either at the beginning or at the end of the batch fermentation when accurate measurements are especially difficult to obtain. Since the deletion of the sugar measurements yield the smallest performance indices in all five cases, one can conclude that sugar measurements may be erroneous. Because the individual deletions of biomass and lysine measurements also result in passing test functions in all cases except the 0-12 h period of run No. 1, the possibility of these two measurements being the sources of errors should also be examined. The performance indices of the 0-12 h period of run No. 1 suggest that equating heat evolution rate to oxygen evolution rate is not valid at the beginning of run No. 1 even when an error of 40% is allowed for the equivalence relation. Compared to the lengthy consistency analysis by Erickson and associates,[2] the proposed χ^2 analysis is capable of simpler unbiased detection and identification of the sources of errors.

The next set of data is reported by Ferrer and Erickson.[4] *Candida lipolytica* was cultivated in a batch fermentor using *n*-hexadecane as the carbon source and ammonium sulphate as the nitrogen source. The rates of substrate and oxygen consumption and biomass and carbon dioxide production were measured. The results of analysis for run No. 4 are shown in Table VII, which is obtained by using the same values of $\gamma_b = 4.291$ (for the equivalent available electrons/C mol biomass) and $\sigma_b = 0.462$ (for the weight fraction of carbon in biomass) as used by Ferrer and Erickson. As before, a measurement error of 10% is assumed. Low CO_2 measurements were judged to be the cause of errors in five instances and a high O_2 measurement was blamed in one instance. Moreover, they occur either at the beginning or at the end of the batch runs. Although not shown, the results of analysis of their data for other runs also display the same tendency of attributing errors to CO_2 measurements. The erroneous CO_2 measurements may have been caused by the dissolved CO_2 that was not properly accounted for.

Finally, the data for this last example of continuous culture of *Aerobacter aerogens* in a glucerol-limited medium was originally reported by Herbert[17] and later analyzed for consistency by Erickson and co-workers.[3] The experimentally measured values of biomass and substrate concentrations at each reported dilution rate are used. Since not all the oxygen uptake rates and

TABLE VII
Performance Indices for a *n*-Hexadecane-Limited Batch Growth of *Candida lipolytica* under Aerobic Conditions (run 4)[a]

	h				
		After deletion of the measurement of			
Time (h)	No deletion	Substrate	O_2	Biomass	CO_2
0	109.94[b]	76.04	36.74	46.30	5.03[d]
2	26.08[b]	24.29	4.06	16.00	1.90[d]
4	21.54[b]	17.24	7.07	7.69	0.00[d]
8	8.35[b]	5.63	4.37	1.12	0.78[d]
10	4.47	1.72	3.65	0.01	1.68
12	0.92	0.83	0.28	0.40	0.00
14	1.05	1.04	0.03	0.89	0.18
16	3.36	3.19	0.00	3.20	1.23
18	2.68	2.66	0.08	2.36	0.65
19.5	2.60	2.60	0.18	2.14	0.46
21.5	1.97	1.69	0.71	0.85	0.00
23.5	3.68	3.67	0.29	3.07	0.74
25.5	5.95[b]	5.21	2.06	3.41	0.23[d]
26.5	9.50[b]	7.19	0.15[c]	8.79	8.68

[a]Source of data is ref. 4.
[b]The test function lies outside the 90% confidence level.
[c]The O_2 measurement is to be deleted.
[d]The CO_2 measurement is to be deleted.

carbon dioxide evolution rates are reported at the same dilution rates as biomass and substrate concentrations, the straight lines passing through the experimental points in the of Φ_{O_2} vs. D and Φ_{CO_2} vs. D plots are used to generate the "measured" values. The same $\gamma_b = 4.291$ and $\sigma_b = 0.462$ as used by Erickson and co-workers[3] are used here, and the results of our analysis are shown in Table VIII. The consistency of every set of data, except the last, is found to be remarkable, and the deviation of the last set of data can also be seen in Herbert's plot, in which the lines drawn through the experimental points totally missed the measurements at $D = 1.01\ h^{-1}$. As Figure 1 shows, the occurrence of extremely large performance indices is rare; so is the occurrence of phenomenally small ones. The consistently small performance indices in this example are due to the application of smoothing of random noises when values read from the "best fit" lines are used as measurements. The objective of this example is to show that biased data manipulation may be suspected if the performance indices are consistently unreasonably small.

CONCLUSIONS

The suggested procedure for detecting and isolating gross errors through the use of statistical hypothesis testing is conceptually simple. It provides a

TABLE VIII
Performance Indices for a Glucerol-Limited Steady-Stage Continuous Culture of *Aerobacter, aerogens*[a]

	h				
		After deletion of the measurement of			
D (h^{-1})	No deletion	Glucerol	O_2	Biomass	CO_2
0.050	0.06	0.06	0.01	0.06	0.00
0.115	0.20	0.14	0.01	0.16	0.12
0.125	0.24	0.14	0.17	0.11	0.03
0.250	0.26	0.24	0.00	0.26	0.07
0.350	0.32	0.28	0.12	0.24	0.01
0.485	0.40	0.31	0.19	0.25	0.02
0.510	0.33	0.33	0.06	0.30	0.00
0.625	0.39	0.34	0.14	0.29	0.01
0.750	0.50	0.36	0.27	0.28	0.05
0.850	0.40	0.37	0.10	0.33	0.00
0.910	0.38	0.37	0.00	0.38	0.06
0.935	0.46	0.37	0.20	0.30	0.03
0.980	1.48	0.38	0.75	0.64	1.31
1.010	5.55[b]	0.37	5.54	0.07[c]	4.26

[a]Source of data is ref. 17.
[b]The test function lies outside the 90% confidence level.
[c]Biomass measurement is to be deleted.

potentially powerful means for the systematic analysis of a set of measurements subjected to equality constraints. It gives a much more reliable and internally consistent set of estimates with only little extra effort in computation. All the necessary calculations and decisions can be readily made by a computer. Thus, it can be incorporated into the existing process of automated data collection, analysis, and computerized control of a biochemical reactor. There is the advantage that an operator may be warned of instrumental malfunctions or process disorders. Its usefulness becomes especially significant as future development in sensors enables us to take more measurements that are related through some constraints. In order to improve the accuracy of the estimates, it can be applied to the on-line continuous estimation in which a Kalman filter is used to estimate the growth parameters of a biochemical reactor.[19] As long as an investigator uses the correct model and makes the proper measurements, it can be employed in both aerobic and anaerobic fermentation processes under various conditions and in different modes of operation, let it be batch, fed-batch, or continuous.

APPENDIX A: A NUMERICAL EXAMPLE

The data of de Kok and Roels[11] will be used to illustrate the numerical aspect of the proposed strategy. *Saccharomyces cerevisiae* CBS 426 was cultivated in a medium in which glucose is utilized as the carbon source and ammonium sulfate as the nitrogen source. The steady-state flows of the substrate, oxygen, biomass, and carbon dioxide were measured for a range of dilution rates. $CH_{1.83}O_{0.56}N_{0.17}$ was used as the elemental composition for the biomass in correcting the experimental data to conform to the elemental balances. No extracellular metabolic product is assumed to be present. Thus, reaction (1) for this problem is:

$$aC_6H_{12}O_6 + bO_2 + cNH_3 \rightarrow BCH_{1.83}O_{0.56}N_{0.17} + dH_2O + eCO_2$$

Matrix $\mathbf{A}'$ and the vectors $\mathbf{x}''$ and $\mathbf{b}'$ of eq. (9) are:

$$\mathbf{A}' = \begin{bmatrix} -6 & 0 & 0 & 1.00 & 0 & 1 \\ -12 & 0 & -3 & 1.83 & 2 & 0 \\ -6 & -2 & 0 & 0.56 & 1 & 2 \\ 0 & 0 & -1 & 0.17 & 0 & 0 \end{bmatrix} \begin{matrix} \cdots \mathrm{C} \\ \cdots \mathrm{H} \\ \cdots \mathrm{O} \\ \cdots \mathrm{N} \end{matrix} \qquad \mathbf{x}'' = \begin{bmatrix} a \\ b \\ c \\ B \\ d \\ e \end{bmatrix} \qquad \mathbf{b}' = \mathbf{0}$$

The net consumption of NH_3 and the net production of H_2O were not measured; so coefficients c and d are eliminated from the above equations to yield $\mathbf{A}\mathbf{x}' = \mathbf{0}$ where

$$\mathbf{A} = \begin{bmatrix} -6 & 0 & 1.0 & 1 \\ 0 & -2 & -0.1 & 2 \end{bmatrix} \qquad \mathbf{x}' = \begin{bmatrix} a \\ b \\ B \\ e \end{bmatrix}$$

IDENTIFICATION OF GROSS MEASUREMENT ERRORS

If the reaction has already been normalized with respect to the biomass stoichiometric coefficient, then we set B equal to 1 and consider it not as a measured variable but as an exact quantity. In this case, the dimension of $\mathbf{x}'$ is reduced by one. Since during the normalization process all other stoichiometric coefficients are divided by B, the error in biomass measurement is distributed among the other stoichiometric coefficients. This has the disadvantage that gross errors in the biomass measurement will not be detected as such but diagnosed as errors in another quantity. When the biomass measurement errors are random, normalization will not affect the result. In this example, the above formulation will be retained so that the possibility of gross biomass errors can be considered, and the equality constraint can be represented as $\mathbf{Ax} = \mathbf{0}$ with $\mathbf{x} = \mathbf{x}'$ and $\mathbf{A}$ as given above.

At a dilution rate of 0.008 h^{-1}, de Kok and Roels reported the measurements on the net flows of glucose, oxygen, biomass, and carbon dioxide. Since a continuous steady-state operation is achieved, there is no accumulation of chemical species, and the net fluxes indeed represent the relative values of the stoichiometric coefficients. Therefore, the measurement vector is:

$$\bar{\mathbf{x}} = \begin{bmatrix} a \\ b \\ B \\ e \end{bmatrix} = \begin{bmatrix} 0.35 \\ 3.80 \\ 1.00 \\ 1.40 \end{bmatrix}$$

The measurement errors used in this example are 6, 11.7, 5, and 11.1% for a (substrate), b (oxygen), B (biomass), and e (carbon dioxide), respectively. Thus, the variance-covariance matrix for δ is:

$$\psi = \begin{bmatrix} (0.35 \times 0.06)^2 & 0 & 0 & 0 \\ 0 & (3.80 \times 0.117)^2 & 0 & 0 \\ 0 & 0 & (1.00 \times 0.05)^2 & 0 \\ 0 & 0 & 0 & (1.40 \times 0.111)^2 \end{bmatrix}$$

Equations (18) and (20) give:

$$\epsilon = -\mathbf{A}\bar{\mathbf{x}} = -\begin{bmatrix} -6 & 0 & 1.0 & 1 \\ 0 & -2 & -0.1 & 2 \end{bmatrix} \begin{bmatrix} 0.35 \\ 3.80 \\ 1.00 \\ 1.40 \end{bmatrix} = \begin{bmatrix} -0.30 \\ 4.90 \end{bmatrix}$$

$$\phi = \mathbf{A}\psi\mathbf{A}^T = \begin{bmatrix} 0.04253 & 0.04805 \\ 0.04805 & 0.88730 \end{bmatrix} \qquad \phi^{-1} = \begin{bmatrix} 25.0480 & -1.3564 \\ -1.3564 & 1.20050 \end{bmatrix}$$

Applying eq. (25) yields for the test function:

$$h = \epsilon^T \phi^{-1} \epsilon = 35.07$$

This value is compared to the χ^2 distribution function of 2 degrees of freedom (two constraint equations). Because $\chi^2_{0.9}(2) = 4.61 < 35.07 = h$, the hypothesis testing fails if a 90% level of confidence is desired. Thus, we can state with 90% of confidence that some of the measurements have gross errors that are much larger than can be reasonably expected from the original variance-covariance structure. Note that this conclusion is reached in a very early stage of the error analysis. Normally, we do not need to proceed further to calculate the maximum likelihood estimate for this entire set of measurements because it will inevitably be erroneous, but we shall calculate it here so that it can be referred to later for the purpose of comparison. Applying eqs. (21) and (22) directly and substituting the required numerical values, we have:

$$\hat{\delta} = \psi A^T \phi^{-1} \epsilon = \begin{bmatrix} 0.0375 \\ -2.4864 \\ -0.0370 \\ -0.0382 \end{bmatrix} \qquad \hat{x} = \bar{x} + \hat{\delta} = \begin{bmatrix} 0.3875 \\ 1.3136 \\ 0.9630 \\ 1.3618 \end{bmatrix} \begin{matrix} \cdots a \\ \cdots b \\ \cdots B \\ \cdots e \end{matrix}$$

Had the hypothesis test been passed, we would stop here. However, since gross errors are suspected, the deletion algorithm is applied to locate the source of errors.

First, consider the deletion of measurement $x_1 = a$ (i.e., the glucose measurement). The matrices are partitioned according to eqs. (27), (28), and (29):

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_u \\ \mathbf{x}_c \end{bmatrix} \qquad \mathbf{x}_u = \begin{bmatrix} 3.80 \\ 1.00 \\ 1.40 \end{bmatrix} \qquad \mathbf{x}_c = [0.35]$$

$$\mathbf{A} = [\mathbf{A}_u \mathbf{A}_c] \qquad \mathbf{A}_u = \begin{bmatrix} 0 & 1.0 & 1 \\ -2 & -0.1 & 2 \end{bmatrix} \qquad \mathbf{A}_c = \begin{bmatrix} -6 \\ 0 \end{bmatrix}$$

$$\psi = \begin{bmatrix} \psi_u & 0 \\ 0 & \psi_c \end{bmatrix}$$

$$\psi_u = \begin{bmatrix} (3.80 \times 0.117)^2 & 0 & 0 \\ 0 & (1.00 \times 0.05)^2 & 0 \\ 0 & 0 & (1.40 \times 0.111)^2 \end{bmatrix} \qquad \psi_c = [(0.35 \times 0.06)^2]$$

The scalar to be inverted in eq. (35) is:

$$\mathbf{A}_c^T \phi^{-1} \mathbf{A}_c = [-6 \quad 0] \begin{bmatrix} 25.0480 & -1.3564 \\ -1.3564 & 1.2005 \end{bmatrix} \begin{bmatrix} -6 \\ 0 \end{bmatrix} = 901.73$$

$$[\mathbf{A}_c^T \phi^{-1} \mathbf{A}_c]^{-1} = 0.00111$$

Equation (35) is used to calculate the variance-covariance matrix for ϵ after the deletion of a:

$$\phi_u^{-1} = \phi^{-1}\{\mathbf{I} - \mathbf{A}_c[\mathbf{A}_c^T \phi^{-1} \mathbf{A}_c]^{-1} \mathbf{A}_c^T \phi^{-1}\} = \begin{bmatrix} 0 & 0 \\ 0 & 1.127 \end{bmatrix}$$

After this, the new h is calculated according to eq. (36), and the hypothesis testing is applied to the set of retained measurements:

$$h_a = \epsilon^T \phi_u^{-1} \epsilon = 27.06$$

Because of the deletion of one measurement, the number of constraint equations is reduced from 2 to 1. The above test function is now compared to $\chi^2_{0.9}(1) = 2.71$. Thus, we see that deleting a only brings the performance index from $h = 35.07$ down to $h_a = 27.06$, which is still far short of what is needed to pass the test.

Next, consider the deletion of the measurement $x_2 = b$ (i.e., the oxygen measurement.) The partitioning of matrices for this case is done as follows:

$$\mathbf{x}_u = \begin{bmatrix} 0.35 \\ 1.00 \\ 1.40 \end{bmatrix} \qquad \mathbf{x}_c = [3.80]$$

$$\mathbf{A}_u = \begin{bmatrix} -6 & 1.0 & 1 \\ 0 & -0.1 & 2 \end{bmatrix} \qquad \mathbf{A}_c = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$$

$$\psi_u = \begin{bmatrix} (0.35 \times 0.06)^2 & 0 & 0 \\ 0 & (1.00 \times 0.05)^2 & 0 \\ 0 & 0 & (1.40 \times 0.111)^2 \end{bmatrix} \qquad \psi_c = [(3.80 \times 0.117)^2]$$

These matrices give:

$$\mathbf{A}_c^T \phi^{-1} \mathbf{A}_c = 4.8019$$

$$[\mathbf{A}_c^T \phi^{-1} \mathbf{A}_c]^{-1} = 0.2083$$

$$\phi_u^{-1} = \phi^{-1}\{\mathbf{I} - \mathbf{A}_c[\mathbf{A}_c^T \phi^{-1} \mathbf{A}_c]^{-1} \mathbf{A}_c^T \phi^{-1}\} = \begin{bmatrix} 23.515 & 0 \\ 0 & 0 \end{bmatrix}$$

The new test function for the case of deletion of b is:

$$h_b = \epsilon^T \phi_u^{-1} \epsilon = 2.12$$

Since $h_b = 2.12$ is less than $\chi^2_{0.9}(1) = 2.71$, we can state with a 90% level of confidence that the measurement b may be erroneous. The maximum likelihood estimates of the measurement errors and the measurements themselves are given by:

$$\hat{\delta} = \psi_u \mathbf{A}_u^T \phi_u^{-1} \epsilon = \begin{bmatrix} 0.0187 \\ -0.0176 \\ -0.1704 \end{bmatrix}$$

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \hat{\delta} = \begin{bmatrix} 0.3687 \\ 0.9824 \\ 1.2296 \end{bmatrix} \begin{matrix} \cdots a \\ \cdots B \\ \cdots e \end{matrix}$$

The reconstructed value for the deleted measurement b is 1.1805.

The same procedure is followed for the cases of deletion of B and e. The resulting test functions after deletions of B and e are $h_B = 26.43$ and $h_e = 34.96$, respectively. Comparing the test functions after each deletion, we find that deleting b gives rise to the lowest test function. Since the test function obtained after deleting b is small enough to pass the hypothesis test, no further simultaneous deletion of two measurements is necessary, nor is it possible to do so due to the limited number of constraint equations available. The error analysis carried out so far indicates that the measurement of b should be deleted. The proposed method of statistical analysis enables us to reach this conclusion smoothly and systematically. De Kok and Roels have also reached the same conclusion on this set of measurements by comparing it to other similar sets of measurements and by relying heavily on their past experience as to what values RQ should have. Thus, we see how the proposed method can be used systematically in the face of uncertainty, and it is this systematized approach that is especially suited for the use in computerized data analysis and control of a biochemical reactor, although human experience, to be sure, is still indispensable.

APPENDIX B: SEQUENTIAL ANALYSIS

The basic results that are essential for the actual implementation are presented here without proof. (See Romagnoli and Stephanopoulos[9] for a detailed discussion.) In the discussion that follows, the subscript i signifies the value for the ith step, and the subscript $i - 1$ represents the value of the last *successful* step. The variables affected by these subscripts are $\hat{\psi}$, $\hat{\delta}$, $\hat{\mathbf{x}}$, and h because they need be updated as each balance equation is processed. The prime mark (′) is reserved for those variables which represent the *incremental* changes as a result of processing the ith step; whereas, the absence of the prime mark on a variable signifies that the variable under concern is the overall cummulative quantity.

First, we partition the $m \times n$ matrix A as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_i \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \tag{37}$$

Now we process each subdivided balance equation in $\mathbf{Ax} = \mathbf{0}$ individually. Each subdivided balance equation is expressed as:

$$\mathbf{A}_i\mathbf{x} = \mathbf{0} \qquad i = 1, 2, \ldots, m \tag{38}$$

Before any balance equation is processed, no correction can yet be applied to the measurements; thus,

$$\hat{\boldsymbol{\delta}}_0 = \mathbf{0} \tag{39}$$

$$\hat{\mathbf{x}}_0 = \bar{\mathbf{x}} + \hat{\boldsymbol{\delta}}_0 = \bar{\mathbf{x}} \tag{40}$$

Naturally, the variance-covariance matrix for $\hat{\mathbf{x}}_0 - \mathbf{x}$ is the same as the original variance-covariance matrix for $\boldsymbol{\delta}$, i.e.

$$\hat{\boldsymbol{\psi}}_0 \equiv E[(\hat{\mathbf{x}}_0 - \mathbf{x})(\hat{\mathbf{x}}_0 - \mathbf{x})^T] = E[\hat{\boldsymbol{\delta}}_0\hat{\boldsymbol{\delta}}_0^T] = E[\boldsymbol{\delta}\boldsymbol{\delta}^T] = \boldsymbol{\psi} \tag{41}$$

The maximum likelihood incremental adjustments for the measurements due to the ith constraint equation is:

$$\hat{\boldsymbol{\delta}}_i' = \hat{\boldsymbol{\psi}}_{i-1}\mathbf{A}_i^T\boldsymbol{\phi}_i^{-1}\boldsymbol{\epsilon}_i' \tag{42}$$

and the corresponding variance-covariance matrix for $\hat{\boldsymbol{\delta}}_i'$, which can also be thought of as the incremental decrease in the variance-covariance of the estimate of $\mathbf{x}$ thus far, is:

$$\hat{\boldsymbol{\psi}}_i' \equiv E[\hat{\boldsymbol{\delta}}_i'\hat{\boldsymbol{\delta}}_i'^T] = \hat{\boldsymbol{\psi}}_{i-1}\mathbf{A}_i^T\boldsymbol{\phi}_i^{-1}\mathbf{A}_i\hat{\boldsymbol{\psi}}_{i-1} \tag{43}$$

where $\boldsymbol{\phi}_i$ and $\boldsymbol{\epsilon}_i'$ are calculated by the following equations:

$$\boldsymbol{\phi}_i = \mathbf{A}_i\hat{\boldsymbol{\psi}}_{i-1}\mathbf{A}_i^T \tag{44}$$

$$\boldsymbol{\epsilon}_i' = \boldsymbol{\epsilon}_i - \mathbf{A}_i\hat{\boldsymbol{\delta}}_{i-1} = -\mathbf{A}_i(\bar{\mathbf{x}} + \hat{\boldsymbol{\delta}}_{i-1}) = -\mathbf{A}_i\hat{\mathbf{x}}_{i-1} \tag{45}$$

Finally, the overall correction up to now, the maximum likelihood estimate of $\mathbf{x}$ at the end of the ith step, and the corresponding variance-covariance matrix are updated according to the following equations:

$$\hat{\boldsymbol{\delta}}_i = \hat{\boldsymbol{\delta}}_{i-1} + \hat{\boldsymbol{\delta}}_i' \tag{46}$$

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} + \hat{\boldsymbol{\delta}}_i' = \bar{\mathbf{x}} + \hat{\boldsymbol{\delta}}_i \tag{47}$$

$$\hat{\boldsymbol{\psi}}_i \equiv E[(\hat{\mathbf{x}}_i - \mathbf{x})(\hat{\mathbf{x}}_i - \mathbf{x})^T] = \hat{\boldsymbol{\psi}}_{i-1} - \hat{\boldsymbol{\psi}}_i' \tag{48}$$

As expected, eqs. (42), (43), (44), (45), (47), and (48) reduce to eqs. (21), (23), (20), (18), (22), and (24), respectively, when only a single step is taken to solve $\mathbf{Ax} = \mathbf{0}$. One of the advantages of using this sequential method of processing is that $\boldsymbol{\phi}_i$, which is the matrix to be inverted, is only a scalar when one equation at a time is considered.

Once again, we can perform a hypothesis testing to detect the presence of gross errors as each equation is processed by formulating a test function for each stage i:

$$h_i = \hat{\boldsymbol{\delta}}_i^T\boldsymbol{\psi}^{-1}\hat{\boldsymbol{\delta}}_i \tag{49}$$

This test function is then compared against $\chi^2_{1-\theta}(k)$, where k is the number of successful steps thus far plus 1 for the current step, i.e. k is i minus the number of unsuccessful steps. Actually, it can be tested before the calculation of the overall correction, $\hat{\delta}_i$, is carried out. This can be done by using the following identities:

$$\begin{aligned} h_i &= \sum_{j=1}^{i} \hat{\delta}_j'^T \psi^{-1} \hat{\delta}_j' = \sum_{j=1}^{i} \epsilon_j'^T \phi_j^{-1} \epsilon_j' \\ &= h_{i-1} + \hat{\delta}_i'^T \psi^{-1} \hat{\delta}_i' = h_{i-1} + \epsilon_i'^T \phi_i^{-1} \epsilon_i' \\ &= h_{i-1} + h_i' \end{aligned} \tag{50}$$

Conceptually, the last equality in the above equation is due to the fact that the term $\hat{\delta}_i'^T \psi^{-1} \hat{\delta}_i' = \epsilon_i'^T \phi_i^{-1} \epsilon_i'$ can be considered as the incremental increase in h as a result of processing the ith subdivided constraint equation. In the above equation, only the terms corresponding to the successful steps are included in the summation.

If the test is successful, we proceed to calculate the adjustments needed for this step and then to the next constraint equation. If the test is failed, one of the measurements included in the ith equation is suspected of having gross errors, and it is then isolated through the process of serial deletions described earlier.

This sequential procedure can be used to analyze many sets of data taken under the same conditions. Estimation can be carried out as soon as the pertinent data become available. The maximum likelihood estimates of the first set can be combined in a weighted manner with the original data of the second set to produce a new set of estimates and an updated variance-covariance matrix. This new set of data can once again be fed through the same estimation and error detection algorithm. Thus, this sequential error analysis algorithm is analogous to the methods of running average or Kalman filtering.

The example of Appendix A is reworked here to illustrate the application of the sequential analysis to error identification. We start by partitioning the matrix A according to eq. (37):

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \qquad \mathbf{A}_1 = [-6 \quad 0 \quad 1 \quad 1] \qquad \mathbf{A}_2 = [0 \quad -2 \quad -0.1 \quad 2]$$

The initial values for $\hat{\delta}$, $\hat{\mathbf{x}}$, $\hat{h}$, and $\hat{\psi}$ are:

$$\hat{\delta}_0 = \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{\mathbf{x}}_0 = \bar{\mathbf{x}} = \begin{bmatrix} 0.35 \\ 3.80 \\ 1.00 \\ 1.40 \end{bmatrix} \qquad h_0 = 0$$

$$\hat{\psi}_0 = \psi = \begin{bmatrix} (0.35 \times 0.06)^2 & 0 & 0 & 0 \\ 0 & (3.80 \times 0.117)^2 & 0 & 0 \\ 0 & 0 & (1.00 \times 0.05)^2 & 0 \\ 0 & 0 & 0 & (1.40 \times 0.111)^2 \end{bmatrix}$$

From eqs. (44) and (45), we can calculate:

$$\phi_1 = \mathbf{A}_1 \hat{\psi}_0 \mathbf{A}_1^T = 0.04253 \qquad \phi_1^{-1} = 23.515$$

$$\epsilon_1' = -\mathbf{A}_1 \hat{\mathbf{x}}_0 = -[-6 \quad 0 \quad 1 \quad 1] \begin{bmatrix} 0.35 \\ 3.80 \\ 1.00 \\ 1.40 \end{bmatrix} = -0.30$$

From eq. (50), the incremental increase in the test function due to the processing of the first constraint equation is given by:

$$h_1' = \epsilon_1'^T \phi_1^{-1} \epsilon_1' = 2.12$$

Thus, the test function after processing the first equation is:

$$h_1 = h_0 + h_1' = 0 + 2.12 = 2.12$$

Since only one equation has been processed so far, the test function is compared to $\chi^2_{0.9}(m = 1) = 2.71 > 2.12 = h_1$, which indicates that the test is passed. Had the test been failed, we would set the incremental changes of $\hat{\delta}$, $\hat{x}$, $\hat{\psi}$, and h for this step to 0. However, since the test shows that this step is successful, we proceed to calculate the incremental changes of $\hat{\delta}$, $\hat{x}$, and $\hat{\psi}$ directly according to eqs. (42) and (43).

$$\hat{\delta}_1' = \hat{\psi}_0 A_1^T \phi_1^{-1} \epsilon_1' = \begin{bmatrix} 0.0187 \\ 0.0000 \\ -0.0176 \\ -0.1704 \end{bmatrix} \begin{matrix} \cdots a \\ \cdots b \\ \cdots B \\ \cdots e \end{matrix}$$

$$\hat{\psi}_1' = \hat{\psi}_0 A_1^T \phi_1^{-1} A_1 \hat{\psi}_0 = \begin{bmatrix} 0.00016 & 0.00000 & -0.00016 & -0.00150 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 \\ -0.00016 & 0.00000 & 0.00015 & 0.00142 \\ -0.00150 & 0.00000 & 0.00142 & 0.01371 \end{bmatrix}$$

Finally, each overall estimate so far is updated.

$$\hat{\delta}_1 = \hat{\delta}_0 + \hat{\delta}_1' = \begin{bmatrix} 0.0187 \\ 0.0000 \\ -0.0176 \\ -0.1704 \end{bmatrix} \begin{matrix} \cdots a \\ \cdots b \\ \cdots B \\ \cdots e \end{matrix} \qquad \hat{x}_1 = \hat{x}_0 + \hat{\delta}_1' = \begin{bmatrix} 0.3687 \\ 3.8000 \\ 0.9824 \\ 1.2296 \end{bmatrix} \begin{matrix} \cdots a \\ \cdots b \\ \cdots B \\ \cdots e \end{matrix}$$

$$\hat{\psi}_1 = \hat{\psi}_0 - \hat{\psi}_1' = \begin{bmatrix} 0.00028 & 0.00000 & 0.00016 & 0.00150 \\ 0.00000 & 0.19767 & 0.00000 & 0.00000 \\ 0.00016 & 0.00000 & 0.00235 & -0.00142 \\ 0.00150 & 0.00000 & -0.00142 & 0.01044 \end{bmatrix}$$

Now, we are ready to analyze the second constraint equation. Similar to the handling of the first constraint equation, we have for this step:

$$\phi_2 = A_2 \hat{\psi}_1 A_2^T = 0.8330$$

$$\phi_2^{-1} = 1.200$$

$$\epsilon_2' = -A_2 \hat{x}_1 = 5.239$$

$$h_2' = \epsilon_2'^T \phi_2^{-1} \epsilon_2' = 32.95$$

$$h_2 = h_1 + h_2' = 2.12 + 32.95 = 35.07$$

The value of $\chi^2_{0.9}(m = 2) = 4.61$ shows that $h_2 = 35.07$ is too high to pass the test. Thus, the presence of gross errors is detected, and no correction on $\hat{\delta}$, $\hat{x}$, $\hat{\psi}$, and h can come from this step. Instead, the second constraint equation is used to find the source of gross error. By going through the serial deletion of the involved measurements b, B, and e as have already been done in the previous section, we find that the measurement of b is erroneous. The same conclusion has been

reached in the batch processing of the same experimental data in the previous section. The estimates obtained in the last successful step become the final results for this set of measurements, which are again consistent with those obtained in the batch processing.

Another way to identify the source of gross error is to argue as follows. Since this second constraint equation involves a new measurement of b that is not included in the first constraint equation, we can conclude that the measurement of b possesses gross errors. Although this reasoning usually strikes out the correct sources of gross errors, our past experiences in analyzing other data shows that it need not always be true. A statistically erroneous measurement may remain undetected during the first encounter because the algorithm does not see the whole picture initially. During the first encounter, certain combinations of the other measurements may soften the aggravation caused by an erroneous measurement and pull the test function just barely within the passing range. Therefore, to be more confident about the judgment, it is best to check for all the measurements involved in the current step before deciding the choice of deletion. If the error is detected during the second or later encounter of that measurement, we need to go back and reanalyze the equation in which the measurement in question is first encountered. For the same statistical reasons, sequential analysis may also occasionally discard some other measurements whose deletions do not give the lowest performance indices in the batch processing. However, it is not very clear, and one will never really know, what precisely constitute the right choices, because, after all, the decisions on deletion are made using statistical concepts.

APPENDIX C: EFFECT OF CORRELATION ON TEST FUNCTION

To see the effect of correlated terms in the covariance matrix ψ on the test function h, we follow a mathematical approach similar to that used in locating the sources of gross errors. The original measurement vector $\mathbf{x}$ is again partitioned into two parts: a uncorrelated measurements which have a purely diagonal covariance matrix ψ_a and b correlated measurements which have a covariance matrix ψ_b that is composed of both a purely diagonal ψ_{bd} and a purely off-diagonal ψ_{bo}.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} \qquad \mathbf{A} = [\mathbf{A}_a \mathbf{A}_b] \tag{51}$$

$$\psi = \begin{bmatrix} \psi_a & 0 \\ 0 & \psi_{bd} \end{bmatrix} \qquad \psi_d = \begin{bmatrix} \psi_a & 0 \\ 0 & \psi_b \end{bmatrix} = \begin{bmatrix} \psi_a & 0 \\ 0 & \psi_{bd} + \psi_{bo} \end{bmatrix} \tag{52}$$

Following eqs. (32) and (33), we get:

$$\phi_d \equiv \mathbf{A}\psi_d\mathbf{A}^T = \phi + \mathbf{A}_b\psi_{bo}\mathbf{A}_b^T \tag{53}$$

where

$$\phi \equiv \mathbf{A}\psi\mathbf{A}^T$$

$$\phi_d^{-1} = \phi^{-1} - \phi^{-1}\mathbf{A}_b[\psi_{bo}^{-1} + \mathbf{A}_b^T\phi^{-1}\mathbf{A}_b]^{-1}\mathbf{A}_b^T\phi^{-1} \tag{54}$$

where the subscript d signifies that the variable involved contains the effect of off-diagonal ψ_{bo}. and the lack of it show that the variable is devoid of ψ_{bo}. Thus, when the measurements are correlated, the test function is:

$$\begin{aligned} h_d &= \epsilon^T\phi_d^{-1}\epsilon = \epsilon^T\phi^{-1}\epsilon - \epsilon^T\phi^{-1}\mathbf{A}_b[\psi_{bo}^{-1} + \mathbf{A}_b^T\phi^{-1}\mathbf{A}_b]^{-1}\mathbf{A}_b^T\phi^{-1}\epsilon \\ &= h_{\text{diagonal}} + h_{\text{correlation}} \end{aligned} \tag{55}$$

The above equation shows that the presence of off-diagonal ψ_{bo} affects the test function by $-\epsilon^T\phi^{-1}\mathbf{A}_b[\psi_{bo}^{-1} + \mathbf{A}_b^T\phi^{-1}\mathbf{A}_b]^{-1}\mathbf{A}_b^T\phi^{-1}\epsilon$. The test function is increased if $[\psi_{bo}^{-1} + \mathbf{A}_b^T\phi^{-1}\mathbf{A}_b]$ is negative definite and decreased if $[\psi_{bo}^{-1} + \mathbf{A}_b^T\phi^{-1}\mathbf{A}_b]$ is positive definite.

The example of Appendix A is reworked here to show the effect of an error in the gas flow rate measurement on the test function. We assume that the gas flow rate contributes 8% to the 11.7%

error in the measurement of b (i.e., O_2 consumption rate) and the 11.1% error in the measurement of e (i.e., CO_2 evolution rate):

$$\bar{x}_b = \begin{bmatrix} b \\ e \end{bmatrix} = \begin{bmatrix} 3.80 \\ 1.40 \end{bmatrix} \quad A_b = \begin{bmatrix} 0 & 1 \\ -2 & 2 \end{bmatrix} \quad \epsilon = -A\bar{x} = \begin{bmatrix} -0.30 \\ 4.90 \end{bmatrix}$$

$$\psi_b = \begin{bmatrix} (3.80)^2(0.117)^2 & (3.80)(1.40)(0.08)^2 \\ (3.80)(1.40)(0.08)^2 & (1.40)^2(0.111)^2 \end{bmatrix}$$

$$\psi_{bo} = \begin{bmatrix} 0 & (3.80)(1.40)(0.08)^2 \\ (3.80)(1.40)(0.08)^2 & 0 \end{bmatrix}$$

$$h_{\text{correlation}} = -\epsilon^T \phi^{-1} A_b [\psi_{bo}^{-1} + A_b^T \phi^{-1} A_b]^{-1} A_b^T \phi^{-1} \epsilon = 4.45$$

$$h_d = h_{\text{diagonal}} + h_{\text{correlation}} = 35.07 + 4.45 = 39.52$$

As can be seen from the expression for ψ_{bo}, the test function is less affected when the fraction of error that is contributed by the gas flow rate measurement is smaller. The effect of correlation for the example in Appendix

Nomenclature

A	the matrix in the balance equation **Ax** = **0** (dimension = $m \times n$)
A_c	partitioned **A** that corresponds to the deleted measurements. (dimension = $m \times c$)
A_u	partitioned **A** that corresponds to the retained measurements. (dimension = $m \times u$)
a, b, c, d, e, f	stoichiometric coefficients in eq. (1); a is in mol substrate/C mol biomass; b is in mol O_2/C mol biomass; c is in mol NH_3/C mol biomass; d is in mol H_2O/C mol biomass; e is in CO_2/C mol biomass; f is in mol product/C mol biomass
B	biomass stoichiometric coefficients in eq. (1)
b	nonhomogeneous term in the balance equation
C_j	concentration of component j in the system (mol/L)
c	number of measurements to be deleted
D	dilution rate (h^{-1})
H_j	heat of formation of species j in eq. (6) (kcal/mol)
h	a constant term added to the measured stoichiometric coefficients to absorb the nonhomogeneous term **b**
h	test function
h_j	test function after deleting measurement j
i	number of equations processed thus far
$i - 1$	the last successful step
J	performance index
k	$k = i -$ (number of unsuccessful steps)
m	number of constraint equations
m_{ATP}	maintenance requirement for ATP (mol ATP C mol^{-1} biomass h^{-1})
n	number of measurements
P/O	efficiency of the oxidative phosphorylation (mol ATP/mol O)
Q	heat evolution in eq. (6) (kcal/C mol biomass)
R	rate of biomass formation; extent of reaction (1) (mol biomass L^{-1} h^{-1})
RQ	respiratory quotient (mol CO_2/mol O_2)
u	number of measurements remaining after deleting c measurements
V	volume (L)
x, y, z	composition of substrate in eq. (1)

$\mathbf{x}$	true parameter vector for measured variables (dimension = n)
$\bar{\mathbf{x}}$	measurement vector (dimension = n)
$\hat{\mathbf{x}}$	estimate of measurement vector (dimension = n)
$\mathbf{x}_c$	measured variables to be deleted (dimension = c)
$\mathbf{x}_u$	measured variables to be retained (dimension = u)
$Y_{\mathrm{ATP}}^{\max}$	maximum growth yield on ATP (C mol biomass/mol ATP)

Greek

$\alpha, \beta, \gamma, \delta$	composition of biomass in eq. (1)
$\alpha', \beta', \gamma', \delta'$	composition of metabolic product in eq. (1)
γ_b	reductance degree of biomass (equivalent available electrons/C mol biomass)
δ	measurement error vector (dimension = n)
$\hat{\delta}$	estimate of δ (dimension = n).
$\hat{\delta}_n$	estimate of δ when gross errors are present in measurements (dimension = n)
$\hat{\delta}_u$	estimate of δ for the undeleted measurements (dimension = u)
ϵ	balance equation residual vector (dimension = m)
E	expected value operator
$1 - \theta$	confidence level of χ^2 distribution
ν_j	stoichiometric coefficient of component j in eq. (6)
σ_b	weight fraction of carbon in biomass (dimensionless)
σ_j	standard deviation of the jth measurement
Φ_j	net rate of input of component j to the system by transport (mol/h)
ϕ	variance-covariance matrix for ϵ with no gross error (dimension = $m \times m$); $\phi = E[\epsilon\epsilon^T]$
ϕ_n	variance-covariance matrix for ϵ when $\Delta\psi$ is present (dimension = $m \times m$)
ϕ_u^{-1}	singular matrix of ϕ_n^{-1} after deletion of c measurements (dimension = $m \times m$); $\phi_u^{-1} \equiv \lim_{(\Delta\psi)_i \to \infty} \phi_n^{-1}$
χ^2	chi-square probability distribution
ψ	variance-covariance matrix for δ (dimension = $n \times n$); $\psi = E[\delta\delta^T]$
ψ_c	ψ for c deleted measurements with no gross error (dimension = $c \times c$)
ψ_n	variance-covariance matrix for δ_n with gross error (dimension = $n \times n$); $\psi_n = E[\delta_n\delta_n^T]$
ψ_u	ψ for u retained measurements with no gross error (dimension = $u \times u$)
$(\Delta\psi)$	gross error (dimension = $c \times c$)
$\hat{\psi}$	variance-covariance matrix for $\hat{\delta}$ (dimension = $n \times n$); $\hat{\psi} = E[\hat{\delta}\hat{\delta}^T]$

Subscripts

c	signifies that it is of the measurements to be deleted
u	signifies that it is of the measurements to be retained
n	signifies that it is of the measurements which contain both random and gross errors

Diacritics

$\bar{}$	measured values
$\hat{}$	estimated values

Partial financial assistance from NSF grant No. CPE-8118877 is gratefully acknowledged.

References

1. C. L. Cooney, H. Y. Wang, and D. I. C. Wang, *Biotechnol. Bioeng.*, 19, 55 (1977).
2. L. E. Erickson, S. E. Selga, and U. E. Viesturs, *Biotechnol. Bioeng.*, 20, 1623 (1978).

3. L. E. Erickson, I. G. Minkevich, and V. K. Eroshin, *Biotechnol. Bioeng.*, **21**, 575 (1979).
4. A. Ferrer and L. E. Erickson, *Biotechnol. Bioeng.*, **22**, 421 (1980).
5. D. R. Kuehn and H. Davidson, *Chem. Eng. Prog.*, **57**, 44 (1961).
6. D. L. Ripps, *Chem. Eng. Prog. Symp. Ser. 55*, **61**, 8 (1965).
7. S. Nogita, *Ind. Eng. Chem. Process Des. Develop.*, **11**, 197 (1972).
8. F. Madron, V. Veverka, and V. Vanéček, *AIChE J.*, **23**, 482 (1977).
9. J. A. Romagnoli and G. Stephanopoulos, *Chem. Eng. Sci.*, **36**, 1849 (1981).
10. J. W. de Kwaadsteniet, J. C. Jager, and A. H. Stouthamer, *J. Theor. Biol.*, **57**, 103 (1976).
11. H. E. de Kok and J. A. Roels, *Biotechnol. Bioeng.*, **22**, 1097 (1980).
12. Th. G. E. Guerts, H. E. de Kok, and J. A. Roels, *Biotechnol. Bioeng.*, **22**, 2031 (1980).
13. J. G. J. Dekkers, H. E. de Kok, and J. A. Roels, *Biotechnol. Bioeng.*, **23**, 1023 (1981).
14. B. O. Solomon, L. E. Erickson, J. E. Hess, and S. S. Yang, *Biotechnol. Bioeng.*, **24**, 633 (1982).
15. A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*, (Hemisphere, Washington, DC, 1975), p. 2.
16. S. E. Selga, G. R. Mezhina, A. A. Lacars, and U. E. Viesturs, *Microbial Biomass and Metabolites* (Zinatne, Riga, Latvia, 1972) (in Russian), p. 21.
17. D. Herbert, *Seventh International Congress for Microbiology: Recent Progress in Microbiology* (Almqvist and Wiksell, Stockholm, 1959), p.381.
18. E. S. Pearson and H. O. Hartley, Eds., in *Biometrika Tables for Statisticians*, *Vol. 1*, 3rd ed. (Cambridge University Press, Cambridge, England, 1966).
19. G. Stephanopoulos and K. Y. San, *Proc. ISCRE-7*, *ASC Symp. Series*, **196**, 155 (1982).

Accepted for Publication March 17, 1983

APPENDIX F

LISTING

OF THE GROSS ERROR IDENTIFICATION CODES

```
c ************************************************************************
c ************************************************************************
c This is used to create a file named RUN.FOR whichc ontains the necessary
c   FORTRAN codes, with the correct dimensions to call the primary subroutine
c   ERROR used for gross error detection, identification and rectification.
c RUN.FOR is then compiled, linked with all other needed subroutines, and run
c   on a VAX computer under VMS operating system.
c ************************************************************************
c ************************************************************************

      PROGRAM CREATE
      write(6,601)
601   format('$the dimension of matrix a(m,n) = : ')
      read(5,501)m,n
501   format(i1,1x,i1)
      open(unit=10,file='run.for',status='new')

      write(10,602)n,n,n,n,n,n,n,n,m,m,m,m,
     *             m,n,n,m,m,n,m,m,m,n,
     *             n,n,m,m,n,n,
     *             n,n,n,n,
     *             m,m,n,n,
     *             m,m,m,m,m,m,m,m,m,m,
     *             n,n,n,n,m,n,n,m,n,n,
     *             m,m,
     *             m,n,
     *             n,n,n,m
602   format(
     *'      program test'/
     *'      dimension pn('i1','i1'),pn0('i1','i1'),pn1('i1','i1'),'
     *               'pn2('i1','i1'),pm('i1','i1'),pmi('i1','i1')'/
     *'      dimension a('i1','i1'),at('i1','i1'),aj('i1'),ai('i1'),'
     *               'aj2('i1',2),aj2t(2,'i1'),ntest('i1','i1')'/
     *'      dimension x('i1'),xx('i1'),crit('i1'),e('i1'),h1('i1'),'
     *               'nj('i1')'/
     *'      dimension d('i1'),dd0('i1'),dd1('i1'),dd2('i1')'/
     *'      dimension tpm1('i1'),tpm2('i1'),tpn1('i1'),tpn2('i1')'/
     *'      dimension tpmm1('i1','i1'),tpmm2('i1','i1'),tpmm3('i1','
     *               i1'),tpmm4('i1','i1'),tpmmi('i1','i1')'/
     *'      dimension tpnn1('i1','i1'),tpnn2('i1','i1'),tpmn1('i1','
     *               i1'),tpnm1('i1','i1'),pni('i1','i1')'/
     *'      dimension tp221(2,2),tp222(2,2),tp2m1(2,'i1'),'
     *               'tpm21('i1',2)'/
     *'      dimension ntpm1('i1'),nindex('i1')'/
     *'      dimension xdel('i1'),pndel('i1','i1'),edel('i1')'/
     *'      dimension a1(4,10),msrmt(10),natest(4,10),xxx(10),'
     *               'tp1(10),tp2(10)')

      write(10,605)m,n
605   format(
     *'      call ERROR(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,'
     *                'aj2t,ntest,'/
     *'     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,'
     *       'tpmm2,'/
```

```
     *'     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,'
     *       'tpm21,'/
     *'     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,'
     *       'tp1,tp2,'/
     *'     *'i1','i1')'/
     *'      stop'/
     *'      end')

      close(unit=10)
      stop
      end


c ***************************************************************************
c ***************************************************************************
c ***************************************************************************
c Rectification of process measurement data in the presence of gross errors.
c ***************************************************************************
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE ERROR(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n)
      dimension pn(n,n),pn0(n,n),pn1(n,n),pn2(n,n),pm(m,m),pmi(m,m)
      dimension a(m,n),at(n,m),aj(m),ai(n),aj2(m,2),aj2t(2,m),ntest(m,n)
      dimension x(n),xx(n),crit(m),e(m),h1(n),nj(n)
      dimension d(n),dd0(n),dd1(n),dd2(n)
      dimension tpm1(m),tpm2(m),tpn1(n),tpn2(n)
      dimension tpmm1(m,m),tpmm2(m,m),tpmm3(m,m),tpmm4(m,m),tpmmi(m,m)
      dimension tpnn1(n,n),tpnn2(n,n),tpmn1(m,n),tpnm1(n,m),pni(n,n)
      dimension tp221(2,2),tp222(2,2),tp2m1(2,m),tpm21(m,2)
      dimension ntpm1(m),nindex(n)
      dimension xdel(n),pndel(n,n),edel(m)
      dimension a1(4,10),msrmt(10),natest(4,10),xxx(10),tp1(10),tp2(10)
      character * 120 name

      ic = 10
      open(unit=ic,file='errdat',status='new')
      open(unit=11,file='data',status='old')

c ***************************************************************************
c Input the comments ********************************************************
c ***************************************************************************
      read(11,6)name
6     format(a120)
      write(ic,8)name
8     format(1x,a119,/)
c If iread=0, there is no reduction of a1 matrix, and a is entered directly.
      read(11,9)iread
9     format(i1)
```

```
      if(iread .eq. 0)go to 150

c  ***************************************************************************
c  Input the original matrix a1 and calculate the working matrix a ************
c  Put biomass term on #5                                          ************
c  msrmt(i) : 0      = not measured                                ************
c             1,2,3..= measurement order                           ************
c            -1      = not used                                   ************
c  ***************************************************************************
      read(11,12)(msrmt(j),j=1,10)
12    format(10i8)
      read(11,11)((a1(i,j),j=1,10),i=1,4)
11    format(10f8.3)
      write (ic,4)msrmt
4     format(' measurement identification:',/,1x,i10,9i12)
      call writa(a1,4,10,'a',1,10,4,ic,'(original)',10)
      call setna(natest,0,4,10)
      irow = 1
      do 110 j=1,10
        if(msrmt(j) .ne. 0)go to 110
        do 120 i=irow,4
120       if(abs(a1(i,j)) .gt. 0.01)go to 130
130     natest(irow,j) = 1
        call swapar(a1,i,irow,4,10)
        call delete(j,irow,4,a1,tp1,tp2,4,10)
        irow = irow + 1
110   continue
      irow = irow - 1
      call writa(a1,4,10,'a',1,10,4,ic,'(after deleting unmeasured varia
     *bles)',37)
c     Copy a1 to a for use in the actual analysis
      do 140 j=1,10
        if(msrmt(j) .le. 0)go to 140
        k = msrmt(j)
        do 135 i=1,4-irow
135       a(i,k)=a1(irow+i,j)
c     Chech to see that no other measurements are unintentially deleted
        do 136 i=1,4-irow
136       if(abs(a(i,k)) .gt. 0.01) go to 140
        write(5,666) k
666     format(' warning: measurement #',i2,' is effectively deleted')
        stop
140   continue
150   if(iread .eq. 0)read(11,13)((a(i,j),j=1,n),i=1,m)
13    format(<n>f8.3)
      call writa(a,m,n,'a',1,10,4,ic,' ',1)

c ****************************************************************************
c Input measurements, covariances, & significance test criteria **************
c ****************************************************************************
      read(11,5)(crit(i),i=1,m)
5     format(<m>f8.3)
      read(11,7)(x(i),i=1,n)
7     format(f12.7)
```

```
      read(11,7)(pn(i,i),i=1,n)
      read(11,7)(d(i),i=1,n)
      do 10 i=1,n
        pn(i,i)=(pn(i,i)*x(i))**2
        if(pn(i,i) .le. 1.e-8) read(11,7)pn(i,i)
10      x(i)=x(i)+d(i)
      call writx(x,n,'x',1,8,4,ic,'measurements:',13)
      call writa(pn,n,n,'pn',2,10,7,ic,'covariances of measurements',27)
      call writx(crit,m,'crit',4,6,2,ic,'criteria vector:',16)


c ***************************************************************************
c Miscellaneous junk ******************************************************
c ***************************************************************************
      call inverq(pni,pn,n)
c     create a m*m identity matrix tpmmi
      do 401 j=1,m
401     tpmmi(j,j) = 1.


c ***************************************************************************
c Check to see that error is within the specified level of significance ******
c ***************************************************************************
      call transp(at,a,m,n)
      call q2abat(pm,a,pn,at,tpnm1,m,n)
      call invert(pmi,pm,tpmm1,tpmm2,tpmm3,tpmm4,m)
      call mulyax(e,a,x,m,n)
      call scaler(e,-1.,e,m)
      call writx(e,m,'e',1,8,5,ic,'error vector:',13)
      call writa(pmi,m,m,'pmi',3,10,5,ic,'inverse error covariances:',26
     *)
      hh = xay(e,pmi,e,tpm1,m)
      write(ic,610)hh
610   format(' test function:',/,' hh = ',f6.2,/)

c ***************************************************************************
c Calculate the least square estimate without deleting any measurement *******
c ***************************************************************************
      call mulyax(tpm1,pmi,   e,m,m)
      call mulyax(tpn1, at,tpm1,n,m)
      call mulyax(   d, pn,tpn1,n,n)
      call addzxy(  xx,  d,   x,n)
      call writx(xx,n,'x',1,8,4,ic,'least square estimate of x:',27)
      if(iread .ne. 0)call UNMSRM(xx,xxx,msrmt,a1,natest,tp1,n,ic)
      if(m .eq. 1) go to 200

c ***************************************************************************
c Locate gross errors by deleting one measurement at a time ******************
c ***************************************************************************
      call DEL1(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n,ic,n1)
```

```
      if(h1(n1) .le. crit(m-1) .or. m .lt. 2) go to 200
c  ***************************************************************************
c  Locate gross errors by deleting two measurements at a time *****************
c  ***************************************************************************
       call DEL2(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n,ic)

c  ***************************************************************************
c  Sequential processing of Ax=0 **********************************************
c  ***************************************************************************
200   call AX0(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n,ic)

c  ***************************************************************************
c  Complete the calculation of the deleted measurements ***********************
c  ***************************************************************************
       call RECONS(xx,a,ntest,tpn1,m,n)
       call writx(xx,n,'x',1,8,4,ic,'estimate of measurements, after reco
     *nstructing the deleted measurements, is:',76)

c  ***************************************************************************
c  Complete the calculation of the unmeasured ones ****************************
c  ***************************************************************************
       if(iread .ne. 0)call UNMSRM(xx,xxx,msrmt,a1,natest,tp1,n,ic)

       close(unit=11)
       close(unit=ic)
       return
       end



c  ***************************************************************************
c  ***************************************************************************
c  Sequential processing of ax=0 **********************************************
c    'this step' variables                       == 0                         *
c    'overall last successiful step' variables == 1                           *
c    'overall' variables                         == 2                         *
c  The original x and pn are saved in xdel and pndel in case one has to go back.
c  ***************************************************************************
c  ***************************************************************************

       SUBROUTINE AX0(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n,ic)
       dimension pn(n,n),pn0(n,n),pn1(n,n),pn2(n,n),pm(m,m),pmi(m,m)
```

```
      dimension a(m,n),at(n,m),aj(m),ai(n),aj2(m,2),aj2t(2,m),ntest(m,n)
      dimension x(n),xx(n),crit(m),e(m),h1(n),nj(n)
      dimension d(n),dd0(n),dd1(n),dd2(n)
      dimension tpm1(m),tpm2(m),tpn1(n),tpn2(n)
      dimension tpmm1(m,m),tpmm2(m,m),tpmm3(m,m),tpmm4(m,m),tpmmi(m,m)
      dimension tpnn1(n,n),tpnn2(n,n),tpmn1(m,n),tpnm1(n,m),pni(n,n)
      dimension tp221(2,2),tp222(2,2),tp2m1(2,m),tpm21(m,2)
      dimension ntpm1(m),nindex(n)
      dimension xdel(n),pndel(n,n),edel(m)
      dimension a1(4,10),msrmt(10),natest(4,10),xxx(10),tp1(10),tp2(10)

      write(ic,612)
612   format(' sequential processing of ax=0 ************************')

c *************************************************************************
c Initialization
c *************************************************************************
      ndele2 = 0
      nrow = 1
      call assgx(xdel,x,n)
      call assgx(xx,x,n)
      call assga(pndel,pn,n,n)
      call assga(pn2,pn,n,n)
      call setna(ntest,0,m,n)
201   index1=0
      indexm=1
      ndele1 = ndele2
      hh1 = 0.
      call setnx(nindex,0,n)
      call setx(dd1,0.,n)
      call setx(dd2,0.,n)

c *************************************************************************
c Process one equation at a time, from eqn #nrow to #m
c *************************************************************************
c   nindex(n) ... The measurement that has been processed
c   index0    ... The no. of measurements that have been processed
c   index1
c   index2
c   indexm    ... The no. of equations that have been processed
c   ndele1    ... The no. of measurements that have been deleted
c   ndele2    ... The no. of going back is identified by
c *************************************************************************

      do 250 i=nrow,m
c *************************************************************************
c Find out how many variables have been processed successively so far
c *************************************************************************
        index2 = index1
        index0 = 0
        do 220 j=1,n
          ai(j) = a(i,j)
          if(abs(ai(j)) .gt. 0.01 .and. nindex(j) .eq. 0)index0=index0+1
220     continue
```

```
        index2=index2+index0

c ***************************************************************************
c Calculation of performance index
c ***************************************************************************
        pm1 = xay(ai,pn2,ai,tpn1,n)
        pm1 = 1./pm1
        ee  = -xy(ai,xx,n)
        hh2 = hh1 + ee*pm1*ee
        write(ic,621)i,hh2
621     format(' after handling ',i1,'-th equation',/,' hh = ',f6.2)
        if(hh2.le.crit(indexm) .or. m.eq.1 .or. ndele1.eq.n-2) go to 230

c ***************************************************************************
c Test is failed.
c Find the least desirable measurement & delete this from the subsequent
c ... rows of the matrix a when the sequential testing is unsuccessful.
c ***************************************************************************
        if(ndele1 .eq. 1)
     *  call del1(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *  x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *  tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *  ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *  m,n,ic,n1)
c     find out which measurement is the least desirable
        do 231 j=1, n
          n1 = nj(j)
231       if(abs(ai(n1)) .gt. 0.01) go to 232

232     if(nindex(n1) .eq. 0) go to 233
c ***************************************************************************
c Detection on the 2nd or later encounter *************************************
c ***************************************************************************
c       Set the measurement to be deleted batch-wise and its covariances to 0
        ndele2 = ndele2+1
        xdel(n1) = 0.
        do 236 k=1, n
          pndel( k,n1) = 0.
236       pndel(n1, k) = 0.
        call assgx(xx,xdel,n)
        call assga(pn2,pndel,n,n)

c     Set flag back to 0 for all the previous sequential deletions
        do 234 j=1,n
          do 234 k=nrow,m
234         if(ntest(k,j) .eq. 1) ntest(k,j)=0
c     Switch rows of the a matrix
        call swapar(a,i,nrow,m,n)
c     Set flag to record when and where batch deletion was made
        ntest(nrow,n1)= 2
c     Actual deletion
        call delete(n1,nrow,m,a,tpn1,tpn2,m,n)

c ***************************************************************************
```

```
c     Set nrow so that the next sequential analysis is performed from nrow
        nrow = nrow+1
        write(ic,625)n1,n1
625     format(' measurement #',i1,' is found to be erroneous during'
     *         ' 2nd or later encounter'/
     *         ' measurement #',i1,' is deleted.'/
     *         ' re-analyze the whole set of data'///)
        call writa(a,m,n,'a',1,5,1,ic,' ',1)
        go to 201

c ***************************************************************************
c Detection on the 1st encounter ******************************************
c ***************************************************************************
c   Correct the measurement and the covariance matrix due to sequental deletion
 233     ndele1 = ndele1+1
        xx(n1) = 0.
        do 235 k=1, n
          pn2(k,n1) = 0.
235       pn2(n1,k) = 0.

c     Set flag to record when and where sequential deletion was made
        ntest(i,n1) = 1
c     Actual deletion
        call delete(n1,i,m,a,tpn1,tpn2,m,n)
        write(ic,626)n1
626     format(' measurement #',i1,' is deleted.')
        go to 249

c ***************************************************************************
c Incremental measurement errors & covariances when test is passed **********
c ***************************************************************************
230     call mulyax(dd0,pn2,ai,n,n)
        call scaler(dd0,pm1*ee,dd0,n)
        call mulaxy(tpnn1,ai,ai,n,n)
        call q2abat(pn0,pn2,tpnn1,pn2,tpnn2,n,n)
        call scalaa(pn0,pm1,pn0,n,n)
c ***************************************************************************
c Update parameters
c ***************************************************************************
        do 221 j=1, n
221     if(abs(ai(j)) .gt. 0.01 .and. nindex(j) .eq. 0)nindex(j)=1
        index1=index2
        indexm=indexm+1
        call addzxy(dd1,dd1,dd0,n)
        hh1=hh2
        call subcab(pn2,pn2,pn0,n,n)
        call addzxy(xx,dd0,xx,n)

c ***************************************************************************
c Output of the result of the sequential analysis of this step
c ***************************************************************************
249     call writx(xx,n,'x',1,8,4,ic,'estimate of measurements:',25)
250   continue
```

```
      return
      end


c ***************************************************************************
c ***************************************************************************
c Locate gross errors by deleting one measurement at a time ******************
c The measurement & its covariances are copied to and worked out in xx & pn2.*
c The modified error covariances are temporarily stored in tpm1 & tpmm2. *****
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE DEL1(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n,ic,n1)
      dimension pn(n,n),pn0(n,n),pn1(n,n),pn2(n,n),pm(m,m),pmi(m,m)
      dimension a(m,n),at(n,m),aj(m),ai(n),aj2(m,2),aj2t(2,m),ntest(m,n)
      dimension x(n),xx(n),crit(m),e(m),h1(n),nj(n)
      dimension d(n),dd0(n),dd1(n),dd2(n)
      dimension tpm1(m),tpm2(m),tpn1(n),tpn2(n)
      dimension tpmm1(m,m),tpmm2(m,m),tpmm3(m,m),tpmm4(m,m),tpmmi(m,m)
      dimension tpnn1(n,n),tpnn2(n,n),tpmn1(m,n),tpnm1(n,m),pni(n,n)
      dimension tp221(2,2),tp222(2,2),tp2m1(2,m),tpm21(m,2)
      dimension ntpm1(m),nindex(n)
      dimension xdel(n),pndel(n,n),edel(m)
      dimension a1(4,10),msrmt(10),natest(4,10),xxx(10),tp1(10),tp2(10)

      write(ic,611)
611   format(' after deletion of one measurement ...')

c ***************************************************************************
c Delete one measuremet at a time, repeat for j=1 to n
c ***************************************************************************
      do 410 j=1,n
        if(pn(j,j) .le. 1.e-6)h1(j)=1.e6
        if(pn(j,j) .le. 1.e-6)go to 410
        do 405 i=1,m
405          aj(i) = a(i,j)
        f = xay(aj,pmi,aj,tpm1,m)
        f = 1./f
        call mulyxa( tpm1,   aj,  pmi,m,m)
        call mulaxy(tpmm1,   aj, tpm1,m,m)
        call scalaa(tpmm1,    f,tpmm1,m,m)
        call subcab(tpmm1,tpmmi,tpmm1,m,m)
        call mulcab(tpmm2,  pmi,tpmm1,m,m,m)
        h1(j) = xay(e,tpmm2,e,tpm1,m)
        write(ic,620) j,h1(j)
620     format(' after deleting ',i1,'-th measurement',/,' hh = ',f6.2)

c ***************************************************************************
c Calculate the least square estimate after deletion of one measurement
```

```
c ***************************************************************************
        call assgx(xx,x,n)
        call assga(pn2,pn,n,n)
        xx(j) = 0.
        do 413 k=1, n
          pn2(k,j) = 0.
413       pn2(j,k) = 0.
        call mulyax(tpm2,tpmm2,   e,m,m)
        call mulyax(tpn1,   at,tpm2,n,m)
        call mulyax(   d,  pn2,tpn1,n,n)
        call addzxy(  xx,    d,  xx,n)

c ***************************************************************************
c     Check to see the dependence of equations
c ***************************************************************************
        call assga(tpmn1,a,m,n)
        call setna(ntest,0,m,n)
        do 411 i=1, m
411       if(abs(tpmn1(i,j)) .gt. 0.01) go to 412
412     ntest(i,j) = 1
        call swapar(tpmn1,i,1,m,n)
        call delete(j,1,m,tpmn1,tpn1,tpn2,m,n)
        do 417 k=1,n
          if(k .eq. j) go to 417
          do 416 i=2,m
            if(abs(tpmn1(i,k)) .gt. 0.01) go to 417
416       continue
          write(ic,643) k
643       format(' measurement #',i1,' is also effectively deleted')
417     continue

c ***************************************************************************
c Reconstruct the deleted measurements
c ***************************************************************************
        call recons(xx,a,ntest,tpn1,m,n)
        call writx(xx,n,'x',1,8,4,ic,'least square estimate of x:',27)
410   continue

c ***************************************************************************
c Check to see which deletion gives rise to the least square objective.
c The result is ordered in array nj and the most erroneous measurement is
c   identified by n1.
c ***************************************************************************
420   call sequen(h1,tpn1,nj,n)
      n1 = nj(1)

      return
      end



c ***************************************************************************
c ***************************************************************************
c locate gross errors by deleting two measurements at a time ****************
```

```
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE DEL2(pn,pn0,pn1,pn2,pm,pmi,a,at,aj,ai,aj2,aj2t,ntest,
     *x,xx,crit,e,h1,nj,d,dd0,dd1,dd2,tpm1,tpm2,tpn1,tpn2,tpmm1,tpmm2,
     *tpmm3,tpmm4,tpmmi,tpnn1,tpnn2,tpnm1,pni,tp221,tp222,tp2m1,tpm21,
     *ntpm1,nindex,tpmn1,xdel,pndel,edel,a1,msrmt,natest,xxx,tp1,tp2,
     *m,n,ic)
      dimension pn(n,n),pn0(n,n),pn1(n,n),pn2(n,n),pm(m,m),pmi(m,m)
      dimension a(m,n),at(n,m),aj(m),ai(n),aj2(m,2),aj2t(2,m),ntest(m,n)
      dimension x(n),xx(n),crit(m),e(m),h1(n),nj(n)
      dimension d(n),dd0(n),dd1(n),dd2(n)
      dimension tpm1(m),tpm2(m),tpn1(n),tpn2(n)
      dimension tpmm1(m,m),tpmm2(m,m),tpmm3(m,m),tpmm4(m,m),tpmmi(m,m)
      dimension tpnn1(n,n),tpnn2(n,n),tpmn1(m,n),tpnm1(n,m),pni(n,n)
      dimension tp221(2,2),tp222(2,2),tp2m1(2,m),tpm21(m,2)
      dimension ntpm1(m),nindex(n)
      dimension xdel(n),pndel(n,n),edel(m)
      dimension a1(4,10),msrmt(10),natest(4,10),xxx(10),tp1(10),tp2(10)

      do 460 k=1,n-1
      do 450 j=k+1,n
        do 420 i=1,m
          aj2(i,1) = a(i,k)
          aj2(i,2) = a(i,j)
420     continue
        call transp( aj2t,  aj2,m,2)
        call q2atba(tp221, aj2t,  pmi, aj2,tpm21,m,2)
        call invers(tp222,tp221,2)
        call q2abat(tpmm2,  aj2,tp222,aj2t,tp2m1,m,2)
        call mulcab(tpmm1,tpmm2,  pmi,m,m,m)
        call subcab(tpmm1,tpmmi,tpmm1,m,m)
        call mulcab(tpmm2,  pmi,tpmm1,m,m,m)
        hh = xay(e,tpmm2,e,tpm1,m)
        write(ic,641)k,j,hh
641     format(' after deletion of two measurements, #',i1,' and',
     *         ' #',i1,' ...',/,' hh = ',f6.2)

c ***************************************************************************
c Calculate the least square estimate after deletion of two measurement
c ***************************************************************************
        call assgx(xx,x,n)
        xx(k) = 0.
        xx(j) = 0.
        call assga(pn2,pn,n,n)
        do 423 l=1, n
          pn2(l,k) = 0.
          pn2(k,l) = 0.
          pn2(l,j) = 0.
423       pn2(j,l) = 0.
        call mulyax(tpm2,tpmm2,   e,m,m)
        call mulyax(tpn1,   at,tpm2,n,m)
        call mulyax(   d,  pn2,tpn1,n,n)
        call addzxy(  xx,    d,  xx,n)
```

```
c ***************************************************************************
c     Check to see the dependence of equations
c ***************************************************************************
        call assga(tpmn1,a,m,n)
        call setna(ntest,0,m,n)
        do 421 i=1, m
421       if(abs(tpmn1(i,k)) .gt. 0.01) go to 422
422     call swapar(tpmn1,i,1,m,n)
        call delete(k,1,m,tpmn1,tpn1,tpn2,m,n)
        ntest(1,k) = 1

        do 424 i=2, m
424       if(abs(tpmn1(i,j)) .gt. 0.01) go to 425
        write(ic,624)
624     format(' reconstruction is not valid')
        go to 449
425     call swapar(tpmn1,i,2,m,n)
        call delete(j,2,m,tpmn1,tpn1,tpn2,m,n)
        ntest(2,j) = 1

        do 419 l=1,n
          if(l .eq. k) go to 419
          if(l .eq. j) go to 419
          do 418 i=3,m
            if(abs(tpmn1(i,l)) .gt. 0.01) go to 419
418       continue
          write(ic,643) l
643       format(' measurement #',i1,' is also effectively deleted')
419     continue

c ***************************************************************************
c Reconstruct the deleted measurements
c ***************************************************************************
        call recons(xx,tpmn1,ntest,tpn1,m,n)
449     call writx(xx,n,'x',1,8,4,ic,'least square estimate of x:',27)
450   continue
460   continue

      return
      end



c ***************************************************************************
c ***************************************************************************
c Delete nd-th column from the subsequent rows of the original a matrix, *****
c   starting from the (i0+1)-th row up to and including the i1-th row.   *****
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE DELETE(nd,i0,i1,a,temp1,temp2,m,n)
      dimension a(m,n),temp1(n),temp2(n)
      do 5 k=1, n
```

```
5       temp1(k) = a(i0,k)
      alpha1 = temp1(nd)
      do 20 j=i0+1, i1
        if(abs(a(j,nd)) .le. 1.e-8)go to 20
        do 10 k=1, n
10        temp2(k) = a( j,k)
        alpha2 = temp2(nd)
        alpha  = alpha1/alpha2
        call scaler(temp2,alpha,temp2,n)
        call subzxy(temp2,temp1,temp2,n)
        do 15 k=1, n
15        a(j,k) = temp2(k)
20    continue
      return
      end



c ***************************************************************************
c ***************************************************************************
c Reconstruct the deleted measurements  ************************************
c ntest shows where deletion was made   ************************************
c   ntest = 1  ... sequential deletion  ************************************
c   ntest = 2  ... batch deletion       ************************************
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE RECONS(x,a,ntest,temp,m,n)
      dimension x(n), a(m,n), ntest(m,n), temp(n)
      do 20 i=m, 1, -1
        do 10 j=1, n
          if(ntest(i,j) .eq. 0) go to 10
          do 5 k=1, n
5           temp(k) = a(i,k)
          x(j) = -xy(temp,x,n)/temp(j)
10      continue
20    continue
      return
      end



c ***************************************************************************
c ***************************************************************************
c Complete the calculation of the unmeasured ones **************************
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE UNMSRM(x,xx,msrmt,a,natest,temp1,n,ic)
      dimension a(4,10),msrmt(10),natest(4,10)
      dimension x(n),xx(10),temp1(10)
      call setx(xx,0.,10)
      do 10 j=1,n
        do 15 i=1,10
```

```
15        if(msrmt(i) .eq. j)go to 16
16      k=msrmt(i)
        xx(i) = x(k)
10    continue
      call recons(xx,a,natest,temp1,4,10)
      call writx(xx,10,'x',1,8,4,ic,'final estimate of coefficients:',
     *31)
      call scaler(xx,1./xx(5),xx,10)
      call writx(xx,10,'x',1,8,4,ic,'normalized final estimate of coeffi
     *cients:',42)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Sequence the array x from the smallest to the largest    ********************
c   nj(i) ...  array index of the i-th smallest element    ********************
c   y(i)  ...  the original array x in the sequenced order ********************
c ***************************************************************************
c ***************************************************************************

      SUBROUTINE SEQUEN(x,y,nj,n)
      dimension x(n),nj(n),y(n)
      do 10 i=1, n
      nj(i) = i
   10 y(i) = x(i)
      do 20 i=2, n
        do 30 j=i, 2, -1
          if(y(j) .ge. y(j-1)) go to 30
          temp   = y(j)
          y(j)   = y(j-1)
          y(j-1) = temp
          ntemp  = nj(j)
          nj(j)  = nj(j-1)
          nj(j-1)= ntemp
   30   continue
   20 continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c ***************************************************************************
c List of subroutines on vector and matrix manipulation
c ***************************************************************************
c ***************************************************************************
c ***************************************************************************
c READC(iout)
c           Read and print comments from the terminal.
c READX(x,n,name,namea,ia,ib,iout,label,labela)
```

```
c           Read a real array x(n) from the terminal.
c READNX(nx,n,name,namea,ia,iout,label,labela)
c           Read an integer array nx(n) from the terminal.
c COPYX(x,n,ia,ib,iin)
c           Read a real array x(n).
c COPYNX(nx,n,ia,iin)
c           Read an integer array nx(n).
c WRITX(x,n,name,namea,ia,ib,iout,label,labela)
c           Write a real array x(n).
c WRITNX(nx,n,name,namea,ia,iout,label,labela)
c           Write an integer array nx(n).
c READA(A,m,n,name,namea,ia,ib,iout,label,labela)
c           Read a real matrix A(m,n) from the terminal.
c READNA(NA,m,n,name,namea,ia,iout,label,labela)
c           Read an integer matrix NA(m,n) from the terminal.
c COPYA(A,m,n,ia,ib,iin)
c           Read a real matrix A(m,n).
c COPYNA(NA,m,n,ia,iin)
c           Read an integer matrix NA(m,n).
c READD(A,n,name,namea,ia,ib,iout,label,labela)
c           Read diagonal matrix A(n,n) from the terminal.
c WRITA(A,m,n,name,namea,ia,ib,iout,label,labela)
c           Write matrix A(m,n).
c WRITNA(NA,m,n,name,namea,ia,iout,label,labela)
c           Write matrix NA(m,n).
c ************************************************************************
c SETX(x,alpha,n)            Set array x(n) to alpha.
c SETNX(nx,nalpha,n)         Set integer array nx(n) to nalpha.
c SETA(A,alpha,m,n)          Set matrix A(m,n) to alpha.
c SETNA(NA,nalpha,m,n)       Set integer matrix NA(m,n) to nalpha.
c ASSGX(y,x,n)               Set array x(n) to array y(n).
c ASSGNX(ny,nx,n)            Set integer array nx(n) to integer array ny(n).
c ASSGA(B,A,m,n)             Set matrix A(m,n) to matrix B(m,n).
c ASSGNA(NB,NA,m,n)          Set integer matrix A(m,n) to integer matrix B(m,n).
c ADDZXY(z,x,y,n)            Vector addition: z(n) = x(n) + y(n).
c ADDCAB(C,A,B,m,n)          Matrix addition: C(m,n) = A(m,n) + B(m,n).
c SUBZXY(z,x,y,n)            Vector subtraction: z(n) = x(n) - y(n).
c SUBCAB(C,A,B,m,n)          Matrix subtraction: C(m,n) = A(m,n) - B(m,n).
c SCALER(y,alpha,x,n)        Multiply a scaler constant to the vector x(n).
c SCALAA(B,alpha,A,m,n)      Multiply a scaler constant to the matrix A(m,n).
c FUNCTION XY(x,y,n)         Evaluate the scalar xt(n)*y(n).
c FUNCTION XAY(x,A,y,wkn,n) Evaluate the scalar xt(n)*A(n,n)*y(n).
c MULAXY(A,x,y,m,n)          Matrix multiplication: A(m,n) = x(m)*y(n).
c MULYAX(y,A,x,m,n)          Matrix multiplication: y(m) = A(m,n)*x(n).
c MULYXA(y,x,A,m,n)          Matrix multiplication: y(n) = x(m)*A(m,n).
c MULCAB(C,A,B,m,l,n)        Matrix multiplication: C(m,n) = A(m,l)*B(l,n).
c Q2ABAT(C,A,B,At,wknm,m,n) Quardratic matrix multiplication: C(m,m)=A*B*At.
c Q2ATBA(C,At,B,A,wkmn,m,n) Quardratic matrix multiplication: C(n,n)=At*B*A.
c SYSFIL(A,n)                Fill in the lower half of the symmetric matrix A.
c TRANSP(At,A,m,n)           Transpose of matrix A(m,n) => At(n,m).
c SQRX2A(A,x,n,ibegin,na,nx)Change an array x(nx) to a matrix A(na,na).
c SQRA2X(A,x,n,ibegin,na,nx)Change a mrtrix A(na,na) to an array x(nx).
c INVERS(B,A,n)              Inverse of a square matrix A(n,n).
c INVERQ(Ai,A,n)             Inverse of diagonal square matrix A(n,n).
```

```
c INVERT(Ai,A,U,B,Ui,Bi,n)  Inverse of square matrix A(n,n).
c SWAPX(x,y,n)              Switch x(n) and y(n).
c SWAPA(A,B,m,n)            Switch A(m,n) and B(m,n).
c SWAPAR(A,i,j,m,n)         Switch two rows i and j of matrix A(m,n).
c SWAPAC(A,i,j,m,n)         Switch two columns i and j of matrix A(m,n).
c ***************************************************************************
c ***************************************************************************


c ***************************************************************************
c ***************************************************************************
c Read and print comments from the terminal.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE READC(iout)
      character * 120 name
      write(6,10)
10    format(' enter the comments')
      read(5,20)name
20    format(a120)
      write(iout,30)name
30    format(1x,a119,/)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read a real array x(n) from the terminal.
c   ia,ib ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE READX(x,n,name,namea,ia,ib,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension x(n)
      id = int(float(n)/10.) + 1
      write(6,3)name
3     format(' enter the array ',a<namea>)
4     write(6,5)((name,i),i=1,n)
5     format(1x,<n>(a<namea>,i<id>,<ia-id-namea>x))
      read(5,10,err=4) (x(i),i=1,n)
10    format(<n>f<ia>.<ib>)
14    write(6,15)
15    format('$if want echo, enter 1; otherwise, enter 0  : ')
      read(5,20,err=14) iecho
20    format(i1)
      if(iecho .eq. 0) return
      call writx(x,n,name,namea,ia,ib,iout,label,labela)
      return
      end
```

```
c ***************************************************************************
c ***************************************************************************
c Read an integer array nx(n) from the terminal.
c   ia    ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE READNX(nx,n,name,namea,ia,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension nx(n)
      id = int(float(n)/10.) + 1
      write(6,3)name
3     format(' enter the array ',a<namea>)
4     write(6,5)((name,i),i=1,n)
5     format(1x,<n>(a<namea>,i<id>,<ia-id-namea>x))
      read(5,10,err=4) (nx(i),i=1,n)
10    format(<n>i<ia>)
14    write(6,15)
15    format('$if want echo, enter 1; otherwise, enter 0  : ')
      read(5,20,err=14) iecho
20    format(i1)
      if(iecho .eq. 0) return
      call writnx(nx,n,name,namea,ia,iout,label,labela)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read a real array x(n).
c   iin   ... input file specification
c   ia,ib ... format specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE COPYX(x,n,ia,ib,iin)
      dimension x(n)
4     read(iin,10,err=4) (x(i),i=1,n)
10    format(<n>f<ia>.<ib>)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read an integer array nx(n).
c   iin   ... input file specification
c   ia    ... format specification
c ***************************************************************************
```

```
c ***********************************************************************
      SUBROUTINE COPYNX(nx,n,ia,iin)
      dimension nx(n)
4     read(iin,10,err=4) (nx(i),i=1,n)
10    format(<n>i<ia>)
      return
      end


c ***********************************************************************
c ***********************************************************************
c Write a real array x(n).
c   ia,ib ... format specification
c   iout  ... output file specification
c ***********************************************************************
c ***********************************************************************
      SUBROUTINE WRITX(x,n,name,namea,ia,ib,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension x(n)
      id = int(float(n)/10.) + 1
      write(iout,20)label
20    format(1x,a<labela>)
      write(iout,30) ((name,i,x(i)),i=1,n)
30    format(1x,a<namea>,'(',i<id>,') = ',f<ia>.<ib>)
      write(iout,40)
40    format(1x)
      return
      end


c ***********************************************************************
c ***********************************************************************
c Write an integer array nx(n).
c   ia    ... format specification
c   iout  ... output file specification
c ***********************************************************************
c ***********************************************************************
      SUBROUTINE WRITNX(nx,n,name,namea,ia,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension nx(n)
      id = int(float(n)/10.) + 1
      write(iout,20)label
20    format(1x,a<labela>)
      write(iout,30) ((name,i,nx(i)),i=1,n)
30    format(1x,a<namea>,'(',i<id>,') = ',i<ia>)
      write(iout,40)
40    format(1x)
      return
      end
```

```
c ***************************************************************************
c ***************************************************************************
c Read a real matrix A(m,n) from the terminal.
c   ia,ib ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE READA(A,m,n,name,namea,ia,ib,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension a(m,n)
      id = int(float(n)/10.) + 1
      write(6,3)name
3     format(' enter the matrix ',a<namea>)
4     write(6,5)((name,i),i=1,n)
5     format(1x,<n>(a<namea>,i<id>,<ia-id-namea>x))
      read(5,10,err=4) ((a(i,j),j=1,n),i=1,m)
10    format(<n>f<ia>.<ib>)
14    write(6,15)
15    format('$if want echo, enter 1; otherwise, enter 0  : ')
      read(5,20,err=14) iecho
20    format(i1)
      if(iecho .eq. 0) return
      call writa(a,m,n,name,namea,ia,ib,iout,label,labela)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read an integer matrix NA(m,n) from the terminal.
c   ia,ib ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE READNA(NA,m,n,name,namea,ia,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension na(m,n)
      id = int(float(n)/10.) + 1
      write(6,3)name
3     format(' enter the matrix ',a<namea>)
4     write(6,5)((name,i),i=1,n)
5     format(1x,<n>(a<namea>,i<id>,<ia-id-namea>x))
      read(5,10,err=4) ((na(i,j),j=1,n),i=1,m)
10    format(<n>i<ia>)
14    write(6,15)
15    format('$if want echo, enter 1; otherwise, enter 0  : ')
      read(5,20,err=14) iecho
20    format(i1)
      if(iecho .eq. 0) return
```

```
      call writna(na,m,n,name,namea,ia,iout,label,labela)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read a real matrix A(m,n).
c   ia,ib ... format specification
c   iin   ... input file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE COPYA(A,m,n,ia,ib,iin)
      dimension a(m,n)
4     read(iin,10,err=4) ((a(i,j),j=1,n),i=1,m)
10    format(<n>f<ia>.<ib>)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read an integer matrix NA(m,n).
c   ia,ib ... format specification
c   iin   ... input file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE COPYNA(NA,m,n,ia,iin)
      dimension na(m,n)
4     read(iin,10,err=4) ((na(i,j),j=1,n),i=1,m)
10    format(<n>i<ia>)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Read a diagonal matrix A(n,n) from the terminal.
c   ia,ib ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE READD(A,n,name,namea,ia,ib,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension a(n,n)
      id = int(float(n)/10.) + 1
      write(6,3)name
3     format(' enter the matrix ',a<namea>)
4     write(6,5)((name,i),i=1,n)
5     format(1x,<n>(a<namea>,i<id>,<ia-id-namea>x))
```

```
      read(5,10,err=4) (a(i,i),i=1,n)
10    format(<n>f<ia>.<ib>)
14    write(6,15)
15    format('$if want echo, enter 1; otherwise, enter 0  : ')
      read(5,20,err=14) iecho
20    format(i1)
      if(iecho .eq. 0) return
      call writa(a,n,n,name,namea,ia,ib,iout,label,labela)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Write matrix A(m,n).
c   ia,ib ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE WRITA(A,m,n,name,namea,ia,ib,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension a(m,n)
      write(iout,29) name,label
29    format(1x,'matrix ',a<namea>,': ',a<labela>)
      write(iout,30) ((a(i,j),j=1,n),i=1,m)
30    format((1x,<n>(f<ia>.<ib>,2x)))
      write(iout,40)
40    format(1x)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Write matrix NA(m,n).
c   ia,ib ... format specification
c   iout  ... output file specification
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE WRITNA(NA,m,n,name,namea,ia,iout,label,labela)
      character * (*) name
      character * (*) label
      dimension na(m,n)
      write(iout,29) name,label
29    format(1x,'matrix ',a<namea>,': ',a<labela>)
      write(iout,30) ((na(i,j),j=1,n),i=1,m)
30    format((1x,<n>(i<ia>,2x)))
      write(iout,40)
40    format(1x)
      return
      end
```

```
c ***************************************************************************
c ***************************************************************************
c Set all the elements of array x(n) to alpha.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SETX(x,alpha,n)
      dimension x(n)
      do 10 i=1, n
        x(i) = alpha
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Set all the elements of integer array nx(n) to nalpha.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SETNX(nx,nalpha,n)
      dimension nx(n)
      do 10 i=1, n
        nx(i) = nalpha
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Set all the elements of matrix A(m,n) to alpha.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SETA(A,alpha,m,n)
      dimension a(m,n)
      do 10 i=1, m
      do 10 j=1, n
        a(i,j) = alpha
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Set all the elements of integer matrix NA(m,n) to nalpha.
c ***************************************************************************
c ***************************************************************************
```

```
      SUBROUTINE SETNA(NA,nalpha,m,n)
      dimension na(m,n)
      do 10 i=1, m
      do 10 j=1, n
        na(i,j) = nalpha
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Set all the elements of array x(n) to array y(n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE ASSGX(y,x,n)
      dimension y(n), x(n)
      do 10 i=1, n
        y(i) = x(i)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Set all the elements of integer array nx(n) to integer array ny(n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE ASSGNX(ny,nx,n)
      dimension ny(n), nx(n)
      do 10 i=1, n
        ny(i) = nx(i)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Set all the elements of matrix A(m,n) to matrix B(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE ASSGA(B,A,m,n)
      dimension b(m,n), a(m,n)
      do 10 i=1, m
      do 10 j=1, n
        b(i,j) = a(i,j)
10    continue
      return
      end
```

```
c ***************************************************************************
c ***************************************************************************
c Set all the elements of integer matrix A(m,n) to integer matrix B(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE ASSGNA(NB,NA,m,n)
      dimension nb(m,n), na(m,n)
      do 10 i=1, m
      do 10 j=1, n
        nb(i,j) = na(i,j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Vector addition: z(n) = x(n) + y(n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE ADDZXY(z,x,y,n)
      dimension z(n), x(n), y(n)
      do 10 j=1, n
        z(j) = x(j) + y(j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Matrix addition: C(m,n) = A(m,n) + B(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE ADDCAB(C,A,B,m,n)
      dimension a(m,n), b(m,n), c(m,n)
      do 10 i=1, m
      do 10 j=1, n
        c(i,j) = a(i,j) + b(i,j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Vector subtraction: z(n) = x(n) - y(n).
c ***************************************************************************
```

```
c ***************************************************************************
      SUBROUTINE SUBZXY(z,x,y,n)
      dimension z(n), x(n), y(n)
      do 10 j=1, n
        z(j) = x(j) - y(j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Matrix subtraction: C(m,n) = A(m,n) - B(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SUBCAB(C,A,B,m,n)
      dimension a(m,n), b(m,n), c(m,n)
      do 10 i=1, m
      do 10 j=1, n
        c(i,j) = a(i,j) - b(i,j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Multiply a scaler constant to the vector x(n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SCALER(y,alpha,x,n)
      dimension  y(n),x(n)
      do 10 j=1,n
           y(j) = alpha*x(j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Multiply a scaler constant to the matrix A(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SCALAA(B,alpha,A,m,n)
      dimension b(m,n), a(m,n)
      do 10 i=1,m
      do 10 j=1,n
        b(i,j) = alpha*a(i,j)
10    continue
      return
```

```
      end


c ***************************************************************************
c ***************************************************************************
c Evaluate the scalar xt(n)*y(n).
c ***************************************************************************
c ***************************************************************************
      FUNCTION XY(x,y,n)
      dimension x(n), y(n)
      xy=0.
      do 10 i=1,n
        xy = xy + x(i)*y(i)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Evaluate the scalar xt(n)*A(n,n)*y(n).
c ***************************************************************************
c ***************************************************************************
      FUNCTION XAY(x,A,y,wkn,n)
      dimension x(n), a(n,n), y(n), wkn(n)
      call mulyxa(wkn,x,a,n,n)
      xay = xy(wkn,y,n)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Matrix multiplication: A(m,n) = x(m)*y(n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE MULAXY(A,x,y,m,n)
      dimension a(m,n), x(m), y(n)
      do 10 i=1, m
        do 10 j=1, n
          a(i,j) = x(i)*y(j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Matrix multiplication: y(m) = A(m,n)*x(n).
c                        where y = vector (m)
```

```
c                               A = matrix (m,n)
c                               x = vector (n)
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE MULYAX(y,A,x,m,n)
      dimension y(m), a(m,n), x(n)
      do 10 i=1, m
        y(i) = 0.
        do 10 j=1, n
          y(i) = y(i) + a(i,j)*x(j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Matrix multiplication: y(n) = x(m)*a(m,n).
c                       where y = vector (n)
c                               A = matrix (m,n)
c                               x = vector (m)
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE MULYXA(y,x,A,m,n)
      dimension y(n), a(m,n), x(m)
      do 10 j=1, n
        y(j) = 0.
        do 10 i=1, m
          y(j) = y(j) + x(i)*a(i,j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Matrix multiplication: C(m,n) = A(m,l)*B(l,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE MULCAB(C,A,B,m,l,n)
      dimension c(m,n), a(m,l), b(l,n)
      do 10 i=1, m
        do 10 j=1, n
          c(i,j) = 0.
          do 10 k=1, l
            c(i,j) = c(i,j) + a(i,k)*b(k,j)
10    continue
      return
      end
```

```
c ***************************************************************************
c ***************************************************************************
c Quardratic matrix multiplication: C(m,m)=A(m,n)*B(n,n)*At(n,m).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE Q2ABAT(C,A,B,At,wknm,m,n)
      dimension c(m,m), a(m,n), b(n,n), at(n,m), wknm(n,m)
      call mulcab(wknm,b,at,n,n,m)
      call mulcab(c,a,wknm,m,n,m)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Quardratic matrix multiplication. C(n,n)=At(n,m)*B(m,m)*A(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE Q2ATBA(C,At,B,A,wkmn,m,n)
      dimension c(n,n), a(m,n), b(m,m), at(n,m), wkmn(m,n)
      call mulcab(wkmn,b,a,m,m,n)
      call mulcab(c,at,wkmn,n,m,n)
      return
      end


c ***************************************************************************
c ***************************************************************************
c Fill in the lower half of the symmetric matrix A.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SYSFIL(A,n)
      dimension a(n,n)
      do 10 i=1, n
      do 10 j=i+1, n
        a(j,i) = a(i,j)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Transpose of matrix A(m,n) => At(n,m)
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE TRANSP(At,A,m,n)
      dimension at(n,m),a(m,n)
      do 10 i=1, m
      do 10 j=1, n
        at(j,i) = a(i,j)
```

```
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Change an array x(nx) to a matrix A(na,na).
c        | 1  2  4  7  .. |
c        |    3  5  8  .. |
c    A = |       6  9  .. |
c        |         10  .. |
c        |             .. |
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SQRX2A(A,x,n,ibegin,na,nx)
      dimension a(na,na),x(nx)
      i = ibegin - 1
      do 10 k=1, n
      do 10 j=1, k
        i = i+1
        a(j,k) = x(i)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Change a mrtrix A(na,na) to an array x(nx).
c        | 1  2  4  7  .. |
c        |    3  5  8  .. |
c    A = |       6  9  .. |
c        |         10  .. |
c        |             .. |
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SQRA2X(a,x,n,ibegin,na,nx)
      dimension a(na,na),x(nx)
      i = ibegin - 1
      do 10 k=1, n
      do 10 j=1, k
        i = i+1
        x(i) = a(j,k)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
```

```
c Caculate the inverse of a square matrix A(n,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE INVERS(B,A,n)
      dimension b(n,n),a(n,n)
      det = a(1,1)*a(2,2) - a(1,2)*a(2,1)
      b(1,1) = a(2,2)/det
      b(1,2) =-a(1,2)/det
      b(2,1) =-a(2,1)/det
      b(2,2) = a(1,1)/det
      return
      end


c ***************************************************************************
c ***************************************************************************
c Determine the inverse of diagonal square matrix A(n,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE INVERQ(Ai,A,n)
      dimension ai(n,n), a(n,n)
      call seta(ai,0.,n,n)
      do 10 i=1, n
        ai(i,i) = 1./a(i,i)
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Determine the inverse of square matrix A(n,n).
c Method used: upper and lower triangular matrix decomposition
c A = B*U          Ai = Ui*Bi
c iwkn=order vector ... keep track of the pivoting
c nn=row dimension exactly as specified in the dimension statement in the
c    calling program.
c iflag=1 ... singlular matrix
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE INVERT(Ai,A,U,B,Ui,Bi,iwkn,n,iflag)
      dimension ai(n,n), a(n,n), u(n,n), b(n,n), ui(n,n), bi(n,n)
      dimension iwkn(1)

      iflag=0

c Initialize the order vector
      do 10 i=1,n
10      iwkn(i)=i
      call seta(b,0.,n,n,nn)
      call seta(u,0.,n,n,nn)
      call seta(bi,0.,n,n,nn)
```

```
      call seta(ui,0.,n,n,nn)

      do 20 i=1, n

c Piviting ... Find the maximum of a(j,i) in the ith column
        xmax=abs(a(i,i))
        imax=i
        do 21 j=i+1, n
          if(abs(a(j,i)) .gt. xmax)then
            xmax=abs(a(j,i))
            imax=j
          endif
21      continue

c Switch i<->imax for matrices A and B and vector iwkn
        if(imax .ne. i)then
          call swapar(a,i,imax,n,nn)
          call swapar(b,i,imax,n,nn)
          itemp=iwkn(i)
          iwkn(i)=iwkn(imax)
          iwkn(imax)=itemp
        endif

c Upper-Lower triangular decomposition
        u(i,i)=1.
        b(i,i) = a(i,i)
        do 25 k=1, i-1
          b(i,i) = b(i,i) - b(i,k)*u(k,i)
25      continue
c Check for singularity
        if(b(i,i) .eq. 0.)then
          iflag=1
          goto 100
        endif

        do 30 j=i+1, n
          b(j,i) = a(j,i)
          u(i,j) = a(i,j)
          do 35 k=1, i-1
            b(j,i) = b(j,i) - b(j,k)*u(k,i)
            u(i,j) = u(i,j) - b(i,k)*u(k,j)
35        continue
          u(i,j) = u(i,j)/b(i,i)
30      continue
20    continue

c Invert Lower triangular matrix
      do 40 i=1, n
        do 45 j=1, i
          if(i .eq. j)then
            bi(i,j)=1.
          else
            bi(i,j)=0.
          endif
```

```
          do 46 k=j, i-1
46          bi(i,j) = bi(i,j) - b(i,k)*bi(k,j)
          bi(i,j) = bi(i,j)/b(i,i)
45      continue
40    continue

c Invert Upper triangular matrix
      do 50 i=n, 1, -1
        do 55 j=i, n
          if(i .eq. j)then
            ui(i,j)=1.
          else
            ui(i,j)=0.
          endif
          do 56 k=i+1, j
56          ui(i,j) = ui(i,j) - u(i,k)*ui(k,j)
55      continue
50    continue

c Calculate Ai=Ui*Bi
      call mulcab(ai,ui,bi,n,n,n,nn,nn)
c Switch colums of Ai according to iwkn to get the Ai corresponding to the
c   original A; also switch the rows of A.
100   do 60 i=1, n
        do 65 j=i, n
          if(iwkn(j) .eq. i)goto 66
65      continue
66      if(imax .ne. i)then
          call swapac(ai,i,j,n,nn)
          call swapar(a,i,j,n,nn)
          itemp=iwkn(i)
          iwkn(i)=iwkn(j)
          iwkn(j)=itemp
        endif
60    continue

      return
      end



c ***************************************************************************
c ***************************************************************************
c Switch the contents of two arrays x(n) and y(n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SWAPX(x,y,n)
      dimension x(n),y(n)
      do 5 i=1, n
        temp = x(i)
        x(i) = y(i)
        y(i) = temp
5     continue
      return
```

```
      end


c ***************************************************************************
c ***************************************************************************
c Switch the contents of two matrices A(m,n) and B(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SWAPA(A,B,m,n)
      dimension a(m,n),b(m,n)
      do 10 i=1, m
        do 5 j=1, n
          temp   = a(i,j)
          a(i,j) = b(i,j)
          b(i,j) = temp
5       continue
10    continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Switch the contents of two rows i and j of matrix A(m,n).
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SWAPAR(A,i,j,m,n)
      dimension a(m,n)
      do 5 k=1, n
        temp   = a(i,k)
        a(i,k) = a(j,k)
        a(j,k) = temp
5     continue
      return
      end


c ***************************************************************************
c ***************************************************************************
c Switch the contents of two columns i and j of matrix A.
c ***************************************************************************
c ***************************************************************************
      SUBROUTINE SWAPAC(A,i,j,m,n)
      dimension a(m,n)
      do 5 k=1, m
        temp   = a(k,i)
        a(k,i) = a(k,j)
        a(k,j) = temp
5     continue
      return
      end
```

APPENDIX G

COMPUTER APPLICATIONS TO FERMENTATION PROCESSES

(The text of Appendix G consists of an article coauthored with G. N. Stephanopoulos which has appeared in *CRC Critical Reviews in Biotechnology*, **2**, No.1, 1-103, 1984.)

CRC CRITICAL REVIEWS in
BIOTECHNOLOGY

Volume 2 Issue 1

TABLE OF CONTENTS

Nam Sun Wang, co-author. B.Sc., University of California at Berkeley, Berkeley; M.Sc., California Institute of Technology, Pasadena. Graduate Research Assistant, Department of Chemical Engineering, California Institute of Technology, Pasadena.

Gregory N. Stephanopoulos, co-author. B.Sc., National Technical University of Athens, Athens, Greece; M.Sc., University of Florida, Gainesville; Ph.D., University of Minnesota, Minneapolis, Associate Professor of Chemical Engineering, California Institute of Technology, Pasadena.

Larry E. Erickson, referee. B.Sc., Ph.D., Kansas State University, Manhattan. Professor of Chemical Engineering, Kansas State University, Manhattan.

A review of real-time digital computer applications to fermentation processes is presented. The state of instrumentation and the principles of operation are briefly examined with a special emphasis on the on-line capabilities. Also reviewed are the recent developments in bioreactor state/parameter estimation with the utilization of various measurements, macroscopic material-energy balances, and the principles of basic biochemistry. The use of these estimates toward control or modeling purposes is discussed. Finally, the advances in the application of modern computer control and optimization techniques to fermentation processes are presented.

Volume 2, Issue 1

COMPUTER APPLICATIONS TO FERMENTATION PROCESSES

Authors: **Nam Sun Wang**
Gregory N. Stephanopoulos
Department of Chemical Engineering
California Institute of Technology
Pasadena, California

Referee: Larry E. Erickson
Department of Chemical Engineering
Kansas State University
Manhattan, Kansas

I. INTRODUCTION

During the last decade we have witnessed the use of computers in fermentation processes expanding at an impressive rate. Although the modern control theory was already well developed by the 1960s, practical applications of the theory were not feasible until an adequate tool for its implementation, i.e., the computer, became available. So fast was the computer's spread into general use after its introduction that people were even prompted to discuss seriously the social effects resulting from its applications.[1]

There are two main reasons for this widespread use of computers: affordability and necessity. Currently, the costs of computer logic and memory devices are plunging at an astonishing rate of 25 to 40%/year.[2] At the same time, the computational speed is also rapidly increasing, and the reliability is proportionately boosted. The advent of affordable microcomputers makes direct digital control (DDC) economically more advantageous than analog control, even for a fermentation process of relatively modest complexity. The break-even point is currently as low as 30 loops and still decreasing.[3] Hereforth, by computer we mean digital computer. Although analog computers are still widely used today, they are increasingly becoming obsolete. Because of the fast-paced evolution in computer technology, one should keep in mind that the terms describing the size of a computer are only relative; for example, a typical microcomputer today has the computing power comparable to that possessed by a minicomputer in the early 1970s and to that possessed by a mainframe computer in the early 1960s. Aside from the economic and technical advances in computer hardware, computers have increasingly become an indispensable instrument to aid those engaged in fermentation research in maximizing their productivity and in widening their scope of observation as a result of shorter time lag between the physical event and the detection. For the fermentation industry, economic pressure and competition make the use of computers still more critical, not only in data acquisition, analysis, and documentation, but also in process optimization and automatic control of a plant.

Although the resistance to computer application coming from the older fermentation technologists has mostly vanished,[4] to be sure, there are still some serious hinderances to a fully computerized and automated fermentation process. One of them is the lack of adequate on-line sensors in monitoring both the physical and biological conditions inside a reactor;[5] another one is the lack of understanding of microbial metabolic pathways and cellular control mechanisms, which is needed in order to model the process and formulate a meaningful process control algorithm so that the final objective of process optimization can be achieved. Because direct measurements on many important physiological variables are either inadequate or simply unavailable, elementary and macroscopic balances must be utilized to the fullest extent for the on-line estimation of many more parameters. Furthermore, efficient on-line

Reprinted from the CRC Critical Reviews in Biotechnology, Vol. 2, Issue 1, pages 1–103, © 1984 by CRC Press, Inc.

noise filtering and error detection/rectification algorithms based on statistical considerations are necessary to complement the inaccurate and unreliable sensors that we are currently forced to use. Much more sophisticated sensors are needed for the direct monitoring of the more complex fermentation processes that utilize multiple substrates and employ mixed cultures.

In addition to instrumentation, many different practical aspects of the problems in computer applications must be carefully considered if one desires to apply computers to fermentation processes successfully. Hardware/software structure, interfacing, monitoring, instrumentation, data acquisition, data analysis, state estimation, parameter identification, mathematical modeling, control, optimization, and many other selected topics must be individually studied and each problem painstakingly solved. Of course, depending on the needs and the objectives of each individual in using a computer, different areas may be emphasized. For instance, primary concern will be placed on the speed if the objective is merely simple data handling; on the other hand, profit generation will be the ultimate goal if process control is used in a production plant.

A review of real-time digital computer applications to fermentation processes is presented in this paper. This paper is not intended to be an exhaustive survey on the topic under consideration. Rather, it will concentrate on the state-of-the-art development with the emphasis placed on real-time operations. Although some of the results of microbial studies can often be directly applied to mammalian and plant cell cultures, the cultivation of these higher organisms may require quite different approaches and techniques and will not be the center of attention of this paper. Biological treatment of wastewater will not be considered as fermentation per se, and its discussion will be restricted to those few studies that potentially have direct applications to fermentation processes.

At first, a very brief historical review is presented to carry us from the beginning of computer applications in fermentation to the present. The subjects covered in the previously published general review papers by various writers from different backgrounds are summarized to bring the readers up to date on the development in the area of computer applications to fermentation processes. It is also hoped that such a summary will widen the perspective and help reduce the degree of bias that is inherently unavoidable in any papers of this nature. Among the first ones to be attacked by the earlier investigators are some of the practical problems in computer hardware/software configurations and interfacing and they will be discussed next. The examination of the state of instrumentation and the principles of operation of some of the typical sensors in use today will follow with a special emphasis on the on-line capabilities. The potential ways of utilizing various measurements to infer the conditions inside a fermentor are pointed out. Then we will discuss how the shortage in on-line direct measurement sensors can be partially alleviated by the use of indirect measurements, which are the result of combining those directly measurable variables, sometimes quite ingeniously. Also reviewed are the recent developments in bioreactor state/parameter estimation with the utilization of various measurements, filtering techniques, macroscopic material-energy balances, statistical theories, and the principles of basic biochemistry. The use of these estimates toward control or modeling purposes is discussed. Finally, the advances in the application of modern computer control and optimization techniques to fermentation processes are presented.

An attempt is made not to dwell extensively on the points covered by previous reviews. Some of the articles referred to in these reviews may be mentioned when it is deemed necessary to maintain the overall cohesiveness or to stress its contribution. Undoubtedly, many people who are actively involved in this area are barred from leaking industrial secrets, and their contribution, although undeniable, may never be recorded.

A. Historical Overview

The chemical process industry began utilizing direct digital computer control in the late 1950s. Since then the use of computer control in petroleum and petrochemical industries has been widespread.[5] On the other hand, the idea of applying computers to control a fermentation process was proposed in the early 1960s,[6] and the fermentation industry has lagged behind the chemical industry in the application of computers by approximately 10 years.[7]

Murao and Yamashita were among the first to discuss the possibility of using computers for the detection of abnormal fermentations.[8] At about the same time, the use of computers specifically for the purpose of modeling a fermentation process also emerged.[9] Subsequently, in 1969, Yamashita et al. reported that sequential control functions such as start-up, sterilization, and shutdowns were accomplished in pilot-scale and commercial batch glutamic acid fermentors by Ajinomoto Co.[10] In addition, they reported that two different approaches to optimization were tried. The first method relied on the detection of changes in the value of a principal parameter of a mathematical model. Although, in effect, an adaptive control scheme was considered, a rigorous treatment was not feasible at the time because of the problems encountered both in on-line identification of process characteristics and in the formulation of an algorithm, owing to the difficulties in deriving mathematical models. Nevertheless, the need and usefulness of an adaptive control scheme in performing self-optimization were emphasized. The second method was heuristic; a standard pattern was chosen based on the accumulated past experiences and each run was duplicated by following the standard pattern. In this first application of a computer to a fermentation process, temperature, pressure, pH, air flow rate, and foam were controlled, and effluent gas composition, glutamic acid concentration, dissolved oxygen, and microbe density were monitored.

Grayson reported in 1969 that direct digital control was conducted on 36 batch penicillin fermentation vessels with 114 control loops in Dista Products Ltd.[11] The controlled variables were similar to those described by Yamashita et al.

When the First International Conference on Computer Application in Fermentation Technology was held in Dijon, France, in 1973,[12] there had only been two production plants and one pilot-scale plant that were described in the published reports. Moreover, not all applications were completely successful. Within the next 5 years, we had witnessed about a dozen more institutions newly engaged, in one way or another and to different degrees of sophistication, in the application of computers to fermentation processes.[13] These institutions include, among others, the University of Pennsylvania, E. R. Squibb, Lord Rank Research Center, Station de Gene Microbiologique, Karolinska Institute, Gesselschaft Fur Biotechnologische Forschung mbH, Fermentation Design, and Massachusetts Institute of Technology.[13-15]

The Second International Conference on Computer Process Control in Fermentation was held in Philadelphia, Penn., in 1978,[16] and the third one was held soon after in Manchester, England in 1981.[17]

Due to corporate secrecy policies, many companies do not report their activities in technical journals. Consequently, a significant fraction of the published papers on this subject are contributed by researchers in academic laboratories. Thus, the relatively moderate level of reporting from industrial settings is not at all indicative of the fermentation industry's attitude toward the use of computers in production or pilot plants. Today, computers are accepted as a component in control systems throughout the fermentation industry,[18] and those few who have not yet adopted computers undoubtedly are seriously considering the possibilities of incorporating one.

B. General Reviews to Date

There have been a few excellent general review papers on the subject of computer control of fermentation processes at different stages of development. The proliferations in the number

of computer applications in the recent years make it difficult for anyone to keep abreast of the most recent advances. Even during the writing of this paper, many new articles are appearing in various journals. Thus, frequent, updated reviewing is useful in a fast evolving field such as this. A brief discussion of these reviews will provide a broader perspective on the progress of this subject and enable one to see the evolution at each stage of the development.

Nyiri was among the first to write a review article.[19] He examined the contributions prior to 1971. The publications he reviewed were mostly concerned with the off-line use of computers such as experimental data analysis, simulation, optimal trajectory calculation, and parameter estimation for kinetic model construction. Only three papers cited in his review dealt with the use of computers for the purpose of process control.[10,11,20] A substantial part of the article was devoted to various miscellaneous, but quite practical and essential considerations such as development of models, the choice of computers and programming languages, methodology, and algorithms for data logging and analysis. He also emphasized the importance of formulating accurate mathematical fermentation models for use in computer control.

In 1977, another outstanding review by Dobry and Jost examined the developments published between 1972 and 1977 from an industrial point of view.[13] They documented 11 existing fermentation systems interfaced to computers specifically for the purpose of monitoring and controlling a fermentor. Of the 11 systems being described, only two originated in industrial companies; the remaining were based in academic and institutional laboratories. The cost of computers was still a major factor in industry's decision to use computers; this was reflected by the fact that both reports from the industry dwelled on the cost justification of introducing computers to their respective fermentation systems. The academic and institutional systems were built with two different kinds of objectives: to study the problems involved in applying computers and to extend the monitoring and control capabilities of the existing laboratory. Employees of companies which supply fermentation or control equipment, notably Chemap, Fermentation Design, and Taylor Instrument, tended to be more idealistic in their approaches to the problem. Many articles from the vendors were aimed at expanding the market for their systems and concentrated on the description of the systems they had designed for various customers. Many articles concerning model development in the areas of biomass production (yeast) and product production (ethanol and antibiotics) were also found. In addition, another set of papers discussed the application of digital computer systems to control fermentors on a real-time basis; however, most of them were of a highly speculative nature, and only very few dealt with the actual on-line implementation of these ideas. Finally, Dobry and Jost elaborated on the more practical industrial concerns such as the choice in the design of a computer-coupled fermentation system (types of control, data storage, computer, etc.) and the system requirements (documentation, data logging, computer languages for programming and the process control, etc.).

In the following year, Weigand published a complementary review,[21] emphasizing the developments in computer application to fermentation from an academic/research viewpoint. The system at Purdue was extensively documented. The paper also superbly reviewed the studies on modeling, control, and parameter estimation, and optimization published from 1973 through 1978 that was missed by Dobry and Jost.[13] Because of the difference in interests resulting from the difference in his background, Weigand's emphasis was highly contrasted with Dobry and Jost's. There exist two main objectives in employing computers: whereas the final goal in an industrial installation may be the increased rate of product synthesis, the main goal in a research institution or a pilot plant may be merely an increased output of information (data).

In 1979, Armiger and Humphrey evaluated the use of the indirect measurement concept published in literature.[7] The merit of using component balancing techniques for the on-line estimation of biomass concentration and growth rates and the possibilities of using these

estimated values directly for control purposes were discussed. However, at the time, few studies were found to venture beyond the stages of on-line data acquisition and analysis. System hardware and software configurations were briefly discussed. There is a categorized list of references at the end of the paper. These references are in the areas of general papers, hardware and software system design, data analysis and modeling, and process control and optimization.

In the same year, Zabriskie gave an insightful review which concentrated on the applications of computers principally for control purposes.[22] Other areas were also discussed from the control perspective. Three representative studies in continuous fermentations and five studies in batch fermentations were each carefully scrutinized. The advantages and disadvantages of the two approaches used in tackling control problems in continuous fermentation, namely, black-box and model inference methods, were argued in the face of practical limitations of response time and sensor availability. He observed that there existed two approaches in batch fermentations: the use of empirical procedures and rigorous optimal control theory. He further proposed an optimal multivariable control strategy for a fed-batch fermentor.

In 1982 Rolf and Lim reviewed the existing hardware and software technology available for the implementation of computer control.[23] General modern control schemes that combined the advanced steady-state optimization algorithm for continuous fermentations and dynamic optimization algorithms for batch and fed-batch fermentations were commented on. This paper's inclusion of on-line estimation/identification and filtering aimed at feedback control was especially notable and revealed the recent surge of interest toward on-line feedback control with the aid of sophisticated instrumentation.

Finally, Hatch provided a review of the latest advances in computer applications in the analysis of fermentor conditions and in the classical type of process control of fermentors.[24] Bull also broadly surveyed the latest developments in computer-coupled instrumentation and the most recent studies in fermentation optimization.[25]

This brings us up to the present. Thus, we see that there are three main evolving stages in the use of computers — data acquisition, analysis, and control — and that each stage is progressively more difficult to solve. Ryu and Humphrey[26] divided the progress into five levels. The problems in the first stage need to be solved before a purposeful attempt can be made to resolve the questions in the second stage. Similarly, before the control function of a computer can be utilized, we must be able to perform an adequate degree of data logging and to analyze the data collected on the fermentation system. In addition to data acquisition, analysis, and control functions, computers are also quite useful in sequencing a predefined series of operations correctly.

When viewed as a whole, each stage is intimately connected to other stages. A successful execution of the functions of one stage depends on the successful implementation of functions not only of the same stage but also of other stages. The step of data acquisition includes such topics as instrumentation and computer interface. Provisions for operator access are also required if measurements are taken on an off-line basis. The step of data analysis involves the application of mathematical equations to the raw data. For example, data collected on the air flow rates and the partial pressures of various components, possibly with the help of temperature and total pressure measurements, are reduced to a set of more meaningful variables such as oxygen consumption rate or carbon dioxide evolution rates. Calculation for a number of indirect measurement variables is also possible through the use of some physical laws such as the concept of material balance. The data analysis step also includes error detections and the sending of alarms. Likewise, documentation and data storage/retrieval for the purposes of other calculations, graphic outputs, or for use in process control[27] fall within this category. The last step of control may encompass mathematical modeling, as well. Built into such a model is a fundamental understanding of the effect a

change in the controlled variables has on the state of the fermentor, may it be physical, chemical, or biological. Various subjects pertaining to the optimization of operating conditions and the implementation of optimal control algorithms belong to this last category as well.

II. COMPUTERS

Although the block diagram shown in Figure 1 is for a typical microcomputer system, other more powerful computers also operate with similar principles. Whether a microcomputer is dedicated to the low level control within a massive computer network or it represents the sole computing means for a small research project, the importance of a microcomputer's role in applying computer technology to fermentation cannot be denied. A microcomputer is composed of a central processing unit (CPU) and primary internal memories. The internal control within the CPU keeps track of the registers and regulates the arithmetic and logic unit. The quartz oscillator synchronizes the entire operation within a computer, with the speed of computation determined by its frequency and word size (i.e., the number of bits processed per cycle). The bus control supervises the flow of information between different components through a set of parallel conductors (bus). Necessary data and instruction codes are stored in memory devices, which may be internal or external to the computer. Some limited numbers of preprogrammed instructions are printed permanently on read-only memory (ROM) chips. Since the random memories (RAM) are relatively costly, their sizes are often severely limited; therefore, RAM can only be treated as temporary storage spaces for instructions and data during program execution. Rarely can all the necessary programs be loaded on the primary memories simultaneously, and a slightly larger program can easily fill the primary memory spaces of a small microcomputer to capacity and render it inoperative. Massive permanent storage of programs and data are provided by external memory devices such as disks and tapes. Various supporting input and output peripherals, process instruments, process controllers, and other computers all can be connected to the computer through some suitable interfacing devices.

A. Hardware

Various researchers have extensively described the computer hardware and software systems in their installations.[28-43] They discussed the mechanics of connecting computers to fermentors, most of which do not differ significantly from that encountered in chemical industries. Various degrees of automation and a wide range of computer selections are currently available. Recently, a market survey was conducted by Fox[44] to find the availability and cost of computer systems suitable for fermentation control. He listed five fermentation control requirements with varying degrees of sophistication ranging from simple laboratory data logging to controlling 20 production fermentors. He asked 50 manufacturers and suppliers to provide him with information and price quotations on each of the suggested systems, and found various proposals at each and every level. Thus, there is no problem in computer availability. There were even reports on reasonably low cost data acquisition and analysis systems using programmable desk-top computers.[39,45-47] However, regardless of the level of sophistication, a well-planned and efficient hardware and software configuration is needed to tie all the functions of data handling and control into one single highly effective and coherent automated fermentation system.

There are basically two kinds of computer-coupled systems in existence today. The first kind is more pilot-plant or production oriented, in which computers are connected to many different fermentors, and software is developed for the concurrent sequential scheduling of plant operations for all the fermentors, each at different stages of operation. Many fermentors are operated simultaneously in industry to produce different products. Even when only one

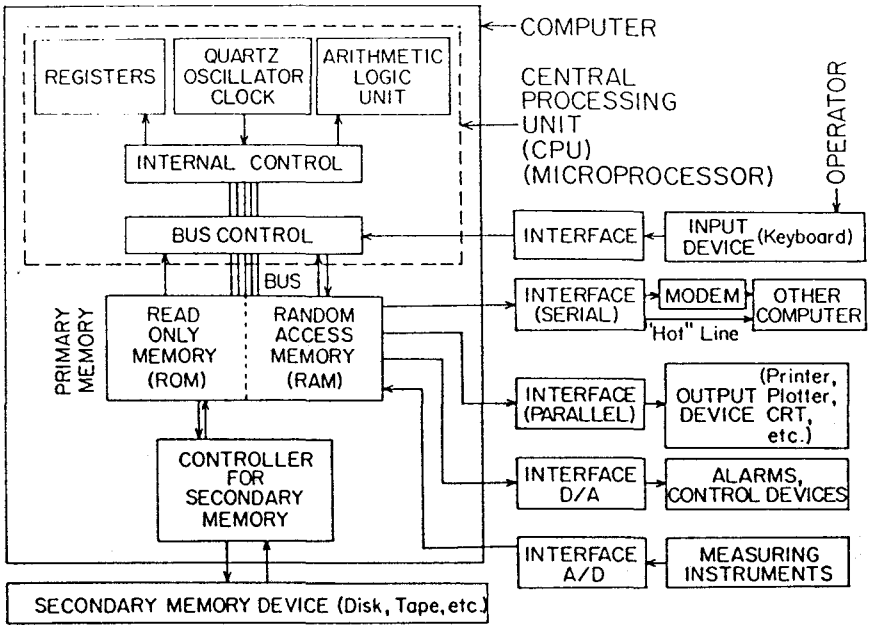


FIGURE 1. Block diagram of the hardware of a microcomputer.

product is produced, a multifermentor system is favored because of the possibility of strain selection. The use of multiple fermentors to study the effect of a large number of variables at the same time is also rather routine in the field of pharmaceuticals.[28] The second kind is more research oriented; usually, only one fermentor is controlled in which all of the data acquisition, data analysis, control, and optimization functions may be carried out simultaneously.

However, regardless of the type of systems, one must structure the hardware and software configurations so that the system can be operated efficiently as an integrated unit. Just as a well-structured computer program can be easily corrected when an error is detected and can be smoothly modified as the need arises, a properly designed hardware and software configuration will more than compensate for the initial small effort in provident planning. Flexibility to accommodate both the anticipated and the unforeseen future expansion and modification is an important factor to consider because new developments in products and fermentation technology will certainly force changes in the future.

1. System Configuration

Recently, a hierarchical computer system, as shown in Figure 2, is becoming quite popular, especially at large installations.[48-55] The advances of such a hierarchical system is listed by Blachere and Peringer[52] and Hennigan et al.[54] This is analogous to the use of subroutines in computer programming in the sense that one can change hardware and/or software in one computer without affecting the assigned routines in other computers; such a structure also facilitates simplified system changes. A battery of smaller computers, as opposed to one large computer, has the advantage that only one fermentor is affected in the event of computer failure. A hierarchical system is divided into different functional levels,[55] and progressively smaller computers are usually used as the level gets lower.[54] The first-level computer generally carries high priority on-line measurement and simple proportional-integral-derivative (PID) control tasks. The second level is frequently reserved for on-line data analysis and sophisticated control algorithms aimed toward optimization. Finally, modeling is carried out

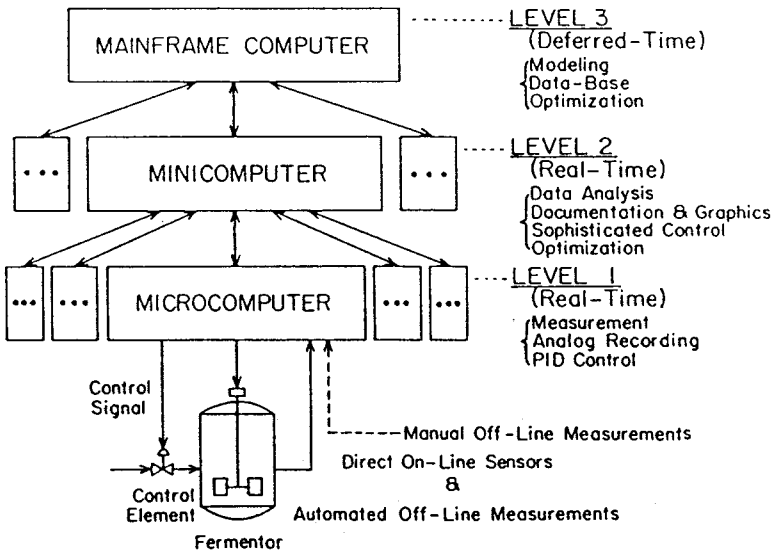


FIGURE 2. Tree root structure of distributed hierarchical computer systems used for a large installation.

at the third level on an off-line basis. Optimization based on the models developed at the third level are also conducted to generate the optimal operating procedures. In general, a higher level holds a broader perspective over the whole fermentation process and requires a deeper knowledge of the microbial behavior.[56] Although an on-line updating of model parameters and an automatic adjustment in the optimal control strategy are desirable, such a scheme has not yet been implemented.

Many variations to the aforementioned scheme are possible. Lundel divided the whole plant organization into four levels of functions and distributed computers to each level according to functions.[50] The computer may be dedicated to only one task such as PID control. The usual PID control chore may be substituted by conventional analog controllers with the set point supplied by a computer, in which case the direct digital control (DDC) degenerates into digital set point control (DSC). The advantages and disadvantages of DDC and DSC have been pointed out by various investigators.[23,48,56,57] Alternatively, all the tasks of each level may be carried out in one computer in smaller installations; however, some software provisions for separation of high priority foreground and low priority background calculations are needed[40] if data handling, modeling, control, and optimization are all to be implemented simultaneously. To the authors' knowledge, no such feat has yet been accomplished, whether the system is large or small. For a less ambitious project, the stand-alone configuration will suffice. In fact, one-computer systems are very popular, and they are often the only means for small, economically deprived research groups to carry out fermentation studies in the state-of-the-art manner.[58] Since metabolic responses and delays are generally much slower than the responses in a typical chemical industry, computation speed is not as problematic, as long as the high priority tasks such as monitoring and control are not severely backed up. Otherwise, the scanning frequency must be decreased accordingly. While 5- to 15-min scanning intervals may be satisfactory for parameters with slow changes and highly accurate measurements, 1-sec intervals are customarily needed for those parameters that have high noise levels and require averaging to achieve the necessary accuracy.[59] In such cases, the computing power of a small microcomputer may be inadequate if many additional functions are to be carried out.

Nevertheless, the reduction in hardware cost and the advances in computer technology make it increasingly more feasible to acquire one or more computers for each functional

level, depending upon the needs of each individual installation. Design requirements are highly dependent on each specific application — those for an antibiotic production are certainly different from those required for single cell protein (SCP) production, and those for academic research are undoubtedly different from those required in a commercial environment.[60] Some of these requirements are discussed by Hampel[59] and Hennigan et al.,[54] but reliability and flexibility are the two most fundamental requirements in all cases.[57] In conclusion, the readers should be cautioned that "Distributed systems by whatever definition are nothing if not versatile. But anything which promises something for everyone is suspect."[3]

2. Interface

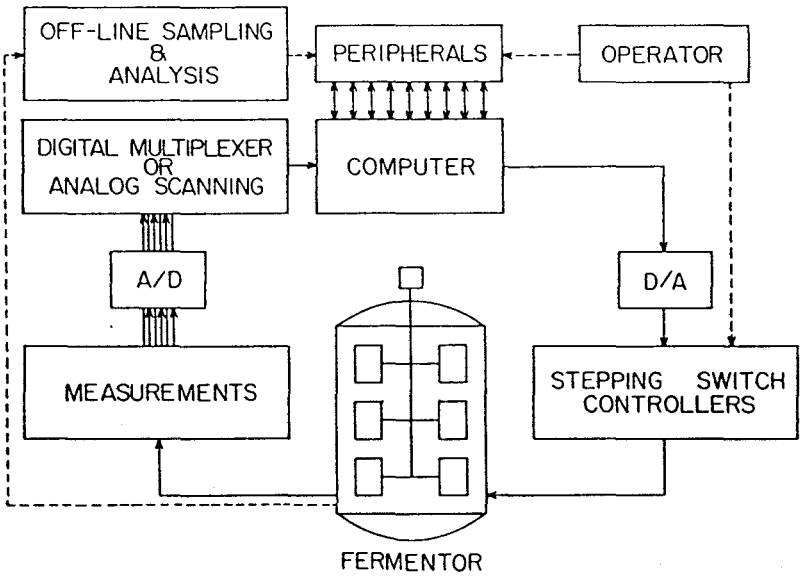
A typical computer-coupled monitoring/control configuration for a highly instrumented fermentor is depicted in Figure 3. The computer is totally powerless unless it is properly connected to communicate with a variety of peripheral devices. The connection between the real physical world and the abstraction of a computer is through interfacing, and an adequate method must be employed to make the computer work as the user desires. There are commonly five kinds of interfacing: sensors to computer for data logging, computer to computer for communication between computers, computer to actuator for the actual control phase, operator to computer for on-line interaction, and computer to operator for documentation. Since the interfacing techniques are not peculiar to the fermentation field and since they can be found from a variety of sources,[48,51,54,61,62] they will not be discussed in depth in this paper.

The readers should be cautioned that a universally accepted industry standard for computers and their accessories does not exist at the present time. This is because computers, especially the affordable microcomputers commonly used in the fermentation field, are still undergoing drastic evolutionary changes and many new products are constantly emerging. We cannot simply plug one component into another and expect them to function in the same way as plugging household appliances into electrical outlets. Thus, to minimize the difficulties arising from interfacing, we should plan carefully and make sure that the modules are compatible before the purchase. The signals from most of the conventional instrumentation devices often need to be amplified/reduced and conditioned before they can be accepted by the existing interfacing units. Similar procedures also apply to the signals sent out to the conventional pneumatically or electronically activated control devices, which usually consist of valves and pumps in the fermentation field.

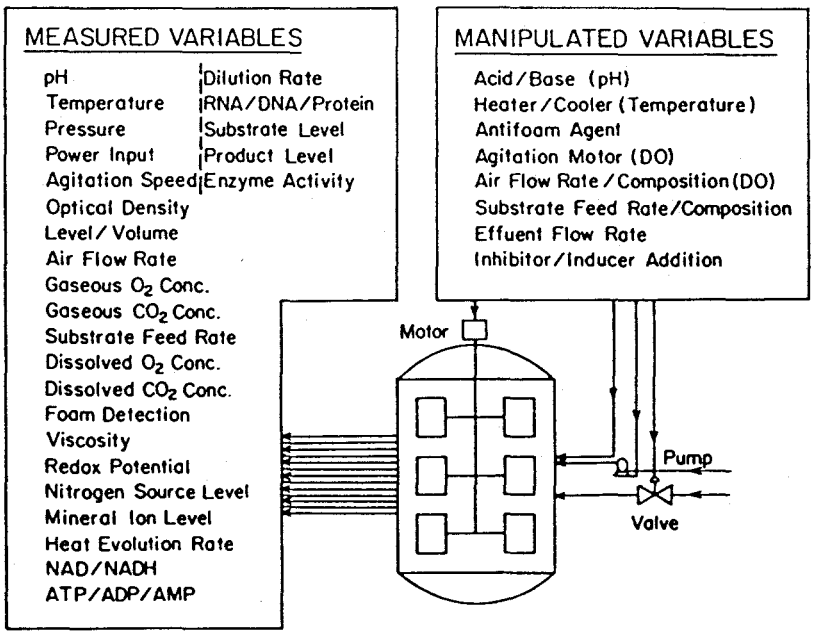
Analog to digital (A/D) and digital to analog (D/A) converters are routinely used in interfacing in the fermentation field as in any other field. Interface can be either serial or parallel; the choices are largely dictated by the physical nature of the devices to be connected. The choices between analog scanning or digital multiplexing are also available; while the former is more economical when the number of channels is large (>25), the latter offers the opportunity of simultaneous logging and channel independence.[7] Furthermore, instrument scanning can be randomly accessed or prefixed.[7,56] Randomly accessed scanning is useful when the dynamic time constants for various measurements differ significantly. In such cases, fast-changing and less reliable measurements are accessed more frequently than the stable ones;[56] however, a price must be paid for this flexibility in terms of a higher hardware cost.

B. Software

The general software structure in a typical computer system, shown in Figure 4, is centered on the operating system which performs most of the drudgery involved in a computer. Among the most elementary but also the most tedious functions carried out by the operating system are the inventory of memory space, management of files, control of program execution, and coordination of all the operations. There are basically three types of programs:



A



B

FIGURE 3. (A) Typical computer-coupled monitoring/control configuration for a fermentation system; (B) highly instrumented and controlled fermentor.

utility, language, and application. Utility programs start up the computer and create new files; language programs allow the use of high level languages; and, finally, application programs accomplish useful tasks. The first two types of programs are widely available from various software vendors to run fermentation application programs. Although such topics as the choice of computer languages have frequently concerned fermentation investigators in the past, they are not peculiar to fermentation applications and will not be discussed here.

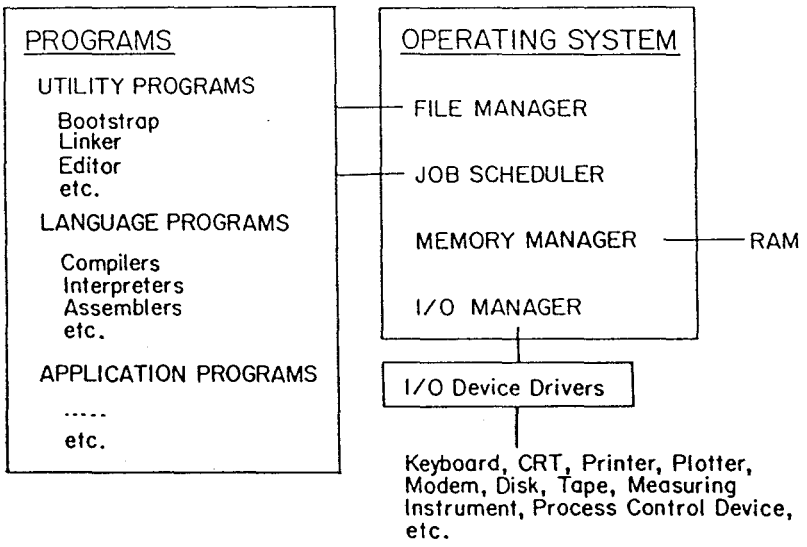


FIGURE 4. Schematic of the software structure of a computer.

Among software, application programs are of the most interest to us. Of course, computers do no more and no less than what they are instructed to do. Thus, many application programs are necessary to carry out a variety of tasks needed to support a sophisticated fermentation control scheme. An organized structural arrangement of the files is essential at every level of computer software to insure satisfactory performance in a computer-coupled fermentation system.[31,63,64] Programs should be divided into manageable, small modules so that each procedure can be developed separately. As the need arises, each procedure should also possess the flexibility to be altered smoothly by any qualified person, especially those who did not write the original program. When the goal of program flexibility is achieved, a large part of the reliability problem is automatically solved. In addition, priority levels should be correctly assigned to each program to insure efficient allocation of computer resources.

A swarming number of articles have indulged in the detailed description of the function and algorithm of various application programs.[29-31,33,34,36,41,42,50,56,58] In the data logging phase, programs are required for scanning measurements sequentially or in any desired pattern, filtering out noises, converting electrical signals to physical units, and storing or retrieving data in a condensed form. Other application programs may also be needed to supply documentation in the desired format, whether alphanumeric or graphical. Although an on-line monitoring of all the variables of interest is ideal, due to many economical and technical reasons this cannot always be achieved. In such cases, the programs should be designed to remind operators when to take a sample and to allow them to enter the data from off-line measurements, which must be merged with the ones obtained on-line.

In the data analysis phase, programs are needed to calculate various derived variables, including indirect measurement quantities, from the conditioned signals obtained in the previous phase. This includes conversion of units, scaling, comparison with a calibration curve, and perhaps the use of correction factors. A set of simple algebraic equations or a collection of routines for integrating complex differential equations may be called upon to assign numerical values to a set of variables. Because of the lack of redundancy at the level of sensors and actuators to check for and detect the source of inoperative valves and instruments, programs may be built in to perform self-analysis and detect errors. For this purpose, a periodical fault analysis can be employed to detect the failure of valves.[65] Ad-

ditional independent measurements can be introduced to detect the instrument failure early enough to permit necessary actions to be taken so as to insure reliable process control.[66]

In the process control phase, software programs are written to compare the measured values to the set point and to generate controller outputs. In a sophisticated control scheme, the optimal operating condition and the dynamic path may be calculated to generate set points from within. Programs are needed to sequence a wide variety of plant operations, including the initiation and termination of fermentation processes. Other functions that need to be considered are the automatic calibration of instruments, sounding of alarms, handling of emergency situations such as power failure and fatal contamination, communication in-between programs and computers, and interaction with operators to reset process variables and to redefine control objectives.

The sequence of software development generally follows the logging-analysis-control order. The task of developing a complete set of software is both time consuming and costly. While many functions are commonly required by all applications, many others are not absolutely necessary, and the requirements for each individual application differ significantly. In a production plant, for example, the economic reward is contributed mainly from the sequencing and scheduling of a series of operations.[60] However, for research scientists, whose purpose is to study the biological response in a well-defined environment in the hope that a cause-effect relationship can be identified and the observed phenomenon explained, the use of computer is for mathematical modeling and data handling. Finally, it is important not to consider software and hardware exclusive of each other.

III. INSTRUMENTATION

Sensors in the fermentation field are generally less reliable than their counterparts in the chemical industry. As always, there continues to be an urgent need for more and better instrumentation in a fermentor. Instrumentation is one area where the needs of fermentation technology are far greater than the needs of the conventional chemical industry. Another area that differs most significantly from its counterpart in the chemical industry, the modeling of microbial behavior in a fermentor, will be discussed in a later section. Unlike the relatively well-defined conditions encountered in most chemical industries, gas, liquid, and solid phases are all involved in a typical microbial environment, and this physical heterogeneity alone can already make sampling and measurement very difficult.[5] Moreover, the dynamics of living organisms has a complexity unparalleled in the chemical industry, where the number of reactions involved are relatively few and the nature of reactions are rather uncomplicated and better understood.

Because of this, the sensor requirements for the study of microorganisms are much more demanding. Some practical considerations in the development of a sensor include the sensor's potential for continuous or fully automated measurements and the capability of yielding appropriate electrical signals for easy interfacing to a computer. Rapid response is a desirable characteristic in a sensor from the point of view of both process modeling and control implementation. A sensor's sterilization capability is critical in achieving an aseptic fermentor operation, which is especially important for a continuous bioreactor. Finally, cost is always the most crucial factor in the final decision of employing or rejecting a particular sensor because the additional benefit that can be gained from the use of the extra piece of equipment should surpass the expenditure involved.

Instruments that either are capable of continuous on-line measurement or possess the possibility of computer-coupled automatic sampling are the main focus of the following discussion. Although certain methods that require batchwise manual analysis may sometimes be the only available alternatives, they are not well suited for computer interfacing. A fully automated scheme for the monitoring of a fermentor cannot be over emphasized if the true

potential of the use of computers in fermentation technology is to be realized. Continuous measurements have many advantages over infrequent, intermittent samplings. Although, understandably, there may be acute storage problems if the large volume of data generated by the continuous measurement is not properly reduced prior to the storage for later analytical or modeling purposes, a continuous on-line measurement is more likely to satisfy more of the sensor requirements listed previously.

First, there is the advantage in speed. Until recently, the state variables of a fermentor were estimated in batchwise analyses with sampling time no shorter than one half hour. These analyses were usually done manually and took an unreasonable amount of time to complete. For example, the determination of biomass concentration alone could easily take anywhere from one half hour for simple cell counts or spectrophotometric measurement, up to 1 day for dry weight measurement. A continuous measurement not only is convenient and requires less man-hours, it also enables a speedy data collection and analysis. This speed is needed if a problem is to be detected early enough to take whatever control actions necessary in order to bring the system back to the desired state. Thus, unless on-line measurements are used, there will be an unacceptable amount of delay and manpower involved, and the control will be rendered worthless.

Also, with on-line measurements, we can observe events that are occurring in a time interval shorter than the previous batchwise sampling interval. A general analogy that may demonstrate the usefulness of this fact is the use of a microscope,[5] which allows us to see smaller objects than previously possible. This capability is undeniably an important asset under many circumstances in which the interested events last for only a short period of time. For example, the formation of a secondary metabolite during a certain phase of microbial growth cannot be satisfactorily monitored when we use the batchwise sampling technique.

Furthermore, the repeated outside access to the fermentor during the batchwise sampling greatly increases the chance of contamination. The problem of contamination is especially serious in a continuous mode of operation, and it is frequently cited as the reason for the termination of the entire operation. When a continuous operation is carried out in a small-scale laboratory fermentor, the batch sampling can easily disrupt the whole system due to high sensitivities of the microorganisms. The data collected this way and the subsequent analysis and modeling of the system, thus, may be quite inaccurate.

Despite the intensive effort spent in developing new biological sensors in recent years, the state of available sensors is not much different from, say, a decade ago. Currently, not many new sensors are reliable enough to be commercially available in a form that can be interfaced directly with a fermentor. The sensor deficiency is notably serious in the measurement of broth composition such as nitrogen source, substrate, precursor, intermediate products, and final product concentrations. Sensors involving enzymes or antibodies are capable of these specific measurements. As a result of the substantial effort expended on the development of these sensors in recent years, they hold great future promises. The commercialization of glucose analyzers by various instrument companies in recent years is especially encouraging.[24] Because of the possibility of faster response time and the capability of measuring a wide range of components in the gas stream and fermentation broth, there is a noticeable trend in employing mass spectroscopy in fermentation analysis. Highly reliable sensors for the biomass concentration measurement are still absent. In addition, there is a serious lack of on-line sensors to measure intracellular macromolecular composition such as DNA, RNA, and protein (including enzyme) levels in microorganisms. Likewise, the elemental biomass composition (i.e., percentages of C, H, O, N, etc.) cannot yet be done on an on-line basis.

Unfortunately, because each microbial process is inherently different and so are the objectives of each individual who uses such a process, the vitally important question of what to measure cannot be answered universally here. Since knowing what to measure presumes

a certain amount of knowledge of the process, it is already a quite significant step toward the final understanding of the process. Whether one's purpose is to formulate a process control scheme or merely to understand the microbial system, he must choose a set of measurements which, when used alone or in combination, directly or indirectly, will yield an adequate picture of the process.

Fermentation sensors are loosely divided into four categories in the following discussion: physical, chemical (extracellular chemical compositions), biochemical (intracellular compositions), and biological (cell genetics). Other methods of classification are also available.[52,67-70] It is hoped that the list of variables in Table 1 will provide readers a partial menu from which to choose and decide the "what" part of the question. We will, however, discuss the methods of measurement available on some of the most typical state and control variables, i.e., answer part of the "how" question. Inevitably, some sensors are more useful than others, and a hierarchy in the sensor development activity is needed for the optimal allocation of resources. By examining the information that can be gained from each of the measurements and by considering the potential utilization of this information in modeling and control, we can assess the relative importance of these sensors and assign priorities accordingly. Since most of the devices used for the physical measurements are the same as those basic units used in the chemical industry, emphasis will be placed on those measurements specific to the fermentation industry.

Finally, a thoroughly compiled and well-categorized list of commercial suppliers of various instruments is published each year along with their detailed addresses by the American Association for the Advancement of Sciences.[71] The 1982 issue devoted a section to biotechnology-oriented instruments. Tannen and Nyiri also have a short list of equipment manufacturers.[68] The readers are referred to the original papers for the more detailed description of the construction of those commercially unavailable instruments.

A few general reviews on fermentation instrumentation have been published in the past.[52,68,72-76] The work by Solomons is especially detailed in the coverage of the physical measurements.[75] Fleischaker et al.,[77,78] have considered the requirements for the computer-coupled instrumentation for mammalian cell cultures, which are slightly different from those for microbial cultures.

A. Physical Measurements

Almost all the measurements in this category can be measured on-line. Although certain sensors are preferred in biochemical engineering studies, the equipment is not much different from that used in conventional chemical industry.

1. Temperature

Thermocouples, thermistors (semiconductor), and metal resistance thermometers (platinum) are all capable of producing an on-line electrical signal. Thermistors are most widely used in fermentation because of the high sensitivity and the absence of the cold junction requirement. It has the disadvantage that the output is highly nonlinear; some "programmers" are sold to overcome this problem. High accuracies (down to 0.001°C) over the range of 25 through 45°C are possible. For other uses of temperature measurements, the reader is referred to the section on calorimeters. Temperature control by the use of external or internal heat exchangers are commonly used; it may also be affected by cooling the nutrient feed when the generation of heat by microbial activity is not too large.[79]

2. Vessel Pressure

Diaphragm gauges are used with transducers to produce an electronic signal. They are highly linear (≈0.25%) and thermally stable, and a wide selection of operation ranges are available. The principle of operation is described elsewhere.[68]

Table 1
FERMENTATION MEASUREMENTS

Physical
Temperature
Pressure
Agitation
Shaft power input
Heat transfer rate
Heat production rate
Gas flow rate
Liquid flow rate
Foaming
Fermentation broth volume
Weight
Viscosity
Density
Volume
Level
Chemical (extracellular)
pH
Redox potential
Ionic strength
Gaseous oxygen concentration
Gaseous carbon dioxide concentration
Dissolved oxygen concentration
Dissolved carbon dioxide concentration
Carbon source concentration
Nitrogen source concentration
Metabolic product concentration
Mineral ion level
Precurser level
Nutrient composition
Broth composition
Biochemical (intracellular molecular)
Cell biomass composition
NAD/NADH level
ATP/ADP/AMP level
Specific protein (enzyme activity)
DNA/RNA level
Total protein level
Carbohydrate level
Intermediates concentration
Biological (cellular genetics)
Morphology (shape)
Size/size distribution
Age/age distribution
Contamination
Mutation

3. Volume/Level

The fermentation broth volume is important in terms of the amount of inoculum introduced. It also serves as the basic unit of a fermentor. For example, feed rate and residence time is more meaningful on a per volume basis for both fed-batch and continuous cultures. Of course, various concentration units have always had volume as the basis. The level, hence, the volume (knowing the fermentor geometry), of the fermentor can be relatively simply measured by a pressure transducer mounted at the bottom of the fermentor. A capacitance level probe or a float-photosensor combination is also sometimes used.[80] Theoretically, the accuracy of these traditional measurements is high, but in actuality it is not at all so. This

is because the liquid is far from being tranquil due to the turbulence caused by agitation, the presence of swarming air bubbles from aeration, and foaming. Unfortunately, this basic unit is overrun by an error of approximately 5%, and in view of the liquid conditions in a fermentor no improvement is likely in the future.

4. Weight

This measurement by an electronically coupled strain gauge needs no further elaboration. However, tubings and connections can make the isolation of a vessel (e.g., reactor or medium holding tank) relatively difficult. Thus, the attainable accuracy is routinely below that of the scale's potential.

5. Density

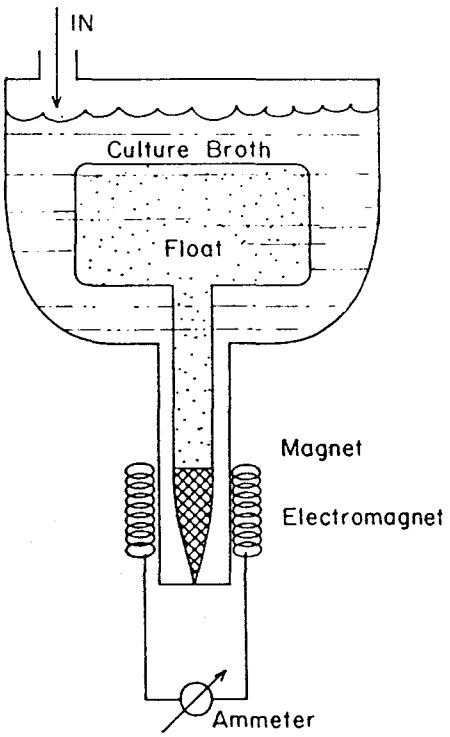
For an anaerobic batch operation, a change in the volume means a change in the density. However, for other modes of operation, the density of the fermentation broth does not equate as simply as above. Although a density measurement is rarely made in fermentation practice, highly accurate continuous flow methods are available for this measurement in the biochemical field.[81] One method uses a float of known volume suspended in the fluid. A magnet is attached to the float and is pulled by an electromagnet. The current needed for the electromagnet to immerse the float totally in the fluid is correlated to the density (see Figure 5). Another method uses the principle of resonance of a tube in wave mechanics. The fluid flows inside the U-shaped capillary tubing whose ends are anchored to a large mass. A magnetic coil attached to the U-tube drives the tube into resonant vibration. The same magnetic coil is also used to detect the frequency of the resonant vibration which depends on the total mass of the tube: the bare tubing and its fluid content.

Since density measurements are seldom conducted by biotechnologists, the potential gains and the potential problems of using such methods, such as the complications arising from the presence of solids, are not yet totally clear at the present. However, some ways of utilizing this measurement can be speculated. The specific gravity of cells are quite close to unity, and, as a result, the settling of microbial cells is relatively slow. An accurate density measurement can aid in predicting the rate of cell sedimentation. If the cell density is reasonably different from that of the fermentation broth, it should also be possible to relate the density measurement to the cell biomass concentration. Studies in this area are recommended.

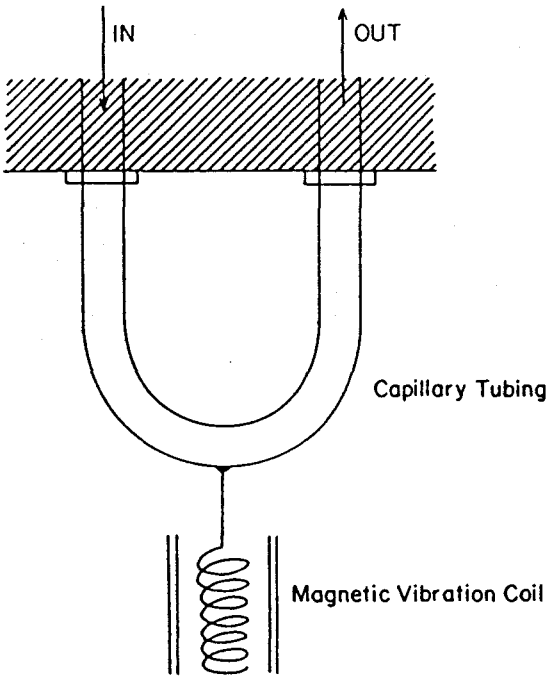
6. Power Input

Because the mass transfer characteristics between gas and liquid phases and, subsequently, the concentrations of oxygen, carbon dioxide, and volatile components in the fermentation broth are intimately affected by the power input to the system, the measurement of this variable is important in the control of microbial respiration. The power input not only serves as a means for the dispersion of gas in the liquid but also ensures the homogeneous mixing of all the components in the fermentation broth.[82] It is also considered in the calculation of power consumption in optimization and in the calculation of the rate of heat removal in temperature control. Furthermore, the power input through the shaft must be known accurately to estimate the rate of heat production by microorganisms in certain instances where such calculations are made based on the energy balance of the whole fermentor.

There are two types of power input instruments in common use today: a torsion dynamometer and a strain-gauge dynamometer. Usually, a torsion dynamometer based on the principle of the Hall effect is attached to the armature of the drive motor located outside the fermentor. However, the actual power input to the fermentation broth through the impeller is less than that indicated by the dynamometer because of the friction losses in bearings between the motor and the impeller. It can be as high as 30% of the total power input into



A



B

FIGURE 5. (A) Continuous density measurement by a float; (B) vibrating tube flow densimeter.

the motor and should not be neglected outright.[74] On the other hand, a strain-gauge type of dynamometer is attached to the shaft of the impeller inside a fermentor; thus, the actual power input through agitation is measured. This type of dynamometer depends on a change in the resistance of the material under longitudinal or transversal stress. Aiba et al. had previously described the principles of operation in detail.[69]

7. Agitation Speed

Like shaft power input measurement, agitation speed also affects dissolved gaseous concentrations. In addition, agitation by the rotating impellers creates a shear force that may upset the microbial cells. Although rpm figures are customarily reported in the literature, one should keep in mind that larger fermentors inevitably have larger impeller diameters; the speed of the impeller tip is proportional to both the rotation speed and the impeller diameter. Tachometers are used for the monitoring of shaft rotation speed.[68] A gear can be mounted on the device shaft outside the fermentor and either a magnetic sensor or an optical sensor can be used to measure the rate at which the magnetic field or light source is interrupted by the rotating teeth. The signals are passed through appropriate transducers to give a continuous electric voltage output. Gears or belts may also be mechanically connected to a tachometer to measure the rotation speed, but the maximum speed that can be accommodated by these mechanical devices is rather limited.

8. Gas Flow Rate

Three kinds of devices for on-line monitoring of gas flow rates are popular among fermentation technologists. The conventional variable-area meter such as a rotameter have limited accuracy but are relatively cheap. The float position in the tapered tube can be readily converted into an electrical signal suitable for computer interface. The second device exploits the thermal capacity/conductivity of the gas. The gas is heated by a coil of wire as it passes by. The change in the temperature of the gas registered by two thermistors located upstream and downstream of the heated wire is used to correlate the mass flow rate (see Figure 6). This flow meter is generally marketed with a flow control device as one unit. A gas flow meter of the third type forces the gas flow profile to be laminar through the insertion of a matrix device in the line. The pressure drop is continuously measured and correlated to the flow rate.[68]

9. Liquid Media Feed Rate

Since no significant changes have appeared in the recent years, the readers are referred to Solomons.[75] Turbine, rotameter, and electromagnetic meters are available. However, one should be cautioned to the fact that manufacturers' quoted accuracy of the instruments applies only under an idealized environment. Since rheological properties of the fermentation broth and media affect the performance of these flow meters, the quoted level of precision can seldom be reached. There are three main types of liquid pumps: propeller, diaphragm, and peristaltic. The diaphragm type of pump gives constant volume dosage each time the diaphragm pulses and is capable of high flow rate, but there may be sterilization problems. Peristaltic pumps use steam sterilizable tubes that are squeezed externally, but the maximum flow rate obtainable is low. Frequently, the meter and the pump are not well distinguished, and the combined units are commercially available for each of the three types. Alternatively, a liquid holding tank may be placed on a load cell that gives a continuous reading on the weight of the vessel. The inference of the instantaneous rate of addition of nutrients, base/acid, or antifoam from the weight measurement is inherently erroneous, and a noise filtering technique is required to obtain reliable information on flow rate. Nevertheless, it is a superb measurement on the cumulative quantity.

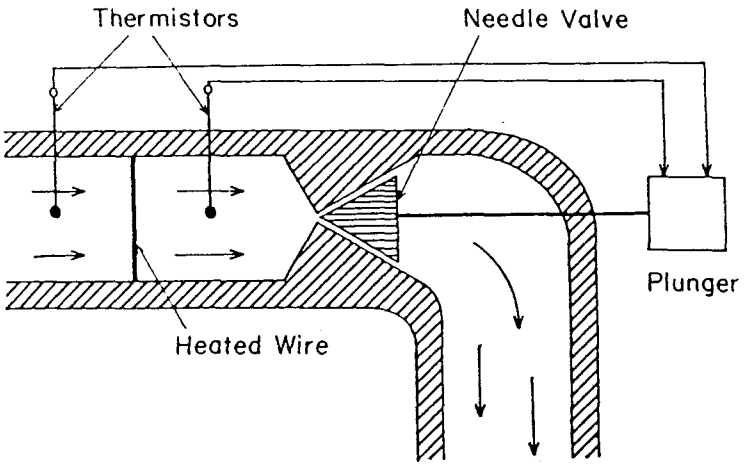


FIGURE 6. Thermal gas flow meter/controller.

10. Viscosity

Due to the great variability of fermentation broths, their viscosity can differ significantly from system to system, its magnitude spanning several orders.[82] Viscosity also varies over the course of a fermentation process as the composition and temperature inside a fermentor change. Understandably, the potential for variation is greater for a batch process than for a steady-state continuous operation. It is further known that the viscosity of the fermentation broth affects the response and accuracy of other sensors.[83] It affects the mixing characteristics of the system, and, subsequently, the mass transfer, heat transfer, and power consumption depend on it as well. Despite this, no extensive systematic effort has been spent either on the sensor development or on the characterization of the effect of viscosity on fermentation variables or the utilization of viscosity as a means of on-line bioreactor monitoring.

Two types of devices are available: capillary and rotational viscometers. For a Newtonian fluid, the viscosity is proportional to the pressure drop across a capillary tube for a constant volumetric flow rate. Since a capillary viscometer requires the measurements of both the volumetric flow rate and the pressure drop,[84] this type is not usually used for on-line monitoring. Furthermore, the calculation of velocity profiles inside the capillary is complicated if the fluid is non-Newtonian. Thus, the rotational type of viscometer is more popular in fermentation applications.[85] Depending on the geometry, many variations of this type are possible: coaxial cylinder (Couvette), Brookfield, cone and plate, and turbine. All are described in detail by Charles.[83] Alternatively, the whole fermentor can be used as a viscometer by combining the measurements on shaft power input (P) and the impeller rotation speed (n) and by calculating the viscosity (v) according to the correlation $P \propto n^2 v$.[78] However, due to the difficulties in maintaining a laminar flow, the simpler correlations derived from laminar flow conditions are generally not applicable to this *in situ* measurement method. In addition, an *in situ* viscosity measurement is plagued with many problems such as the presence of air bubbles and solids in the broth as well.[86]

It has long been known that the presence of microorganisms, especially fungi and *Actinomycetes*, changes the viscosity of the broth.[69,87] Thus, it is theoretically possible to use viscosity as an indicator of cell concentration or cell morphology.[70,78,89] In actuality, this has not been successfully done. For example, Perley et al.[84] attempted to measure yeast concentration by this method with only limited success. Due to changes in morphology and various other factors affecting the viscosity, the correlation between cell concentration and

viscosity was poor except at high cell concentrations (>20 g/ℓ). Roels et al.[87] defined a "morphological factor" which took into consideration the length to diameter ratio of the mycelial filaments. However, as pointed out by Blanch and Bhavaraju,[88] the measurable parameter was the "modified morphological factor", which was not directly related to the length/diameter ratio; actually, it related the viscosity to the mycelial concentration. Shimmons et al.[89] have conducted a preliminary feasibility study to demonstrate the possibility of using viscosity as a biomass sensor to control the yeast concentration during the blending of concentrated baker's yeast with water. The control was successfully accomplished, but the use of viscosity measurement has not been carried out in a more realistic fermentation system. The presence of some fermentation products also changes fluid viscosity. LeDuy et al.[90] and Thomson and Ollis[91] have studied the changes in the Newtonian behavior of the fermentation broth as polysaccharide is produced.

The technical aspects of the rheological properties of microbial cultures have been extensively and critically evaluated by Charles.[83] Blanch and Bhavaraju have also reviewed this aspect[88] and suggested the control of viscosity by dilution with water or by addition of nutrient shots; however, the side effects resulting from such control strategies were not considered. The viscosity was controlled at a constant value by the manipulation of nutrient flow rate in a continuous microbial polysaccharide fermentor,[92] but it was not clear what was to be gained from such a control. In future investigations, we need to identify both the variables that affect the viscosity and those that are affected by it. Cell concentrations, morphology, and product concentrations are some possible candidates for correlation.

11. Foaming

Since the state of foaming is not physically well defined, the measurement as such is not generally made. Nevertheless, the detection and control of foam are crucial in assuring a smooth fermentor operation. The problem may be exceptionally severe when growth media have a high protein content.[93] The presence of foam can be detected by conductance/impedance measurements, and it can be controlled with a multistage foam control system.[94] Production scale fermentors usually have some built-in mechanical provisions to aid in the breaking up of the foam. (Actually, the foam is compressed.) Antifoam agents, either silicone based or hydrocarbon based, may be added alone or in conjunction with the use of mechanical foam breakers. The advantages and the disadvantages of different antifoam agents have been discussed by Wang et al.[70] The cost of antifoam agents may become intolerable if the size of the fermentor is large. Antifoam agents are not easily sterilized and this can sometimes pose dangers of contamination. Furthermore, some liquid phase sensors such as the dissolved oxygen electrode, as well as the gas-liquid mass transfer characteristics, are severely affected if the concentration of antifoam exceeds 2000 ppm.[93]

Soifer et al.[95] formed a foam index that takes into consideration the volume, ease of formation, and stability of the foam layer. They studied the effect of antibiotic level in the broth and the effect of nutrient composition on foaming and found the metabolites of the microorganisms to be critical in the formation of foam. Recently, attempts have been made to exploit the protein-concentrating property of foams to achieve a highly effective separation of enzymes or antibiotics from the fermentation broth.[96,97] Thus, the ability to detect and control foam may be invaluable in a combined operation in which the continuous separation and recovery of antibiotics from the fermentation broth through foam removal not only solve the foaming problems but also help repress product inhibition and stimulate further product formation; at the same time, the removal of foam enables better aeration quality.

B. Chemical Measurements

Measurements that reflect the extracellular chemical environment fall in this category. By performing a simple component balance, we can infer the rate of production or utilization

of certain compounds such as substrate or products through concentration and flow rate measurements (see Figure 7). At present, the detection and the measurement of the concentrations of various compounds in the aqueous phase are still very much done on an off-line basis. The methods of enzymatic analysis of various fermentation substrates and products, including a wide range of organic acids and nucleic acid derivatives, are frequently used and are very valuable in monitoring the catabolic, anabolic, and energy metabolisms in microbial cells.[98] As mentioned before, off-line chemical methods are relatively slow and cumbersome, and the permissible number of analyses is rather limited. Thus, continuous on-line instruments, especially of the probe type, are preferred.

In the recent years, suppliers of laboratory analytical instruments have been spending substantial effort in utilizing microprocessors to automate the sampling process. A Technicon® auto analyzer is one such example. In the past, sampling, gas chromatography, mass spectrometer, and a variety of other instruments have also been automated by microprocessors; however, automation is inevitably accompanied by a quantum jump in the price. Automated analyzers capable of giving out electrical signals to a computer can also contribute significantly to the modeling and control of a fermentation process, not to mention the saving in labor. To be sure, automated analyzers still perform analyses on a batchwise basis, and the disruption caused by a high rate of continuous sample withdrawal that is required for a large number of analyses is sometimes intolerable, especially when the fermentor is small.

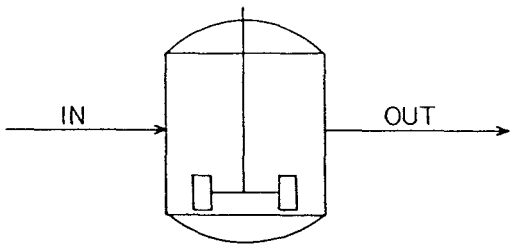
Redox potential, which is related to the ionic strength of the fermentation broth, is also included here. Unlike the instruments for physical conditions, the instruments in this section are peculiar to fermentation technology. Even a conventional pH probe needs to be totally redesigned to allow sterilization. Since the methods for monitoring the substrate and the product will be determined by the nature of each process, they will not be listed for each individual compound. Rather, the general measurement concepts, such as enzyme electrode and automated gas chromatography, will be discussed.

1. pH

pH is one of the most widely monitored variables in fermentation. Steam sterilizable and autoclavable pH probes are commercially available in the form of a combined unit for the H^+ half-cell and reference half-cell. Cheaper separate units of glass and reference electrodes are also available. Principles of operation can be found in Aiba et al.,[69] and the construction of a glass pH electrode is detailed by Eisenman.[99]

2. Biomass

The central element of a fermentation is, of course, the microbial cells. Therefore, biomass concentration is probably the most important parameter in various stages of the study of all fermentation processes, ranging from modeling to control, from inoculation to termination, and from single cell protein production to antibiotic production. Since the biomass concentration affects both the substrate consumption rate and the product formation rate, it should be closely and continuously monitored in almost all process control situations. Despite this fact, its on-line determination has not yet been accomplished in a totally satisfactory manner. Many methods have been tried in the past, and they can be roughly divided into the following categories: optical, chemical, physical, thermal, mechanical, and manual, depending on the principles behind the measurement. Needless to mention, each method actually measures different properties related to the biomass; some methods may be better suited for certain purposes on certain species under certain conditions than others. Optical methods include light absorbance-transmittance and scattering. Chemical methods include ATP,[100-104] NADH,[105-110] nucleic acid, and proteins.[52,111] Physical methods include viscosity,[70,78,83,84,89] broth density, and ionic strength.[78,112] The description of chemical, physical, and thermal methods can be found in other sections in this paper and will not be repeated.



ACCUMULATION RATE = (RATE IN – RATE OUT) – UPTAKE RATE

$$\frac{d(CV)}{dt} = R_jV + \Phi_j$$

C_j = concentration of component j in the system

R_j = volumetric rate of production of component j

Φ_j = net rate of input of component j

FIGURE 7. Macroscopic component balance around a fermentor.

There are three optical methods,[113] depending on the angle between the incident light and the position of the detector: light transmission (spectrophotometer), side-scattering (nephelometry), and retroreflective scattering (split-fiber optics).

Of the three methods, the spectrophotometric method that measures the transmittance/absorbance of the fermentation broth is the most popular. Quite a few instrument companies market spectrophotometers that can accommodate a flow-through cell to generate continuous signals. Miniaturized probes that can be directly inserted into a fermentor have also been developed.[113,114] The one developed by Ohashi et al.[114] is steam sterilizable and even has built-in foam and bubble elimination mechanisms. Since most of the scattered light is toward the direction of the incident light, a bifurcated fiber-optics light pipe can be used to detect the back-scattered light[115] (see Figure 8).

Because the relationship between the optical density and the cell concentration becomes increasingly nonlinear beyond an optical density of 0.5 (or a biomass concentration of approximately 0.5 g/ℓ) all of the above devices are limited to a low range of microbial cell density unless linearity is maintained either by dilution or by the use of a shorter light path.[116]

The deposition of microorganisms on the flow cell walls has always been a source of signal drift; it needs to be removed before the build-up becomes significant. The design by Hancher et al.[115] provides a periodic cleaning of the flow cell by flushing with a high-intensity jet of water. The changes in the size, shape, and opacity of the microbial cells make optical correlations vary from one run to another,[78] and it has been reported that even during the same run the optical property depends on the specific growth rate.[117]

Counting and sizing can be rapidly performed with a Coulter® counter.[118] A schematic of a Coulter® counter is shown in Figure 9. A prefixed volume of electrolyte solution in which cells are suspended is drawn through a minute opening. As a cell passes through the opening, it displaces the electrolyte solution with its own nonconducting body. This increases the electrical resistance of the fluid within the aperture, and a voltage pulse is sent to the signal analyzer. Since the voltage pulse is proportional to the cell volume, a size distribution can also be obtained with a multichannel analyzer. Although a computer can be easily interfaced to store and process the large amount of data generated, automatic sampling and operation may prove to be difficult. First, the fermentation culture has to be diluted with an electrolyte solution; secondly, the obstruction of the small aperture, which leads to totally

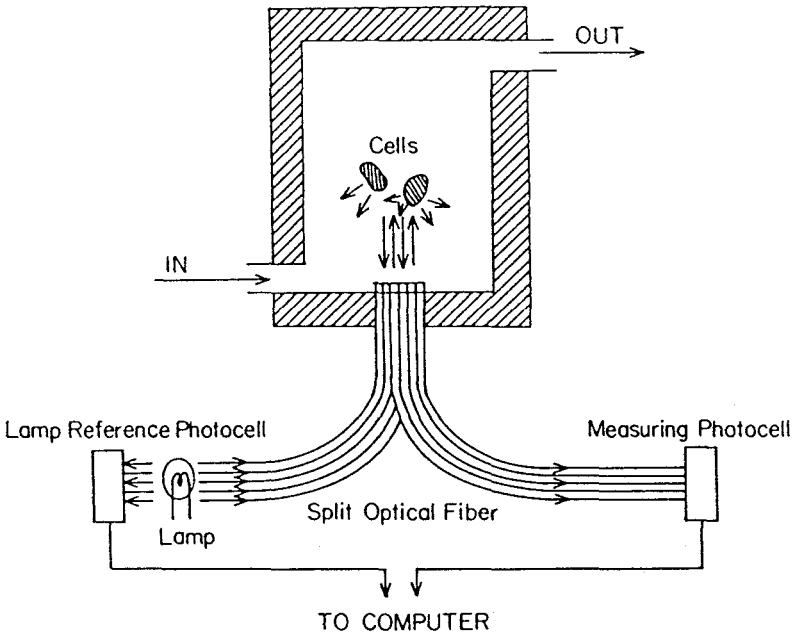


FIGURE 8. Schematic of a retroreflective turbidometer.

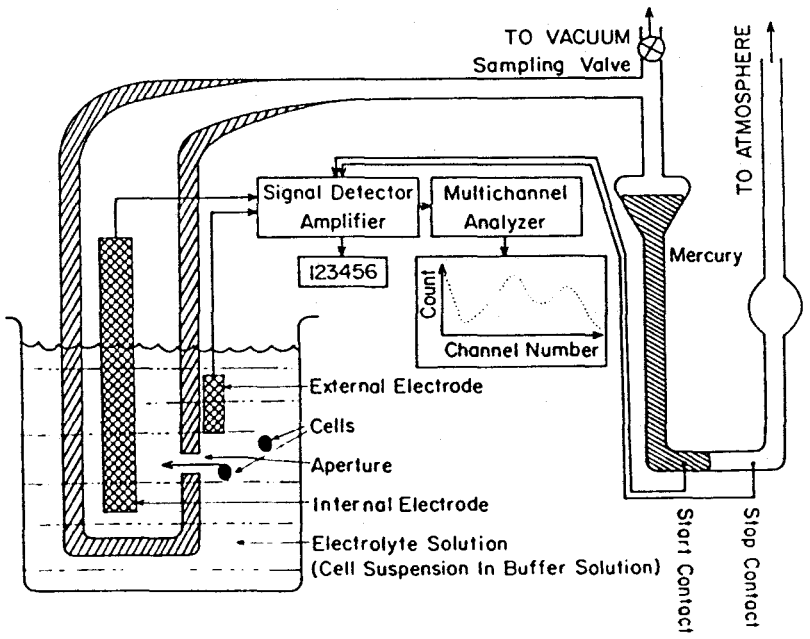


FIGURE 9. Schematic of a Coulter counter.

erroneous readings, can occur very frequently even when there are no large particles in the solution. Nevertheless, its capability of a fast size distribution measurement makes this instrument an extremely powerful tool in a variety of studies, especially if the sampling process is automated. It can be a vital instrument, for example, in the study of the dynamics of a mixed culture where the sizes of two microorganisms are different.

As expected, the above optical and counting methods fail completely when the fermentation broth contains suspended solids or is colored; they cannot be used on mycelium, either. Since the media for many industrial processes contain insoluble materials, and since filamentous fungi or molds are frequently used in antibiotic products, the application of the optical methods is actually rather restricted. In such cases, one may be forced to use other chemical, physical, or thermal methods. Finally, an indirect measurement method that employs the macroscopic elemental balance concept to infer the biomass concentration will be discussed in later sections.

Since no direct biomass sensor is available to monitor the mycelial concentration, Nestaas and Wang[119-121] have utilized the filtration characteristic of the mycelia to construct a "filtration probe" shown in Figure 10. The filter cake volume (actually the height) is monitored by a photocell, and the filtrate volume (actually the weight) is monitored by a load cell. The filtration behavior was predicted from a mathematical analysis of a model, and its validity was confirmed experimentally. The filtration resistance was found to be proportional to the filtrate volume for a constant cell concentration, and it is proportional to the cell concentration for a constant morphology.[119] They found that the cell concentration estimated from the filtration data was in good agreement with the measured cell dry weight. One of the basic assumptions used in the correlation was that the specific cake volume be known. However, it was affected by the impeller shear rate,[120] mycelial morphology,[120] and ionic strength of the broth.[121] The "filtration probe" was interfaced to a computer, and the probe operation was made semicontinuous and automatic in a later report.[122]

3. Gas Oxygen and Carbon Dioxide Concentrations

Because the respiratory activity is closely related to the metabolism and the growth of cells, because the measurement needs not be performed *in situ*, and because appropriate instruments are commercially available, the continuous measurement of gaseous oxygen concentration with a paramagnetic O_2 analyzer and carbon dioxide concentration with an infrared CO_2 analyzer has become quite routine in fermentation processes. The gaseous O_2 and CO_2 concentrations are among the very few measurements that can be carried out on a truly continuous on-line basis. These analyzers are commercially available and rather affordable by most researchers and industry. Furthermore, because respiratory activities have profound effects on microbial behavior, an overwhelming fraction of the control scheme to be described in later sections is based on these two measurements. The principles of operation can be found in Solomons,[75] and the basic construction has not changed for the last decade.

Because in most circumstances air, which is routinely used as the source of oxygen, is sparged into the broth at a generous rate to ensure enough oxygen being available for cell growth and maintenance, the difference in the oxygen concentrations between the inlet and outlet is relatively small. Since an oxygen analyzer yields signals in terms of oxygen partial pressure, a slight drift in the ambient barometric pressure caused by the opening and closing of a door or the turning on and off of a ventilation system can lead to an unacceptable magnitude of errors in the calculation of oxygen uptake rate if the barometric pressure is not monitored simultaneously to calculate the needed compensation.[27,87] This compensation is especially indispensable if the control decision is solely based on the calculated values of the respiratory quotient. Furthermore, a paramagnetic oxygen analyzer is also sensitive to the gas flow rate; thus, it also needs to be controlled well.[123]

Since the presence of water vapor can cause the drift of oxygen partial pressure in a paramagnetic O_2 analyzer and the condensation in the measurement cell of an infrared CO_2 analyzer, a desiccation device is necessary prior to the measurements. When combined with the data on the gas flow rate, the gaseous O_2 and CO_2 concentrations can yield oxygen uptake rate (OUR), carbon dioxide evolution rate (CER), and the respiratory quotient (RQ). Other methods, such as the use of gas chromatography for both O_2 and CO_2,[124] oxygen

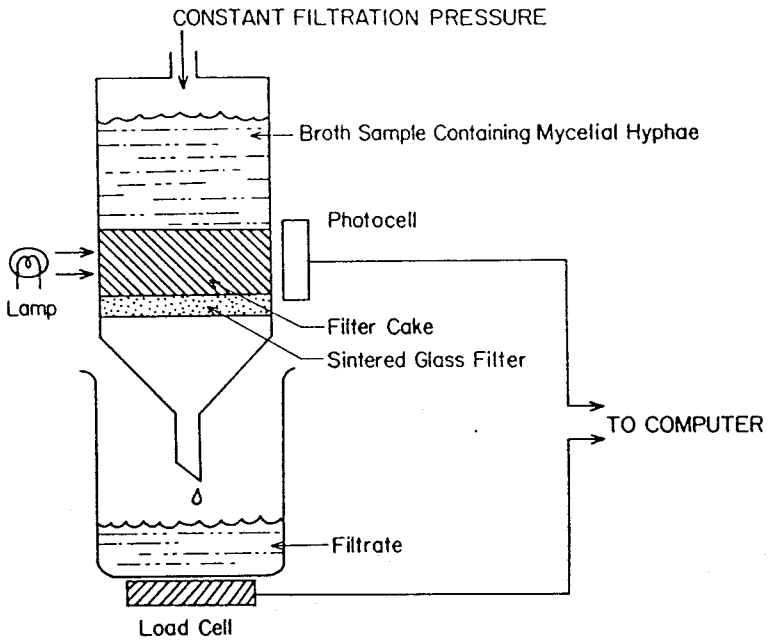


FIGURE 10. Schematic for a filtration probe.

electrodes for O_2,[125] and thermal conductivity for CO_2, are also available but not as frequently used. Mass spectrometry has also been recently introduced.[126-128] The basic instrument is substantially more expensive than the combination of a paramagnetic O_2 analyzer and an IR CO_2 analyzer, but it offers the additional capability of monitoring many other gases at the same time such as nitrogen, ammonia, and others. Cost considerations have limited the use of mass spectrometer, to date, mainly to industrial laboratories.

As a point of interest, oxygen generated by electrolysis has been used in the past to obtain a high oxygen concentration. In these systems the current used for electrolysis, which is directly related to the oxygen consumption, is continuously recorded,[68] and the measurement of gaseous O_2 concentration is effectively eliminated.

4. Dissolved Oxygen

Dissolved oxygen concentration in a fermentor is generally measured with an oxygen electrode. Since the first autoclavable dissolved oxygen electrode was described by Johnson et al.,[129] many variations and improvements have been incorporated into today's commercialized versions.[130-132] Even today the search for a better probe is continuing.[114] Various articles have appeared in the past that extensively reviewed the operation, the construction, and the dynamic response characteristics of various oxygen electrodes.[133-137] Oxygen electrodes can be classified as either potentiometric (galvanic) or amperometric (polarographic or Clark type), depending on whether a current is forced through them or not. The basic construction of each type is shown in Figure 11. It should be noted that these electrodes actually measure the activity, or the equivalent partial pressure, of the dissolved oxygen (oxygen tension) and not the concentration.

In a galvanic type of dissolved oxygen electrode, oxygen diffuses to the cathode through a gas permeable membrane and is spontaneously reduced on a platinum surface by a highly reductive metal (Pb, Zn, Cd, etc.). The oxidation-reduction reaction is as follows:

$$\text{Cathode:} \quad {}^1/_2O_2 + H_2O + 2e \xrightarrow{Pt} 2OH \qquad (1a)$$

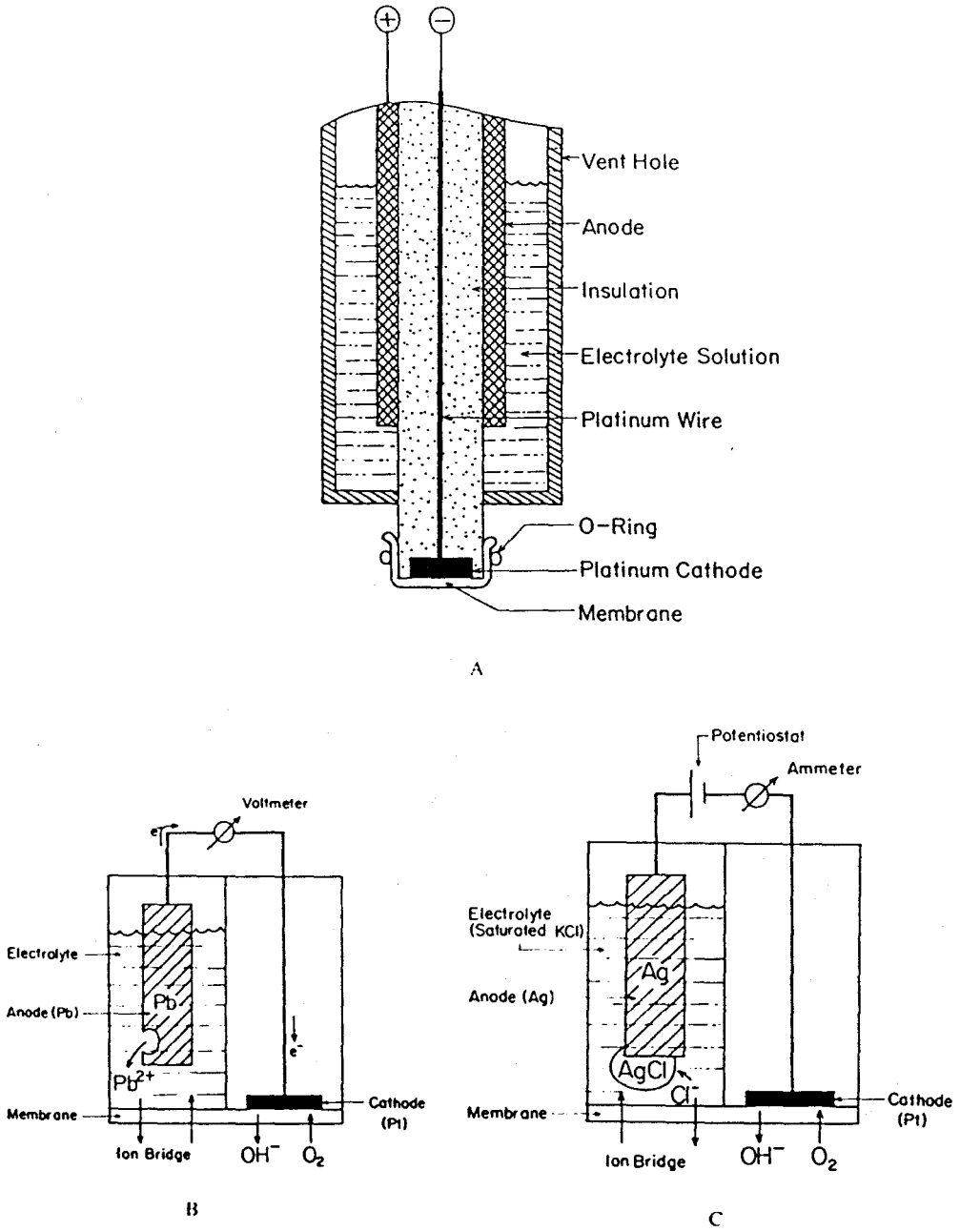


FIGURE 11. Schematic of a dissolved oxygen electrode. (A) Probe construction; (B) galvanic half-cells; (C) polarographic half-cells.

$$\text{Anode:} \qquad Pb \rightarrow Pb^{2+} + 2e \tag{1b}$$

Only a small amount of current is drawn from the electrode to permit a voltage measurement. The voltage generated from the above reaction is correlated to the oxygen tension. Although the electrolyte is not consumed, in time the accumulation of Pb^{2+} ions can become a source of drift.

In a polarographic type, a constant voltage is applied between the anode and the cathode, and the current forced through the electrodes is measured. The oxidation-reduction reaction is essentially the same as that of a galvanic type:

$$\text{Cathode:} \quad {}^1/_2O_2 + H_2O + 2e \xrightarrow{Pt} 2OH \tag{2a}$$

$$\text{Anode:} \quad Ag + Cl \rightarrow AgCl + e \tag{2b}$$

(Actually, two separate steps are involved on the cathode; H_2O_2 is formed first, then reduced further to OH^-.) The current is measured and correlated to the oxygen tension. Since a much higher current passes through a polarographic electrode than a galvanic electrode, OH^- generated by the cathodic reaction tends to make the broth slightly alkaline; meanwhile, Cl^- ions in the electrolyte gradually become depleted, and the probe life partly depends on the amount of Cl^- available.

Because the oxygen molecules have to be brought to the membrane to react, a minimum fluid velocity of approximately 1.8 ft/sec is required for a commercial polarographic electrode to work.[138] Since the reaction occurring on the electrode surface is limited by the mass transfer of oxygen through a thin boundary layer whose thickness depends on, among other things, the bulk fluid velocity and the oxygen diffusivity, the validity of a significant fraction of the published data on dissolved oxygen is, in the authors' opinion, somewhat questionable. The effect of boundary layer thickness on the rate of mass transfer is well known in chemical engineering, and the effect of fluid velocity on the boundary layer thickness is also well documented in the studies using a rotating electrode.[139] Thus, it seems that the best way a stable dissolved oxygen measurement from a polarographic electrode can be achieved is by rotating the electrode to establish a stable boundary layer.

Due to a variety of factors, the signal from both types of electrodes often drifts with time. The drift usually is not too serious for a short-batch operation, but for a long-lasting continuous operation, calibration can be difficult because the removal of the probe introduces the risk of contamination. To overcome this problem, the probe can be mechanically modified to allow *in situ* calibration.[140] Because electrochemical reactions are sensitive to temperature, the probe needs to be compensated for temperature variations if an accurate measurement is to be maintained.[125] A thermistor is sometimes included in an oxygen probe for this purpose. The dynamic behavior of an oxygen electrode has been studied from time to time,[141-143] and it was found that the contact of bubbles and liquid turbulence can cause an error of 20%.[88,140] Consequently, finding a space in a fermentor where probe-bubble interactions can be eliminated is not a trivial proposition.[144]

Although a great majority of the measurement on dissolved oxygen is now carried out with an electrode system of one of the above types, other methods are also available. One of these methods uses a coil of Teflon® tubing immersed in the fermentation broth,[145,146] but it is now rarely used because it requires an extensive auxiliary apparatus, including an O_2 analyzer or a mass spectrometer,[147] to measure the desorbed O_2 concentration in the carrier gas stream.

There are a wide variety of studies based on the measurement from a dissolved oxygen probe. The solubility of oxygen in fermentation broth[148] and the oxygen transfer coefficient under ideal situations,[149-152] in a fermentor,[114,153,154] and in trickling filter slimes have been studied.[155] The effect of oxygen tension on bacterial growth has also been investigated.[156] By comparing the desired set point to the dissolved oxygen concentration measured by an electrode, the control of dissolved oxygen level can be accomplished through the manipulated changes in the agitation speed,[157] aeration rate,[158] or both of the above.[159-161] Another method used to control the dissolved oxygen level is to shift the temperature in such a direction that the change in metabolic rate affects the oxygen consumption rate and restores the dissolved

oxygen concentration to the desired level.[27] However, the benefit resulting from such a control scheme is not clear. It may also be affected by the use of oxygen enrichment,[161,162] which is advantageous when the upshift of air flow rate and agitation speed alone cannot bring up the dissolved oxygen level during the period of high oxygen demand.

As cells grow and multiply in a batch or fed-batch fermentor, the demand for oxygen increases, and dissolved oxygen can often become the growth-limiting factor if it is not maintained properly. Automatic control of the dissolved oxygen level was attempted by Yano et al.,[160] however, some control parameters had to be changed manually. Subsequent use of a microcomputer by Kobayashi et al.[161] and Nyiri et al.[163] successfully achieved the objective of keeping the dissolved oxygen at a constant level. For example, in Kobayashi et al.'s study[161] of growing *Candida brassicae* in a fed-batch mode, three variables (the agitation speed, the air flow rate, and the oxygen flow rate) were manipulated. Moreover, a sharp decrease in oxygen consumption, which was interpreted as the depletion of a carbon source, could be detected. When it happened, the nutrient pump was switched on to avoid starvation. A significant point in their approach was the use of a computer to determine the optimum choice of a set of operating points for the three manipulated variables such that the lowest running cost was realized. Although both aeration and agitation can be used to enhance the interfacial oxygen mass transfer, they exert different side effects on the fermentation broth, such as shear force and foam formation. Thus, aside from the consideration of operating cost, a judicial selection of the combination may be required for microorganisms with poor mechanical strength.[164]

5. Dissolved Carbon Dioxide

Due to the unavailability of reliable steam-sterilizable sensors for this measurement until very recently, dissolved carbon dioxide electrodes are rarely used in fermentation. This is partly because the carbon dioxide concentration in the aqueous phase is often assumed to be in equilibrium with that of the gas phase, and a readily available gas analyzer is generally used to deduce the dissolved carbon dioxide concentration. However, this is not always true due to the slow rate of desorption of CO_2 from the liquid phase,[165] which is, in addition, pH dependent. Supersaturation of CO_2 both in the fermentation broth and inside the cells has been observed.[166] A carbon balance on the whole fermentor has, in the past, often indicated that some missing carbon may be in the form of dissolved carbon dioxide. This measurement may be very useful in closing the carbon mass balance in a fermentor. The consideration of dissolved carbon dioxide is especially important when fermentation is carried out under anaerobic conditions and when an exit gas stream does not exist.

The commercially available electrode which measures HCO^{3-} ions requires the liberation of HCO^{3-} into CO_2 gas and operates on a very strict pH range. Laboratory-constructed steam-sterilizable electrodes can be simply made by adapting a common pH electrode.[167,168] Sodium bicarbonate solution is sandwiched between the glass membrane of the original pH electrode and the newly added Teflon® or silicone membrane, which is permeable to CO_2 (actually HCO^{3-}). The HCO^{3-} concentrations in the sandwiched liquid layer then influence the pH of that liquid, which is, in turn, sensed by the pH electrode. Similar to an oxygen electrode, by this method it is actually the activity, or the partial pressure of CO_2, which is measured and not the concentration. Another method uses a coil of porous Teflon® tubing immersed in the broth.[146,169] The dissolved carbon dioxide diffuses through the membrane and is carried away by a stream of inert carrier gas, which is subsequently passed through an infrared CO_2 analyzer. However, the slow CO_2 desorption rate through the Teflon® tubing causes a significant delay in the response. A faster response is possible with a mass spectrometer system in which a vacuum tube covered with a silicone rubber membrane at one end is directly inserted into the fermentor as a probe.[147,165]

6. Mineral Ions

On-line specific-ion probes for a series of mineral ions are commercially available for Na^+, K^+, Mg^{2+}, and Ca^{2+} concentrations. Other ions present in the fermentor such as PO_4^{3-}, SO_4^{2-}, and Cl^- can also be monitored. The construction and operating principles of ion-specific glass electrodes for various types of ions can be found elsewhere and will not be covered here.[99] Generally, these probes can be operated only at certain limited pH ranges and are not steam sterilizable. The readers should be cautioned that these electrodes measure the ion activity, not the concentration. The name "ion-specific" is rather misleading because no electrode can truly respond to only one species of ion to the total exclusion of all other ions;[76] it may be more appropriate to call them ion-selective instead.

Since these mineral ions are vital for the growth and maintenance of microbial cultures, they actually impart very significant effects on microorganisms. Some of these effects have long been studied qualitatively in the microbiological field. However, more detailed quantitative studies suitable for fermentation control of microorganisms are scarce. Some studies on the effect of phosphate on *Saccharomyces* sp.,[170] *Chlorella pyrenoidosa*,[171,172] and *Selenastrum pyrenoidosa*[172] have been conducted in the past. Otherwise, the fermentor is routinely supplied with an overwhelming amount of these essential mineral ions when only a trace amount will be sufficient to ensure a healthy growth. Although the motivation in doing so is to eliminate any possible complications arising from a shortage of these mineral ions, it is conceivable that the vast amount of overfeeding may cause undesirable side effects such as inhibition. Besides being uneconomical, unnecessary overfeeding of these ions creates a high ionic strength environment which may affect product formation as well as the cell growth. For example, a derepression of the enzymes involved in antibiotic synthesis occurred as phosphate in the broth became exhausted.[173] The cells reversed the metabolic activities from antibiotic production back to biomass production when phosphate was added to the culture broth.[173] Despite the pronounced effect mineral ions exert on microbial cells, so far no systematic investigation has been attempted to study these effects in a fermentor.

7. Nitrogen Source

Ion-selective electrodes are available for both the nitrate (NO_3^-) and ammonium (NH_4^+) ions. Although the use of these electrodes are becoming the standard methods for measuring the nitrogen content in water and wastewater applications,[174] their use in the fermentation field is not widespread due to the sterilization problems and the strict working pH range. Since the measurement of nitrogen source is still a problem, an excess amount of nitrogen is routinely loaded into a fermentor to ensure that nitrogen limitation does not occur. The same problems mentioned in the discussion on mineral ions can infest this measurement as well.

8. Ionic Strength

A continuous measurement of ionic strength requires only a relatively simple apparatus. This is done by performing a conductivity measurement on the fermentation broth. An alternating current of 1 to 10 kHz (to avoid polarization) and below 1 V (to avoid electrolysis) is supplied between two platinum plate electrodes that form a capacitor (C).[78] A change in the ionic strength of the broth is reflected in a change in the resistive component (R) of an equivalent RC circuit. The R component and the C component can be isolated from the combined RC measurement by electronically analyzing the phase shift between the input signal and the output signal.[78] Thus, the impedance/conductance of this RC configuration provides an indirect measurement on the ionic strength.

During the course of fermentation, microorganisms can convert noncharged substances to charged species and vice versa. An example of the first case is the conversion of glucose to lactate or acetate; an example of the second case is the microbial utilization of NH_4^+ from

the broth. Because of the aforementioned fact, Hardy et al.[175] have proposed to use this measurement to monitor cell metabolism. However, because the media are routinely loaded with ionic species, sometimes to the saturation point, the change in the ionic strength due to cell metabolism is not significant compared to the total value.[112] Similarly, since the capacitance constant is affected by the nature of the broth that is sandwiched between two plate-electrodes, the cell concentration can be closely followed by a continuous conductance measurement. Fleischaker et al.[78] have reported a good agreement between the conductance measurement and cell count for transformed human fibroblast. Gencer and Mutharasan have also reported the same findings with yeast,[112] but nitrogen source and salts were absent from the culture medium. The disadvantage of conductance measurement for cell concentration monitoring is that it is highly nonspecific. One is not certain of the exact source of contribution to the changes in conductivity, because the overall effect of various chemical species, and even temperature and aeration, is also indicated by this measurement, besides microbial cells. For instance, the uptake and/or production of ionic species during the course of fermentation will cause a background drift in the biomass concentration measurement.[78]

Since the extracellular ionic strengths affect both the intake of essential mineral ions and the secretion of metabolic end products through membrane transport, it is a potentially important variable that has long been neglected. The infrequency of its use in fermentation studies is not due to the ability to perform a conductance measurement; rather, it is due to the difficulty in interpreting the result.

9. Redox Potential

The theory of redox potential can be found in any standard college freshman chemistry textbook. Kjaergaard has also described, among other aspects, the theory of the measurement.[176] In biotechnology applications, redox potential is usually measured with a platinum electrode in conjunction with a commonly used calomel or Ag/AgCl reference electrode. This measurement is highly nonspecific in the sense that the total overall oxidation-reduction capacity of the system is read. Thus, any inorganic or organic compound capable of oxidation and/or reduction, not just oxygen and hydrogen, will affect the reading.

Glass redox electrodes have been proposed to minimize the influence of oxygen for systems under intensive aeration.[177] The influence of certain redox substances can be eliminated or reduced by covering the electrode surface with membranes impermeable to those substances.[178] Thus, the influence of oxygen can be separated, e.g., by using a commercially available membrane that is permeable only to gases.[178] In addition to nonspecificity, the measured redox potential, strictly speaking, is not correct from a thermodynamic point of view because a fermentation system is definitely not in equilibrium.[176,177] Despite these shortcomings, several investigators have concluded that the redox potential can be a valuable indication of the metabolic activities inside a fermentor,[176,179-184] and the information on the oxidative status is more readily available from this measurement than from the dissolved oxygen measurement.

Dahod has mentioned many reasons for favoring redox potential measurements.[180] Among them are the erratic dissolved oxygen probe signals, rupture of the Teflon® membrane, and the membrane's contribution to the oxygen diffusion resistance. The range of operation of a redox potential probe is also wider than that of an oxygen electrode. When the dissolved oxygen signal drops to zero, no further information on the oxidation state of the broth is available; however, a negative redox potential is possible.[179] The disadvantage of nonspecificity can also work as an advantage in many situations. For example, redox potential can be a better indicator of the oxygen availability to cells when severe mass transfer limitation creates a discrepancy between the oxygen concentration in the bulk phase and the oxygen availability.[183] Redox potential is the only means to study the overall oxidation-reduction conditions in an anaerobic fermentation where a dissolved oxygen probe totally fails. It is

also better suited to the monitoring of the facultative microorganisms because the concept of oxygen limitation, although frequently mentioned, is vague when applied to these microorganisms. In addition to pH and dissolved oxygen level,[184] fermentation product level has also been correlated to the redox potential measurement.[183]

The control of the redox potential has been suggested,[176,181,182] but the experimental actualization has been rather limited.[185] Theoretically, this can be accomplished by adding chemical oxidizers and reducers or by sparging with a gas whose nitrogen content can be controlled. It is also possible to accomplish this by controlling the addition of nutrient, such as glucose, to the fermentor; however, oxygen supply may have to be adjusted, in addition, in order to maintain the redox potential for a prolonged period of time as cell concentration increases in a batch fermentation.[185] So far, the redox potential measurement has not yet been coupled to a computer for control purposes. In conclusion, this measurement can be easily and inexpensively obtained, but it is seldom used. The importance of this measurement has already been demonstrated,[183] but the interpretation of the result remains difficult, partly because of the low level of enthusiasm toward this measurement, which is, in turn, partly due to the difficulty encountered in interpreting the result.

10. Enzyme-Coupled Sensors

During the last decade, there have been numerous activities in the development of specific sensors that are coupled with immobilized enzymes. Because of an enzyme's specificity for a particular chemical compound, a very wide range of biochemicals can be assayed selectively by employing proper enzymes. The research activity in enzyme-coupled sensors started initially in the field of clinical chemistry to determine the concentration of various amino acids, glucose, urea, uric acid, ethanol, etc. in blood serum. No less than 100 scientific publications have dealt with these sensors; Everse et al.[186] have compiled a partial list of the activities up to 1977. Guilbault[187,188] also has an extensive list for enzyme electrodes which appeared before 1980. Practical applications of enzyme-coupled sensors have been restricted mainly to the areas of clinical analysis where discrete samples are generally used. However, their uses in fermentation fields have just begun and are expanding rapidly in the recent years, as can be seen in Table 2. Various reviews have been given for enzyme-coupled sensors by Guilbault,[187,188] Barker and Somers,[206] Everse et al.,[186] and Ianniello and Jespersen.[207]

In an enzyme-coupled probe, an appropriate enzyme is immobilized by using one of the various immobilization techniques. The substrate level is determined by observing certain effects of the enzyme-catalyzed reaction that are characteristic of that reaction. These analytically observable effects generally include the generation of heat during the enzymatically catalyzed reactions or the appearance and disappearance of those species participating in the reactions.

Depending on the method of detection, enzyme probes can be classified into enzyme electrodes (detection by electrochemical methods), enzyme thermistors (by thermal methods), enzyme mass spectrometers (by electromagnetic methods), enzyme spectrophotometers (by optical methods), enzyme transistors (by electronic methods), etc.[208] Of the above sensors, enzyme electrodes and enzyme thermistors have been overwhelmingly dominating in the number of studies. Moreover, if the feed substrate concentration is known, then the above methods can be easily applied to the study of the kinetics of enzyme immobilization with only slight modifications. One such example is the continuous spectrophotometric monitoring of the effluent concentration of one of the species involved in the reaction inside an immobilized enzyme reactor.[209,210]

In an enzyme electrode, an appropriate enzyme system capable of selectively catalyzing the reaction of the substrate is chosen with the following considerations: the degree of specificity for the substrate whose concentration is to be measured, the stability of the

Table 2
ENZYME-COUPLED SENSORS FOR SUBSTRATES

Type	Enzyme	Detection	Year	Ref.
Cholesterol	Cholesterol oxidase	Pt(O_2 electrode)	1977	189
Penicillin	Penicillinase	pH	1979	190
			1978	191
	β-Lactamase	pH	1976	192
Glucose	Glucose oxidase	O_2 electrode	1979	193
			1980	194
			1982	195
			1983	196
	Glucose isomerase	Thermistor	1979	197
	Glucose dehydrogenase	Pt electrode	1979	198
Phenol	Phenol hydroxylase	O_2 electrode	1980	199
Catechol	Catechol oxygenase	O_2 electrode	1980	200
L-Lysine	Lysine decarboxylase	CO_2 electrode	1983	201
Sucrose	Invertase-mutarotase-glucose oxidase	O_2 electrode	1976	202
NAD	Alcohol dehydrogenase	O_2 electrode	1979	203
Alcohol	Alcohol oxidase-catalase	O_2 electrode	1983	204
Ethanol	Alcohol dehydrogenase	Pt electrode	1979	198
Lactose	Lactase-glucose oxidase	Spectrophotometer	1979	205
	Lactase-glucose oxidase-ammonium molybdate	Iodide electrode	1979	205
	Lactase-glucose oxidase-catalase	O_2 electrode	1979	205

enzyme so that prolonged continuous analysis is possible, the involvement of observable chemical species, and the availability and cost of enzymes. Immobilization is sometimes preferred but is not absolutely necessary. The enzyme is placed in the proximity of an ion-selective electrode, and the chemical species participating in the reaction is closely monitored (see Figure 12). For example, glucose concentration can be determined by reacting with glucose oxidase according to the following reaction:[187,211]

$$C_6H_{12}O_6 + O_2 \xrightarrow{\text{glucose oxidase}} C_6H_{10}O_6 + H_2O_2 \quad (3)$$

Sometimes a single step gives no readily measurable species. In such cases, another enzymatic reaction may be coupled to the first one so as to produce some directly measurable effects.[194,202,205] There still exist a few problems with these enzyme electrodes, and not all reported methods are completely suitable for the direct use on a fermentation broth. Many studies are carried out with artificially synthesized samples, and many need cleaning with phosphate buffer solutions after each measurement which makes the measurement discontinuous. For example, Kernevez et al.[196] reported the use of a dynamic method in which computer-coupled measurements could be made at the rate of one analysis per minute with better than 1% accuracy. However, liquid rinsing (to purge glucose from the electrode) and gas rinsing (to restore O_2 to membranes) were necessary, and the possibility of the presence of O_2 in the glucose sample solution was not considered. Saturating the membrane with O_2 is frequently necessary for glucose measurement.[202] Sterilization problems are rarely considered and the measurement is destructive. Computer-coupled, continuous, on-line enzyme electrodes in a probe form, capable of being directly inserted in a fermentor, are presently rare.

A variation to the above scheme is presented by Volesky and Emond[205] in which the electrode is located downstream to a series of flow-through beds of immobilized enzymes.

Enzyme thermistors have been introduced by Mosbach and co-workers and others.[197,208,212-217] As shown in Figure 13, the construction of an enzyme thermistor is relatively simple. Two thermistors are placed in the inlet and outlet of a reaction column

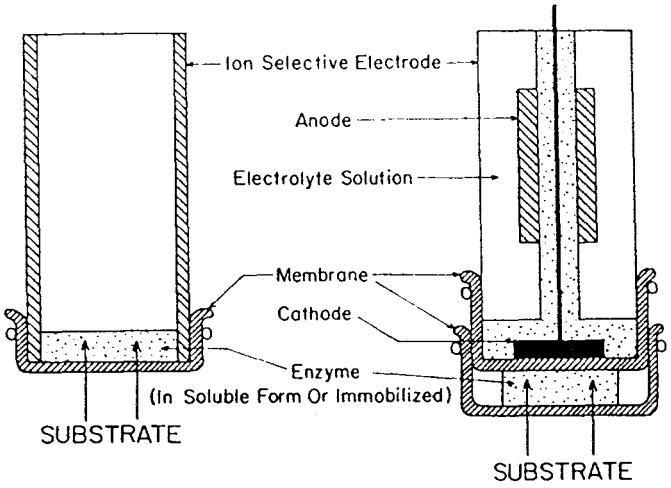


FIGURE 12. Schematic of an enzyme electrode.

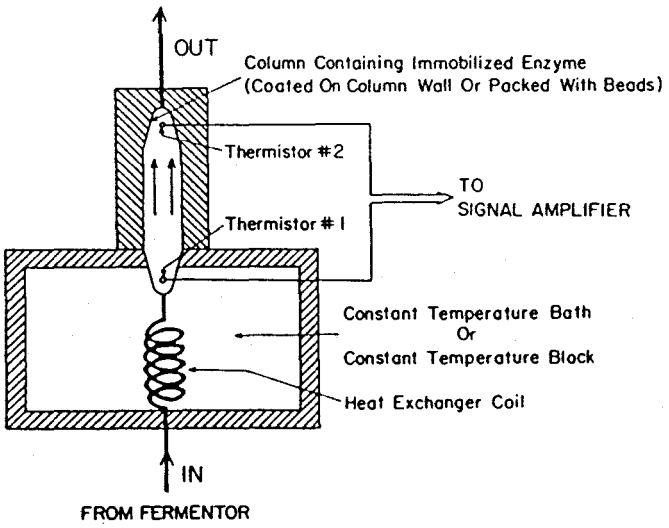


FIGURE 13. Schematic of an enzyme thermistor.

packed with an immobilized enzyme preparation. A small stream is diverted from a fermentor and passed through a coil of heat exchanger immersed in a constant-temperature bath. The stream is then passed through a bed of immobilized enzyme packings or a column coated with immobilized enzymes on the inner walls. The change in the stream temperature resulting from the enzymatically catalyzed reactions is monitored by the two sensitive thermistors. The signal from these thermistors is first amplified by a Wheatstone bridge and sent to the next IC amplifier. The amount of heat generated by the reaction is then correlated to the temperature difference. Finally, the concentration is determined. For a more accurate determination, an identical reference column may be placed next to the working column to minimize the background.

In order to obtain an accurate measurement, the reaction process should proceed to completion in the flow cell, that is, the reaction time should be short compared to the residence time for the liquid in the flow cell. As long as the enzymatic reaction reaches

completion, the amount of heat evolved is directly proportional to the quantity of reactant present and independent of the activity of the enzyme if the enzyme used is highly specific. One wants to ensure complete reaction, but, on the other hand, a short residence time is preferable so that the concentrations in the fermentor can be monitored closely. Therefore, as the enzyme becomes deactivated, a longer residence time, thus, longer delay, is necessary. An immobilized enzyme of high specific activity will improve the accuracy and sensitivity of the measurement because the reactor volume and, consequently, the heat capacity may be reduced.

The considerations in the choice of an enzyme system are similar to those of an enzyme electrode, except that the heat of reaction is of interest here. If the amount of heat evolved from the desired reaction is small, the primary reaction can be coupled with other heat-producing reactions to achieve a chemical amplification effect. For example, Reaction 3 catalyzed by glucose oxidase releases approximately 20 kcal/mol of heat.[212] However, the heat effect can be more than doubled if some catalase is present to carry out Reaction 3 one step further:

$$H_2O_2 \xrightarrow{\text{catalase}} \frac{1}{2} O_2 + H_2O \qquad (4)$$

The enthalpy for H_2O_2 decomposition is approximately 30 kcal/mol.[212] Thus, by the appropriate coupling of reactions, the heat effect can sometimes be greatly amplified, although the measurement may be somewhat less direct.

Since the above glucose conversion requires oxygen, some means of oxygen supply needs to be provided to avoid the problem of incomplete conversion due to oxygen exhaustion. In the past, attempts were made to generate oxygen in an enzyme-based electrode by electrolysis to prevent problems caused by oxygen depletion.[218] The same precaution also applies to other substrate-enzyme systems. The advantages of an enzyme thermistor are similar to those of an enzyme electrode, including enzyme stabilization in the immobilized state and the possibility for repeated measurements and continuous analysis. Moreover, unlike other existing methods, this method does not require any elaborate preparation of the biological samples. Particularly, there is the advantage that the sample needs not be optically clear.

Some of the immobilized enzyme electrodes (notably glucose electrode) are commercially available in self-contained units.[24,187] Many more are currently under intensive development. Although they are mainly aimed at the highly profitable clinical market, the fermentation industry will undoubtedly benefit from these electrodes, as well. This is one area where the sensor development is truly active, and the future for enzyme-coupled probes is bright.[208]

11. Microbe-Coupled Sensors

Because of low stability and inflexibility of enzyme sensors and other demands that cannot be filled by the conventional instruments, the use of immobilized whole microbial cells or organelles as a sensor has also been investigated quite intensively. The construction and the principles of operation of a microbial sensor are similar to those of an enzyme-coupled sensor. Classification by the methods of detection can also be carried out similarly: microbial electrodes, microbial thermistors, etc. However, microbial electrodes constitute a great majority of the sensors constructed. A review of the literature in microbial sensors has been given by Mattiasson.[219] Karube and Suzuki have also provided a survey of microbial sensors for detecting a variety of substrates in fermentation processes.[220]

In microbial electrodes, the immobilized cell paste may be sandwiched in-between an electrode and a membrane,[221] or the microbe-collagen membrane may directly cover an electrode.[222] Many microbial electrodes have appeared. Oxygen electrodes in conjunction with immobilized aerobic microorganisms, based on the determination of microbial respi-

ration, have been used for the measurements of glucose, acetic acid, alcohols, ammonia, nitrate, and biochemical oxygen demand (BOD); biochemical fuel cells in conjunction with immobilized anaerobic microorganisms, based on the detection of electroactive metabolites, have been used for cell population and vitamins; pH electrodes and CO_2 electrodes with immobilized whole cells, based on the detection of hydrogen ions, have generally been used for antibiotic and glutamic acid.[223] A partial list of these microbial electrodes is available,[224] and Table 3 shows some of the recent additions from the area of biotechnology. Of these studies, the use of a computer to compensate for signal drift in real time by Yasuda is noteworthy.[232] Although a microbial electrode is generally constructed for the purpose of measuring substrate levels, it has many side uses as well. One such use is in the testing of the microorganism's ability to assimilate certain substrates.[234]

12. Calorimeters

All biological processes are accompanied by heat effects, usually the evolution of heat. The measurement of the heat quantity involved in a reaction has always been important in thermodynamic studies, but its direct use in fermentation studies has been considerably delayed because of the small rate of heat evolution in a typical microbial system. Recently, fast and ultrasensitive microcalorimeters have been developed and marketed,[235] and calorimetry has become a potentially powerful analytical tool in biological applications.

There is a wide variety of calorimeters based on different principles of operation and capable of continuous measurements. The constructional details and theories on modern calorimeters have been discussed in many books or articles solely devoted to this instrument.[81,235-240]

Calorimeters can be roughly classified into two major categories: adiabatic calorimeters and heat-conduction calorimeters. In an adiabatic calorimeter, the measuring cell is thermally insulated so that the amount of heat exchange between the measuring cell and the surrounding environment is as small as possible. The heat effect during a reaction is registered as a change in the temperature of the system. Usually, a calibration curve for a calorimeter is obtained by following the changes in temperature when a known amount of heat is dissipated through an electrical resistor placed inside the calorimeter. An adiabatic calorimeter of the crudest construction has its wall thickly lined with thermal insulators. On the other hand, a more advanced calorimeter has an adiabatic shield consisting of a thin wall made of a good conductor, usually gold, and a heater winding over the surface. The heater winding's heat output is automatically controlled so that the shield is at the same temperature as the measuring cell. In this way, no heat will be lost through the cell wall.

In a heat-conduction calorimeter,[241] the heat generated inside a measuring cell is conducted through the wall to the surrounding constant-temperature heat sink. The rate of heat flux through the wall is generally proportional to the difference in the temperature between the cell and the heat sink. This temperature difference is registered with a thermopile constructed from a long string of thermocouples soldered end to end in series. These thermocouples also act as heat conductors. In order to have a fast response, the thermocouples should be made of materials of small heat capacity and large heat conductivity.

There are also many other types of calorimeters based on concepts in-between the above two extremes. One of these is the thermoelectric heat-pump calorimeter. It is half insulated and half conducting. In this type of calorimeter, heat is supplied or withdrawn electrically from the measuring cell so as to keep the temperature inside the cell at a constant value. By monitoring the rate of heat input and/or output, the heat effect of a reaction is determined. The temperature changes in an endothermic reaction can easily be compensated by passing an electrical current through a resistor. For an exothermic reaction, heat is pumped out by a thermoelectric cooler. The thermoelectric cooler is based on the principle of Peltier effect, which is essentially the reverse of the operating principle of a thermocouple. As an electric

Table 3
MICROBIAL SENSORS FOR SUBSTRATES

Type	Microorganism	Detection	Year	Ref.
Nitrogen dioxide	Nitrifying bacteria	O_2 electrode	1983	225
BOD	*Clostridium butyricum*	O_2 electrode	1977	222
	Hansenula anomala	O_2 electrode	1980	221
	Trichosporon cutaneum	O_2 electrode	1980	226
			1981	227
Phenol	*T. cutaneum*	O_2 electrode	1979	228
Nitrilotriacetic acid	*Pseudomonas* sp.	Ammonia electrode	1981	229
		CO_2 electrode		
Methanol	*T. brassicae*	O_2 electrode	1979	230
Ethanol	*T. brassicae*	O_2 electrode	1981	224
Acetic acid	*T. brassicae*	O_2 electrode	1981	224
			1981	227
Ammonia	*T. brassicae*	O_2 electrode	1981	224
	Nitrosomonas sp.	O_2 electrode	1981	227
Cephalosporins	*Citrobacter freundii*	pH electrode	1979	223
Glucose	*P. fluorescence*	O_2 electrode	1979	223
			1979	231
Glutamic acid	*Escherichia coli*	CO_2 electrode	1981	227
			1982	232
Formic acid	*Clostridium butyricum*	Fuel cell	1980	233

current passes through a thermocouple in an appropriate direction, heat is absorbed at the junction inside the measuring cell and is released at the junction in contact with the heat sink. When the intrinsic resistance of the junction is taken out of consideration, the rate at which the heat is absorbed is related to the electrical current.

Alternatively, the whole fermentor can be regarded as a calorimeter. In this case, the measurement is usually carried out by an *in situ* dynamic procedure in which the rise in the broth temperature is recorded when the temperature control is turned off.[242-244] This method is well suited for monitoring in batch fermentation, but it is not as useful in a continuous fermentor because the large time constant involved for the temperature measurement prevents the use of a quasi-steady-state assumption for the energy balance. Another possibility is to employ flow calorimetry in which the heat effect is measured by withdrawing a side stream from the fermentor, passing it through a calorimeter, and returning it to the fermentor after the measurement is completed. However, this method fails when wall growth is present or when cell density is so high that oxygen or certain nutrients may be exhausted during the transport to the flow cell.[245] It is also unsuitable for use with filamentous organisms or non-Newtonian fluids.[243]

Currently, the number of calorimetric studies in the fermentation field is relatively small, and calorimetry has not yet gained the attention commensurate with its importance. There are three main uses of calorimetric data. One is in the calculation of the cooling requirement in the design and scale-up of fermentation equipment as well as during the actual operation;[246] the second use is in the study of microbial growth,[247] dynamics,[248] and thermogenesis;[249] the third use is in the bioassay of proteins, lipids,[250] enzymes,[251] and antibiotics.[252] Using a dynamical method, Cooney et al.,[242] Luong and Volesky,[244] and Volesky and Thambimuthu[253] have performed experiments with bacteria, yeasts, and molds on various substrates and found that the rate of heat production during metabolism correlates well with the rate of oxygen consumption. Thus, by using the fact that oxygen evolution rate is also closely correlated with cell growth in many instances, calorimetric measurements were used to estimate the cell concentration.[254] Wang et al.[255] applied the dynamic calorimetry to monitor the growth of *Saccharomyces cerevisiae*. In the past, the ratio of heat production per oxygen consumption

has also been proposed as a physiological variable to monitor metabolic activities such as cell respiration efficiency and growth yields.[243] However, because it is almost constant under most commonly encountered circumstances, it is presently regarded as a physiological variable of only limited value. At one end, the efficiencies of energy conversion were found to be significantly different under different conditions of nutritional limitation;[256] at the other end, they were found to be nearly constant under different dilution rates.[257] The thermogram of *Escherichia coli* grown on glucose showed the diauxic growth on the acetate product after depletion of glucose.[245,256] Thermograms can also be used to study growth kinetics.[258] The typical magnitude of heat evolution was 1 kcal/0.3 g of cell produced,[243] and the heat of combustion ranged from 5.0 to 6.4 kcal/g dry cell.[258,259] However, since the heat of combustion is defined on the basis of dry cell weight, which includes ash, the variance is believed to be the result of nonuniformity in the cell composition from species to species and the result of dependence of cell composition upon growth conditions even within the same strain.

13. Automatic Sampling and Inoculation

Many measurements cannot be performed *in situ*, and many also require the sample to be cell-free and solid-free; therefore, sampling and sample preparation cannot be totally avoided. Aseptic inoculation[260] and sterile sample withdrawal procedures can be readily automated by various mechanical configurations.[52,260-263] Three methods are available for the elimination of particulate matter: membrane filtration, mechanical filtration, and centrifugation. Continuous on-line filtration (dialysis) can be easily accomplished with a membrane, usually in a by-pass tubular form[39,264-269] or sometimes directly in the fermentor;[270] however, finding a suitable membrane may be difficult in some circumstances, and the response time is long due to the mass transfer resistance of the membrane and the liquid film.[266] Depending on the flow rate, the permeability of the tubing, and the Henry's constant for each individual component, the delay is about 2 to 10 min.[78] Although the risk of contamination is practically nil unless the membrane breaks, attachment of cells on the broth side of the membrane can restrict dialysis rates or yield false readings. A continuous selective separation of molecules from the broth into an inert carrier gas stream or a liquid stream is possible with a suitable membrane. On-line preparation of a sample by filtering and centrifugation for measurement purposes is still a difficult, if not impossible, mechanical engineering feat.

14. Other Methods

Finally, a great variety of off-line wet chemical and microbiological procedures are available for the analysis of a wide range of fermentation substrates and products.[271,272] Gas chromatography,[273] flame ionization detectors,[146,267] and spectrophotometers[271] are some of the widely used means for detection. Some of the measurements that require no extensive preparation can be readily coupled to a computer; a mass spectrometer for the determination of gas phase components and volatile liquid phase components is one such example.[126,127,147,274-277] Those that require extensive sample preparation can be and have been incorporated into automated analysis units; the use of an auto analyzer is one such example.[39] These units consist of a series of tubings, valves, timers, pumps, and switches to perform a series of tasks: sampling, necessary pretreatment, mixing, dilution, etc. The sequentially prepared samples are finally analyzed with conventional specific detection devices such as spectrophotometers, electrodes, liquid chromatography, gas chromatography, mass spectrometers, and many others. Gas chromatography has recently been automated with a computer-controlled electropneumatic injector,[278] and the same principle can be applied to high performance liquid chromatography (HPLC) as well.

Besides the examples discussed previously, various unique sensors have also been tried. An affinity sensor based on an optical detector can be used to measure the level of sugars

present.[279] Accurate, continuous determination of ethanol concentration during aerobic cultivation of yeasts has been accomplished by sensing its vapor in the existing gas stream with a smoke detector[280] or a semiconductor gas sensor.[281] A waterproof solid-state chip with a series of MOSFETs capable of determining a wide range of metal ions is currently under development.[282] Finally, automation can be applied to areas other than instrumentation; even culture screening can be automated.[283]

C. Biochemical Measurements

Now, measurements into the cells are considered. Measurements described in this section are primarily concerned with the molecular chemistry inside the cells. They are RNA, DNA, protein, carbohydrate, and lipid levels. The macromolecular composition in cells is known to be influenced by the environmental conditions.[284] NAD, NADH, ATP, ADP, AMP concentrations, and various enzyme activities need to be measured as well if an in-depth understanding of the kinetics, the mechanics, and the pathways of microbial metabolism is to be reached. Unfortunately, no truly on-line instruments exist for any of these measurements except for NADH. In addition, in most cases cells must be lysed prior to the analysis of the contents. Some units capable of automated sampling are under heavy development; however, they are aimed mainly at the biological and biochemical field of study.

1. ATP/ADP

It is a well-known fact that adenosine triphosphate (ATP) is present in all living cells as the currency of available energy. Because ATP participates in virtually all the important metabolic reactions, the importance of the measurement of ATP levels inside a cell cannot be overemphasized. The energy generated in one reaction step is stored in ATP, which subsequently participates in another reaction that requires the release of the stored energy. Its occurrence is related to the total amount of viable biomass and the metabolic activity.

The dynamics involving ATP is fast compared to the growth of cells; it takes a growing cell only about 1 sec to consume the whole pool of ATP in its body if regeneration were not occurring simultaneously.[78,100] Thus, continuous, highly sensitive, on-line ATP measurements may be necessary to follow closely fast intracellular dynamic behavior in a satisfactory manner. Unfortunately, a truly on-line measurement of the ATP levels in a cell currently does not exist, and the measurement is presently carried out by an essentially off-line batch sampling technique. Because the ATP dynamics is relatively fast, a rapid sampling technique is required. The reaction must be stopped to preserve the existing conditions if the analysis cannot be performed almost instantaneously. An automatic sampler capable of extremely rapid sampling and a quick quenching with phosphoric acid (to preserve the ATP content), though, is currently commercially available. Harrison and Harmes have also described the operation of a rapid sampler.[285] The final measurement of ATP level can be carried out on the quenched samples at a later time with a biometer that utilizes luminescence. Many investigators have shown that ATP concentration is a good indicator of cell viability;[100,286] ATP is absent from nonviable cells. Thus, the ATP measurement, when combined with other total biomass measurement, may afford the distinction between viable and nonviable fractions. In the past, many studies have correlated this measurement to biomass concentrations.[100,104]

Methods for off-line batch analysis of ADP and AMP have also been developed.[104,287,288] Bioluminescence analyzer is used to monitor and integrate the light emission from the sample.[288] Instead of the absolute ATP level, it is sometimes more revealing to use a quantity called energy charge, EC, defined by:

$$EC = \frac{[ATP] + \frac{1}{2}[ADP]}{[ATP] + [ADP] + [AMP]} \quad (5)$$

Hence, it represents approximately the number of basic energy units per adenosine base. A truly on-line method or a fully automated sampling-analysis scheme would be desperately needed if the ATP measurement is to be incorporated in computer control.

2. NAD/NADH

Just as the extracellular redox potential is closely related to the oxygen concentration in the broth, the intracellular redox potential is closely related to the NADH content within the cell. Many reactions involved in the cell metabolism are of the oxidation-reduction types. Nicotinamide adenine nucleotides have been known to be the coenzymes of a good fraction of these oxidation-reduction reaction steps. Hence, by following NAD/NADH levels inside a cell, important information can be extracted that can be very helpful in elucidating the underlying model and in formulating a process control strategy.

Harrison and Chance have demonstrated the use of an on-line fluorometer in the measurement of NADH level.[107] Since then, many investigators have followed with the same method.[105,108,285,289-291] A schematic drawing of the apparatus used by them is shown in Figure 14. The filtered ultraviolet (UV) light (366 nm) is shone on the culture broth through a window in the fermentor vessel wall. The UV light excites NADH to a higher energy state; fluorescent light with a wavelength of 460 nm is detected by a photomultiplier as the excited NADH is restored to its original state. In the past, several investigators have correlated the cell biomass concentration to this luminescence measurement.[105,107,108] However, this measurement is not accomplished without any problems. Although the selectivity of the measurement is relatively high, the background signal from, for example, the media used for cell cultivation also contributes somewhat.[289] The signal instability caused by the presence of air bubbles may force the use of a more expensive laser light source instead of a cheaper UV light source.[285] In addition, the drift in the intensity of the UV light lamp may require some means of normalization of the output signal.[78]

Since the intensity of culture fluorescence intensity depends on the number of cells containing NADH as well as the level of NADH in the cells, the fluorescence intensity is directly related to the cell biomass only if the level of internal NADH stays unchanged during the course of fermentation. The environmental conditions such as temperature, pH, dissolved oxygen concentration, and various nutrient concentrations all affect both the cell metabolic rates and pathways, and the effect of these changes may be reflected in the changes of the NAD/NADH level.[105,108] For example, when the withholding of oxygen supply prevented the oxidation of NADH to NAD in aerobically growing yeast, the culture fluorescence, which was related to the level of fluorescent NADH, jumped to a new value within 1 to 2 min.[105,292] Because of the fast and sensitive response of NADH level to dissolved oxygen concentration in a fermentation broth, culture fluorescence measurement was used to detect the spatial nonhomogeneities of gas-liquid mass transfer inside a fermentor.[292] In another example, the NADH level decreased as the depletion of glucose prevented the replenishment of the high-energy NADH reserve.[105] Adding appropriate chemicals to block certain NADH regenerating metabolic pathways. Zabriskie and Humphrey[108] were able to estimate that approximately 50% of the culture fluorescence was contributed by the intracellular NADH; the rest were contributed mostly by unknown sources. In yet another example, the glucose uptake and mixing times were studied by using the fluorometric measurement of NADH.[291] Recently, this measurement was also used in an attempt to unveil the substrate uptake mechanisms for *Candida tropicalis* and the baker's yeast.[290]

The fluorescence measurement of NAD/NADH has long been used by biochemists to study the effect of environmental conditions on cells, but it is not as regularly monitored in the fermentation field as are pH or oxygen and carbon dioxide concentrations. As a consequence, a detailed interpretation of the culture fluorescence data has not yet been accomplished. Since many intracellular reaction steps can involve NAD/NADH, this measurement

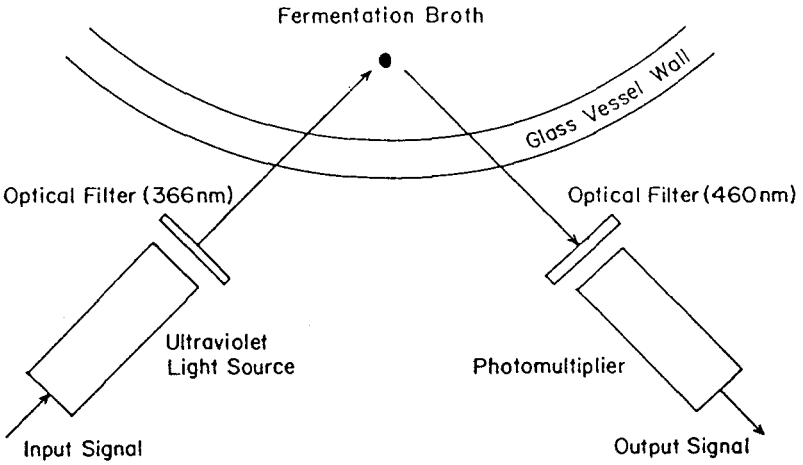


FIGURE 14. Schematic diagram of a fluorometer.

can give an indication of the overall reaction in a cell. Thus, it may be used as a basis in the control and on-line optimization studies.

Besides the optical method, an electrochemical method has also often been employed by investigators in the fermentation field.[106,109,110,293] The double-electrode system, shown in Figure 15, takes advantage of the oxidation-reduction property of NAD/NADH. Two sets of electrodes are identical except that the anode of the determination probe is exposed to the fermentation broth, whereas the anode of the reference probe is covered with a cellulose dialysis membrane. In the more sophisticated system, a built-in saturated calomel electrode (S.C.E.), which is routinely used in electrochemical studies, can provide the reference voltage for each of the probes. For a high precision measurement, a thermistor may be attached to each probe to correct for the temperature effect on the electrochemical reactions. As a constant voltage in the range of 220 to 280 mV (vs. S.C.E.) is applied to each of the electrodes, NADH on the microbial cell surface is oxidized to NAD on the surface of the exposed anode as the cells are brought to the anode by agitation in the fermentor.[109] Thus, the current produced by the reaction is related to the NADH levels on each cell, and, thereby, the cell concentration. The reference probe, with anode covered by a cellulose dialysis membrane which prevents cells from reaching the platinum anode surface, can measure the contribution from unknown background sources, including NADH and other oxidizable compounds in the fermentation broth.

Since the electrochemical probe takes the NADH from the surface of the cells, the use of this method to study the internal metabolic reactions may be complicated. Furthermore, the output signal may depend heavily on the mass transport characteristics near the electrode surface because the cells must be carried to the electrode surface to donate electrons. Thus, the response time of the electrodes is approximately 15 min, possibly too long for the study of fast reaction dynamics. However, this steam-sterilizable electrode can be obtained at a much cheaper price than the fluorescence system described earlier. Although the output signal from the polarographic electrode system was found to be sensitive to temperature and pH[106,110] as in culture fluorescence measurement, the effect of dissolved oxygen was only slight[106] as opposed to the culture fluorescence method. This method has been applied to monitor directly the microbial cell numbers.[106,109,110,293] Although not so specifically stated, the electrochemical determination of cell populations from the oxidation of electroactive substances as employed by Matsunaga et al.[293] works in similar principles.

Wallace et al.[294] have recently used a unique amperometric technique; however, it is not as commonly used among biotechnologists as the above two methods.

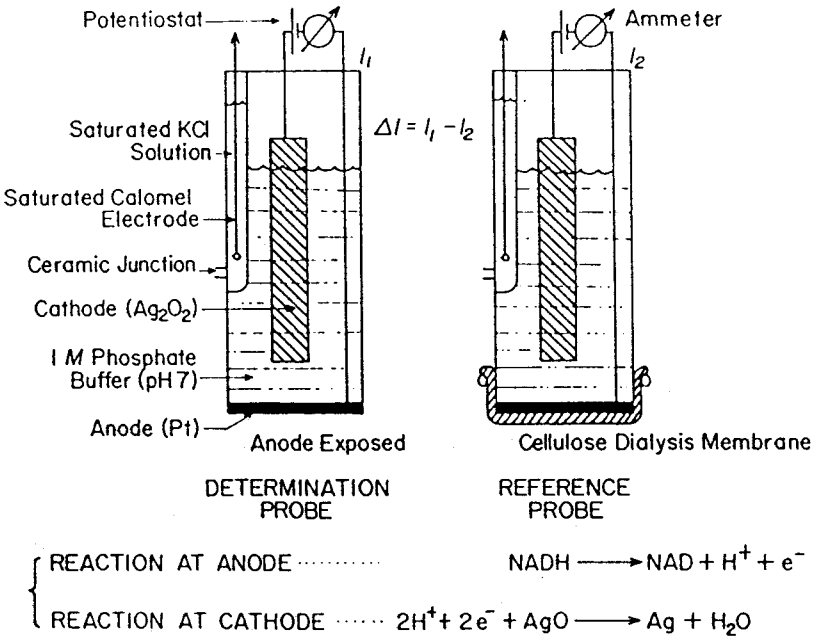


FIGURE 15. Schematic of a double-electrode system for NADH measurement.

Although the measurement of NAD/NADH level by either culture fluorescence or polarographic electrode system depends on biomass concentration, metabolic activity, and a variety of intracellular and extracellular environmental conditions, it may be adopted to control purposes.[105] Ristroph et al.[289] has employed the fluorescence measurement to determine the substrate feed rate in a fed-batch single cell protein (SCP) fermentation. In their study, *Candida utilis* was grown on ethanol. A control of feed rate in this fermentation is necessary because an overfeeding leads to acetate formation which reduces the SCP yield on ethanol. Similarly, an underfeeding leads to a slow cell growth rate which reduces the volumetric productivity. Since the depletion of a carbon source can be detected by a drop in culture fluorescence as mentioned before, a computer can be programmed to activate the feed pump to add more ethanol into the fermentor when the culture fluorescence drops to a predetermined value.

3. Cytophotometry and Cytofluorometry

Although we generally employ unstructured, unsegregated models in the study of fermentation, it is well known that cells in a population do not all have the same size or a uniform composition. Cells behave differently at different stages of development; the metabolic activity shifts over the growth cycle of a cell. Such tendency of treating the microbial system as a homogeneous population is reflected in the instrumentation we use; all the instruments we have considered thus far, with the only exception of the Coulter® counter, measure the bulk property of the whole population. A structured approach must be considered if a more fundamental understanding of microorganisms is to be gained, and a segregated approach is a must in studying the behavior of a mixed culture.

Recently, a computer-coupled instrument, a laser flow microfluorometer, has been developed to measure the distribution function of, among other things, the macromolecular contents and the cell sizes in a population. By labeling a sample of cells with a fluorescent stain, the frequency functions for a variety of macromolecular components such as proteins, certain specific enzymes, RNA, and DNA can be rapidly measured by a flow fluorometer.

As shown in Figure 16, the stained sample is carried through a capillary by a laminar stream of sheath flow that confines the cells to the center of the flow in a straight file. A helium-neon laser is directed at a right angle to the stream. For fluorescence measurements an argon laser may be added. The absorbance/scattering at various angles and the fluorescence intensity at different wavelengths can all be simultaneously monitored at a rate of as many as 3×10^4 cells per second.[78]

This instrument has been applied to the measurement of many organisms: mammalian,[295,296] yeast,[297-302] and bacterial cells.[303-306] Fazel-Madjlessi and Bailey have shown that the morphology and biochemical content of bacterial walls change drastically during the course of a batch fermentation.[303-305] In addition, the effect of inoculum age, temperature, and aeration rate on bacterial growth rate and product formation has also been studied using this technique.[305]

Furthermore, Hatch et al.[307] and Cadman[308] have proposed to discriminate between the two species of *Corynebacterium glutamicum* and *Candida utilis* by using the fact that the ratio of light scatter to light absorbance is different for different species. With additional modifications, the flow fluorometer can be used in rapid cell sorting according to a prescribed criterion. Thus, flow microfluorometer has been useful in providing a detailed insight into cellular metabolism at the molecular level, but its popular use has been hampered by the high cost of the equipment.

D. Biological Measurements

Biological measurements detect the changes in contamination, mutation, morphology, and physiology. The Coulter® counter and cytometry measurements may reveal some of the morphological properties such as shape, size/size distribution, and age/age distribution. These measurements have been discussed previously, and they are quite critical in the formulation of a reliable mathematical model. For example, both the average size of cells in a yeast population and the difference between the mean sizes of daughter cells and mother cells were found to be highly dependent on specific growth rates.[309] The cell size and age, in turn, determine the macromolecular content and influence the rates of various metabolic activities.[301,309]

The ability to conduct strain selection has a profound economic effect on fermentation companies, and contamination can result in major economic losses. Very often, the determination of contamination or mutation is an extremely difficult task, even by manual means, and no single method is totally infallible. Direct observation through an optical microscope seldom gives conclusive indications of contamination, especially at early stages. Numerous tests involving tedious procedures and countless plate transfers may be required before mutation can be detected. The trend in the industry is to automate contamination detection and elimination procedures so that a rapid detection may enable operators to salvage a fermentor batch and minimize losses. One such technique used for antibiotic production is an automatic bacterial detection system in which CO_2 is monitored for the content of the labeled ^{14}C that is originally introduced in the substrate during inoculation.[310] The technique utilizes a flow cytometer to detect the presence of microbial contaminants.[311]

The use of the off-line determination of various compounds with a gas chromatograph has been proposed as a means for taxonomic identification and for contamination detection;[68] however, no report on the actual application of this idea has yet been published. Currently, there is a strong effort in the water and public health areas to automate rapid counting methods with a specificity for coliforms, especially fecal coliforms such as *Escherichia coli*.[312] Such methods have the potential of being applied to the fermentation field with slight modifications.

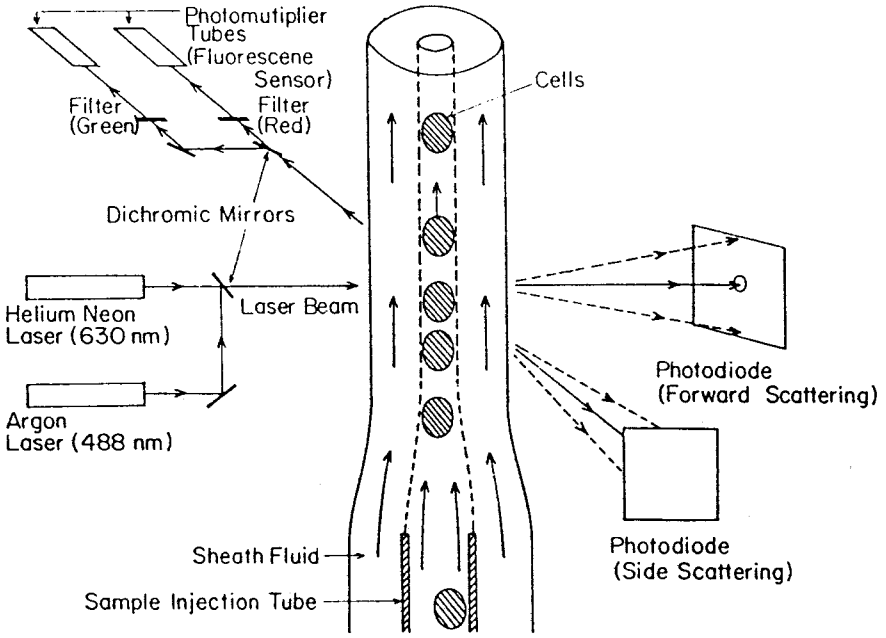


FIGURE 16. Schematic of cytofluorometry.

IV. STATE ESTIMATION AND PARAMETER IDENTIFICATION

Data collection for the sake of collection is, of course, totally fruitless. The data collected must be exploited through a set of manipulations to achieve some predetermined goals. However, most of the data collected cannot be put to use without some further processing. A formidable amount of data can be generated in a short period of time and can easily overwhelm everyone, including the brightest scientists and the computer storage of any size. The essential part of the collected data must be extracted and condensed through the application of some suitable algorithms prior to the actual use or storage. In addition, the data from several sources may be combined to achieve a set of more comprehensible units. Material balance around a fermentor, whose concept is shown in Figure 7, is extremely valuable in this respect. The condensed set of data is further compressed to yield a concise set of parameter values, such as specific biomass growth rate or cell yield, through parameter identification and state estimation algorithms. Under certain circumstances, some quantities that are not directly measured, such as biomass concentrations, can be estimated through the use of elemental balances.

Although many specialized sensors are presently under development for the continuous measurement of cell mass and substrate/product concentrations and some have even succeeded in being marketed commercially, generally, they have not yet been fully tested and their reliability is sometimes questionable. Since the state of instrumentation is still relatively poor, many measurements contain a significantly high level of noise. The rate of failure may be high and the signals may even be totally erroneous. Thus, raw measurement signals should be conditioned through the use of suitable filters before they are made available to control a fermentor.

A. Indirect Measurement

Although some of the data collected from the fermentor can be directly used, many have very little physical significance when standing alone. For example, temperature measurements can be directly fed back to actuate the controlling devices for the heater/cooler, pH

measurements can be used to activate the pumps for acid/base addition, or dissolved oxygen measurements can be used to control the gas flow rate or the impeller rotation speed. On the other hand, gas flow rate data are relatively uninformative unless they are combined with other measurements to form the so-called "gateway" sensors, originally coined by Humphrey.[313] Indirect measurements and gateway sensors provide information at an easily digestable mental level on cellular metabolism and fermentation conditions. The consumption rates or production rates of many components can be calculated by taking a simple component balance around the fermentor, as shown in Figure 7. Oxygen utilization rate (OUR), carbon dioxide evolution rate (CER), and respiratory quotient (RQ) are easily acquired on-line and are among the most frequently calculated quantities. They are very useful indicators of the cellular respiratory activities. Other indirect measurement variables include overall oxygen mass transfer rate, metabolic heat evolution rate, specific growth rate, cell yield, substrate utilization rate, and secondary metabolite production rate. A list of the calculated variables are given by Armiger and Humphrey,[7] and a set of straightforward step-by-step data analysis schemes are given by Nyiri.[19,20] Some of the more complicated schemes are discussed below. However, the authors wish to comment that the delay time for various instruments must be taken into consideration before different measurements are combined; otherwise, the indirect measurements may suffer unnecessarily from large errors.

The first example is the indirect measurement of specific growth rate. An indirect measurement of specific growth rate can be realized in a turbidostat system in which the cell concentration is tightly controlled between b_{max} and b_{min} by an on-off feeding of fresh nutrient solutions via a continuous optical density measurement.[80] In this simple on-off control, when the cell concentration reaches b_{max}, the feed pump is actuated to dilute the culture until b_{min} is reached. The specific growth rate is calculated semicontinuously by combining the on-line measurements of the time interval between successive nutrient addition and the optical density. A typical saw teeth signal curve generated from this on-off control scheme is shown in Figure 17. This method is applicable to both fed-batch and continuous bioreactors, although some careful material balance analysis may be required for the latter case.

A similar procedure can be applied to a pH control scheme to determine the specific product formation rate if the product possesses acidic or basic properties. The product concentration required for the calculation of specific product formation rate can either be measured directly by one of the methods discussed earlier or be estimated through a careful monitoring of the cumulative amount of acid/base added to achieve neutralization. In the latter case, other processes such as the neutralization of produced carbon dioxide or the protons released when ammonia is taken up by the cells from the ammonium salts of the medium must be carefully accounted for. Acetic acid production by *E. coli* growing in glucose has been studied this way in the authors' laboratory.[314] The same technique has been applied to gluconic acid production as well.[80]

The second indirect measurement to be discussed is the oxygen mass transfer coefficient. An adequate supply of oxygen is often vital to the growth and maintenance of many microorganisms, and the availability of oxygen in a fermentor depends on the volumetric oxygen mass transfer coefficient, k_la, as described by the following equation:

$$q_{O_2}(t) = (k_la)(t)[C^*_{O_2}(t) - C_{O_2}(t)] \quad (6)$$

where $q_{o_2}(t)$ is the volumetric oxygen transfer rate, $C^*_{o_2}(t)$ is the liquid phase oxygen concentration in equilibrium with the gas phase, and $C_{o_2}(t)$ is the liquid phase oxygen concentration.

In the above equation, it is important to note that k_la is a function of time in order to account for changes in fermentor conditions such as agitation, air dispersion, and rheological properties. Because the properties of the fermentation broth are constantly changing due to biomass synthesis, product formation, substrate depletion, and antifoam addition, the esti-

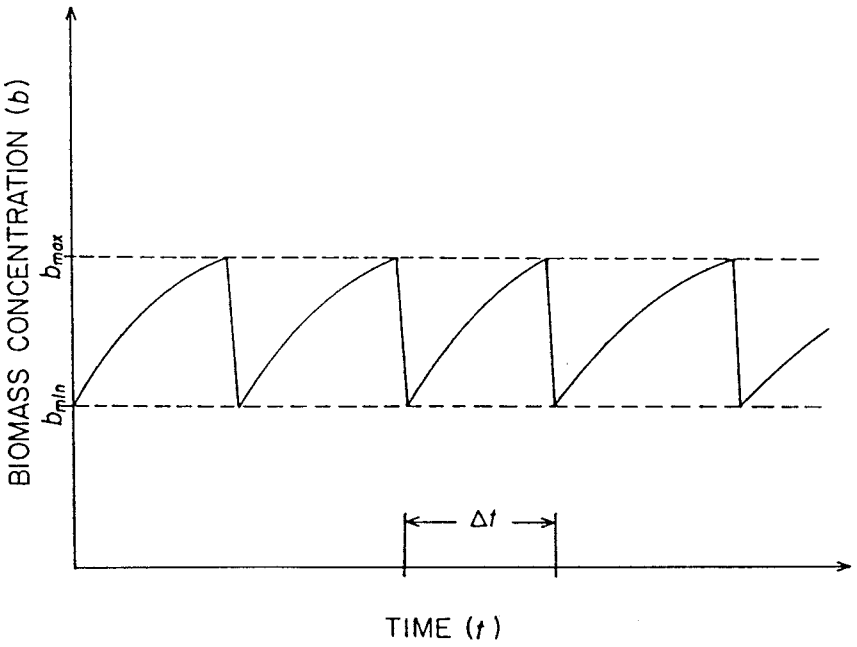


FIGURE 17. Use of on-off turbidostat to calculate the specific growth rate.

mate of oxygen transfer coefficient in the fermentor should be constantly updated, i.e., monitored on-line, by combining the measurements on gas flow rate, gas phase oxygen concentration, and dissolved oxygen concentration.

There are two general methods for the estimation of k_1a: the static (balancing) method and the dynamic (gassing-out) method. Both methods assume that the dynamics of oxygen is much faster than the dynamics of liquid phase-bound species such as biomass, substrate and products. This assumption enables one to employ quasi-steady-state approximation for the liquid phase. Furthermore, both methods assume that the fermentor is well mixed, and k_1a is treated as a function of time but not space. In the first method, $q_{o_2}(t)$ is calculated from the gas flow rate and the oxygen concentration in the inlet and exit stream. By inferring $C_{o_2}(t)$ from Henry's law and by directly measuring $C_{o_2}(t)$, one can calculate k_1a continuously.[154,315] These measurements have been coupled to a computer to yield an on-line estimate of k_1a.[157] After a sensitivity analysis, the precision of this method was found to be limited by the precision of the oxygen analyzer.[157] In the second method, the transient response of the dissolved oxygen is continuously monitored after the air supply is interrupted for a short period of time.[114,316-318] This method is well recognized,[133] but it requires the deliberate disturbance of the system and generally yields only one single estimate for k_1a. The on-line estimation of k_1a by this second technique was attempted by Yoshida et al.[151] He was able to generate k_1a continuously and showed that a good convergence from the erroneous initial guess to the true value was possible if an appropriate algorithm was used. A more elaborate algorithm that took into account the inertia of the measuring instruments was also published.[319]

Thus, by utilizing the existing relationships between the measurements, the number of observable variables can be greatly increased for the same number of raw measurements. The gateway sensor concept is important in deciding what measurements to select so that biologically important information can be extracted. A simple examination of the necessary instrumentation for realizing the interested indirect measurements can point out the priority in sensor development.

B. Macroscopic Material/Energy Balances

The principles of macroscopic material balance and energy balances have long been known in chemical engineering. Their applications to fermentation fields have been summarized by Roels.[320,321] The use of on-line material/energy balance is considered a great step forward in fermentation technology. Because of the availability of continuous gas analyzers for oxygen and carbon dioxide, these material balancing methods depend heavily on the measurement of gas exchange conditions in a fermentor. The on-line calculation of the associated parameters of gas exchange is not new.[322] Since the pioneering work by Cooney et al.,[323] Wang et al.,[324] Zabriskie et al.,[325] and Zabriskie and Humphrey,[326] many studies based on elemental balances have been published.[327-338]

There are two variations of the material balance approach. The first method, adopted by Cooney et al.[323] and Wang et al.,[324] is based on the concept of conservation of mass and chemical reaction stoichiometry. The basic feature of the method is to represent the biological conversion of substrate to cell mass and metabolic product by a chemical reaction as follows:

$$a\ C_xH_yO_z + b\ O_2 + c\ NH_3 \rightarrow B\ C_\alpha H_\beta O_\gamma N_\delta + d\ H_2O + e\ CO_2 + f\ C_{\alpha'}H_{\beta'}O_{\gamma'}N_{\delta'} \quad (7)$$

where $C_xH_yO_z$, $C_\alpha H_\beta O_\gamma N_\delta$, and $C_{\alpha'}H_{\beta'}O_{\gamma'}N_{\delta'}$ are the chemical formulas for the substrate, cell biomass, and product, respectively. Usually, it is only the ratio, not the absolute value, of the composition of the cell biomass that can be determined; therefore, α can be set equal to 1 without any loss of generality.

For the moment, all the chemical formulas are assumed known and constant, although there are reports that the chemical composition of cells may be affected by drastic changes in growth rates and by the nature and composition of the medium.[339,340] Herein lies the disadvantage of this method. For example, a 10% change in the cell nitrogen content during the course of the fermentation forced the on-line estimation of cell concentration to deviate from the true values in Wang et al.'s study.[324]

Note that a, b, c, d, e, and f are the six unknown stoichiometric coefficients for the reaction, and the stoichiometric coefficient for biomass can be normalized to 1. Although Cooney et al.[323] have used two separate chemical reaction equations, one for biomass synthesis and one for product formation, the combined single-equation approach is essentially equivalent to their approach, and it is algebraically slightly simpler. The principle of elemental balances gives four equations, one for each of the elements C, H, O, and N.

$$\text{C:} \quad xa = \alpha + e + \alpha' f \quad (8)$$

$$\text{H:} \quad ya + 3c = \beta + 2d + \beta' f \quad (9)$$

$$\text{O:} \quad za + 2b = \gamma + d + 2e + \gamma' f \quad (10)$$

$$\text{N:} \quad c = \delta + \delta' f \quad (11)$$

A quasi-steady-state approximation applied to the conservation equations for O_2 and CO_2 gives a fifth equation needed to solve for the six unknowns:

$$R = \frac{Y_{O_2}}{V} OUR = \frac{Y_{CO_2}}{V} CER \quad (12)$$

where R is the total rate of growth, Y_{o_2} and Y_{co_2} are the yields of biomass with respect to oxygen and carbon dioxide, respectively, and V is the volume of the culture reactor (in liters). The quasi-steady-state approximation for O_2 and CO_2 is valid for all three modes of operation (batch, fed-batch, and continuous) with respect to the liquid medium. The reason is that the time constants of the dynamic equations for O_2 and CO_2, which are of the order

of the broth volume divided by the volumetric gas flow rate, are much smaller compared to that of the dynamic equations for biomass and substrate concentrations, which are of the order of the broth volume divided by the volumetric medium flow rate. Equation 12 relates the coefficients b and e in Equation 7, since the yield of biomass with respect to the component under consideration is essentially the reciprocal of the corresponding stoichiometric coefficient properly adjusted for the differences in molecular weights.

There are only five unknowns if we do not consider product formation; the five independent equations presented heretofore can be used to solve fully the five unknowns. From these stoichiometric coefficients, all the yield relationships can be obtained. Thus, this material balance method does not require the assumption of constant cellular yield coefficients. As a matter of fact, no kinetic models are used at all.

On the other hand, one more unknown must be determined if single product formation is considered. An additional relationship is needed in order to solve for the last unknown. Among the possible measurements are nitrogen source, substrate, product, and fermentation heat. However, a word of caution on the possibility of the use of direct substrate or product measurements should be mentioned here. Although these are direct measurements of the states of a biochemical reactor, they cannot be used directly to solve for the stoichiometric coefficients in Equation 7 if their time rate of change cannot be simultaneously monitored with a certain degree of accuracy. A simple difference between two consecutive measurements cannot be used to calculate the time rate of change because of the presence of noise inherent in any real measurement. Thus, in this case, an estimate on the time rate of change of those states that are directly measured also needs to be obtained. The use of the heat of fermentation, Q, together with an enthalpy balance was considered by the authors as another possible source of independent equation:

$$- a H_c - b H_{O_2} - c H_{NH_3} + H_b + d H_{H_2O} + e H_{CO_2} + f H_p + Q = 0 \quad (13)$$

However, a sensitivity analysis has revealed that, because of the close relationship between the heat evolution and oxygen utilization,[341-343] Equations 8 to 11 and 13 form a nearly singular set of linear equations in most cases. Singularity problems can also occur if the degree of reductance of the substrate and the degree of reductance of the product or biomass are the same or similar. One such example is the conversion of glucose ($C_6H_{12}O_6$) to acetic acid ($C_2H_4O_2$) by *E. coli*.

The second method of indirect estimation of biomass concentration and growth rate through on-line material balance was advanced by Zabriskie et al.[325,345] and Zabriskie and Humphrey.[326] This approach is based on the material balance of only one chemical component. With the help of a mathematical kinetic model which relates the formation/utilization of the chemical component under consideration to biomass growth, biomass and/or substrate concentrations can be continuously estimated. Clearly, the accuracy of the estimates obtained by this method depends on the validity of the mathematical model employed.

In the following discussion, the example of Zabriskie and Humphrey[326] will be used to contrast the two approaches. Because oxygen can be continuously measured both in the gaseous and aqueous phases and because it is closely related to the amount of energy produced or consumed by cell biosynthesis and catabolism, it represents a component very well suited for this purpose. As in the first method, the quasi-steady-state assumption is applied to oxygen in order to calculate its consumption rate (OUR). The biomass concentration, b, is estimated in real time by integrating the following limiting substrate yield and maintenance model of Pirt.[117]

$$OUR = \frac{1}{Y_{O_2}} \frac{db}{dt} + m_{O_2} b \quad (14a)$$

After an estimate of biomass has been obtained by integrating the above first-order linear differential equation, an estimate of the specific growth rate can be obtained by writing Equation 14a as:

$$\mu = \frac{1}{b}\frac{db}{dt} = \frac{Y_{O_2}}{b} OUR - m\, Y_{O_2} \qquad (14b)$$

The disadvantage of this method is that instead of assuming a known constant biomass composition used in the first method, a knowledge of the constants Y_{o_2} and m_{o_2} is assumed. The values of these parameters are determined with off-line biomass sampling in the previous experiments carried out preferably under similar conditions as the real-time run. A criticism to this approach is that the accuracies of the biomass and specific growth rate estimates depend heavily on the reproducibility of the experimental conditions so that the same Y_{o_2} and m_{o_2} can be obtained. Thus, the ability to duplicate the previous experiments basically determines the success of this method.

Zabriskie and Humphrey[326] applied this algorithm to batch cultures of *Thermoactinomyces* sp., *Streptomyces* sp., and *Saccharomyces cerevisiae*. The on-line correlations for the metabolically simple *Thermoactinomyces* and *Streptomyces* were successfully accomplished, but the metabolically more complex *S. cerevisiae* assimilated glucose through different pathways during the course of fermentation. Because of the shift in the metabolic behavior (e.g., the production of ethanol and acetate and the corresponding diauxic growth), the values of Y_{o_2} and m_{o_2} were not truly constant but varied, and a correction factor based on a careful mathematical analysis of the representative pathways was needed to obtain good agreement between the estimate and the actual biomass data. The correction factor was expressed as a function of OUR and CER (or RQ) whose functional form was derived from a reference sequence of known yeast metabolic energy producing pathways such as the Embdem-Meyerhof-Parnas (EMP) pathways and the tricarboxylic acid (TCA) cycle.

Another variation to the material and energy balances is the concept of the degree of reductance and that of an electron balance.[346] The latter is actually the result of a linear combination of the elemental balance Equations 8 to 11. In this approach, energy requirement is divided into three parts: growth, maintenance, and product formation. Some important regularities such as 27 kcal/g-equiv oxygen utilized and 4.291 for the degree of reductance of biomass are identified.[342,343] The regularity between heat evolved and oxygen uptake means that the heat evolution rate can be accurately correlated to oxygen measurements. At the same time, it implies that a heat balance may not be very useful as an additional independent equation in providing more information to indirect measurement due to the singularity problems encountered in solving linear algebraic Equations 8 to 11 and 13, as noted previously. If an energy balance is used in an attempt to estimate more stoichiometric coefficients, the resulting estimates will be very sensitive to measurement errors if such heat-oxygen regularity holds.

The first method of macroscopic material balance was also applied to a batch glutamic acid fermentation with *Brevibacterium flavum* by Constantinides and Shao.[331] The substrate (glucose) concentration was chosen as the additional measurement besides OUR and CER; however, a time lag of one half hour in the on-line estimate caused by the off-line glucose analysis was unavoidable due to the lack of an on-line enzymatic glucose analyzer at the time. The same method was also used by Swartz and Cooney[328] to monitor the growth of *Hansenula polymorpha* on methanol in a continuous fermentor. As before, an additional relationship was needed because of the presence of a product. A constant heat-to-oxygen ratio was used to satisfy this requirement because the heat measurement was unreliable. Control action was implemented to avoid methanol accumulation through an on-line adjustment of dilution rate when the available oxygen level was reduced. However, their use of a constant heat-to-oxygen ratio to circumvent the sensitivity problems encountered in

calculating the stoichiometric coefficients is, in the authors' opinion, questionable. This is because the constant heat-to-oxygen ratio is actually a manifestation of the singularity problems arising in solving simultaneously the set of elemental balance equations of Equations 8 to 11 and the energy balance equation.

A variation of the second method of macroscopic material balance is to relate CER to the specific growth rate (μ), and the substrate (glucose) consumption to μ through a semi-empirical yield and maintenance model.[330] The necessary information on the true growth yield and the maintenance coefficients are evaluated from previous experiments. The on-line data on CER are used to calculate μ, which, in turn, is used to calculate the substrate consumption rate. Given appropriate initial conditions, biomass and substrate concentrations could be estimated.

A combination of the previously described two methods is also possible. In a study of penicillin fermentation by Cooney and Mou,[333] instead of oxygen, carbon dioxide was used as the component for balancing because CER was found to be indicative of the cell growth rate. The calculation of biomass based on CER and the experimentally determined carbon dioxide yield coefficient worked well during the fast growth phase of a penicillin fermentation, but it failed during the transition phase and the antibiotic production phase. During the slow growth phase, a carbon balance was employed, with the assumption of a complete recovery of the carbon content in the nutrient as biomass, penicillin, and CO_2. Thus, the second method was used during the beginning of the fed-batch fermentation when yield coefficient stayed relatively constant; the first method was switched on when maintenance and endogenous metabolism caused the yield coefficient to vary. Biomass concentrations and instantaneous specific growth rates could be calculated, and an experimentally determined specific product formation rate was used to approximate the penicillin level. A feedback control strategy was outlined in which the glucose feed rate was manipulated to maximize the specific growth rate in order to accumulate cells in the beginning of the fermentation.[335] Feedback information was used to prevent glucose overfeeding which would result in side product formation and repress antibiotic synthesis in the next stage. As the penicillin production phase was reached, a slower growth rate was selected and maintained at a constant level by the control action, which at the same time also attempted to depress the glucose level to avoid catabolic repression. The preprogrammed desired cell growth path could be duplicated by the on-line feedback control, even when complex medium was utilized.[336] Finally, these balancing methods were also applied to model the penicillin fermentation in terms of a set of unstructured dynamic equations for biomass, substrate, and product by Heijnen et al.[347]

C. On-Line Estimation

Well-organized estimation studies in the biochemical engineering field have been quite limited in the past. An efficient estimation-filtering algorithm is especially indispensable in the face of the lack of reliable sensors for the direct observation of the important biological and physiochemical parameters needed for control and regulation. A good estimation scheme will continue to contribute significantly in a wide variety of processes even as sensor development progresses. For example, a biomass sensor developed in the future may be useless in measuring the state of immobilized cells or those inside hollow fibers. Any fermentation substrate or product sensors to be developed will necessarily be specific, and they are useless in other substrates for which they are not designed; nor will they be useful in an undefined medium.

Because the variety of currently available continuous measurements is rather limited, no information contained in them should be wasted. For example, a pH measurement can be used not only to activate the pH controller, but also to estimate a few other important bioreactor parameters such as the rate of formation of a metabolic product with acidic/basic

properties and the gas mass transfer rate from the analysis of the effect of dissolved carbon dioxide. Similarly, a calorimetric measurement provides other information in addition to the cooling needs of a bioreactor. The limited assortment of measurements must be employed and interpreted, together with all other information available on the bioreactor, in order to estimate accurately as many important variables as possible. To this end, indirect measurement and material balancing methods prove to be extremely valuable. Thus, a sophisticated parameter estimation package is becoming increasingly important, especially as the number of parameters and variables expands as a result of a deeper understanding of the biological behavior.

Both balancing methods described in the previous section suffer from the inaccuracies of the available instruments. The error in the primary measurement is often large; its magnitude may be as large as over 100% of the measurement itself.[326] These errors can have profound effects on the accuracy of the estimates obtained from these measurements. In one case, for example, the propagation of measurement errors compounded the deviation of the on-line biomass estimates from the off-line assay value as fermentation progressed.[331] Noisy oxygen measurements were cited as one of the causes for the need of biomass concentration reinitialization in the midst of fermentation as the deviation became unacceptably large.[324] Furthermore, these balancing methods must be supplied with initial conditions which are, at best, rough guesses. Despite the claim by Zabriskie and Humphrey[326] that the sensitivity of biomass estimates to the initial guess rapidly diminished in a batch fermentation, this is not generally true as shown by Stephanopoulos and San through statistical variance analysis.[348]

In material balancing techniques, the biomass concentration in a batch bioreactor is estimated by integrating either Equation 14 or the following equation:

$$\frac{db(t)}{dt} = R(t); \qquad b(0) = b_0 \tag{15}$$

However, the presence of errors in the various measurements produces rather noisy values for R(t) which, in turn, give rise to large fluctuations in the estimate of b(t) obtained from the integration of Equation 15. A useful analogy to the above procedure of estimating b(t) from the measured values of R(t) exists here: given the initial position of an object and the measurements of its velocity at various time instants, estimate the position of the object as a function of time. Because the initial conditions are not accurately known and the measurements are corrupted by noises in actuality, Equation 15 should be written as:

$$\frac{db(t)}{dt} = R(t) + \zeta(t); \qquad b(0) = b_0 + \zeta_0 \tag{16}$$

where, for the sake of simplicity, $\zeta(t)$ can be considered as a Gaussian white noise with zero mean and intensity σ^2, and, similarly, ζ_0 the uncertainty involved in the guess of b_0 with a variance of σ_0^2.

It can be shown that b(t) will be a normally distributed random variable characterized by the following expected values and variances:[348]

$$E[b] = \int_0^t R(\tau)d\tau \tag{17}$$

$$Var[b] = \sigma_0^2 + \sigma^2 t \tag{18}$$

Fed-batch and continuous fermentors can also be analyzed in a similar manner. The variance of the biomass concentration estimates increases with time for a batch or fed-batch bioreactor, whereas it approaches a limiting value of $\sigma^2/2D$ for a continuous fermentor, which is still quite high. Thus, it is highly desirable to employ a good noise filtration algorithm to boost the reliability of the estimated values before they are used for modeling or control purposes.

The use of moving average to reduce the noise at the instrument level has always been very common due to its simplicity.[322] Measurements of oxygen concentration in the exit gas, for example, are routinely scanned at a higher rate then averaged for a certain number of times before being used. However, the averaging methods are poorly suited to the estimation of derivative quantities such as the specific growth rate, substrate uptake rate, and product formation rate. Although a simple average can reduce the noise level by a factor of $\sqrt{n}$, where n is the number of measurements averaged, there is a practical limit to this number due to the considerations of time lag and time-varying nature of the quantities involved. No consideration to optimality is taken by the averaging method; nor is the estimation carried out for various variables which are not directly measured.

Since the operating conditions of most fermentors are strictly due to the sensitivity of microbial cultures to the surrounding environment, the control of the environment with a small degree of allowable excursion requires a more sophisticated estimation technique than a simple averaging to yield estimates that are almost noise-free. In a total absence of models, a recursive least-square filter may be used which yields estimates for the coefficients of the power series employed for the description of the independent variables. Thus, it is well suited for the smoothing of a series of raw measurements. This technique was used to smooth the noisy biomass optical density measurements by Jefferis et al.[349] They expressed the biomass concentration as a second-order polynomial function of time.

$$b = a_0 + a_1 t + a_2 t^2 \tag{19}$$

The object of the whole scheme was to estimate the coefficients a_0, a_1, and a_2 while the discrete measurements were weighed in such a way that the contribution of each measurement decreased exponentially with time. After the coefficients were estimated, the growth rate was simply calculated from the following equation:

$$R = \frac{db}{dt} = a_1 + 2a_2 t \tag{20}$$

A similar algorithm was applied to other raw measurements such as the weight of nutrient reservoir and flow rate.[73] The order of polynomial can be increased when sufficient data are available.[350] It is also possible to interlock more than one state. A kinetic model may be incorporated into the least-square scheme to obtain a more rapid response,[349] or the least-square scheme may be combined with a Kalman filter to reach better estimates for certain variables when some empirical models can be applied to the raw measurement.[73]

In yet another algorithm by Reuβ et al.,[351] the nonlinear model equations were quasi-linearized and the unknown parameters were each assigned a dynamic equation which was set to zero. Thus, the parameters were treated as constants. In one on-line batch experiment,[351] oxygen uptake rate was used to predict the biomass concentration, which was then compared to the intermittent automated optical density measurements. The deviation of the calculated values from the measurements was, in turn, used to iterate on the parameters (yield coefficient and maintenance coefficient) until the sum of the squares of the deviation was minimized. In another on-line experiment involving a column reactor,[352] a distributed model was used to predict the dissolved oxygen concentration, which varied along the reactor. The dissolved oxygen profiles were measured, and a similar quasi-linearization method was employed to estimate the unknown parameters, in this case, the Stanton number and the Peclet number.

Some soundly based and yet simple approaches, such as the material balance around a fermentor, as shown in Figure 7, are available for certain fermentation state variables. In such cases, a Kalman filter is very useful. Since its inception in the early 1960s, Kalman filters have been applied to chemical reactors.[353,354] The extended (linearized) Kalman filter was applied to bioreactor state estimation and kinetic model parameter identification by

Svrcek et al.[355] They assumed that cell or substrate concentration measurements were available and treated the substrate-to-biomass yield coefficient to be constant and exactly known, which is not very likely in real situations. In their simulation, all the initial conditions used to commense estimation coincided with actual values. The scheme was essentially an open-loop integration, and there was no indication that a wrong guess in the initial condition would eventually converge to the true value, especially if a wrong substrate-to-biomass yield coefficient was guessed initially.

An adaptive Kalman-Bucy filter supplied with information readily obtainable from the on-line material balance was proposed by Stephanopoulos and San.[332,348,356,357] The essence of the Kalman filter is briefly summarized below. For a more complete discussion, the reader is referred to the work of Stephanopoulos and San.[332,348,356,357]

The dynamics of a nonlinear system and the measurements can be expressed in the general form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u}) + \zeta(t) \tag{21}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \xi(t) \tag{22}$$

where $\mathbf{x}$ is the state vector of a dynamic system, $\mathbf{u}$ is the nonrandom input vector, $\zeta(t)$ is the random disturbance, and $\xi(t)$ is the random noise in the measurement $\mathbf{y}$. The linearized Kalman estimate, $\hat{\mathbf{x}}$ of the true state $\mathbf{x}$ is described by the following set of vector and matrix filtering differential equations:

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{f}(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{P}[\mathbf{h}_x(\hat{\mathbf{x}})]^T\mathbf{S}^{-1}[\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}})] \tag{23}$$

$$\frac{d\mathbf{P}}{dt} = [\mathbf{f}_x(\hat{\mathbf{x}},\mathbf{u})]\mathbf{P} + \mathbf{P}[\mathbf{f}_x(\hat{\mathbf{x}},\mathbf{u})]^T + \mathbf{Q} - \mathbf{P}[\mathbf{h}_x(\hat{\mathbf{x}})]^T\mathbf{S}^{-1}[\mathbf{h}_x(\hat{\mathbf{x}})]\mathbf{P} \tag{24}$$

where $\mathbf{P}$ is the symmetric covariance matrix of the estimation error, and $\mathbf{Q}$ and $\mathbf{S}$ are the positive definite matrices which are measures of the intensities of the noise processes ζ and ξ, respectively.

In this scheme, unknown parameters such as the specific growth rate and cell yields may also be estimated simultaneously by treating them as additional "state variables" and setting their derivatives equal to zero. For example, when two state variables, namely, biomass and substrate concentrations, are of interest in a continuous biochemical reactor, we may express the corresponding kinetic equations as:

$$\frac{db}{dt} = -Db + \mu b \tag{25}$$

$$\frac{ds}{dt} = D(s_f - s) - \frac{\mu b}{Y_s} \tag{26}$$

The parameters μ and Y_s contained in the above equations are not known, and can be estimated alongside with b and s by setting up superficial dynamic equations for them.

$$\frac{d\mu}{dt} = \zeta_1 \tag{27}$$

$$\frac{dY_s}{dt} = \zeta_2 \tag{28}$$

The performance of the adaptive Kalman filter can be improved significantly by treating ζ_1 as colored noise and by implementing the following equations:

$$\frac{d\mu}{dt} = c + \zeta_1 \tag{29}$$

$$\frac{dc}{dt} = -\frac{c}{\tau} + \zeta_1 \tag{30}$$

In the above equations, c is the colored noise correction variable, and τ is the time constant associated with c, which is usually of the same order of magnitude as the residence time constant. In other situations where Y_s is suspected of being highly variant, we can couple another superfluous "state" to Y_s as is done for μ in Equations 29 and 30.

From the material balance, R and Y_s can be indirectly measured. Thus, the measurement vector **y** is composed of:

$$y_1 = R = h_1(b, s, \mu, c, Y_s) = \mu b \tag{31}$$

$$y_2 = Y_s = h_2(b, s, \mu, c, Y_s) = Y_s \tag{32}$$

A notable feature of this approach is the adaptive detection of the deviation of the predicted values from the measurement. The intensities of the process noise ζ_1 and ζ_2 in the variance-covariance matrix **Q** are adjusted on-line, depending on the magnitude of the deviation.[332] The estimation can be further improved, at the cost of a longer computation time, by smoothing[358,359] or by relinearization for severely nonlinear systems.[360]

This general scheme can be applied to a wide variety of situations to estimate not only the states but also the associated kinetic parameters on-line continuously as long as they are observable. This scheme is especially useful for estimating highly noisy derivative variables such as the specific growth rate. Computer simulations have shown close agreement between the estimate and the true values,[332,348,356,357] and wrong initial guesses also converged quickly. Its application to an actual fermentor was equally successful.[361] Using the data that are readily available from applying material balances to the exit gas measurements, the authors have tried a similar technique to estimate the substrate yield coefficient and the maintenance coefficient simultaneously with very promising results. Finally, the Kalman filter technique has also been applied to waste treatment.[362,363]

D. Consistency Checks and Measurement Error Detection

The discussion on estimation, thus far, applies to situations in which the number of measurements is just sufficient to determine a set of variables. The system is overdetermined when more independent measurements are taken than absolutely required for the full identification of the system by the application of the available constraints. Because of the sensor inadequacy, highly accurate and reliable measurements on a fermentation process are difficult to obtain and are frequently limited to the ones absolutely necessary for the estimation of the bioreactor state. However, in some cases there may actually exist a redundancy in instrumentation when well-founded relationships such as material/energy balances can be applied to the measurement data. This redundancy is highly desirable and should be exploited whenever technically possible, especially in the fermentation field where the accuracy of sensors is extremely critical but also rather questionable. Sensor redundancy can be exploited in many ways. For example, suitable analytic methods can be applied to the redundant measurements to check the data consistency, to detect the presence of gross errors, to locate the source of gross errors, to minimize the noise level, to estimate the level of confidence in the experimental results, and to detect changes in models and model parameters.

Because of the various types of errors that may originate in random or systematic sources, the obtainable independent measurements rarely form a consistent set of data in the sense of satisfying the material/energy balances or other model constraints imposed on them. These errors may be significantly large, or, in some cases, the instrument may fail completely. In such cases, the biased data will lead to false conclusions and inaccurate fermentation models for microbial growth kinetics. Any optimization based on these unreliable models will certainly be invalid. Furthermore, process control functions will possibly be rendered inoperative if grossly erroneous measurements are used for feedback control, resulting in unstable or unproductive operations and economic losses.

In a long series of publications, Erickson and co-workers[342,364-370] have persistently advocated that fermentation data should be checked for consistency by using macroscopic elemental balances combined with the regularities of oxygen-heat correlation and the degree of reductance of biomass. Consistency tests were applied to literature data from various fermentation processes, ranging from single-cell protein production[342] to extracellular product formation,[365,366,369] from batch culture[368,370] to continuous culture,[364] and in a diverse collection of media containing hydrocarbon substrates[367] or nitrate as the nitrogen source.[369] However, the above analyses did not consider the variation of errors contained in the measurements, and no criteria were given for assessing the reliability of the measurements. Likewise, when parameter estimation was attempted, different estimates were obtained for the same parameter, depending on the combination of the data set.

A single consistent set of estimates that conform to the macroscopic elemental balance equations can be obtained by analyzing the raw experimental data with a scheme that incorporates the concepts from statistics and optimization. The presence of random errors was recognized by de Kwaadsteniet et al.[371] in their calculation of Y_{ATP}^{max}, m_{ATP} and the 95% confidence intervals. The maximum likelihood principle was employed by de Kok and Roels,[372] Geurts et al.,[373] and Dekkers et al.[340] to condition the raw data before indirect measurement variables such as RQ, Y_{ATP}^{max}, and P/O were calculated from the adjusted, consistent data. A slightly different approach that also used the maximum likelihood principle was taken by Solomon et al.[374] to estimate these variables in one step.

Because random errors in the measurements are unavoidable, a certain degree of deviation in the consistency test is expected and is perfectly acceptable. To see just how large a deviation can be before the data are rejected, a statistical hypothesis testing is carried out. Madron and co-workers[375,376] introduced a powerful, yet simple, multidimensional chi square (χ^2) test to indicate the level of inconsistency. Finally, Wang and Stephanopoulos[66] have employed an on-line method to detect the presence of gross measurement errors, to identify the source of these errors if they are, indeed, found to exist, to adjust the inconsistent set of fermentation data according to the maximum likelihood principle, and to rectify the erroneous data in an optimal manner. This method combines the use of macroscopic material/energy balances, optimal minimization techniques, statistical hypothesis testing methods, and a highly efficient algorithm that was originally proposed by Romagnoli and Stephanopoulos.[377] The method contains many computational improvements to facilitate its real-time data processing. These improvements permit sequential processing of the measured data and greatly minimize the effort in matrix computation. Such capabilities are especially valuable when the dimension of the problem is large, and they may prove to be critical in the actual on-line implementation of the combined methodology of error detection and data analysis.

In Wang and Stephanopoulos' scheme, the balances that must be satisfied by the directly or indirectly measured variables are written in the compact matrix form:

$$\mathbf{A}\mathbf{y} = \mathbf{O} \tag{33}$$

where y is the vector of the measured variables. For example, OUR, CER, substrate consumption rates, biomass synthesis rate, and product formation rate can all sometimes be measured. In this case the stoichiometric coefficients b, e, a, B, and f can be determined by employing balances similar to that of Equation 12. These coefficients must still satisfy the elemental balances of Equations 8 to 11 and enthalpy balance of Equation 13. Therefore, in the formulation of Equation 33, the stoichiometric coefficients will be the components of vector y and Equations 8 to 11 and 13 the row of matrix A for this example.

The above equation represents a set of m linearly independent equality constraints applicable to a set of n measured variables y with $1 \leq m < n$. Because the vector of measured values contains random measurement errors δ the true value for the measured variable is

$$\mathbf{y} = \bar{\mathbf{y}} + \delta \tag{34}$$

where, for simplicity, δ can be assumed to be normally distributed with a zero mean and with a variance-covariance matrix ψ. The presence of δ in the measurements will force Equation 33 not to be exactly satisfied but will produce a residual ϵ in the equality constraint.

$$\epsilon = -\mathbf{A}\bar{\mathbf{y}} \tag{35}$$

Since the number of equality constraints is less than the number of unknowns, δ cannot be estimated by solving Equation 33 directly; however, a maximum likelihood estimate of δ can be obtained by minimizing the sum of the weighted error squares subject to the equality constraint of Equation 33:

$$\delta = \psi \mathbf{A}^{\mathrm{T}} \varphi^{-1} \epsilon \tag{36}$$

where

$$\varphi = \mathbf{A}\psi\mathbf{A}^{\mathrm{T}} \tag{37}$$

Based on statistical theories, the following test function, h, was formulated to test whether the measurement vector y contains grossly biased errors such that the residual in the equality constraint is significantly large:

$$h = \epsilon^{\mathrm{T}} \varphi^{-1} \epsilon \tag{38}$$

It can be shown that h is characterized by a χ^2 distribution with m degrees of freedom. Thus, the presence of gross errors can be detected with any desired level of confidence of $1 - \vartheta$ in a straightforward manner by comparing h to $\chi^2_{1-\vartheta}(m)$. When the test is failed, the source of the gross error can be easily identified by the use of a serial elimination algorithm in which one measurement at a time is deleted to form new test functions. When a significant drop in the value of the test function is observed, the corresponding deleted measurement is identified as suspect. The computational effort in the serial elimination algorithm is further greatly reduced by the introduction of a useful matrix inversion lemma. Another advantage of Wang and Stephanopoulos' approach is that gross errors can be eliminated before Equation 36 is finally used to yield a consistent set of optimal estimates for the measurements. The error detection and identification potential of the scheme was demonstrated in the analysis of a number of literature data. The scheme was able to identify the same source of gross errors as suspected by the original investigators, except that, instead of relying on the past experiences or comparing to other sets of data, the scheme was able to yield the correct judgement systematically and consistently. The scheme's capability of systematic detection and rectification of gross measurement errors is especially valuable in the automated data collection, analysis, and computerized control of a bioreactor.

E. Estimation of Basic Biochemical Parameters

A macroscopic material balance involving Reaction 7 is not only useful in estimating the biomass concentration or the specific growth rate. It can also be used for the estimation of some other fundamental parameters involved in the metabolic pathways, thus, elucidating a given biological process. The decomposition of the overall Reaction 7 can be carried out in various ways. For example, it can be decomposed into two separate reactions, one for biomass synthesis and one for product formation.[323] A separate reaction for the respiratory metabolism may be added.[378] ATP generation by direct substrate oxidation and oxidative phosphorylation can be considered,[340] and a reaction for the diauxic growth on the metabolic product after the exhaustion of the primary substrate is also possible.

For example, after an analysis of the dominant metabolic pathways, the growth of yeast cells on glucose ($C_6H_{12}O_6$) with the production of ethanol ($C_2H_6O_6$) can be expressed as the following set of simultaneous reactions:[340,379]

$$\text{TCA/glycolysis} \quad \xi_1\text{: } C_6H_{12}O_6 + 6\,H_2O \rightarrow 6\,CO_2 + 12\,NADH_2 + 4\,ATP \tag{39a}$$

$$\text{Respiratory chain} \quad \xi_2\text{: } 2\,NADH_2 + O_2 \rightarrow 2\,H_2O + 2\,P/O\,ATP \tag{39b}$$

$$\text{Ethanol formation} \quad \xi_3\text{: } C_6H_{12}O_6 \rightarrow 2\,C_2H_6O + 2\,CO_2 + 2\,ATP \tag{39c}$$

$$\text{Biomass synthesis} \quad \xi_4\text{: } \frac{1}{6}C_6H_{12}O_6 + \delta\,NH_3 + \rho\,NADH_2 + Y_{ATP}\,ATP$$

$$\rightarrow CH_\beta O_\gamma N_\delta + (1-\gamma)H_2O \left(\text{where } \rho = \frac{\beta - 2\gamma - 3\delta}{2}\right) \tag{39d}$$

$$\text{Cell maintenance} \quad \xi_5 = m_{ATP}\text{: } ATP \rightarrow ADP \tag{39e}$$

where ξ_is are the extents of the respective reactions and ρ is related to the degree of reductance of biomass. For the sake of simplicity, some of the participating chemical compounds such as ADP, P, H_2O, and NAD are sometimes omitted in the above equations to avoid undue complications. The generation of ATP through TCA cycle is biochemically well established in biochemistry. What is not so clear is the phosphorylation efficiency as indicated by the P/O ratio in the respiratory chain, the ATP yield coefficient (Y_{ATP}) in the biosynthesis reaction, and the ATP requirement for cell maintenance (m_{ATP}). Note that various linear combinations of the above set of equations are possible, and some may be more meaningful than others. The following are some of the more popular permutations.

$$(39a)\ \&\ 6 \times (39b)\text{yields: } C_6H_{12}O_6 + 6O_2 \rightarrow 6H_2O + 6CO_2$$

$$+ (4 + 12P/O)ATP \tag{40a}$$

$$(39a)\ \&\ \frac{\xi}{6} \times (39b)\text{yields: } \frac{1}{6}(1+\xi)C_6H_{12}O_6 + \delta\,NH_3 + \left(Y_{ATP} - \frac{2}{3}\xi\right)ATP$$

$$\rightarrow CH_\beta O_\gamma N_\delta + (1 - \gamma - \xi)\,H_2O + \xi\,CO_2 + (2\xi - \rho)\,NADH_2 \tag{40b}$$

Equation 40a represents the internal combustion of glucose, and Equation 40b is another commonly used form of biomass synthesis in which the evolution of CO_2 is included. However, the concept of linear independency indicates that no more and no less information is to be gained from the use of these linearly dependent reactions.

In the decomposed formulation, some information on reaction stoichiometry is available, but the extents at which these parallel reactions participate in the overall process are unknown and can be related to the stoichiometric coefficients of the overall Reaction 7. By balancing the nine components appearing in Equations 39a to 39e, we can obtain nine equations for each of the components. Of the nine equations, one equation is consumed by normalizing $B = \xi_4 = 1$; another four equations are the linear combinations of the elemental balances on C, H, O, and N; and another three equations relate ξ_1, ξ_2, and ξ_3 to the stoichiometric coefficients of Reaction 7:

$$\xi_1 = \frac{1}{6}(e - f) \tag{41a}$$

$$\xi_2 = b \tag{41b}$$

$$\xi_3 = \frac{1}{2} f \tag{41c}$$

Thus, the decomposed formulation in this example introduces only one additional independent equation that has not been available in the previous overall formulation. This equation is basically the ATP balance:

$$\frac{f}{b} + 2\frac{e}{b} + 6\ P/O - \frac{3}{b}(Y_{ATP} + m_{ATP}) = 0 \tag{42}$$

Note that the above equation is gained only after the introduction of three additional unknown parameters: P/O, Y_{ATP}, and m_{ATP}. Therefore, this additional equation can contribute in the material balance scheme described earlier only when all the above parameters are known. Equation 42 can be rearranged to yield the following equation:

$$\frac{f}{b} = \frac{6(Y_{ATP} + m_{ATP}) - 2\rho}{6(Y_{ATP} + m_{ATP}) + \rho}(RQ - 1) - \frac{2\rho(1 + 3\ P/O)}{6(Y_{ATP} + m_{ATP}) + \rho} \tag{43}$$

A similar form for ethanol production with baker's yeast has been derived by Wang et al.[324] without considering the set of parallel Reactions 39a to 39e in metabolic pathways:

$$\frac{f}{b} = (RQ - RQ_0) \tag{44}$$

where RQ_0 is the respiratory quotient when no ethanol is produced. Because the average degree of reductance of biomass is about 4.291,[338] $\rho = (4.291 - 4)/2 \approx 0.15$, and for a plot of f/b vs. RQ Equation 43 predicts a slope of slightly less than 1 and an intercept on the RQ axis of slightly larger than 1. In contrast, Equation 44 predicts a slope of 1 if RQ_0 is truly a constant. (Evidently, RQ_0 is not truly constant because it is actually a function of b in Wang et al.'s formulation.) The experimental result in our laboratory confirmed this prediction, and a similar result, but with a much wider data scattering, was also previously observed.[13] A sensitivity analysis has shown, however, that both the intercept and the slope of the plot of f/b vs. RQ are not very strong functions of Y_{ATP}, m_{ATP}, and P/O;[344] therefore, the relationship of Equation 43 is generally valid for yeasts following Reactions 39a to 39e. Of course, it must be reformulated if the microorganisms do not undergo such pathways. The usefulness of this extra equation remains to be seen. However, the authors have some reservation about its contribution to the macroscopic elemental balance techniques, because the calculation of stoichiometric coefficients may be rather sensitive to the slope and intercept of the f/b vs. RQ plot. This sensitivity may be partly related to the sensitivity resulting from the oxygen-heat regularity.

The above example of yeast growth is cited in order to demonstrate an uncomplicated methodology of how cell metabolism, in terms of the extents of reactions, can be monitored in real time with the existing instrumentation without prodding deeply into cells. The generalized systematic approach to solving a set of decomposed parallel reactions is summarized by Roels.[321] However, it should be recognized that Reactions 39a to 39e are actually a set of starting assumptions. Since a different set of assumptions on the individual reactions will result in a different set of estimates for the extents of reactions, these assumptions must be independently verified through some other methods. In other words, these reactions must be established as the predominant representative of the intracellular metabolism despite the fact that the validification of these assumptions are seldom practiced.

V. MODELING, CONTROL, AND OPTIMIZATION

All real processes are forced by random inputs, their mathematical models are frequently inaccurate, and the values of the important parameters are seldom known. The above problems are more pronounced in the case of a bioreactor because of the generally poor understanding of biological processes, indicating that feedback information must be utilized to obtain a satisfactory operation. Open-loop control schemes which do not utilize any feedback of information are subject to the effect of numerous uncertainties and perturbations which eventually will lead to instabilities and economic losses. Therefore, control strategies derived from the solution to optimal control problems cannot be entrusted for the bioreactor operation unless a feedback parameter adjustment mechanism is provided for.

A schematic block diagram for the formulation of an interactive estimation-control-optimization scheme is shown in Figure 18. The separation principle can be invoked to facilitate the solution of the combined estimation-control problem by considering the function of each element separately. Even though the validity of this principle is limited to linear systems only, it is usually extended to nonlinear systems as well in order to avoid the overwhelming complications that surface when all the functions of the problem are considered simultaneously. Hence, the structured approach is preferred and may be the only means to solve the control-optimization problem without significant loss of optimality.

In this structured approach, first a set of measurements on a biochemical reactor are made. These noisy measurements are then passed through an estimation-filtering scheme, and the relevant information is extracted to yield a set of on-line estimates for the state variables and growth parameters of the reactor. These estimates are subsequently fed back and compared to the set points, and the control signals optimizing some predetermined performance criteria are sent out to actuate controlling devices. The control signal is calculated by treating the estimates from the filters as if they were deterministic and by applying the well-developed deterministic control theories. The functions in the complete inner loop are carried out in real time with the highest priority.

In addition to control, the estimated values for the states and parameters can also serve as a basis for on-line process modeling. In this function the estimates are passed through the outer loop as well, to generate feed-forward-control signals. In the outer loop, the biochemical process is continuously modeled, new values of the model parameters are estimated, and the biological model itself is constantly updated. Based on this updated mathematical model, the future state of the reactor is predicted and the optimal control strategy is reformulated periodically to minimize the projected value of the objective function. If modeling is too formidable and the rigorous dynamic optimization technique extremely complex to implement, an experimental optimization approach that requires no model can be carried out simultaneously. In this black-box approach, the optimal set points are constantly reevaluated through deliberate and systematic excursions from the current ones. Finally, the performance criteria may also be updated by an operator from outside as the

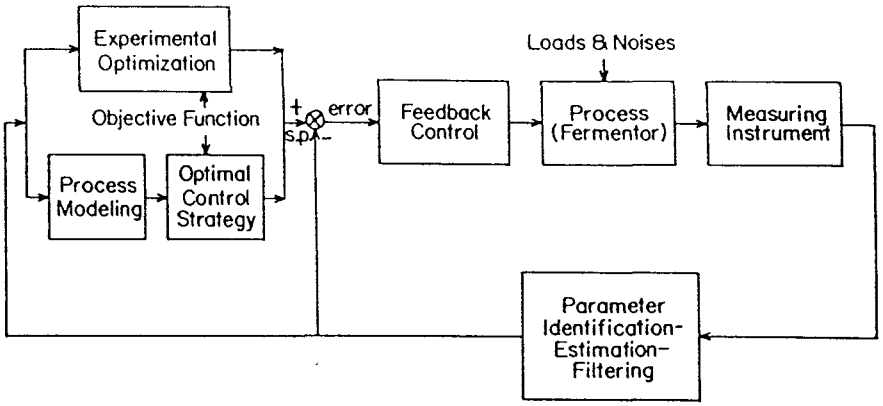


FIGURE 18. Flow structure of the measurement-estimation-optimization-control configuration.

general objective of the operation or the cost structure changes. Thus, the whole feedback scheme is to a certain extent adaptive and self-optimizing in nature.

By breaking the whole problem into many small manageable ones, a certain degree of optimality may be sacrificed in exchange for the ability to perform such a combined feat. For example, it can be shown that only when the system dynamics and measurements are linear, and the random noise is Gaussian, can the separation of the functions of filtering and control still result in an optimal overall performance.[380] Nevertheless, for a nonlinear system, which is the case most often encountered in fermentation, the separation principle is expected to yield a nearly optimal operation if the dynamic equations are linearized about the current states and if the nonlinearity is not exceedingly severe. Some promising results on nonlinear control based on the separation principle have been published in past years.[343,354]

The inner loop without the filtering and parameter identification-estimation block is typical of classical process control applications. The need to include the extra block in the inner loop has already been explained. The outer loop, although omitted from most chemical process control applications, is quite essential in the determination of the optimal operating conditions and the values of set points. If the process model and its parameters are known accurately, this determination can be made mathematically once when a set of performance criteria, such as a high productivity and a high yield efficiency, are given. Only then can the outer loop be deleted without significantly affecting the optimum operation of the process. However, as mentioned before, microorganisms are very sensitive to many parameters, some of them unforeseeable. Thus, the activity of an enzyme may be different from the predicted values, or the behavior of the microorganism may change gradually over a period of continuous operation. In view of these uncertainties, an accurate analytical expression for the performance function in terms of this large set of parameters is presently impossible. Any attempt to do so is likely to produce only an incomplete expression. It is, therefore, necessary to build a scheme for the reevaluation of various individual kinetic parameters, to update the control algorithm, and to readjust various settings when changes are detected. This is especially true for a sensitive system where a tight control is essential for a successful performance, and most biological reactions fall in this category.

A. Modeling

Mathematical modeling in the biological field is a vast subject that requires the total devotion of a whole book. Mathematical modeling per se is not reviewed in this article. Rather, the subject is reviewed here as it relates to computer applications in the areas of fermentor control and optimization.

Mathematical models are routinely used in chemical engineering in various phases of operations from preliminary equipment design to the final specification of operating conditions and optimization. In contrast, the existing mathematical models are rarely fully trusted by investigators in the biochemical engineering field. This is because the biological behavior has a complexity unparalleled in the chemical industry, and, consequently, its prediction from information about the environmental conditions is extremely difficult. Only quite recently has it been possible to predict in a limited way the impact of changes in control variables on the response of the microbial culture.

Nevertheless, mathematical models, being the representation of our understanding and the condensed version of our knowledge, are necessary for bioreactor design and the successful formulation of control strategies. In many instances, the objective of developing a mathematical model is explicitly aimed at providing the basis for controlling the performance of a bioreactor.[381] In this case, the effects of control variables (temperature, pressure, pH, concentration levels, etc.) on the process must be identified and described by models which can be referenced in making control decisions in a computerized bioreactor. Furthermore, they are absolutely indispensable in the calculation of optimal operating conditions, both static and dynamic.

Computers are used in both stages of model formulation and model application. In the first stage, the formulation of a model requires systematic study and identification of the effect of the various model parameters. The study can be accomplished by experimenting in a fully controlled environment so that the effect of other variables are eliminated or, at least, minimized. Such full control and close monitoring is best achieved by using computers, especially in view of the large number of reactors usually employed for this purpose. Computers are also employed in the further manipulation of the collected data and in the identification of the model parameters as discussed in previous sections. Once a model has been constructed, simulations with computers are routinely carried out in biochemical engineering,[382] and they have been summarized elsewhere.[383] The use of a computer in the application of a mathematical model to control an optimization is the subject of later sections.

1. Types of Growth Models

The strategy of modeling has been discussed by Roels.[384,385] A good starting point is usually the consideration of well-founded macroscopic component balances and elemental balances which require meticulous accounting of the participating chemicals. Creative thinking can simplify things by selecting the most relevant state variables and the associated kinetic equations. Many additional constitutive equations for the conversion rate expressions are also needed at this stage. The most widely used state variables are biomass and substrate, with a Monod rate expression for biomass conversion and a yield coefficient-maintenance combination for substrate utilization.

$$R = \frac{\mu_{max}\, s}{K_s + s} \tag{45}$$

$$R_s = \frac{1}{Y_s} R + m_s b \tag{46}$$

Many variations to the Monod model have been proposed and are summarized by Fredrickson and Tsuchiya.[386]

The sophistication of models ranges widely, and many classification criteria are available to categorize models in biochemical engineering. They can be classified as either statistical or deterministic, depending on whether random effects are considered; segregated or unsegregated, depending on whether the distribution of properties in a cell population is

differentiated; or lumped or distributed, depending on whether spatial variations in bioreactor conditions exist. Another classification, perhaps the most fundamental, is given by Kossen.[387] According to this classification, models are divided into phenomenological and mechanistic (or structured) categories. Phenomenological models, sometimes also called formal, empirical, or black box models, are usually unstructured and are mainly used to describe the overall observed microbial response. These models are usually specific to the system for which they were developed and generally not applicable in other environmental conditions. On the other hand, mechanistic models structure the cell into many components in order to provide explanations for the observed phenomena. Thus, it is more encompassing than an empirical model. The success of a mechanistic model lies in its ability to predict microbial response under different environmental conditions; the test of its validity, therefore, is more stringent. In a structured model, the macromolecular components such as proteins, DNA/RNA, and carbohydrates are distinguished, and the concentrations of various compounds and enzymes involved in different metabolic pathways are followed.[388] As can be seen from the discussion on instrumentation, these internal states of cells are difficult to observe, let alone quantify in real time. Thus, most models proposed to date in literature are deterministic, unsegregated, and unstructured.

2. Considerations in Modeling and the Use of Models in Computer Control

Some of the considerations in modeling cell growth and metabolism are shown in Figure 19. As evidenced by this figure, there are many aspects that a model must account for, and these aspects are discussed below in the context of bioreactor modeling.

First, the flow and mass transfer processes, although not strictly the problems exclusive of fermentation technology, are quite dominating inside a bioreactor. Gas-liquid interfacial transport and the flow characteristics, influenced by rheological and surface properties, determine the mixing of broth, affect the heat-mass transfer properties, and create gradients in pH, temperature, and concentrations within the same fermentor. The relationships between flow structure and reactor behavior and the methods of measurement of flow pattern, bubble swarm behavior, and turbulence properties were discussed by Schügerl et al.[389] The variation of oxygen mass transfer coefficient inside an air lift tower loop reactor was studied recently by Luttmann et al.[390,391] by using a distributed parameter model and by measuring the dissolved oxygen concentration profiles along the tower. Besides the oxygen mass transfer coefficient, the variations in gas pressure and velocity were also taken into consideration in estimating the kinetic parameters.

Because gas bubbles are sparged, broken up, and dispersed by agitation, aeration intensity is a function of both the flow pattern and mass transfer characteristics. The effect of aeration intensity on the energetics of baker's yeast growth was studied by Oura by keeping other environmental factors constant;[392] however, the Y_{ATP} and P/O ratio could not be determined simultaneously and independently. The effect of aeration on the yeast metabolism,[393] oxidative enzyme activities,[394] and glycolytic and pentose phosphate enzyme activities[395] in the absence of catabolite repression was also investigated.

The effect of the extracellular physical and chemical environment on cell metabolism is well recognized. For example, the temperature effect on microbial metabolism is very profound because the enzymes inside a cell can only operate within a very narrow temperature range.[396] The metabolic activities at different temperatures are not the same unless the temperature dependencies of the enzymes are all the same, which is highly unlikely. Similarly, different enzymatic reactions have different optimum pH ranges; thus, different pathways may be favored during the course of fermentation, and the kinetic rate expression may also shift as a result. The increased significance of dormant parallel shunting pathways as the pH is changed can be used to explain qualitatively the existence of the so-called "shoulder effect" in a batch fermentation.[397] For the same reason, maintaining the pH at a constant

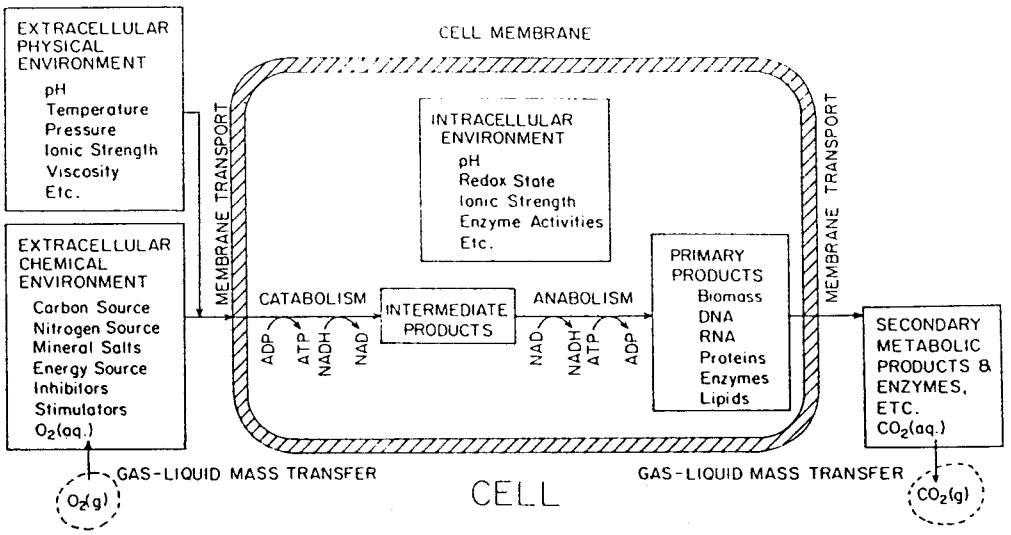


FIGURE 19. Considerations in modeling cell metabolism.

value throughout the entire batch or fed-batch fermentation, in spite of its popular practice, is generally nonoptimal.[398] Although pH is mentioned as an example, the same general principle applies to other variables as well.

The interface between the extracellular environment and the intracellular environment is the cell membrane, and the membrane transport process supplies a microbial cell with the necessary nutrients and removes harmful products of metabolism from it. Although cell behaviors are routinely correlated to and modeled in terms of extracellular environment, it is actually the intracellular environment that a cell responds to. Each cell can be viewed as a complex chemical reactor in which thousands of enzymatically catalyzed reactions with intimate interactions take place along with internal feedback controls and inhibition/repression/activation.[399] These reactions can be roughly divided into two categories: those which break up the nutrient compounds to derive energy (catabolism) and those which assimilate carbon sources to and from cell biomass (anabolism). These reactions are the heart of the fermentation process, and there is an urgent need to understand fundamental microbial behaviors in terms of catabolic and synthesis pathways. The imbalance between catabolism and anabolism results in internal osmotic, redox, and energy imbalances.[400] The excretion of partially oxidized secondary metabolic product, the increased rate of respiration, and the shift in the respiratory mechanism to avoid excessive ATP generation are some manifestations of the cell's effort to level these imbalances.[400] An understanding of these processes will have a profound impact on the formulation of a process control scheme. Toward this goal, multicompartment mathematical models are frequently used.[171,401] In a multicompartment model, intracellular material is divided into many parts. The criteria for classification and the number of compartments depend on the specific process being modeled, but the most popular division is along the lines of macromolecular components (RNA/DNA, protein, carbohydrate contents, etc.). Individual considerations of each and every chemical compound are not practical; instead, lumping techniques should be utilized. The possible exceptions to this rule are the carbon source, ATP/ADP, NAD/NADH, and certain important intermediate products. Usually, various intracellular biochemical mechanisms are considered in a structured modeling approach;[388] however, the sophisticated approach is fruitful only when the model is centered around a few carefully selected predominant mechanisms.

Mathematical models are used to formulate control strategies. The sophistication of the resulting strategies varies widely. Among the straightforward ones may be an adjustment of

wine fermentation temperature in order to avoid overload on cooling; the heat generated can be predicted based on a set of simple kinetic equations that take into account the ethanol inhibition and the temperature effects on various model parameters.[402] Because yeast cell growth and sugar uptake rate is slower at lower temperature, it is possible to spread the heat generation into a longer period of time so that the overall cooling requirement can be met by the existing equipment. Among the slightly more involved control strategies are those based on gas exchange values, especially the RQ. The use of RQ as an indicator of cell physiology and control based on RQ has often been proposed.[322] For example, RQ can be directly used to estimate the extents of reactions for yeast growing on ethanol.[403] The interpretation of RQ in terms of physiological conditions is slightly more complicated for yeasts growing on glucose (see Equation 43). Wang et al.[324] interpreted RQ > 1.0 as ethanol formation, $0.9 < RQ < 1.0$ as oxidative growth, $0.7 < RQ < 0.8$ as endogenous metabolism, and $RQ < 0.6$ as ethanol utilization. The use of models in the sophisticated computer control of fermentation processes has been discussed by Yoshida and Taguchi.[404]

The characteristics of a proposed control strategy can be studied through the use of mathematical analysis and the application of control theories. A preliminary evaluation of the feasibility of realizing the proposed scheme can be carried out by examining the computer-simulated transient behavior and the stability of the system. The effect of different modes of regulation on the bioreactor behavior was analyzed by Chirkov et al.[405] and a different mode was recommended for each different objective. Many revisions of the model may be necessary, and each time a new control strategy may be formulated. For example, some additional parameters that have been neglected in the original models, such as the maintenance term on substrate and endogenous decay, change the steady-state condition in a continuous fermentor. More importantly, these neglected parameters can affect the transient behavior; while a simple Monod model with a constant Y_x cannot exhibit damped oscillation behaviors, a Monod model with a maintenance term can.

3. Model Identification and Parameter Estimation

Mathematical estimation of model parameters is generally carried out by minimizing a selected objective function by some sort of optimization technique. Normally, a single set of parameter values are obtained through a least-square type of algorithm.[406,407] Linear static parameter estimation methods for fitting experimental data to a model are well established.[408] However, there are no generally applicable theories for nonlinear parameter estimation. Alternatively, a sensitivity function can be formulated to study the relative effect of changes in various parameters on the observed state variables. A black-box (experimental) approach with least-square fitting criteria was used by Meiners and Rapmundt[409] to identify some of the most important parameters in relation to the order of dynamic model and the time delay. Various interactions between input variables and output variables can be identified through the use of a sensitivity function, and the estimated parameters can be used to construct an empirical model of the dynamic system which, in turn, can be used to formulate overall control strategies and design controller gains.

In yet another black-box systems engineering approach adapted by Young and Bungay,[410] a block diagram of the bioreactor system was first constructed from a set of Laplace-transformed dynamic equations for biomass and limiting substrate. The specific growth rate was expressed as a function of substrate, temperature, and pH; similarly, the yield coefficient was expressed as a function of the specific growth rate. A quick glance at the block diagram enabled a researcher to see what types of experiments were needed to determine the dynamic relationships in each of the blocks. The response of biomass, glucose, RNA, and protein concentrations to separate step changes in feed glucose concentration, dilution rate, pH, and temperature were monitored semicontinuously in a continuous fermentor with *Saccharomyces cerevisiae*. The response diagrams were analyzed by the use of classical control theories,

and the model parameters were evaluated by minimizing an objective function with an iterative procedure. However, due to linearization in Laplace transformation, the range of validity of the derived model was limited to the vicinity of the steady state around which linearization was carried out. Unfortunately, this is generally true, and some model parameters can be identified only at certain restricted conditions.[363]

Nonlinear programming methods are used by Alvarez and Ricanō,[411] to determine the maximum specific growth rate, the Michaelis-Menten constant, and the overall substrate-to-cell yield as polynomial functions of temperature and pH. They pointed out the prevalent lack of statistical analysis on the estimated parameters and applied the Z-test to examine the randomness of the residue. Furthermore, a Student-Fisher test was employed to detect and delete those parameters of the polynomial regression that had no physical meaning because of the high standard deviation associated with them.

Some model parameters are continuously changing despite the fact that they are commonly treated as constant. When these parameters cannot be empirically expressed in a closed functional form in terms of other independent variables, they are sometimes treated as dummy-state variables and their values are estimated as if they were state variables. The specific growth rate and substrate-to-biomass yield coefficient are sometimes treated this way, as mentioned earlier in the section of on-line parameter estimation. In an attempt to improve the agreement between the prediction of a simple Monod model and the experimental data, Takamatsu et al.[412] introduced an "activity" variable, which was related to the specific growth rate.

Inversely, when there are unmeasurable or uncontrollable variables, the order of system should be reduced through linearization and the complexity of the system should be reduced through the use of simplifying assumptions.[404] When the number of model parameters is too large, the on-line estimation may become unfeasible. (For example, 87 min of CPU on a relatively powerful IBM 370/135 computer were consumed to estimate 24 parameters in a SCP respiration model![413]) In such a case, the number of parameters must be reduced to a manageable level.[414] If some uncertain parameters still remain after the simplification effort, a control strategy can sometimes be formulated in such a way that the effect of these uncertain parameters on the overall performance is minimized.[415]

In closing, it should be pointed out that a good agreement between the model and data is, although definitely necessary, far from sufficient to claim the validity of a particular model, especially if the data is available for only one set of conditions. For example, four models (Monod, Contois, linear-specific growth rate, and enzyme kinetic model of growth) could not be distinguished from one another by comparing the simulated batch growth curves, because the curves for each model based on the appropriately chosen parameter values all closely agreed with the fermentation data.[407] Model differentiation is often a very difficult task, requiring dedicated efforts. The demonstration of more general behaviors such as oscillations, limit cycles, and hystereses are sometimes used, but some ingenious techniques are frequently contrived for one-time use. Quite often, one becomes too indulged in fitting the data of a very limited number of experiments into a preconceived model that does not exhibit or cannot explain the observed global behavior. The number or the nature of the data are frequently insufficient or inadequate to identify the model parameters properly. Thus, the degeneration of creative modeling into systematic curve fitting is forewarned. Furthermore, the validity of the model is rarely checked under different sets of experimental conditions. Since a mathematical model is only an approximation of the actual phenomenon, it should be simple enough for us to understand, yet sophisticated enough to display all the pertinent behaviors correctly. In addition, a well-formulated model has an inherent capacity for future refinement and improvement by guiding researchers to design revealing experiments and by allowing them to observe the hitherto unnoticed phenomena.

Finally, it is our opinion that modeling should somehow reflect the present state of knowledge. It should never proceed beyond such a point where the increased complexity

no longer contributes to the true understanding. It is almost always much easier to fit the same experimental data to a sophisticated model containing many dynamic equations and a multitude of adjustable parameters than to a simple model containing only a handful of these dynamic equations and a comparatively small number of parameters. The overuse of parameters without adequate justification is especially tempting in the rigorous structured modeling approach where even the minutely significant metabolism or chemical components are considered. The additional information gained is hardly worth the greatly increased complexity. Approximations and assumptions should be critically utilized to eliminate many nonessential parameters and states by carefully selecting the more relevant variables according to, for example, time scale[384,385] and length scale.[385,387] Although the extremely complex nature of the biological system may require complicated models, we should resist oversophistication because it tends to defy the very purpose of modeling by obscuring the essence of the model and makes the prediction of microbial behavior exceedingly difficult or even beyond human comprehension. In conclusion, one should be skeptical with models that contain an overwhelming number of parameters whose numerical values cannot be experimentally evaluated, for, as pointed out by Roels,[384] such detailed modeling is hardly rewarding. History has shown that simplicity shall prevail, and herein lies the difficulty in modeling, i.e., how to explain intricate responses with simple lucid models.

B. Control

In the past, pH, temperature, dissolved oxygen, viscosity, substrate concentration, biomass concentration, RQ, and a wide variety of other variables have been controlled at constant values, in combination or alone, by various investigators with the aid of computers. In the studies of hydrogen or methane-oxidizing microorganisms, it may sometimes require the fermentor gas composition to be controlled automatically so that the gas exchange parameters can be meaningfully determined.[416] Fermentation vessel pressure and foam are routinely regulated but not really controlled.

Except when specifically designed to demonstrate the technical feasibility of performing such control tasks, it should be cautioned that control for the sake of control accomplishes very little if not nothing at all. As pointed out in the previous sections, some control studies seem to lack purpose because they give no indication of the motivation behind them. If the benefit from a proposed additional control cannot be clearly stated, then setting a set of chosen variables at constant values merely degenerates into regulation and can no longer be truly regarded as control. The implementation of not only control functions but also regulatory functions often needs to be justified; the economic truth is that even sophisticated control algorithms are warranted only if their implementation can improve the performance commensurate with the extra effort.[417] Thus, although regulatory controls are useful in modeling studies where the undesirable influences of other variables are to be eliminated, simply maintaining a set of variables at constant values may not always be rewarding in an industrial environment.

Computer control structures for microbial cultures are shown in Figure 20. They can be represented by three different block diagrams. The first case (Figure 20A), which is presently the most widely used structure, is to institute separate controls for different variables with totally independent loops. The conventional on-off or PID controllers employed to regulate such individual variables as pH, temperature, and vessel pressure generally fall within this first category. In the second case (Figure 20B), the interactive effects of the manipulated variables on the controlled variables are recognized. The simultaneous computer control of biomass and substrate concentrations affected through dilution rate and feed composition manipulations is one such example. In the third case (Figure 20C), the interactions between different controlled variables are also considered in the programmed controller. Typically, many levels of cascaded loops are contained in the much more sophisticated controller. In

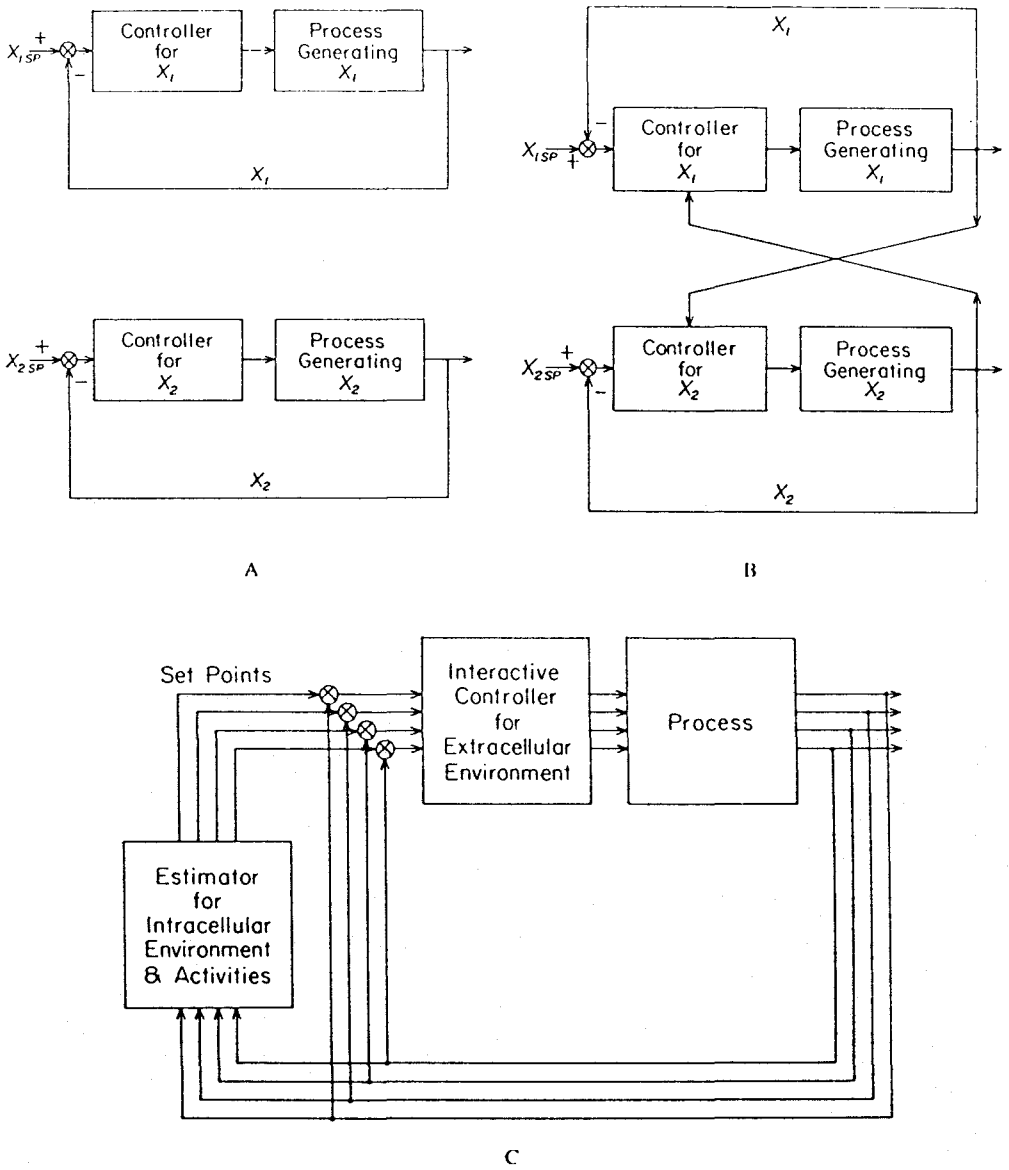


FIGURE 20. Process control loops in biochemical engineering. (A) Separated controls with independent loops; (B) separated controls with interacting loops; (C) cascaded interactive controls.

the most general manner, the intracellular environment and metabolic activities are directly measured or indirectly estimated in the outer loop. They are compared to the desired pattern stored in a computer, and set-points in terms of the extracellular environmental variables are generated for the inner loop. This cascaded control configuration is the manifestation of the fact that we can only hope to influence the intracellular environment through manipulations of the extracellular environmental variables. Since the intracellular environment, over which we have no direct control, is what cellular metabolisms respond to, the importance of understanding the effects of extracellular variables on the intracellular environmental variables and metabolic activities is again emphasized. Unfortunately, the identification of the interactions has been hampered by the complex and nonlinear linking between various variables. An example of this third case is the attempt to suppress the formation of undesirable secondary metabolic products by monitoring gas exchange conditions, which act as indicators

for the intracellular metabolic activities; set-points are then generated for biomass, substrate, and/or dissolved oxygen concentrations, and the controls are finally realized by altering nutrient feed rate, aeration rate, etc.

The control of a microbial system represents an ultimate challenge for many people. Because a fermentation process is sensitive to many variables, in theory an interactive multivariable, multiloop control should thrive in such a process, but in actuality such controls are rare. At present, controls are rarely carried beyond manipulations of the extracellular environmental conditions of temperature, pH, dissolved oxygen, and substrate concentration. Nevertheless, more imaginative multivariable interactive controls designed to shift intracellular mechanisms through the manipulation of the extracellular environmental conditions are expected to appear in the future. As more sophisticated control schemes are devised, conventional on-off or PID controllers will no longer be adequate and computers are expected to be indispensable in carrying out such schemes.

Finally, fermentation product recovery processes, although not extensively covered in this review, may be even more cost intensive than the fermentation process itself, and its control cannot be neglected in an integrated scheme. Of course, widely differing control strategies are possible depending on the specific method of recovery used. An example of the control of a lyophilization process (drying by sublimation under vacuum conditions) was discussed by Jefferis in which the temperature and pressure of the drying chamber were controlled to achieve a shorter drying time.[418] This example offered a unique method of detecting eutectic freezing by an electrical resistivity measurement on the fermentation product.

In the following discussion, recent studies in the general multivariable control algorithms, simple direct environmental feedback controls, and cascaded environmental controls aided by computers will be reviewed. Although they are centered on microbial cultures, most of the fundamental engineering principles and computer utilization can be directly applied in the same manner to animal or plant cultures as well. The major differences between microbial cells and animal cells that one needs to consider in computer application are listed by Nyiri.[419]

1. General Control Algorithms

The general algorithms for an optimal feedback controller for fermentation processes were adapted from the well-known optimal control theories by Fawzy and Hinton to handle disturbances in state variables.[420] First, a set of state equations and measurements are expressed similar to Equations 21 and 22. Then, they are linearized to yield:

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}\mathbf{x}(t) + \mathbf{C}\mathbf{u}(t) + \mathbf{G}\mathbf{W}; \qquad \mathbf{x}(0) = 0 \tag{47}$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) \tag{48}$$

where **x** is now the vector of the deviations of the states from the desired values, **u** is the control vector, **W** is the disturbance in state variables, and **y** is the vector of the measured variables. Three possible situations were each analyzed. The first scheme assumed that all the states were measured. The optimal control signals for this case were calculated by minimizing the following quadratic cost function:

$$J = \frac{1}{2}\int_{t_0}^{t_f} [\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{u}^T\mathbf{B}\mathbf{u}]dt \tag{49}$$

where **A** and **B** are the weighing matrices. The second scheme assumed that only some of the states were measured, and a solution minimizing a cost function similar to the above could be obtained accordingly. The last scheme, which corresponds to the most frequently encountered situations in fermentation processes, assumed that the states were not directly

measurable and that the measurements and system dynamics contained noises. The separation principle was applied to estimate the state variables with a Kalman filter (Equations 23 and 24) cascaded with the deterministic approach of the first scheme. The performance of the optimal controller for each situation was evaluated by computer simulations in which biomass and substrate concentrations were maintained at the desired constant levels by manipulating the flow rate. As expected, the results revealed that the performance was superior when the states were directly measured.

In the next study, Fawzy and Hinton[421] proposed an optimal controller for a set of multistage continuous fermentors. The control structure was derived to minimize an appropriate performance index during the transient from arbitrary initial conditions to the desired steady state by controlling the residence time in each fermentor separately. In order to reduce the computer memory requirement and facilitate computation, a multilevel computation approach was introduced as an alternative to the conventional single-level technique.

However, to the authors' knowledge, no actual control has yet been implemented in which the controller output is determined according to the above optimal strategies. Instead, control studies to date are concerned with the demonstration of the ability to reach the desired states without any optimality considerations. It should be pointed out that the actual implementation of the optimal controller strategies may not be feasible if perturbation from the desired values is frequent and if a large fraction of the computer resource is needed to calculate the optimal trajectory. Thus, the absence of optimal controllers in the existing fermentation processes is a reflection of the point stressed earlier that sophistication must be justified by the realizable improvement in performance.

2. Direct Environmental Controls

Some processes are inherently unstable under the conditions one wishes to operate, and disturbances in the state and manipulated variables cannot be totally eliminated. As a result, information feedback is necessary to close the control loop and ensure satisfactory operation. A difficulty here is the limitation of sensors capable of being interfaced to a computer to yield continuous, instantaneous information on the fermentor status, thus, prohibiting the realization of comprehensive feedback control schemes.

A study by Boyle[422] examined the possibility of manipulating the dilution rate and the feed substrate level for the simultaneous control of the biomass and substrate concentrations at constant values in a quasi-steady-state fed-batch fermentor. Following the linearization of the dynamic state equations in the form of Equation 47, a mathematical analysis on the interaction between the state variables and control variables showed that the dilution factor should be used to control the substrate concentration and that the feed composition should be used to control the biomass concentration. The optimal controller settings were obtained according to the first scheme just described above, but the feedback gains were unrealistically high. To overcome this problem, a more direct approach was utilized to determine the controller law. In a subsequent application of these findings by Kalogerakis and Boyle,[423] biomass and substrate concentrations were experimentally controlled to achieve a quasi-steady-state fed-batch yeast fermentation in the face of various externally imposed disturbances.

The dynamics of a continuous bioreactor subject to substrate inhibition was studied by Edwards et al.,[424] and the responses of a turbidostat and a nutristat under proportional control of flow rate were simulated. With the help of computers, Rolf et al.[51] studied the effect of dissolved oxygen level on μ and Y_s in a batch fermentor cultivating a methanol utilizing bacterium and concluded that μ as a function of the limiting substrate (dissolved oxygen) was nonmonotonic. Thus, there existed multiple steady states under a continuous operation, and the unstable steady state could be maintained only when feedback control was enforced. The dissolved oxygen control was actuated by first altering agitation speed then adjusting aeration intensity if necessary.

Takamatsu et al.[425] simulated the single-variable control of constant biomass concentration in a continuous bioreactor by adjusting the dilution rate. They concluded that a superior performance was obtained when the biomass concentration was directly measured and that divergence was possible when the dissolved oxygen level measurement was substituted for the biomass concentration measurement. A noninteracting multivariable control was also simulated in which both the biomass and substrate concentrations were kept constant by individually manipulating two substrate flow streams of different concentrations (i.e., equivalently the dilution rate and the feed composition). Again, the controller performance was better when the states were both directly observable. Subsequently, an estimation method based on a mathematical model was proposed to derive the biomass and substrate concentrations from the oxygen uptake measurement in an activated sludge reactor used in a sewage treatment plant.[426] The action of PI controllers regulating the recycle rate was similarly simulated with a different set of models, and Takamatsu et al. noticed that even in the presence of a time lag of 30 min and despite the variations in the inlet flow rate and substrate levels, the outlet substrate concentration could be controlled almost as satisfactorily as under the idealistic deterministic conditions with exactly known biomass concentrations. However, these results were not verified experimentally.

Although mixed culture processes dominate the natural environment, few mixed culture systems are used in the fermentation industry, and to the authors' knowledge none is currently being controlled in terms of microbial interactions. Hatch et al.[307] and Hatch and Cadman[427,428] have considered the possibility of multivariable computer control of a continuous competitive mixed culture fermentation by monitoring the concentration of each microorganism with a laser cytophotometer on a semicontinuous basis. In order for both competing organisms A and B to coexist in a continuous steady-state fermentor, the dilution rate has to be carefully controlled at D_{ss}, the common value of the specific growth rate of the two species, as shown in Figure 21. The simultaneous control of steady-state cell densities, cell density ratios, and the growth rates is desired. The control strategy based on two manipulative variables (substrate feed rate and the dilution rate) cannot be implemented manually because of the sensitivity of the system; it can only be accomplished through a continuous on-line feedback control. The response of the classical proportional control algorithm was simulated with a computer, wherein the substrate feed rate was directly coupled to the concentration of the faster growing culture and the dilution rate was directly coupled to that of the slower growing culture.[307,428] This structure produced an excessively long response time (50 hr) for the system to approach to another steady state. Realizing that the dilution rate affected both microorganisms and that the substrate concentration affected their relative growth rates, the same researchers devised an improved scheme where the dilution rate was coupled to the total cell density and the substrate feed rate was coupled to the difference in cell densities. The scheme was further improved by coupling the difference in cell densities to the desired substrate concentration, which was, in turn, used to calculate the substrate feed rate through a linearized kinetic equation. After a few trials, the new scheme was able to shorten the response time to 4 hr. Regretably, the control scheme was not carried beyond the hypothetical computer simulation.

Using a computer, Whaite and Gray performed an on-line optimization study to maximize the biomass productivity in a steady-state continuous fermentor.[429] From Figure 22, in which the schematic plot of cell concentration and cell productivity vs. dilution rate is shown, it can be readily concluded that the maximum biomass productivity is realized by operating at a dilution rate of D_{max} with the corresponding biomass concentration at b_{max}, provided that the system parameters are known to enable the construction of such a curve and that they remain constant during the fermentation. In their study, the optimal biomass concentration was found experimentally and used as the set point. The biomass concentration was monitored in real time with continuous optical density measurement, and the dilution rate

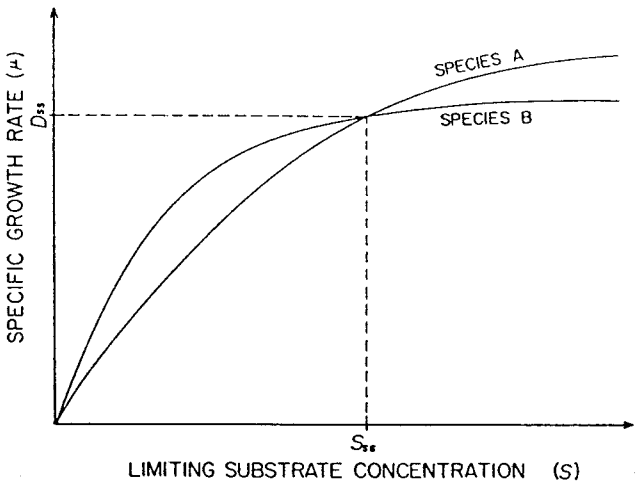


FIGURE 21. Effect of the limiting substrate concentration on the specific growth rates for a mixed culture.

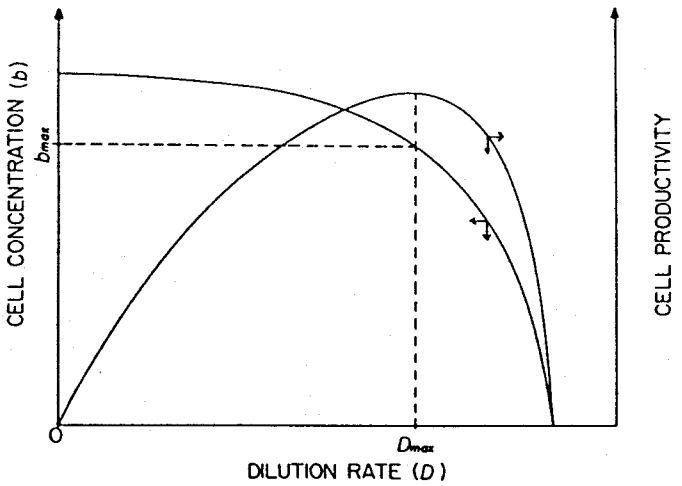


FIGURE 22. Control of a continuous fermentor at the condition of maximum cell productivity.

was adjusted accordingly to keep the cell concentration at the desired value so as to actualize maximum productivity. Derivative action was added to minimize the oscillation caused by the delay in the response of cell concentration to dilution rate changes.

Danielsson et al.[197] have achieved controlling the product (glucose) concentration at a desired constant value as the substrate (lactose) concentration in the feed stream to a lactose enzyme reactor was changing in either a stepwise or a continuous manner. A standard PI controller was used to adjust the feed pump speed based on the continuous glucose measurement performed by an enzyme thermistor containing co-immobilized glucose oxidase and catalase. They also performed an experiment under more realistic conditions in which whey was used as the feed. The enzyme in the reactor was gradually deactivated due to the precipitates caused by the protein present in the whey. They demonstrated that the glucose

concentration could be controlled rather precisely at the desired level under gradual enzyme deactivation in the reactor until the operation was discontinued because of the deactivation of the enzyme.

In another experiment,[216] production of ethanol by an immobilized yeast was controlled by signals from glucose and sucrose enzyme thermistors containing glucose oxidase-catalase and invertase, respectively, such that the substrate (glucose and sucrose) concentrations in the fermentor were kept at constant levels.

Though not the primary concern of this paper, computer control has also been widely applied to wastewater treatment plants and dairy product processing factories. One of the examples in the latter field was supplied by Franks et al.[274] in the production of cheddar cheese. They proposed to control the temperature according to a prespecified profile determined from the mathematical analysis of the kinetic model of syneresis for a multiorganism culture so that the variation in the product quality could be reduced. Some examples in the areas of industrial brewing and wine and soy sauce fermentations also exist, but they will not be elaborated here.

3. Feed-on-Demand Controls for Biomass Production

In recent years, feed-on-demand type of control has become very popular as a means of manipulating the cellular metabolic activities, especially in the area of biomass production. To achieve an optimal result, the nutrient feed rate often needs to be regulated so that the substrate concentration in the fermentor is ideal for cell growth. For example, it is well known that an overfeeding of sugar to yeasts results in ethanol formation even under aerobic conditions (commonly known as aerobic fermentation, glucose effect, or Crabtree effect). The same considerations regarding overfeeding and underfeeding also apply to other microorganisms utilizing other substrates. In general, the formation of undesirable product not only lowers the cell yield but also inhibits cell growth. On the other hand, an underfeeding results in cell starvation and specific growth rate deterioration, reducing the volumetric productivity. There are two types of feedback control policies to regulate the nutrient feed rate. In the first type, an indirect control parameter that can be continuously monitored is used to serve as an indicator of the metabolic activity. RQ,[124,322,327,329,419,430-434] CER, OUR,[431,435] DO,[160,268,436,437] and pH[438] are often used for this purpose. In the second type, the formation of the undesirable side product is directly detected by monitoring its concentration in the broth[268] or in the exit gas.[432,436] Alternatively, the substrate concentration in the fermentor may also be directly monitored as a basis for actuating the feed pump.[267]

Because the gas analyzers for O_2 and CO_2 are readily available, easily operated, and reasonably reliable and sensitive, RQ measurements have widely been used as the indicator of ethanol formation in a baker's yeast fermentation employing glucose as the main carbon source. As shown in Equation 43, RQ should be near 1.0 in the absence of ethanol formation. The direct measurement and control of the sugar level in the fermentor may be more straightforward but may prove more difficult to implement. Although some glucose sensors are available, the direct on-line measurement of the substrate level may be presently impossible if the substrate is composed of not pure glucose but a mixture of various sugars. Furthermore, even when pure glucose is used as the sole carbon source, there still exists the problem of determining the set point for substrate concentration such that the aerobic fermentation by yeast is prevented.

Aiba et al.[431] and Nagai et al.[434] were among the first to perform computer control of substrate feeding in baker's yeast fermentation with RQ as the control parameter. Aiba et al. were able to show that for a fed-batch operation and for an RQ value of near unity, the feed rate is explicitly dependent on the oxygen consumption rate. The disadvantage of this method was that the feed law used to match the substrate demand contained a constant that depended on the substrate to ethanol yield; thus, the success of control depended heavily

on the precise knowledge of this parameter. Despite these shortcomings and the lack of true feedback controls, they were able to control RQ at values between 1.1 and 1.2.

Another control scheme of repeated fed-batch reactor aimed at reducing the effect of catabolite inhibition in baker's yeast fermentation was proposed by Peringer and Blachere.[439] The strategy was essentially to keep the substrate and the dissolved oxygen concentrations constant by feeding the fermentor with nutrient in an exponential manner. The optimal feeding policy could be expressed as a function of RQ and OUR; however, the expression involved many parameters that had to be specified. Thus, the same problems encountered in Aiba et al.'s work[431] also limited the usefulness of this approach.

Although Wang et al.[324] remarked on the possibility of using RQ as the control parameter in baker's yeast fermentation, there was no evidence that they followed this method to control the feed rate of sugar. Furthermore, the control did not utilize the various parameters so painstakenly estimated from on-line macroscopic elemental balances. In a subsequent study, Wang et al.[327,329] derived an expression for the anticipated substrate demand, and the substrate feed was controlled to match this demand. The demand for substrate was a function of cell yield, specific growth rate, and biomass concentration. A noteworthy point on their approach was that instead of arbitrarily assigning constant values, these parameters and states were continuously estimated in real-time by using the macroscopic balance concepts described earlier. However, the use of RQ value alone in the feed forward, or anticipatory, control was not totally adequate, and a proportional feedback control law had to be added based on Equation 44. They were able to maintain both a high cell yield and a high volumetric productivity in spite of various process perturbations such as oxygen starvation and variations in inoculum and feed sugars.

The respiratory quotient was similarly used as the control parameter in a continuous baker's yeast fermentation by Spruytenburg et al.[124] A longer delay was encountered in this study because gas chromatographic analysis of the O_2 and CO_2 concentrations in the exhaust gas was employed for the RQ determination. Overshoot could not be entirely eliminated and, subsequently, resulted in significant hysteresis behavior in metabolism. It was found that the microorganisms could quickly switch to aerobic fermentative metabolism in the presence of excess sugar, but the reversal to the more efficient oxidative growth was a slow process. Woehrer et al.[432] compared the performance of RQ-based and ethanol-based computer control of fed-batch baker's yeast fermentation. In RQ-based glucose feed control, ethanol formation could not be totally prevented, and the yield was lower than in the ethanol-based control.

The respiratory quotient was also found to be a sensitive indicator of the physiological state of a *Candida utilis* culture in a single-cell protein production utilizing sugars.[322] An optimum carbon-to-nitrogen ratio along with suppression of ethanol formation was achieved by controlling RQ close to 1.0.[419,430] When ethanol was used as the carbon source in SCP production, it was also desirable to suppress ethanol level in the fermentor to minimize inhibitory effects and yet maintain it at a reasonable level to enhance productivity. A comparison of the performance of automatic control in which either the dissolved oxygen or vapor phase ethanol concentration was employed as the control parameter for substrate addition was conducted for SCP cultivation in ethanol by Huang and Chu,[436] and the direct measurement of ethanol gave superior performance in terms of preventing ethanol accumulation.

Yano et al.,[160] Kobayashi et al.,[268] and Hopkins,[437] all used dissolved oxygen level as an indirect physiological indicator for methanol-utilizing microorganisms. However, the relationship between the observed dissolved oxygen behavior and the biological interpretation was not always one-to-one. For example, an increasing oxygen concentration could be interpreted either as depletion of the substrate, in which case the substrate feed rate was to be increased, or as damaged metabolic capability, in which case the substrate feed rate was

to be decreased.[437] To overcome problems arising from the use of indirect process indicators, Yano et al.[269] and Yamané et al.[267] both measured the substrate concentration automatically and directly used it to adjust the substrate feed rate.

Finally, the feed-on-demand control is critical in biological wastewater treatment processes. Because industrial wastewater contains toxic substances, the metabolic mechanisms may be severely impaired when there is an upshift in the toxic load. On-line computer control of a phenol treatment reactor with activated sludge recycle was accomplished by manipulating the feed rate based on the oxygen uptake rate measurement in such a way that the highest phenol utilization rate was achieved.[435] Experimental results showed that the controller was responsive to the changing phenol load conditions. However, this control scheme did not ensure that the phenol level in the outlet was always below the value permitted by the government regulations; a more practical control might have resulted if various realistic constraints were considered.

4. Specific Growth Rate Controls for Antibiotic or Enzyme Production

Another area of fermentation in which cascaded environmental control strategies are often used is in fed-batch antibiotic or enzyme production. In these processes, there are roughly two phases: a growth phase and a product formation phase. In the growth phase, the usual strategy is to keep the cell growth rate at a maximum so that cells can accumulate in the shortest time possible. After enough cells are present in the fermentor, a product formation phase is initiated usually by shifting the specific growth rate to a lower value or by adding additional inducing agents. Thus, the attention in antibiotic or enzyme fermentations is presently centered on the control of specific growth rate affected by adjusting the nutrient feed rate, although the simultaneous manipulation of a combination of agitation and aeration such that the respiration intensity is maintained at the maximum level is also occasionally used.[440,441] As in biomass production, the substrate feed rate must be carefully controlled in many antibiotic or extracellular enzyme productions to avoid catabolite repression resulting from overfeeding. Similarly, underfeeding causes starvation and may induce severe irreversible damages to the product formation mechanisms.

Due to the inability to conduct direct measurements of the specific growth rate or the substrate level in a fermentor, indirect correlations, often purely empirical, are routinely used. In a fed-batch cellulase production study by Waki et al.,[442] because of the observed linear relationship between the specific carbon dioxide evolution rate and the specific growth rate, the former was chosen as the control parameter. The optimum specific growth rate for the production of cellulase was maintained by the controlled feeding of a mixed substrate composed of cellulose and cellobiose or glucose.

Another study by Lundell[417] also used CER as well as RQ as the control parameters on which the intermittent addition of carbon source was based. One noteworthy point of this study of β-galactosidase production was that different sets of conditions of temperature, pH, agitation, and aeration were maintained between growth and production phases. Another practical point covered in this study was the economic emphasis of employing computer controls. Four different control strategies were proposed, and a comparison was made based on the experimentally evaluated figures of productivity and energy requirement resulting from each strategy. In these strategies, the fed-batch run was either terminated or additional nutrient was added to induce further enzyme production when the production rate fell below the preset value. It was concluded that a properly chosen strategy (fed-batch/continued-batch combination) could improve productivity over the conventional batch process by more than two times, at the same time substantially reducing the energy requirement.

Through the use of a unique on-line filtration technique,[119,120] the build-up and the degradation of mycelial biomass could be quantitatively characterized during a fed-batch penicillin fermentation. During the antibiotic production period, it was discovered that hyphae

density and cell maintenance activity, as indicated by the respiratory rate, decreased due to the loss of cytoplasmic components (chiefly protein) while the cell wall remained relatively unchanged. It was also found that the rate of penicillin synthesis was correlated to the hyphae density.[119] A control strategy via nutrient feed manipulation was formulated by Nestaas and Wang which followed a predetermined desired growth profile that prevented the specific growth rate from dipping below ≈ 0.01 hr^{-1}, at which value the penicillin synthesis rate started to deteriorate rapidly.[121,443] The open-loop control based on the previously determined feed schedule functioned quite satisfactorily most of the time but failed under certain conditions. The deviation from the desired growth profile was minimized by using a closed-loop control scheme incorporating the difference between the desired cell mass and the cell mass measured directly by the filtration probe. The necessity of a closed-loop control with feedback of information on biomass concentration obtained from the filtration probe was exemplified in a situation where biomass concentration estimation by material balance failed due to the presence of a large amount of residual complex nutrient.

C. Optimization

Numerous studies and extensive reviews on optimization in fermentation processes have been published in recent years.[13,21-23,52,444,445] Dobry and Jost[13] summarized some of the earlier studies in optimization. Weigand[21] followed with an extensive survey of the topic and emphatically debated the positive and the negative aspects of the rigorous mathematical modeling approach. Static and dynamic optimization were discussed by Blachere and Peringer.[52] A review of two optimization approaches (the analytical approach with the maximum principle of Pontryagin and the direct experimental approach of trial and search) was offered by Aiba.[444] An excellent detailed survey of various optimization studies was given by Constantinides.[445] Rigorous optimization techniques were contrasted with the black-box approach in an overview by Zabriskie,[22] which emphasized the application of real-time computers. Rolf and Lim[23] also presented a review on the steady-state static optimization for continuous bioreactors and unsteady-state dynamic optimization for batch and fed-batch bioreactors.

Optimal process control is perhaps the ultimate objective of applying computers to fermentation processes. All the works discussed thus far, though may not be specifically stated so, are to a certain extent aimed at achieving this final grand goal. In view of the system variability, improved on-line data acquisition methods are of central importance to optimization and are being systematically developed. However, at the present time Kalman filtering techniques and interacting schemes of multivariable adaptive control are rarely applied to real fermentation processes in real time. On-line optimization is rarer still, and the application of the optimal control configuration of Figure 18 is practically nonexistent.

The delays in the realization of optimization strategies in actual fermentation processes are caused by many problems. In addition to the two main problems originated in the availability of sensors and mathematical models as repeatedly pointed out throughout this paper, there is also a lack of proven incentives. Due, mainly, to the small contribution of the manufacturing cost to the total cost of present fermentation products, the improvement in the performance resulted from the implementation of an optimal control strategy, which is derived with much effort, is frequently disappointingly minuscule. This is certain to change in the future as large volume applications of biotechnology come in line. Another reason of delay is the relatively small number of research groups studying optimization of a fermentation process. The on-line application is also often limited by the available computer resource. Because fermentation processes are generally confronted with nonlinear equations, time-consuming nonlinear programming techniques are needed to solve for the optimal set-points or dynamic paths. The automatic on-line optimization, carried out simultaneously with a fermentation process, requires ample dedication from the part of research investigators, but its successful implementation should prove to be highly rewarding.

In order to be accepted by the manufacturing industries, optimization must be based on a realistically and correctly formulated objective function. In the past, the objective was formulated to maximize the biomass productivity and substrate-to-cell yield in the case of fed-batch baker's yeast or SCP fermentations or to maximize the product productivity or product concentration in the case of antibiotic or extracellular enzyme fermentations. The implication is that these formulations generally coincide with the objective of maximizing profit; however, this is not always so, especially in an antibiotic production process where fermentation cost is only a relatively small fraction of the total cost. In actuality, any meaningful optimization should consider not merely the fermentor but also other interrelated equipment and processes in the same plant. The influence of fermentor conditions on subsequent recovery processes and the reverse influence of recovery processes on the fermentor operation via recycle were discussed by Ash and Topiwala.[446] The optimal termination time in a batch enzyme fermentation process was studied with respect to the lag phase, the biomass accumulation phase, and the product formation phase, with due consideration to the nutrient cost and various operating costs by Attia et al.,[447] and, as expected, a fermentor optimized so as to yield a maximum rate of enzyme production may not produce the maximum profit. Even external factors such as the market demand, the raw material cost, and government regulations greatly influence the formulation of the objective function and, thereby, the optimal control strategy. It is not rare that an overproduction causes the market value of the product to slump and the inventory to stockpile; hence, an increased productivity may sometimes bring economic losses instead. It is quite possible that the inadequate formulation of an objective function may prevent the advanced optimization and control strategies developed in academia from being applied to fermentation industries.

Within the framework of fermentor optimization, there exist two major types of optimization techniques. The static (point) optimization techniques are generally applied to steady-state continuous bioreactors, and the dynamic (path) optimization techniques are applied to unsteady-state batch or fed-batch bioreactors. Each set of techniques can be modified to allow on-line optimization when disturbances to the fermentor occur, but updating the optimal trajectory without delay for a fed-batch or batch bioreactor may be more involved than calculating a new set of optimal steady-state for a continuous bioreactor. There is little shortage of the necessary mathematical tools. The static optimization problems can be solved by the application of calculus or some direct hill-climbing search algorithms. They include methods of Box,[448] Box and Wilson,[449] Hooke and Jeeves,[450] Rosenbrock,[451,452] Powell,[453] Spendley et al.,[454] and Nelder and Mead.[455] The dynamic optimization procedure is centered on the application of the continuous maximum principle of Pontryagin,[456] and the general dynamic optimal control theory is reasonably well developed to handle the type of problems encountered in fermentation processes.

The optimization structure is shown in Figure 23. We see that on the top level the workings of market demand and economics influence the corporate decision on the manufacturing of certain products via a fermentation process as an alternative to other chemical processes. It should be stressed that fermentation is only one of a long series of steps in an actual production process. Thus, an optimization effort which does not consider the other steps is of limited value and must be forewarned against. The objective function used in optimization should take both the preceding and subsequent steps into consideration. Within the fermentation process, there is the optimization of fermentor scheduling. Fermentor scheduling concerns with the optimum allocation of fermentors to produce a line of products, i.e., the assignment of a given fermentor for a given period of time to the production of a given set of products. Computer-aided scheduling of multiproduct plants was studied by Prokop and Votruba.[457] Then there is the sequencing of various events including sterilization, inoculation, and many others.

Again, within the actual cultivation step, there is the choice of the mode of operation. The optimum choice of continuous, fed-batch, and batch operations or a combination thereof

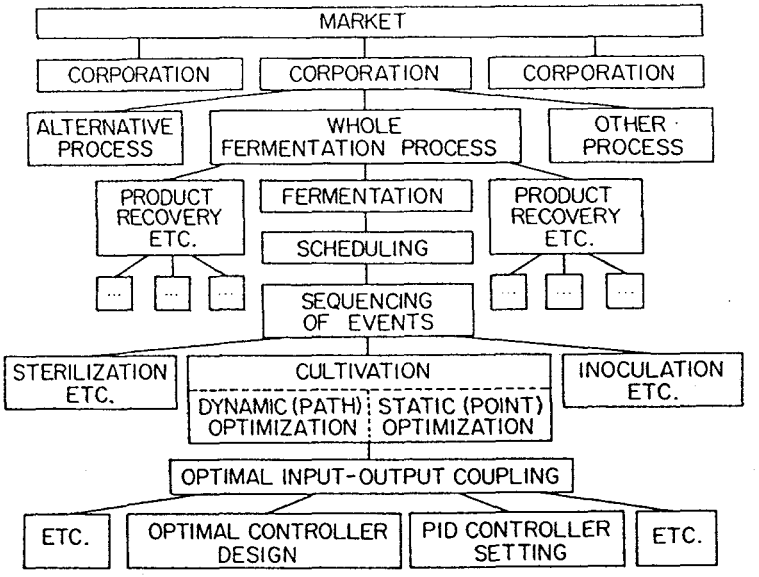


FIGURE 23. Hierarchical optimization structure in a large industrial operation.

was studied theoretically by Ohno et al.[458] and experimentally by Lundell.[417] The characteristics of batch and continuous processes are fundamentally different. For example, batch cultures are synchronized to some extent with similar culture ages, whereas steady-state continuous cultures have an average age. One mode of operation may be preferred over the other depending on the microorganisms and the product involved. In general, continuous operations are suited to large-scale SCP productions, and batch and fed-batch operations are well adapted to antibiotic fermentations.[459] The choice between surface fermentation and submerged fermentation may also need to be considered in certain processes such as citric acid production.[460] So are the shape and size of a fermentation vessel[461] and various other miscellaneous considerations.[462] Finally, at the bottom of the optimization structure comes the optimization of the actual fermentation step, which is of the most interest to the majority of investigators. Once the optimum (static or dynamic) environmental conditions are determined, the optimal controller can be designed to approach the desired conditions in the optimum way.

Thus, we see that there are many levels of hierarchy in a well-posed fermentation optimization structure. Accounting for all the vertical activities from market to the controller and all the lateral activities from raw materials through final products frequently makes the solution of the problem exceedingly difficult, if not impossible. The approach taken to attack this problem is not significantly different from that taken in the chemical industry.[463-466] A large problem is simply divided into a set of smaller ones; each of the smaller problems is further divided into a set of still smaller ones until the size of the problems becomes manageable. This approach is similar to Okabe and Aiba's approach[467-471] in a grand scale. For example, the objective function of the global process is expressed as the sum of the objective functions of the n individual subprocesses,

$$J = \sum_{i=1}^{n} J_i \tag{50}$$

Optimization is subsequently carried out for each subprocess individually, and later recombined to reach the global optimum conditions.

In the following sections, recent global optimization studies will be surveyed first. This will be followed by a brief review of the static optimization studies and the on-line search of optimum conditions for steady-state operations. The dynamic optimization studies will be listed, but they will not be discussed in great detail because the present role of a computer is generally limited to solving a set of two-point boundary value problems or their analogs.

1. Global Optimization

Optimization by the method of decomposition is illustrated by a series of papers by Okabe et al.[472] and Okabe and Aiba.[467-471] Production of penicillin, which involved the sterilization of media and air, batch fermentation, filtration, multistage extraction, and multistage evaporating-crystallization-drying steps, was painstakingly studied. At first, the complex method of minimum searching techniques of Box[448] was modified to find the optimum aeration rate and agitation intensity.[472] The second study examined the optimization of two combined systems: continuous sterilization and batch fermentation.[467] This was accomplished by decomposing the system under consideration into two subsystems, and optimization was carried out in two separate hierarchical levels. Each subsystem had its own objective function which depended on the input, output, and the manipulated variables of the other subsystem. The subsystem objective functions were first minimized individually, then the combined overall objective function, which was a linear combination of each objective function plus the equipment and the utility cost function, was minimized at the higher level. Thus, two separate levels of recursive calculations were required. Successful convergence was reported. The fact that different search procedures resulted in the same optimum conditions was reassuring.

Another subsystem covering auxiliary operations and equipment, such as the dissolution vessels, preculture vessel, and boiler, were subsequently included in the decomposed optimization formulation in addition to the two subsystems previously analyzed.[468] Next, the values of the rotation speed of a rotary filter and the rate of filter-aid addition for the continuous filtration of an Actinomycete's broth were easily optimized.[469] The next step after filtration was extraction. The combined process of filtration and extraction was solved in the same manner as that of sterilization and fermentation.[470] Finally, the last paper coordinated all the previous efforts for a global optimization of the entire antibiotic fermentation plant.[471] Although four subsystems of fermentation, filtration, extraction, and evaporation-crystallization-drying were used in this study, the decomposition algorithm could be easily expanded to include sterilization, solid-waste disposal, and many other unit operations of significance.

As pointed out by Weigand,[21] much of the time spent in calculation will be wasted if care is not exercised in the formulation of proper objective functions. In Okabe and Aiba's studies, the cost to produce a prescribed amount of product annually was minimized; however, it might have been more realistic if the profit had been maximized instead, because, in general, minimum cost is generally not equivalent to maximum profit as mentioned earlier.

Another reason for using the decomposition method is that the classical one-step optimization algorithm may fail to find the true minimum, especially in the case of nonlinear and nonconvex objective function. The existence of numerous implicit inequality constraints and local minima may further complicate the finding of the true optimum conditions.[473] Blachere et al.[473] also used an algorithm similar to Okabe and Aiba's to globally optimize a yeast fermentation plant. The subsystems used were medium preparation, fermentation, and extraction.

Constantinides et al.[474,475] and Blanch and Rogers[476] also considered problems of this type. For example, Blanch and Rogers[476] applied the concept of discrete maximum principle to the multistage continuous production of gramillicin S in which each stage was considered as a subsystem and evaluated the optimal pH, temperature, and the number of stages based on an objective function that accounted for, among others, raw material cost, product value,

and, most notably, the cost of antibiotic extraction. They also calculated the price of the product at which the operation ceased to be profitable.

2. Static (Point) Optimization

When reliable models are available, static optimization can be solved in a straightforward manner quite simply by applying the principles of calculus. The objective of static optimization is to minimize a nonlinear objective function:

$$J = J(\mathbf{x},\mathbf{u}) \tag{51}$$

subject to a set of generally nonlinear equality constraints derived from mathematical models:

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = 0 \tag{52}$$

Physical inequality constraints on the control variables and system states often have not received proper attention in the formulation of the minimization problem, and this has resulted in unrealistic and impractical controller settings. If solutions cannot be obtained analytically in a closed form, simplex and complex (*co*nstrained s*implex*) search techniques are generally relied on to yield solutions for simple linear or highly nonlinear problems.

Typical model-based optimization studies that are solved mathematically are the determination of holding time and recycle ratio to minimize the total cost during the design of a biological wastewater treatment plant[477] and the determination of the reactor pH and temperature and/or substrate concentration in the feed to maximize the biomass productivity in a chemostat.[478,479] Veres et al.[80] experimentally determined the empirical correlations of specific cell growth rate and specific product synthesis rate as functions of temperature and pH in a steady-state gluconic acid fermentation by *Acetobacter suboxydant*. The optimum operating conditions of temperature and pH were calculated through the use of these correlations and controlled with the aid of a computer.

When the proposed model is crude and contains unknown parameters whose values depend on the environmental conditions, the rigorous mathematical approach fails and experiments must be carried out to evaluate these parameters. Numerous batch experiments in which the initial glucose concentration and inoculum concentration were individually varied between each run were conducted by Endo et al.,[480] who investigated the effect of these variables on the performance of aerobic alcoholic fermentation by brewer's yeast. Semicontinuous three-dimensional surfaces of glucose-to-alcohol yield, alcohol level at the end of the batch, and the time required to complete the batch were generated over a region of the controllable variables. Thus, given an objective function in terms of the yield, alcohol level, and fermentation time, the optimum initial condition for the batch fermentation could be easily deduced.

When no model is available at all, the black-box approach to optimization relies totally on the hill-climbing search methods. Recently, a hypothetical example of this approach was given by Saguy,[481] who used the complex method of Box[448] to search for the most profitable set of values for the seven controllable variables in the "fermentation process development game", which was originally introduced by Mateles.[482] Schröder and Weide[483] carried out the experimental determination of the optimum salt proportions in the nutrient for cultivating a yeast. Their selection of experimental conditions was guided by the Box-Wilson gradient method[449] of (2^n)-factorial planning, where n is the number of parameters. The next study on optimum nutrient media formulation was performed by Votruba et al.[484] Using the direct search method of Rosenbrock,[451,452] they were able to reduce significantly the number of experiments required for a step improvement in the objective function from 2^n to $n + 1$.

All the examples mentioned thus far in this section are basically off-line optimization studies. On-line optimization of the operating conditions of a steady-state continuous bio-

reactor can be achieved by treating the whole fermentor as a black box. In this approach, computers are utilized to record the effect of different sets of conditions on the objective function. Upon the establishment of each steady state, the computer calculates the new value for the objective function, analyzes the dependence of the resulting patterns, and systematically determines a new set of conditions to be implemented next according to the search algorithm. By this way, the optimum conditions are sought out continuously through small excursions from the presently known set of best conditions while the production process is in progress. The process of on-line optimization is repeated until the potential gain from a still better performance is balanced with the economic loss induced by the systematic excursion.

The generally poor reproducibility implies that many iterations may be required to detect conclusively the improvement in performance through statistical analysis and hypothesis testings; the generally slow microbial response means that a long time is needed to reach the new steady state after each excursion. Hence, the on-line optimization of a continuous bioreactor is an extremely time-consuming process, and the improvement is likely to fade after the first few rounds unless the conditions used initially are grossly nonoptimal.

The continuous production of SCP from methanol was optimized with respect to the two-dimensional variable space of temperature and pH by Udén and Hedén.[485] The structured search pattern was constructed from Powell[453] and Hooke-Jeeves[450] algorithms by an on-line computer. However, no significant improvement over the initially used conditions was reported. Another study by Nyeste et al.[486] also searched the two-dimensional temperature and pH space with the Nelder-Mead simplex method[455] for the continuous *Hansenula anomala* fermentation. They tried to maximize the specific growth rate, or, equivalently, to minimize the generation time. Similar study was also reported by Szigeti et al.,[487] who achieved some success despite the inconveniences caused by the limited computer memory.

Poor reproducibility enhances the need for statistical analysis of variance to account for experimental variations encountered in the black-box approach of modeling and on-line optimization. This is especially so in the fermentation field where the experimental variations are large, and many repeated experiments are needed to detect even significant changes. Slow response can be partially countered by the careful statistical design of experiments so that a large amount of information can be extracted from the same number of experiments. The topic of statistical design of experiments has been covered in many books.[488-490] These ideas as applied to fermentation were discussed by Porter.[491] Among these ideas, one of the most widely accepted and commonly practiced methods of continuous optimization in other industrial processes is evolutionary operation (EVOP), originally presented by Box and Draper.[489]

EVOP was used by Chu et al.[492] in a two-stage waste treatment system with activated sludge recycle to test the effect of the volume ratio and inlet flow rate on the treatment efficiency. Within three rounds, they were able to improve the removal of chemical oxygen demand (COD) from the initial low of 55 to 82% at the conclusion of the third round. Bernard et al.[493] varied the temperature in a continuous biomass production process according to an optimization pattern, and they were able to reduce the objective function value by 37% over the original value in ten trials. In another study by Albrecht et al.,[494] the optimal operating points for a SCP production from methane were updated by cycling small increments of variations in the control variables (stirrer speed, methane-to-air ratio in the feed, and fermentation pressure). The variations in the natural gas composition were unavoidable, and this represented a source of disturbance. They estimated a saving of 5 to 7% in the cost as the result of implementing the computer-aided on-line optimization strategy. Another method that employed a model was also tried by them. In this method, the unknown disturbances are sometimes reflected in the model parameter. Thus, by updating the model parameters, they were able to maximize the SCP production from methane.

To minimize the potential loss from unsuccessful experiments, the experimental optimization need not really be constantly carried out. Instead, it can be performed during only a

certain assigned duration of the total operation time in order to assure that the operating conditions are not too far away from the optimum ones. Alternatively, the approach suggested by Muzychenko et al.[495] can be taken. They advocated the use of simple growth models and urged the estimation of model parameters from the data collected during the transient start-up period. However, they did not specify how the optimal temperature and pH, for example, can be obtained this way.

3. Dynamic (Path) Optimization

The dynamic optimization problem can be solved by employing optimal control theories.[358,496] The essence of the problem is to minimize:

$$J = \varphi[\mathbf{x}(t_f),t_f] + \int_{t_0}^{t_f} L[\mathbf{x}(t),\mathbf{u}(t),t]dt \tag{53}$$

subject to the system dynamic constraints:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},t) \qquad \text{given I.C.: } \mathbf{x}(t_0) \tag{54}$$

The straightforward way of solving the above problem is to formulate a Hamiltonian composed of the Lagrangian of Equation 53 and a vector of influence functions, λ:

$$H = L + \lambda^T\mathbf{f} \tag{55}$$

The necessary conditions for an optimal solution are given by:

$$\frac{d\lambda^T}{dt} = -\frac{\partial H}{\partial \mathbf{x}} \qquad \text{with the terminal condition: } \lambda(t_f) = \left[\frac{\partial \varphi}{\partial \mathbf{x}}\right]^T_{t=t_f} \tag{56}$$

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \tag{57}$$

Equations 54 and 56 form a two-point boundary value problem, the solution of which is generally carried out numerically by a computer. The integration of Equations 53 to 57 is usually not an easy task. Various final-value iteration methods or control vector iteration methods exist and must be exercised with extreme caution in searching for the optimal trajectory. The above methods approach iteratively the solution, moving in a direction determined by the gradient of the Hamiltonian. Rather frequently, however, flat plateaus of the Hamiltonian exist resulting in insensitive profiles that may be incorrectly mistaken for the optimal solution. The terminal time t_f is not always given and may be included in the set of variables to be optimized. This basic formulation can be expanded to include various constraints, e.g., equality constraints at the terminal time, equality and/or inequality constraints on state variables and/or control variables, interior point constraints, integral constraints, etc. In each case, the formulation of J, H, and the necessary conditions can be easily modified to take these extra constraints into consideration.

The optimal controller synthesis for approaching the desirable conditions has already been discussed in the control section. The optimal trajectory of the dilution rate for returning a continuous process to the optimum steady-state conditions was calculated by D'Ans et al.[497-499] The singular control problem resulted from the Hamiltonian function being linear in the control variable (dilution rate) was solved by transforming the time integral into a line integral, which was further transformed into a surface integral by applying Green's theorem. The maximum principle and Green's theorem were similarly utilized to calculate the optimum approach to steady state from start-up by Takamatsu et al.[500] They considered the problem of maximizing amino acid production in a given period of operation.

Green's theorem was also applied in the calculation of the optimum substrate feed rate to an amino acid (lysine) process by Ohno et al.[501] The tedious transformation procedure was circumvented by Yamané et al.,[502] who posed the problem in such a way that the trajectory of the specific growth rate was optimized, and the feeding policy was later related to this optimal trajectory. Weigand et al.[503,504] developed an optimal substrate feed policy for a repeated fed-batch bioreactor for biomass production when the initial and maximum reactor volume, the substrate concentration in the feed, and the final biomass concentration were prescribed. That was essentially a minimum time problem. In a more realistic situation, perhaps these prescribed variables should be optimized as well, based on an objective function maximizing the profit.

Singular control problem was also encountered in a study by Fishman and Biryukov.[505] The penicillin biosynthesis in a fed-batch fermentor was modeled with due consideration to the cell age and optimized with regards to the substrate addition rate. However, in the authors' opinion, their formulation of the optimization problem did not properly take into consideration the change of volume which is characteristic of a fed-batch operation. It is the neglect of dilution effects that enabled them to arrive at an analytical solution. Andreyeva and Biryukov[398] modeled the effect of pH on a similar process and calculated the optimal substrate feeding profile. It should be pointed out that the optimization would have been more practical if the terminal time was not fixed but allowed to vary and be optimized simultaneously.

Constantinides et al.[474] developed a mathematical model for batch penicillin fermentation and estimated the model parameters from the experimental data, considering especially the temperature dependence of these parameters. In a following study,[475] the optimal temperature profile was determined based on the model developed earlier. King et al.[506] considered the same penicillin problem except that only piecewise constant temperature settings were allowed. This suboptimal strategy was much simpler to implement and resulted in only a 2% decrease in penicillin production compared to the optimal trajectory.

A similar sequence of steps of model construction and model parameter estimation and the application of the Pontryagin's continuous maximum principle was followed by Rai and Constantinides[507,508] in the optimization of a batch gluconic acid fermentation with respect to temperature and pH. They found that the optimum temperature and pH profiles stayed at constant values over most part of the fermentation duration.[507] This point prompted Cheruy and Durand[111] to conduct an experiment in which both pH and temperature were set at constant values throughout the batch erythromycin fermentation. The experimental behavior resulted from the implementation of the simplified control strategy was about the same as that resulted from the optimal control policy, which was determined after a similar sequence of steps of model construction, parameter estimation, and mathematical optimization. They also used many statistical criteria to evaluate the validity of the proposed model based on the experimental data.

Nyeste et al.[509] proposed a structured mathematical model for a batch gluconic acid fermentation. The parameters' temperature dependence was considered by performing a series of runs at different temperatures, and the model parameters were subsequently expressed empirically as polynomial functions of temperature, with the data fitting performed by the method of Hooke and Jeeves.[450] Pontryagin's maximum principle was applied to find the optimum temperature trajectory during the batch run.

Based on a set of simple mathematical models for an antibiotic (turimycin) fermentation, Guthke and Knorre[510] calculated the optimal substrate profile and compared the performance of the corresponding suboptimal, but easily implemendable, trajectories. The improvement of performance resulting from the optimal trajectory over the suboptimal trajectories is only minimal, as concluded consistently in other optimization studies.

The optimal temperature and pH profiles for an enzyme reactor subject to deactivation were determined by Ho and Humphrey.[511] The substrate inhibition and enzyme deactivation

effects were considered by San and Stephanopoulos[512] to obtain a time-varying singular feeding policy in a continuous reactor. The modification of the control policy to include feed-forward control in the presence of time lag was also considered in a later study.[513]

An optimal quality control for fed-batch baker's yeast production was recently carried out by Dairaku et al.[514] This study was aimed at achieving a low value of the ratio of budding cells to total cells (so that the fermentative activity in the making of bread was maximized) by following an optimal trajectory of the specific growth rate. The trajectory was determined by the method of sensitivity function analysis based on a dynamic population model.

Because of selection and enrichment, the microbial behavior and properties are likely to vary over the years, or even from one run to another. The model parameters are also destined to change from time to time because of various unknown disturbances to the system. Furthermore, microbial conditions depend not only on the present state but also on the past histories. For example, the inoculum may not be uniform, and the duration of the lag phase after inoculation may vary. Thus, the periodic updating of model parameters is certainly desirable. However, unlike the black-box approach used in optimizing steady-state conditions, systematic search methods are not applicable to dynamic optimization problems. Nevertheless, by comparing the actual behavior to the expected behavior calculated from an existing model, it is possible to program the computer to elucidate the changes in the values of the parameters. Regretably, the on-line reevaluation of the kinetic parameters, which requires some creative thinking, is not presently practiced.

The effectiveness of the optimal control strategies are rarely verified through actual experiments. Many optimal problems are based on highly hypothetical assumptions. For example, the validity of the starting model is often not established in advance. Many state variables included in the optimization consideration are not practically measurable, and this prevents meaningful studies using closed loop feedback control from being actually carried out.

Even when the optimal path is implemented experimentally, the control is actually a feed-forward open-loop control. Disturbances to the system and uncertainties in the model and its parameters are likely to cause the actual path to deviate away from the calculated path. Therefore, there is a need to monitor the state of the system and to use the measurements to complete the feedback control loop. Nestaas and Wang[121,443] were able to overcome these difficulties by monitoring the state of the system in a penicillin fermentation with an ingenious filter sensor.

Weinrich and Lapidus[515] proposed a feedback controller whose feedback gain was designed in such a way that the variation in model parameters could be inferred from the combination of the known control history and the measured deviation of the actual state away from the nominal path. The required adjustment in the controller output due to parameter variation was calculated by linearizing the controller output about the nominal optimal control trajectory. That controller was, therefore, well suited for small variations in parameters, but more significant variations in model parameters could also be accommodated with only a slight loss in optimality.

Because optimal control strategies are rarely carried out, it is not yet conclusive as to how much more can be gained from the implementation of such policies. The improvement from the use of sophisticated optimal control may not be substantial, the optimal control policy may be difficult to implement due to observability problems, the model used may not be totally reliable, or the model parameters may not be accurately estimated. In all these cases, the use of suboptimal policy may be justified and, in fact, preferred. Thus, once again, the possible gain from optimization must be balanced by the cost in the added manipulation, and simple control may occasionally come ahead.

VI. FUTURE TRENDS OF COMPUTER APPLICATION

Computers will definitely be an integral and natural part of the fermentation system in the years to come. They will prove to be a highly vital instrument for both of those engaged in research and production. However, the success of computer application, as pointed out by Dobry and Jost[13] and Armiger and Humphrey,[7] depends on the development of better on-line instrumentation and reliable mathematical models. The prerequisite to a true and meaningful optimization of a fermentation process is an accurate mathematical model that is capable of describing various behaviors of the microbial system under consideration. The prerequisite to a successful mathematical model, on the other hand, is the availability of sensors to detect and accurately record, either directly or indirectly, various types of relevant microbial responses. This relationship between computers and sound biological models is a two-way interaction. Not only computer-implemented optimization of a fermentation process requires accurate models, but the construction of representative models can be accomplished only through the extensive utilization of computers. This is especially true in the study of fundamental microbial metabolic and synthesis pathways where the intracellular feedback control mechanisms work at a relatively short time scale compared to the frequency of sampling in the studies of these internal regulatory mechanisms. Therefore, by employing on-line sensors and fully utilizing computers for data conditioning and analysis, it is possible that the unique complexity of biological processes will ultimately be understood to allow for the formulation of fundamentally correct and accurate models. Then, we can have a real chance of approaching optimization problems realistically with highly efficient and productive process designs and reliable control strategy formulations.

Presently, the application of closed-loop control with a computer has been limited considerably because of the scarcity of reliable on-line sensors for monitoring key parameters in a fermentor. The succeeding work on modeling, control, and optimization will proceed to the extent allowed by the development of the necessary sensors. As instrumentation technology improves, the emphasis in the use of computers will gradually shift from collecting and storing experimental data to that of interpreting the microbial dynamic behavior, preferably in the context of simple mathematical models which contain no ambiguous, experimentally indeterminable parameters. Although not as rigorous as the theoretically based mathematical models, accurate empirical correlations capable of predicting the effects of a change in control variables on the response of microbial system may also be very useful in the formulation of control strategies. After modeling, the next logical challenge in this field will come from the formulation of a broadly applicable "smart" adaptive control algorithm that is capable of self-optimization. When all these are done, a fermentation technologist may leave the tedious routine to the machine and devote more of his time and energy to the more creative and challenging task of developing new, novel microbial processes to meet the needs of the future.

VII. CONCLUSIONS

An attempt has been made in this paper to provide the readers with a relatively broad perspective on recent developments in the application of computers to fermentation processes. Among the topics reviewed in this paper were applications in the areas of on-line data acquisition, analysis, process modeling, control, and optimization. At present, we are still some distance away from reaching our final intended objective of optimally controlling the whole fermentation process. Nevertheless, intense effort by both industry and academia has fruitfully resulted in significant progress in many aspects of computer applications. If the early 1970s mark the beginning of the application of computers in the fermentation field, then we now stand at the end of the beginning. Problems directly associated with computers such as hardware, software, and interfacing seem to have been thoroughly solved. To be

sure, there are still areas where improvements can be made by tapping more of the power that can be realized by applying computer technology, but, overall, it seems that we are now properly equipped and are ready to advance to the next step. In this context, the optimistic views expressed by Armiger and Humphrey[7] that the time has come for us to upgrade computer applications to fermentation technology to the present level of computer technology and possibly beyond are closer to realization than ever before.

NOMENCLATURE

A	The matrix in the balance equation $\mathbf{Ax} = \mathbf{O}$ (dimension = m × n)
a,b,c,d,e,f	Stoichiometric coefficients in Equation 7; a [=] mole substrate/C mole biomass; b [=] mole O_2/C mole biomass; c [=] mole NH_3/C mole biomass; d [=] mole H_2O/C mole biomass; e [=] mole CO_2/C mole biomass; f [=] mole product/C mole biomass
B	Biomass stoichiometric coefficients in Equation 7
b	Biomass concentration; b [=] g/ℓ
C	Linearization matrix in Equation 47
C_j	Concentration of component j in the system; C_j [=] mol/ℓ
$C^*_{o_2}$	Liquid phase oxygen concentration in equilibrium with the gas phase oxygen concentration; $C^*_{o_2}$ [=] mol O_2/ℓ
C_{o_2}	Oxygen concentration in the fermentation liquid; C_{o_2} [=] mol O_2/ℓ
CER	Carbon dioxide evolution rate; CER [=] mole CO_2/ℓ–hr
D	Dilution rate D [=] hr^{-1}
D_{ss}	Dilution rate at steady state; D_{ss} [=] hr^{-1}
EC	Energy charge, as defined in Equation 5
F	Linearization matrix in Equation 47
f	State dynamic equation
G	Linearization matrix in Equation 47
H	Linearization matrix in Equation 48
H	Hamiltonian in Equation 55
H_j	Heat of formation of species j in Equation 13; H_j [=] kcal/mol
h	Measurement equation
h	Test function
J	Performance index
k_La	Volumetric mass transfer coefficient; k_La [=] hr^{-1}
L	Lagrangian in Equation 53
m	Number of constraint equations
m_{ATP}	Maintenance requirement for ATP; m_{ATP} [=] mole ATP/C mole biomass-hr
$m_{o_2,s}$	Oxygen-to-cell maintenance coefficient
m_s	Substrate-to-cell maintenance coefficient
n	Number of measurements
OUR	Oxygen utilization rate; OUR [=] mole O_2/ℓ-hr
P	Variance-covariance matrix for $\hat{\mathbf{x}}$; $\mathbf{P} = \mathbf{E}[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T]$
P/O	Efficiency of the oxidative phosphorylation; P/O [=] mole ATP/mole O
Q	Variance-covariance matrix for $\boldsymbol{\zeta}$; $\mathbf{Q} = \mathbf{E}[\boldsymbol{\zeta}\boldsymbol{\zeta}^T]$
Q	Heat evolution in Equation 6; Q [=] kcal/C mole biomass
q_{o_2}	Volumetric oxygen transfer rate; q_{o_2} [=] mole O_2/ℓ-hr
R	Rate of biomass formation; extent of Reaction 7; R [=] mole biomass/ℓ-hr
RQ	Respiratory quotient; RQ [=] mole CO_2/mole O_2
S	Variance-covariance matrix for $\boldsymbol{\xi}$; $\mathbf{S} = \mathbf{E}[\boldsymbol{\xi}\boldsymbol{\xi}^T]$
s	Limiting substrate concentration; s [=] g/ℓ
t	Time
t_o	Initial time
t_f	Final time
u	Control vector in Equation 47
V	Volume; V [=] liter
W	Disturbance vector in Equation 47
x	State vector

x,y,z	Composition of substrate in Equation 7
y	True parameter vector for measured variables (dimension = n)
$\bar{y}$	Measurement vector (dimension = n)
Y_{ATP}^{max}	Maximum growth yield on ATP; Y_{ATP}^{max} [=] C mole biomass/mole ATP
Y_{x/o_2}	Oxygen to cell yield coefficient
$Y_{x/s}$	Substrate to cell yield coefficient

Greek Letters

$\alpha,\beta,\gamma,\delta$	Composition of biomass in Equation 7
$\alpha',\beta',\gamma',\delta'$	Composition of metabolic product in Equation 7
δ	Measurement error vector (dimension = n)
δ	Estimate of δ (dimension = n)
ϵ	Balance equation residual vector (dimension = m)
E	Expected value operator
ζ	Random disturbance to the dynamic system
$1 - \vartheta$	Confidence level of χ^2 distribution
λ	Influence function in Equation 55
ξ_1—ξ_5	Extents of reactions in Equations 39a to 39e
μ	Specific growth rate; μ [=] hr^{-1}
ν_j	Stoichiometric coefficient of component j
ξ	Random noise in measurement
Φ_j	Net rate of input of component j to the system by transport; Φ_j [=] mol/hr
ϕ	Variance-covariance matrix for ϵ with no gross error (dimension = m × m); $\phi = E[\epsilon\epsilon^T]$
ϕ	Terminal function be minimized in Equation 53
σ	Standard deviation
χ^2	Chi-square probability distribution
ψ	Variance-covariance matrix for δ (dimension = n × n); $\psi = E[\delta\delta^T]$

ACKNOWLEDGMENT

Financial support for this work was provided by the National Science Foundation Grant Number CPE-8118877.

REFERENCES

1. **Flynn, D. S.**, Social aspects of computer applications in fermentation, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 9.
2. **Toong, H. D. and Gupta, A.**, Personal computers, *Sci. Am.*, 247(6), 87, 1982.
3. **Smith, R. J.**, Computers for fermentation. What else can we do?, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 1.
4. **Falch, E. A., Hjortkjaer, P., and Pedersen, P.**, Computer applications in industry: justification and planning, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 207.
5. **Cooney, C. L.**, Computer application in fermentation technology — a perspective, *Biotechnol. Bioeng. Symp.*, 9, 1, 1979.
6. **Fuld, G. J.**, Control applications in fermentation processes, *Adv. Appl. Microbiol.*, 2, 351, 1960.
7. **Armiger, W. B. and Humphrey, A. E.**, Computer applications in fermentation technology, in *Microbial Technology*, Vol. 2, 2nd ed., Peppler, H. J. and Perlman, D., Eds., Academic Press, New York, 1979, chap. 15.
8. **Murao, C. J. and Yamashita, S.**, Fermentation processes, *Soc. Instrum. Control Eng. J. (Tokyo)*, 6, 735, 1967.

9. **Koga, S., Burge, C. R., and Humphrey, A. E.,** Computer simulation of fermentation systems, *Appl. Microbiol.*, 15, 683, 1967.
10. **Yamashita, S., Hoshi, H., and Inagaki, T.,** Automatic control and optimization of fermentation processes: glutamic acid, in *Fermentation Advances*, 3rd Int. Fermentation Symp., Perlman, D., Ed., Academic Press, New York, 1969, 441.
11. **Grayson, P.,** Computer control of batch fermentations, *Process. Biochem.*, 4 (March), 43, 1969.
12. Proc. 1st European Conf. on Computer Process Control in Fermentation, Dijon, France, September 3 to 5, 1973.
13. **Dobry, D. D. and Jost, J. L.,** Computer applications to fermentation operations, in *Annual Reports on Fermentation Processes*, Vol. 1, Perlman, D., Ed., Academic Press, New York, 1977, chap. 5.
14. **Humphrey, A. E.,** Computer-assisted fermentation development, *Dev. Ind. Microbiol.*, 18, 58, 1977.
15. **Humphrey, A. E.,** Computer-assisted fermentation development, *Dev. Ind. Microbiol.*, 18, 58, 1977.
16. Computer applications to fermentation processes, in *Biotechnology and Bioengineering Symp. Series*, No. 9, John Wiley & Sons, New York, 1979.
17. *Computer Application in Fermentation Technology*, 3rd Int. Conf. on Computer Application in Fermentation Technology, Society of Chemical Industry, London, 1982.
18. **Falch, E. A., Hjortkjaer, P., and Pedersen, P.,** Computer applications in industry: justification and planning, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London 1982, 207.
19. **Nyiri, L. K.,** Application of computers in biochemical engineering, in *Advances in Biochemical Engineering*, Vol. 2, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1972, chap. 2.
20. **Nyiri, L. K.,** A philosophy of data acquisition, analysis, and computer control of fermentation processes, in Proc. of LABEX Symp. on Computer Control of Fermentation Processes, London, 1971, 16.
21. **Weigand, W. A.,** Computer applications to fermentation processes, in *Annual Reports on Fermentation Processes*, Vol. 2, Perlman, D., Ed., Academic Press, New York, 1978, chap. 3.
22. **Zabriskie, D. W.,** Real-time applications of computers in fermentation processes, *Ann. N.Y. Acad. of Sci.*, 326, 223, 1979.
23. **Rolf, M. J. and Lim, H. C.,** Computer control of fermentation processes, *Enzyme Microb. Technol.*, 4, 370, 1982.
24. **Hatch, R. T.,** Computer applications for analysis and control of fermentation, in *Annual Reports on Fermentation Processes*, Vol. 5, Tsao, G. T., Ed., Academic Press, New York, 1982, chap. 8.
25. **Bull, D. N.,** Automation and optimization of fermentation processes, in *Annual Reports on Fermentation Processes*, Vol. 6, Tsao, G. T., Ed., Academic Press, New York, 1983, chap. 13.
26. **Ryu, D. D. Y. and Humphrey, A. E.,** Examples of computer-aided fermentation systems, *J. Appl. Chem. Biotechnol.*, 23, 283, 1973.
27. **Dobry, D. D. and Jost, J. L.,** IMPAC/FORTRAN schemes for fermentation data, *Biotechnol. Bioeng. Symp.*, 9, 39, 1979.
28. **Corrieu, G., Blachere, H., and Geranton, A.,** Acquisition and handling of data by computer in fermentation installations, *Biotechnol. Bioeng. Symp.*, 4, 607, 1973.
29. **Cordonnier, M., Kernevez, J. P., Lebeault, J. M., and Kryze, J.,** Microprocessor in fermentation control, *Biotechnol. Bioeng. Symp.*, 9, 227, 1979.
30. **Meiners, M. and Rapmundt, W.,** Some practical aspects of computer applications in a fermentor hall, *Biotechnol. Bioeng.*, 25, 809, 1983.
31. **Jefferis, R. P., III,** Software and file structures for computer-coupled pilot fermentation plants, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 21.
32. **Breame, A. J. and Spier, R.,** Down market computers for research, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 13.
33. **Lee, S. E. and Berenbach, B. A.,** The use of a minicomputer at Lederle's fermentation pilot plant, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 75.
34. **Jokela, J., Oinas, R., and Meskanen, A.,** Combining microprocessor based instrumentation systems with a fermentation oriented mini-computer system, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 85.
35. **Kuratani, T. and Gato, Y.,** Computer system of fermentation processes, *Kagaku Kogyo*, 31, 900, 1980.
36. **Unden, A., Rindone, W. P., and Heden, C.-G.,** The computer in fermentation process research, *Proc. Biochem.*, 14(2), 8, 1979.
37. **Bowski, L., Perley, C., and West, J. M.,** A minicomputer system for analyzing and reporting pilot plant fermentor data, *Biotechnol. Bioeng.*, 25, 1237, 1983.

38. **Nyiri, L. K., Toth, G. M., Krishnaswami, C. S., and Parmenter, D. V.**, Assessment of on-line computer operations with fermentors, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 31.
39. **Hampel, W., Wöhrer, W., Bach, H. P., and Röhr, M.**, Automatic analysis of fermentation parameters, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 32.
40. **Armiger, W. B., Zabriskie, D. W., and Humphrey, A. E.**, Design and operational experience with a highly instrumented computer coupled fermentation system, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 33.
41. **Jefferis, R. P., III**, Software structure for a computer-coupled fermentation pilot plant, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 34.
42. **Metz, H. and Wenzel, F.**, Computer controlled fermentation pilot plant, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 35.
43. **Müller, F.**, Instrumentation for process control, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 36.
44. **Fox, R. I.**, A market survey of computer systems for fermentation control, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 91.
45. **Hampel, W., Bach, H. P., and Röhr, M.**, Low cost data acquisition and analysis with on-line desk calculators, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 47.
46. **Forrest, E. H., Jansen, N. B., Flickinger, M. C., and Tsao, G. T.**, A simple hobby computer-based off-gas analysis system, *Biotechnol. Bioeng.*, 23, 455, 1981.
47. **Bungay, H. R.**, Inexpensive computers aid fermentation, *Chem. Eng. Prog.*, 76(4), 53, 1980.
48. **Mohler, R. D., Hennigan, P. J., Lim, H. C., Tsao, G. T., and Weigand, W. A.**, Development of a computerized fermentation system having complete feedback capabilities for use in a research environment, *Biotechnol. Bioeng. Symp.*, 9, 257, 1979.
49. **Halme, A.**, Computer control of the Pekilo protein process, *Biotechnol. Bioeng. Symp.*, 9, 369, 1979.
50. **Lundell, R.**, Computer control of fermentation plants, *Biotechnol. Bioeng. Symp.*, 9, 381, 1979.
51. **Rolf, M. J., Hennigan, P. J., Mohler, R. D., Weigand, W. A., and Lim, H. C.**, Development of a direct digital-controlled fermentor using a microminicomputer hierarchical system, *Biotechnol. Bioeng.*, 24, 1191, 1982.
52. **Blachere, H. T. and Peringer, P.**, New developments in instrumentation and control in fermentation, in *Biotechnology*, DECHEMA Monograph Band 82, 1st European Congr. Biotechnology, Verlag Chemie, New York, 1978, 65.
53. **Alford, J. S.**, Evolution of the fermentation computer system at Eli Lilly & Co., in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 67.
54. **Hennigan, P. J., Rolf, M. J., Weigand, W. A., and Lim, H. C.**, A microminicomputer control system for fermentation research, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 99.
55. **Valentini, L., Andreoni, F., Buvoli, M., and Pennella, P.**, A computer system for fermentation research: objectives, fulfilment criteria and use, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 175.
56. **Armiger, W. B. and Moran, D. M.**, Review of alternatives and rationale for computer interfacing and system configuration, *Biotechnol. Bioeng. Symp.*, 9, 215, 1979.
57. **Flynn, D. S.**, Computer control of fermentation processes, *Biotechnol. Bioeng. Symp.*, 4, 597, 1973.
58. **Meskanen, A.**, Design of the man-machine interface for computer coupled fermentation, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 73.
59. **Hampel, W. A.**, Application of microcomputers in the study of microbial processes, in *Advances in Biochemical Engineering*, Vol. 13, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1979, 1.
60. **Beaverstock, M. C. and Trearchis, G. P.**, Fermentation control systems — a time for change, *Biotechnol. Bioeng. Symp.*, 9, 241, 1979.
61. **Sargent, M. and Shoemaker, R. L.**, *Interfacing Microcomputers to the Real World*, Addison-Wesley, Reading, Mass., 1981.
62. **Horowitz, P. and Hill, W.**, *The Art of Electronics*, Cambridge University Press, Cambridge, 1980.
63. **Selva, G. and Zerbetto, P.**, Software engineering experience in a consulting agency, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 11.
64. **Jefferis, R. P., III, Klein, S. S., and Drakeford, J.**, Single-board microcomputer for fermentation control, *Biotechnol. Bioeng. Symp.*, 9, 231, 1979.

65. **Jefferis, R. P., III**, Fault analysis methods for production fermentation control, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 199.
66. **Wang, N. S. and Stephanopoulos, G.**, Application of macroscopic balances to the identification of gross measurement errors, *Biotechnol. Bioeng.*, 25, 2177, 1983.
67. **Undén, A.**, Instrumentation for fermentation monitoring and control, *Biotechnol. Bioeng. Symp.*, 9, 61, 1979.
68. **Tannen, L. P. and Nyiri, L. K.**, Instrumentation of fermentation systems, in *Microbial Technology*, Vol. 2, 2nd ed., Peppler, H. J. and Perlman, D., Eds., Academic Press, New York, 1979, chap. 14.
69. **Aiba, D., Humphrey, A. E., and Mills, N. F.**, Operation, measurement and control, in *Biochemical Engineering*, Academic Press, New York, 1965, chap. 10.
70. **Wang, D. I. C., Cooney, C. L., Demain, A. L., Dunnill, P., Humphrey, A. E., and Lilly, M. D.**, Instrumentation and control, in *Fermentation and Enzyme Technology*, John Wiley & Sons, New York, 1979, chap. 11.
71. Guide to Scientific Instruments, supplement of *Science*, 194(4267A), 1976; 197(4309A), 1977; 1978—79; 1978; 1980—81, 1980; 1981—82, 1981; 1982—83, 1982; published each year by American Association for the Advancement of Science, Washington, D.C.
72. **Vincent, A. and Priestley, G.**, The measurement of process variables, in *Topics in Enzyme and Fermentation Biotechnology*, Vol. 4, Wiseman, A., Ed., Ellis Horwood, Sussex, England, 1980, chap. 5.
73. **Jefferis, R. P., III**, Parameter estimation and digital filter techniques, *Ann. N.Y. Acad. of Sci.*, 326, 241, 1979.
74. **Harmes, C. S., III**, Design criteria of a fully computerized fermentation system, *Dev. Ind. Microbiol.*, 13, 146, 1972.
75. **Solomons, G. L.**, *Material and Methods in Fermentation*, Academic Press, New York, 1969.
76. **Mor, J.-R.**, A review of instrumental analysis in fermentation technology, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 109.
77. **Fleischaker, R. J., Giard, D. J., Weaver, J., and Sinskey, A. J.**, Progress with computer-coupled mammalian cell culture investigations, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 425.
78. **Fleischaker, R. J., Weaver, J. C., and Sinskey, A. J.**, Instrumentation for process control in cell culture, in *Advances in Applied Microbiology*, Vol. 27, Perlman, D., Ed., Academic Press, New York, 1981, 137.
79. **Wick, E.**, Controlling temperature by direct liquid-liquid heat exchange in continuous bioreactors, *Biotechnol. Bioeng.*, 23, 1399, 1981.
80. **Veres, A., Nyeste, L., Kurucz, I., Kirchknopf, L., Szigeti, L., and Holló, J.**, Automated fermentation equipment. I. Program-controlled fermentor, *Biotechnol. Bioeng.*, 23, 391, 1981.
81. **Jolicoeur, C.**, Thermodynamic flow methods in biochemistry: calorimetry, densimetry, and dilatometry, in *Methods of Biochemical Analysis*, Vol. 27, Interscience, New York, 1981, 171.
82. **Brauer, H.**, Power consumption in aerated stirred tank reactor systems, in *Advances in Biochemical Engineering*, Vol. 13, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1979, 87.
83. **Charles, M.**, Technical aspects of the rheological properties of microbial cultures, in *Advances in Biochemical Engineering*, Vol. 8, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1978, 1.
84. **Perley, C. R., Swartz, J. R., and Cooney, C. L.**, Measurement of cell mass concentration with a continuous-flow viscometer, *Biotechnol. Bioeng.*, 21, 519, 1979.
85. **Bongenaar, J. J. T. M., Kossen, N. W. F., Metz, B., and Meijboom, F. W.**, A method for characterizing the rheological properties of viscous fermentation broths, *Biotechnol. Bioeng.*, 15, 201, 1973.
86. **Solomons, G. L.**, Fermentation equipment, in *Advances in Applied Microbiology*, Vol. 14, Perlman, D., Ed., Academic Press, New York, 1971, 169.
87. **Roels, J. A., van den Berg, J., and Voncken, R. M.**, The rheology of mycelial broths, *Biotechnol. Bioeng.*, 16, 181, 1974.
88. **Blanch, H. W. and Bhavaraju, S. M.**, Non-Newtonian fermentation broths: rheology and mass transfer, *Biotechnol. Bioeng.*, 18, 745, 1976.
89. **Shimmons, B. W., Svrcek, W. Y., and Zajic, J. E.**, Cell concentration control by viscosity, *Biotechnol. Bioeng.*, 18, 1793, 1976.
90. **LeDuy, A., Marsan, A. A., and Coupal, B.**, A study of the rheological properties of a non-Newtonian fermentation broth, *Biotechnol. Bioeng.*, 16, 61, 1974.
91. **Thomson, N. and Ollis, D. F.**, Extracellular microbial polysaccharides. II. Evolution of broth power-law parameters for xanthan and pullulan batch fermentation, *Biotechnol. Bioeng.*, 22, 875, 1980.
92. **Silman, R. W. and Bagley, E. B.**, The viscostat: productstat method of feedrate control in continuous fermentations, *Biotechnol. Bioeng.*, 21, 173, 1979.

93. **Kok, R. and Zajic, J. E.**, A mechanical foam breaker, *Biotechnol. Bioeng.*, 17, 273, 1975.
94. **Gualandi, G.**, Mechanical devices and control instruments in pilot plant fermentors, *Biotechnol. Bioeng. Symp.*, 4, 799, 1973.
95. **Soifer, R. D., Gorskaya, S. V., and Ivankova, T. A.**, Foaming as a control factor of fermentation, *Biotechnol. Bioeng. Symp.*, 4, 755, 1973.
96. **Lalchev, Zdr. and Exerowa, D.**, Concentration of proteins by foaming, *Biotechnol. Bioeng.*, 23, 669, 1981.
97. **Lee, S. B. and Ryu, D. D. Y.**, Separation of gentamicin by foaming, *Biotechnol. Bioeng.*, 21, 2045, 1979.
98. **Seidel, H.**, Enzymatic analysis and microbiological process control, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 27.
99. **Eisenman, G., Ed.**, *Glass Electrodes for Hydrogen and Other Cations*, Marcel Dekker, New York, 1967.
100. **Chapman, A. G. and Atkinson, D. E.**, Adenosine nucleotides concentrations and turnover rates, their correlation with biological activity in bacteria and yeast, *Adv. Microbiol. Physiol.*, 15, 253, 1977.
101. **Forsberg, C. W. and Lam, K.**, Use of ATP as an indicator of the microbial biomass in rumen contents, *Appl. Environ. Microbiol.*, 33, 528, 1977.
102. **Holm-Hansen, O. and Karl, D. M.**, Biomass and adenylate energy charge determination in microbial cell extracts and environmental samples, in *Methods of Enzymology*, Vol. 57, DeLuca, M. A., Ed., Academic Press, New York, 1978, 73.
103. **Hendy, N. A. and Gray, P. P.**, Use of ATP as an indicator of biomass concentration in the *Trichoderma viride* fermentation, *Biotechnol. Bioeng.*, 21, 153, 1979.
104. **Hysert, D. W., Knudsen, F. B., Morrison, N. M., Van Gheluwe, G., and Lom, T.**, Application of a bioluminescence ATP assay in brewery wastewater treatment studies, *Biotechnol. Bioeng.*, 21, 1301, 1979.
105. **Zabriskie, D. W.**, Use of culture fluorescence for monitoring of fermentation systems, *Biotechnol. Bioeng. Symp.*, 9, 117, 1979.
106. **Sakato, K., Tanaka, H., and Samejima, H.**, Electrochemical measurements of cell populations, *Ann. N.Y. Acad. Sci.*, 369, 321, 1981.
107. **Harrison, D. E. F. and Chance, B.**, Flourimetric technique for monitoring changes in the level of reduced nicotinamide nucleotides in continuous culture of microorganisms, *J. Appl. Microbiol.*, 19, 446, 1970.
108. **Zabriskie, D. W. and Humphrey, A. E.**, Estimation of fermentation biomass concentration by measuring culture fluorescence, *Appl. Environ. Microbiol.*, 35, 337, 1978.
109. **Matsunaga, T., Karube, I., and Suzuki, S.**, Electrochemical microbioassay of vitamin B_{12}, *Anal. Chim. Acta*, 98, 25, 1978.
110. **Matsunaga, T., Karube, I., and Suzuki, S.**, Electrode system for the determination of microbial populations, *Appl. Environ. Microbiol.*, 37, 117, 1979.
111. **Cheruy, A. and Durand, A.**, Optimization of erythromycin biosynthesis by controlling pH and temperature: theoretical aspects and practical application, *Biotechnol. Bioeng. Symp.*, 9, 303, 1979.
112. **Gencer, M. A. and Mutharasan, R.**, Determination of biomass concentration by capacitance measurement, *Biotechnol. Bioeng.*, 21, 1097, 1979.
113. **Lee, Y. H.**, Pulsed light probe for cell density measurement, *Biotechnol. Bioeng.*, 23, 1903, 1981.
114. **Ohashi, M., Watabe, T., Ishikawa, T., Watanabe, Y., Miwa, K., Shoda, M., Ishikawa, Y., Ando, T., Shibata, T., Kitsunai, T., Kamiyama, N., and Oikawa, Y.**, Sensors and instrumentation: steam-sterilizable dissolved oxygen sensor and cell sensor for on-line fermentation system control, *Biotechnol. Bioeng. Symp.*, 9, 103, 1979.
115. **Hancher, C. W., Thacker, L. H., and Phares, E. F.**, A fiber-optic retroreflective turbidimeter for continuously monitoring cell concentration during fermentation, *Biotechnol. Bioeng.*, 16, 475, 1974.
116. **Lee, C. and Lim, H.**, New device for continuously monitoring the optical density of concentrated microbial cultures, *Biotechnol. Bioeng.*, 22, 639, 1980.
117. **Pirt, S. J.**, The maintenance energy of bacteria in growing cultures, *Proc. R. Soc. London Ser. B*, 163, 224, 1965.
118. **Kubitschek, H. E.**, Counting and sizing micro-organisms with the Coulter counter, in *Methods in Microbiology*, Vol. 1, Norris, J. R. and Ribbons, D. W., Eds., Academic Press, New York, 1969, chap. 17.
119. **Nestaas, E. and Wang, D. I. C.**, A new sensor, the "filtration probe", for quantitative characterization of the penicillin fermentation. I. Mycelial morphology and culture activity, *Biotechnol. Bioeng.*, 23, 2803, 1981.
120. **Nestaas, E., Wang, D. I. C., Suzuke, H., and Evans, L. B.**, A new sensor, the "filtration probe", for quantitative characterization of the penicillin fermentation. II. The monitor of mycelial growth, *Biotechnol. Bioeng.*, 23, 2815, 1981.
121. **Nestaas, E. and Wang, D. I. C.**, A new sensor, the "filtration probe", to monitor and control antibiotic fermentations, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 433.

122. **Nestaas, E. and Wang, D. I. C.,** A new sensor — the "filtration probe" — for quantitative characterization of penicillin fermentation. III. An automatically operating probe, *Biotechnol. Bioeng.*, 25, 1981, 1983.
123. **Swartz, J. R. and Cooney, C. L.,** Instrumentation in computer-aided fermentation, *Process Biochem.*, 13 (February), 3, 1978.
124. **Spruytenburg, R., Dunn, I. J., and Bourne, J. R.,** Computer control of glucose feed to a continuous aerobic culture of *Saccharomyces cerevisiae* using the respiratory quotient, *Biotechnol. Bioeng. Symp.*, 9, 359, 1979.
125. **Fireoved, R. L., Mutharasen, R., and Lee, Y. H.,** Measurement of gas-phase oxygen concentrations with an oxygen electrode, *Biotechnol. Bioeng.*, 24, 2109, 1982.
126. **Pungor, E., Schaefer, E., Weaver, J. C., and Cooney, C. L.,** Direct monitoring of a fermentation in a computer-mass spectrometer-fermentor systems, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 393.
127. **Reuβ, M., Piehl, H., and Wagner, F.,** Application of mass spectrometry to the measurement of dissolved gases and volatile substances in fermentation, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 25.
128. **Buckland, R. C. and Fastert, H.,** Analysis of fermentation exhaust gas using a mass spectrometer, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 119.
129. **Johnson, M. J. and Borkowski, J.,** Steam sterilizable probes for dissolved oxygen measurement, *Biotechnol. Bioeng.*, 6, 457, 1964.
130. **Harrison, D. E. F. and Melbourne, K. V.,** An autoclavable version of the Mackereth oxygen probe, *Biotechnol. Bioeng.*, 12, 633, 1970.
131. **Aiba, S. and Huang, S. Y.,** Some consideration on the membrane-covered electrode, *J. Ferment. Technol.*, 47, 372, 1969.
132. **Elsworth, R.,** The value and use of dissolved oxygen measurement in deep culture, *Chem. Eng.*, No. 258, 63, 1972.
133. **Lee, Y. H. and Tsao, G. T.,** Dissolved oxygen electrodes, in *Advances in Biochemical Engineering*, Vol. 13, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1979, 35.
134. **Krebs, W. M. and Haddad, I. A.,** The oxygen electrode in fermentation systems, *Dev. Ind. Microbiol.*, 13, 113, 1972.
135. **Lessler, M. A. and Brierley, G. P.,** Oxygen electrode measurements in biochemical analysis, in *Methods of Biochemical Analysis*, Vol. 17, Interscience, New York, 1969, 1.
136. **Degn, H., Lundsgaard, J. S., Petersen, L. C., and Ormicki, A.,** Polarographic measurement of steady state kinetics of oxygen uptake by biochemical samples, in *Methods of Biochemical Analysis*, Vol. 26, Interscience, New York, 1980, 47.
137. **Nakanoh, M. and Yoshida, F.,** Transient characteristics of oxygen probes and determination of $k_L a$, *Biotechnol. Bioeng.*, 25, 1653, 1983.
138. **O'Brien, W. J., Agnew, R. W., and Chun, M. R.,** A continuous flow automatically recording respirometer, *Biotechnol. Bioeng.*, 13, 721, 1971.
139. **Albery, W. J. and Hitchman, M. L.,** *Ring-Disk Electrodes*, Oxford University Press, London, 1971.
140. **Kok, R.,** An *in situ* dissolved oxygen probe calibrator, *Biotechnol. Bioeng.*, 18, 729, 1976.
141. **Kok, R. and Zajic, J. E.,** Dynamic response of a polarographic oxygen probe, *Biotechnol. Bioeng.*, 17, 527, 1975.
142. **Linek, V., Sinkule, J., and Vacek, V.,** Oxygen electrode dynamics: three-layer model — chemical reaction in the liquid film, *Biotechnol. Bioeng.*, 25, 1401, 1983.
143. **Merta, K. and Dunn, I. J.,** Oxygen electrode characteristics, *Biotechnol. Bioeng.*, 18, 591, 1976.
144. **Linek, V. and Vacek, V.,** Measurement of fermentor aeration capacity by a fast-response oxygen electrode in medium-air dispersions, *Biotechnol. Bioeng.*, 21, 907, 1979.
145. **Phillips, D. H. and Johnson, M. J.,** Measurement of dissolved oxygen in fermentations, *J. Biochem. Microbiol. Technol. Eng.*, 3, 261, 1961.
146. **Heinzle, E., Bolzern, O., Dunn, I. J., and Bourne, J. R.,** A porous membrane-carrier gas measurement system for dissolved gases and volatiles in fermentation systems, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 439.
147. **Heinzle, E., Furukawa, K., Dunn, I. J., and Bourne, J. R.,** Experimental methods for on-line mass spectrometry in fermentation technology, *Biotechnology*, 1, 181, 1983.
148. **Lehmann, J., Nothnagel, J., Piehl, H., Schultze, R., and Steven, W.,** Measurement of the solubility of oxygen during fermentations, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 453.

149. **Heineken, F. G.,** On the use of fast-response dissolved oxygen probes for oxygen transfer studies, *Biotechnol. Bioeng.*, 12, 145, 1970.

150. **Wernau, W. C. and Wilke, C. R.,** New method for evaluation of dissolved oxygen probe response for $K_L a$ determination, *Biotechnol. Bioeng.*, 15, 571, 1973.

151. **Yoshida, T., Ohasa, S., and Taguchi, H.,** On-line estimation of $k_L a$ by an analog computer, *Biotechnol. Bioeng.*, 22, 201, 1980.

152. **Nakanoh, M. and Yoshida, F.,** Transient characteristics of oxygen probes and determination of $k_L a$, *Biotechnol. Bioeng.*, 25, 1653, 1983.

153. **Tuffile, C. M. and Pinho, F.,** Determination of oxygen-transfer coefficients in viscous Streptomycete fermentations, *Biotechnol. Bioeng.*, 7, 849, 1970.

154. **Spriet, J. A., Botterman, J., De Buyser, D. R., De Visscher, P. L., and Vandamme, E. J.,** A computer-aided noninterfering on-line technique for monitoring oxygen-transfer characteristics during fermentation processes, *Biotechnol. Bioeng.*, 24, 1605, 1982.

155. **Chen, Y. S. and Bungay, H. R.,** Microelectrode studies of oxygen transfer in trickling filter slimes, *Biotechnol. Bioeng.*, 23, 781, 1981.

156. **Harrison, D. E. F., MacLennan, D. G., and Pirt, S. J.,** Responses of bacteria to dissolved oxygen tension, in *Fermentation Advances*, 3rd Int. Fermentation Symp., Perlman, D., Ed., Academic Press, New York, 1969, 117.

157. **Shu, P.,** Control of oxygen uptake in deep tank fermentations, *Ind. Eng. Chem.*, 48, 2204, 1956.

158. **Field, J., Morris, C. A., and Rogers, P. J.,** Construction of an exponential oxygen controller, *Biotechnol. Bioeng.*, 21, 1469, 1979.

159. **Archibald, F.,** Cost-effective dissolved oxygen monitor-controller for continuous culture, *Biotechnol. Bioeng.*, 21, 1553, 1979.

160. **Yano, T., Kobayashi, T., and Shimizu, S.,** Fed-batch of methanol-utilizing bacterium with DO-stat, *J. Ferment. Technol.*, 56, 416, 1978.

161. **Kobayashi, T., Yano, T., and Shimizu, S.,** Automatic control of dissolved oxygen concentration with microcomputer, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 413.

162. **Fuchs, R. and Wang, D. I. C.,** Simple system for controlling dissolved oxygen concentration in laboratory fermentors, *Biotechnol. Bioeng.*, 16, 1529, 1974.

163. **Nyiri, L. K., Jefferis, R. P., III, and Humphrey, A. E.,** Application of computers to the analysis and control of microbiological processes, *Biotechnol. Bioeng. Symp.*, 4, 613, 1973.

164. **Van Hemert, P.,** Application of measurement and control of dissolved oxygen pressure to vaccine production, *Biotechnol. Bioeng. Symp.*, 4, 741, 1973.

165. **Reuss, M., Piehl, H., and Wagner, F.,** Application of mass spectrometry to the measurement of dissolved gases and volatile substances in fermentation, *Eur. J. Appl. Microbiol.*, 1, 323, 1975.

166. **Barford, J. P. and Hall, R. J.,** Investigation of the significance of a carbon and redox balance to the measurement of gaseous metabolism of *Saccharomyces cerevisiae*, *Biotechnol. Bioeng.*, 21, 609, 1979.

167. **Puhar, E., Einsele, A., Bühler, H., and Ingold, W.,** Steam-sterilizable pCO_2 electrode, *Biotechnol. Bioeng.*, 22, 2411, 1980.

168. **Shoda, M. and Ishikawa, Y.,** Carbon dioxide sensor for fermentation systems, *Biotechnol. Bioeng.*, 23, 461, 1981.

169. **Yagi, H. and Yoshida, F.,** Desorption of carbon dioxide from fermentation broth, *Biotechnol. Bioeng.*, 19, 801, 1977.

170. **Toda, K. and Yabe, I.,** Mathematical model of cell growth and phosphatase biosynthesis in *Saccharomyces carlsbergensis* under phosphate limitation, *Biotechnol. Bioeng.*, 21, 487, 1979.

171. **Nyholm, N.,** A mathematical model for microbial growth under limitation by conservative substrates, *Biotechnol. Bioeng.*, 18, 1043, 1976.

172. **Nyholm, N.,** Kinetics of phosphate limited algal growth, *Biotechnol. Bioeng.*, 19, 467, 1977.

173. **Martin, J. F., Liras, P., and Demain, A. L.,** Control by phosphate and ATP of antibiotic formation, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 146.

174. **LeDuy, A. and Samson, R.,** Testing of an ammonia ion selective electrode for ammonia nitrogen measurement in the methanogenic sludge, *Biotechnol. Lett.*, 4, 303, 1982.

175. **Hardy, D., Kraeger, S. J., Dufour, S. W., and Cady, P.,** Rapid detection of microbial contamination in frozen vegetables by automated impedance measurements, *Appl. Environ. Microbiol.*, 34, 14, 1977.

176. **Kjaergaard, L.,** The redox potential: its use and control in biotechnology, *Advances in Biochemical Engineering*, Vol. 7, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1977, 131.

177. **Balakireva, L. M., Kantere, V. M., and Rabotnova, I. L.,** The redox potential in microbiological media, *Biotechnol. Bioeng. Symp.*, 4, 769, 1973.

178. **Jacob, H.-E.,** Reasons for the redox potential in microbial cultures, *Biotechnol. Bioeng. Symp.*, 4, 781, 1973.

179. **Akashi, K., Ikeda, S., Shibai, H., Kobayashi, K., and Hirose, Y.,** Determination of redox potential levels critical for cell respiration and suitable for L-leucine production, *Biotechnol. Bioeng.*, 20, 27, 1978.
180. **Dahod, D. K.,** Redox potential as a better substitute for dissolved oxygen in fermentation process control, *Biotechnol. Bioeng.*, 24, 2123, 1982.
181. **Kjaergaard, L. and Joergensen, B. B.,** Redox potential as a state variable in fermentation systems, *Biotechnol. Bioeng. Symp.*, 9, 85, 1979.
182. **Kjaergaard, L. and Joergensen, B. B.,** The redox potential, a hitherto seldom used parameter in fermentation systems, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 371.
183. **Shibai, H., Ishizaki, A., Kobayashi, K., and Hirose, Y.,** Simultaneous measurement of dissolved oxygen and oxidation-reduction potentials in the aerobic culture, *Agric. Biol. Chem.*, 38, 2407, 1974.
184. **Ishizaki, A., Shibai, H., and Hirose, Y.,** Basic aspects of electrode potential change in submerged fermentation, *Agric. Biol. Chem.*, 38, 2399, 1974.
185. **Kjaergaard, L. and Jorgensen, B. B.,** Maintenance of a constant redox potential during fermentation by automatic addition of glucose, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 24.
186. **Everse, J., Ginsburgh, C. L., and Kaplan, N. O.,** Immobilized enzymes in biochemical analysis, in *Methods of Biochemical Analysis*, Vol. 25, Interscience, New York, 1979, 135.
187. **Guilbault, G. G.,** Applications of enzyme electrodes in analysis, *Ann. N.Y. Acad. Sci.*, 369, 285, 1981.
188. **Guilbault, G. G.,** Enzyme electrodes in analytical chemistry, in *Comprehensive Analytical Chemistry*, Vol. 8, Svehla, G., Ed., Elsevier, New York, 1977, chap. 1.
189. **Satoh, I., Karube, I., and Suzuki, S.,** Enzyme electrode for free cholesterol, *Biotechnol. Bioeng.*, 19, 1095, 1977.
190. **Hewetson, J. W., Jong, T. H., and Gray, P. P.,** Use of an immobilized penicillinase electrode in the monitoring of the penicillin fermentation, *Biotechnol. Bioeng. Symp.*, 9, 125, 1979.
191. **Nilsson, H., Mosbach, K., Enfors, S.-O., and Molin, N.,** An enzyme electrode for measurement of penicillin in fermentation broth: a step toward the application of enzyme electrodes in fermentation control, *Biotechnol. Bioeng.*, 20, 527, 1978.
192. **Enfors, S. O. and Nilsson, H.,** Application of an enzyme electrode for penicillin analysis in fermentation, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 23.
193. **Bourdillon, C., Bourgeois, J.-P., and Thomas, D.,** Chemically modified electrodes bearing grafted enzymes, *Biotechnol. Bioeng.*, 21, 1877, 1979.
194. **Gondo, S., Morishita, M., and Osaki, T.,** Improvement of glucose sensor performance with immobilized glucose oxidase-glucose isomerase system, *Biotechnol. Bioeng.*, 22, 1287, 1980.
195. **Chotani, G. and Constantinides, A.,** On-line glucose analyzer for fermentation applications, *Biotechnol. Bioeng.*, 24, 2743, 1982.
196. **Kernevez, J. P., Konate, L., and Romette, J. L.,** Determination of substrate concentrations by computerized enzyme electrode, *Biotechnol. Bioeng.*, 25, 845, 1983.
197. **Danielsson, B., Mattiasson, B., Kralsson, R., and Winqvist, F.,** Use of an enzyme thermistor in continuous measurements and enzyme reactor control, *Biotechnol. Bioeng.*, 21, 1749, 1979.
198. **Chen, A. K., Liu, C. C., and Schiller, J. G.,** Potentiometric method for substrate analysis using immobilized NAD^+ dependent oxidoreductase enzymes, *Biotechnol. Bioeng.*, 21, 1905, 1979.
199. **Kjellén, K. G. and Neujahr, H. Y.,** Enzyme electrode for phenol, *Biotechnol. Bioeng.*, 22, 299, 1980.
200. **Neujahr, H. Y.,** Enzyme probe for catechol, *Biotechnol. Bioeng.*, 22, 913, 1980.
201. **Tran, N. D., Romette, J. L., and Thomas, D.,** An enzyme electrode for specific determination of L-lysine: a real-time control sensor, *Biotechnol. Bioeng.*, 25, 329, 1983.
202. **Satoh, I., Karube, I., and Suzuki, S.,** Enzyme electrode for sucrose, *Biotechnol. Bioeng.*, 18, 269, 1976.
203. **Malinauskas, A. A. and Kulys, J. J.,** Flow-through NAD sensor, *Biotechnol. Bioeng.*, 21, 513, 1979.
204. **Verduyn, C., Van Dijken, J. P., and Scheffers, W. A.,** A simple, sensitive, and accurate alcohol electrode, *Biotechnol. Bioeng.*, 25, 1049, 1983.
205. **Volesky, B. and Emond, C.,** Continuous-flow monitoring of lactose concentration, *Biotechnol. Bioeng.*, 21, 1251, 1979.
206. **Barker, A. S. and Somers, P. J.,** Enzyme electrodes and enzyme based sensors, in *Topics in Enzyme and Fermentation Biotechnology*, Vol. 2, Ellis Horwood, Sussex, England, 1978, chap. 3.
207. **Ianniello, R. M. and Jespersen, N. D.,** Thermistor probes, in *Comprehensive Analytical Chemistry*, Vol. 12B, Jespersen, N. D., Ed., Elsevier, New York, 1982, chap. 5.
208. **Danielsson, B. and Mosbach, K.,** The prospects for enzyme-coupled probes in fermentation, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 137.
209. **Bovara, R., Pasta, P., and Cremonesi, P.,** Continuous monitoring in enzyme immobilization, *Biotechnol. Bioeng.*, 18, 1017, 1976.

210. **Sae, A. S. W. and Bowen, J. D.,** Simple device for the spectrophotometric assay of immobilized enzymes, *Biotechnol. Bioeng.*, 22, 227, 1980.
211. **Nilsson, H., Akerlund, A.-C., and Mosbach, K.,** Determination of glucose, urea and penicillin using enzyme-pH-electrodes, *Biochim. Biophys. Acta*, 320, 529, 1973.
212. **Schmidt, H.-L., Krisam, G., and Grenner, G.,** Microcalorimetric methods for substrate determination in flow systems with immobilized enzymes, *Biochim. Biophys. Acta*, 429, 283, 1976.
213. **Mosbach, K., Danielsson, B., Borgerud, A., and Scott, M.,** Determination of heat changes in the proximity of immobilised enzymes with an enzyme thermistor and its use for the assay of metabolites, *Biochim. Biophys. Acta*, 403, 256, 1975.
214. **Mosbach, K. and Danielsson, B.,** An enzyme thermistor, *Biochim. Biophys. Acta*, 364, 140, 1974.
215. **Johansson, A., Lundberg, J., Mattiasson, B., and Mosbach, K.,** The application of immobilized enzymes in flow microcalorimetry, *Biochim. Biophys. Acta*, 304, 217, 1973.
216. **Danielsson, B., Mandenius, C. F., Winquist, F., Mattiasson, B., and Mosbach, K.,** Fermentor monitoring and control by enzyme thermistors, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 445.
217. **Mattiasson, B., Danielsson, B., Mandenius, C. F., and Winquist, F.,** Enzyme thermistors for process control, *Ann. N.Y. Acad. Sci.*, 369, 295, 1981.
218. **Enfors, S. O.,** A glucose electrode for fermentation control, *Appl. Biochem. Biotechnol.*, 7, 113, 1982.
219. **Mattiasson, B.,** Application of immobilized whole cells in analysis, in *Immobilized Microbial Cells*, ACS Symp. Series No. 106, American Chemical Society, Washington, D.C., 1979, 203.
220. **Karube, I. and Suzuki, S.,** Application of biosensor in fermentation processes, in *Annual Reports on Fermentation Processes*, Vol. 6, Tsao, G. T., Ed., Academic Press, New York, 1983, chap. 8.
221. **Kulys, J. and Kadziauskiene, K.,** Yeast BOD sensor, *Biotechnol. Bioeng.*, 22, 221, 1980.
222. **Karube, I., Matsunaga, T., Mitsuda, S., and Suzuki, S.,** Microbial electrode BOD sensors, *Biotechnol. Bioeng.*, 19, 1535, 1977.
223. **Suzuki, D. and Karube, I.,** Microbial electrode sensors for cephalosporins and glucose, in *Immobilized Microbial Cells*, ACS Symp. Series No. 106, American Chemical Society, Washington, D.C., 1979, 221.
224. **Suzuki, S. and Karube, I.,** Microbial sensors for fermentation control, in *Advances in Biotechnology*, Vol. 3, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 355.
225. **Okada, T., Karube, I., and Suzuki, S.,** NO_2 sensor which uses immobilized nitrite oxidizing bacteria, *Biotechnol. Bioeng.*, 25, 1641, 1983.
226. **Hikuma, M., Suzuki, H., Yasuda, T., Karube, I., and Suzuki, S.,** Amperometric estimation of BOD by using living immobilized yeasts, *Eur. J. Appl. Microbiol. Biotechnol.*, 8, 289, 1980.
227. **Hikuma, M. and Yasuda, T.,** Application of microbial sensors to the fermentation process, *Ann. N.Y. Acad. Sci.*, 369, 307, 1981.
228. **Neujahr, H. Y. and Kjellén, K. G.,** Bioprobe electrode for phenol, *Biotechnol. Bioeng.*, 21, 671, 1979.
229. **Kobos, R. K. and Pyon, H. Y.,** Application of microbial cells as multistep catalysts in potentiometric biosensing electrodes, *Biotechnol. Bioeng.*, 23, 627, 1981.
230. **Hikuma, M., Kubo, T., Yasuda, T., Karube, I., and Suzuki, S.,** Microbial electrode sensor for alcohols, *Biotechnol. Bioeng.*, 21, 1845, 1979.
231. **Karube, I., Mitsuda, S., and Suzuki, S.,** Glucose sensor using immobilized whole cells of *Pseudomonas fluorescens*, *Eur. J. Appl. Microbiol. Biotechnol.*, 7, 343, 1979.
232. **Yasuda, T.,** Use of computerized glutamic acid sensor, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 167.
233. **Matsunaga, T., Karube, I., and Suzuki, S.,** A specific microbial sensor for formic acid, *Eur. J. Appl. Microbiol. Biotechnol.*, 10, 235, 1980.
234. **Hikuma, M., Suzuki, H., Yasuda, T., Karube, I., and Suzuki, S.,** A rapid electrochemical method for assimilation test of microorganisms, *Eur. J. Appl. Microbiol. Biotechnol.*, 9, 305, 1980.
235. **Calvet, E. and Prat, H.,** *Recent Progress in Microcalorimetry*, MacMillan, New York, 1963.
236. **Beezer, A. E., Ed.,** *Biological Microcalorimetry*, Academic Press, New York, 1980.
237. **Spink, C. and Wadsö, I.,** Calorimetry as an analytical tool in biochemistry and biology, in *Methods of Biochemical Analysis*, Vol. 23, Interscience, New York, 1976, 1.
238. **Forrest, W. W.,** Microcalorimetry, in *Methods in Microbiology*, Vol. 6B, Norris, J. R. and Ribbons, D. W., Eds., Academic Press, New York, 1972, chap. 10.
239. **Eatough, D. J., Izatt, R. M., and Christensen, J. J.,** Titration and flow calorimetry: instrumentation and data calculation, in *Comprehensive Analytical Chemistry*, Vol. 12B, Jespersen, N. D., Ed., Elsevier, New York, 1982, chap. 2.
240. **Schifreen, R. S.,** Flow reactors, in *Comprehensive Analytical Chemistry*, Vol. 12B, Jespersen, N. D., Ed., Elsevier, New York, 1982, chap. 4.

241. **Ishikawa, Y., Shoda, M., and Maruyama, H.,** Design and performance of a new microcalorimetric system for aerobic cultivation of microorganisms, *Biotechnol. Bioeng.*, 23, 2629, 1981.
242. **Cooney, C. L., Wang, D. I. C., and Mateles, R. I.,** Measurement of heat evolution and correlation with oxygen consumption during microbial growth, *Biotechnol. Bioeng.*, 21, 269, 1968.
243. **Mou, D.-G. and Cooney, C. L.,** Application of dynamic calorimetry for monitoring fermentation processes, *Biotechnol. Bioeng.*, 18, 1371, 1976.
244. **Luong, J. G. T. and Volesky, B.,** Determination of the heat of some aerobic fermentations, *Can. J. Chem. Eng.*, 58, 497, 1980.
245. **Eriksson, R. and Holme, T.,** The use of microcalorimetry as an analytical tool for microbial processes, *Biotechnol. Bioeng. Symp.*, 4, 581, 1973.
246. **Waki, T., Suga, K., Hamuro, K., and Ichikawa, K.,** Dynamic analysis and control of yeast production process, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 18.
247. **Goma, G.,** Heat balance, analytical tool in fermentation, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 19.
248. **Lovrien, R., Jorgenson, G., Ma, M. K., and Sund, W. E.,** Microcalorimetry of microorganism metabolism of monosaccharides and simple aromatic compounds, *Biotechnol. Bioeng.*, 22, 1249, 1980.
249. **Chirkov, I. M.,** Thermophysical measurements in studies of microbiological synthesis processes, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 21.
250. **Eatough, D. J., Rehfeld, S. J., Izatt, R. M., and Christensen, J. J.,** Titration and flow calorimetry: application to proteins and lipids, in *Comprehensive Analytical Chemistry*, Vol. 12B, Jespersen, N. D., Ed., Elsevier, New York, 1982, chap. 7.
251. **Grime, J. K.,** Enzymatic enthalpimetry, in *Comprehensive Analytical Chemistry*, Vol. 12B, Jespersen, N. D., Ed., Elsevier, New York, 1982, chap. 8.
252. **Perry, B. F.,** Microcalorimetry in antibiotic bioassay, in *Comprehensive Analytical Chemistry*, Vol. 12B, Jespersen, N. D., Ed., Elsevier, New York, 1982, chap. 9.
253. **Volesky, B. and Thambimuthu, K. V.,** Study of heat of selected aerobic fermentations, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 20.
254. **Luong, J. G. T. and Volesky, B.,** Indirect determination of biomass concentration in fermentation processes, *Can. J. Chem. Eng.*, 60, 163, 1982.
255. **Wang, H., Wang, D. I. C., and Cooney, C. L.,** The application of dynamic calorimetry for monitoring growth of *Saccharomyces cerevisiae*, *Eur. J. Appl. Microbiol. Biotechnol.*, 5, 207, 1978.
256. **Ishikawa, Y., Nonoyama, Y., and Shoda, M.,** Microcalorimetric study of aerobic growth of *Escherichia coli* in batch culture, *Biotechnol. Bioeng.*, 23, 2825, 1981.
257. **Ishikawa, Y. and Shoda, M.,** Calorimetric analysis of *Escherichia coli* in continuous culture, *Biotechnol. Bioeng.*, 25, 1817, 1983.
258. **Hashimoto, M. and Takahashi, K.,** Calorimetric studies of microbial growth: quantitative relation between growth thermograms and inoculum size, *Agric. Bio. Chem.*, 46, 1559, 1982.
259. **Wang, H. Y., Mou, D.-G., and Swartz, J. R.,** Thermodynamic evaluation of microbial growth, *Biotechnol. Bioeng.*, 18, 1811, 1976.
260. **Lelieveld, H. L. M.,** Equipment for the automatic aseptic inoculation and sampling of microbial cultures, *Biotechnol. Bioeng.*, 13, 157, 1971.
261. **Donelli, G.,** An experimental fermenting apparatus designed for the automatic sampling of microbial cultures, *Biotechnol. Bioeng.*, 16, 1407, 1974.
262. **Ferrara, R., Grassi, S., and Del Carratore, G.,** An automatic homocontinuous culture apparatus, *Biotechnol. Bioeng.*, 17, 985, 1975.
263. **Vogelmann, H., Eppert, K., and Wagner, F.,** Bioreactor coupled automated analysis of substrates and metabolites — system description, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 28.
264. **Zabriskie, D. W. and Humphrey, A. E.,** Continuous dialysis for the on-line analysis of diffusible components in fermentation broth, *Biotechnol. Bioeng.*, 20, 1295, 1978.
265. **Dairaku, K. and Yamané, T.,** Use of the porous Teflon tubing method to measure gaseous or volatile substances dissolved in fermentation liquids, *Biotechnol. Bioeng.*, 21, 1671, 1979.
266. **Yamané, T., Matsuda, M., and Sada, E.,** Application of porous Teflon tubing method to automatic fed-batch culture of microorganisms. I. Mass transfer through porous Teflon tubing, *Biotechnol. Bioeng.*, 23, 2493, 1981.
267. **Yamané, T., Matsuda, M., and Sada, E.,** Application of porous Teflon tubing method to automatic fed-batch culture of microorganisms. II. Automatic constant-value control of fed substrate (ethanol) concentration in semibatch culture of yeast, *Biotechnol. Bioeng.*, 23, 2509, 1981.
268. **Kobayashi, T., Yano, T., Mori, G., and Shimizu, S.,** Cultivation of microorganisms with a DO-stat and a silicone tubing sensor, *Biotechnol. Bioeng. Symp.*, 9, 73, 1979.
269. **Yano, T., Kobayashi, T., and Shimizu, S.,** Silicone tubing sensor for detection of methanol, *J. Ferment. Technol.*, 56, 421, 1978.

270. **Dostálek, M. and Häggstrom, M.**, A filter fermenter — apparatus and control equipment, *Biotechnol. Bioeng.*, 24, 2077, 1982.
271. **Marten, J.**, Automatic and continuous assessment of fermentation parameters, in *Methods in Microbiology*, Vol. 6B, Norris, J. R. and Ribbons, D. W., Eds., Academic Press, New York, 1972, chap. 11.
272. **Dawes, E. A., McGill, D. J., and Midgley, M.**, Analysis of fermentation products, in *Methods in Microbiology*, Vol. 6A, Norris, J. R. and Ribbons, D. W., Eds., Academic Press, New York, 1971, chap. 11.
273. **Richardson, B., Bobbitt, T. F., and Orcutt, D. M.**, Use of gas chromatography to monitor sugars and sugar alcohols in microbial media, *Biotechnol. Bioeng.*, 13, 453, 1971.
274. **Franks, P. A., Hall, R. J., and Linklater, P. M.**, The development of control techniques in cheddar cheese production, in *Advances in Biotechnology*, Vol. 2, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 479.
275. **Tonge, G. M.**, Data handling in fermentation processes, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 227.
276. **Pungor, E., Perley, C. R., Cooney, C. L., and Weaver, J. C.**, Continuous monitoring of fermentation outlet gas using a computer coupled MS, *Biotechnol. Lett.*, 2, 409, 1980.
277. **Heinzle, E. and Lafferty, R. M.**, Continuous mass spectrometric measurement of dissolved H_2, O_2, and CO_2 during chemolithoautotrophic growth of *Alcaligenes eutrophus* strain H16, *Eur. J. Appl. Microbiol. Biotechnol.*, 11, 17, 1980.
278. **Comberbach, D. M. and Bu'lock, J. D.**, Automatic on-line fermentation headspace gas analysis using a computer-controlled gas chromatograph, *Biotechnol. Bioeng.*, 25, 2503, 1983.
279. **Schultz, J. S. and Sims, G.**, Affinity sensors for individual metabolites, *Biotechnol. Bioeng. Symp.*, 9, 65, 1979.
280. **Bach, H. P., Woehrer, W., and Roehr, M.**, Continuous determination of ethanol during aerobic cultivation of yeasts, *Biotechnol. Bioeng.*, 20, 799, 1978.
281. **Lee, J. M., Woodward, J. C., Pagan, R. J., and Rogers, P. L.**, Evaluation of semiconductor gas sensor system for on-line ethanol estimation, *Biotechnol. Lett.*, 3, 251, 1981.
282. **Briggs, R.**, Developments in sensors and systems for monitoring and control of wastewater treatment, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 127.
283. **Calam, C. T.**, Automation in screening, in *Fermentation Advances*, Perlman, D., Ed., 3rd Int. Fermentation Symp., Academic Press, New York, 1969, 31.
284. **Alroy, T. and Tannenbaum, S. R.**, The influence of environmental conditions on the macromolecular composition of *Candida utilis*, *Biotechnol. Bioeng.*, 15, 239, 1973.
285. **Harrison, D. E. F. and Harmes, C. S., III**, Control of culture system for ultimate process optimization, *Process Biochem.*, 7 (April), 13, 1972.
286. **Shimz, K.-L. and Holzer, H.**, Rapid decrease of ATP content in intact cells of *Saccharomyces cerevisiae* after incubation with low concentrations of sulfide, *Arch. Microbiol.*, 121, 225, 1979.
287. **Hysert, D. W. and Morrison, N. M.**, Studies on ATP, ADP, and AMP concentrations in yeast and beer, *J. Am. Soc. Brew. Chem.*, 35, 160, 1977.
288. **Schimz, K.-L., Rütten, B., and Tretter, M.**, Determination of adenosine nucleotides with luciferin/luciferase from crude firefly lantern extract on a bioluminescence analyzer, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 457.
289. **Ristroph, D. L., Watteeuw, C. M., Armiger, W. B., and Humphrey, A. E.**, Experience in the use of culture fluorescence for monitoring fermentations, *J. Ferment. Technol.*, 55, 599, 1977.
290. **Einsele, A., Ristroph, D. L., and Humphrey, A. E.**, Substrate uptake mechanisms for yeast cells: a new approach utilizing a fluorometer, *Eur. J. Appl. Microbiol. Biotechnol.*, 6, 335, 1979.
291. **Einsele, A., Ristroph, D. L., and Humphrey, A. E.**, Mixing times and glucose uptake measured with a fluorometer, *Biotechnol. Bioeng.*, 20, 1487, 1978.
292. **Gschwend, K., Beyeler, W., and Fiechter, A.**, Detection of reactor nonhomogeneities by measuring culture fluorescence, *Biotechnol. Bioeng.*, 25, 2789, 1983.
293. **Matsunaga, T., Karube, I., and Suzuki, S.**, Electrochemical determination of cell populations, *Eur. J. Appl. Microbiol. Biotechnol.*, 10, 125, 1980.
294. **Wallace, T. C., Leh, M. B., and Coughlin, R. W.**, Amperometric assay of coenzyme-dependent oxidoreductase enzymes in a flow-through cell, *Biotechnol. Bioeng.*, 19, 901, 1977.
295. **Horan, P. K. and Wheeles, L. L.**, Quantitative single cell analysis and sorting, *Science*, 198, 149, 1977.
296. **Stoehr, M., Gebhardt, U., and Goerttler, K.**, Computer assistance in multiparameter flow microphotometry of mammalian cells, *Biotechnol. Bioeng.*, 18, 1057, 1976.
297. **Slater, M. L., Sharrow, S. O., and Gart, J. J.**, Cell cycle of *Saccharomyces cerevisiae* in populations growing at different rates, *Proc. Natl. Acad. Sci. U.S.A.*, 74, 3850, 1977.

298. **Gilbert, M. F., McQuitty, D. N., and Bailey, J. E.,** Flow microfluorometry study of diauxic batch growth of *Saccharomyces cerevisiae*, *Appl. Environ. Microbiol.*, 36, 615, 1978.
299. **Agar, D. W. and Bailey, J. E.,** Continuous cultivation of fission yeast: classical and flow microfluorometry observations, *Biotechnol. Bioeng.*, 23, 2217, 1981.
300. **Agar, D. W. and Bailey, J. E.,** Measurements and models of synchronous growth of fission yeast induced by temperature oscillations, *Biotechnol. Bioeng.*, 24, 217, 1982.
301. **Alberghina, L., Mariani, L., Martegani, E., and Vanoni, M.,** Analysis of protein distribution in budding yeast, *Biotechnol. Bioeng.*, 25, 1295, 1983.
302. **Agar, D. W. and Bailey, J. E.,** Continuous cultivation of fission yeast: analysis of single-cell protein synthesis kinetics, *Biotechnol. Bioeng.*, 23, 2315, 1981.
303. **Fazel-Madjlessi, J. and Bailey, J. E.,** Analysis of fermentation processes using flow microfluorometry: single-parameter observations of batch bacterial growth, *Biotechnol. Bioeng.*, 21, 1995, 1979.
304. **Fazel-Madjlessi, J., Bailey, J. E., and McQuitty, D. N.,** Flow microfluorometry measurements of multicomponent cell composition during batch bacterial growth, *Biotechnol. Bioeng.*, 22, 457, 1980.
305. **Fazel-Madjlessi, J. and Bailey, J. E.,** Analysis of fermentation processes using flow microfluorometry: amylase and protease activities in *Bacillis subtilis* batch cultures, *Biotechnol. Bioeng.*, 22, 1657, 1980.
306. **Bailey, J. E., Fazel-Madjlessi, J., McQuitty, D. N., Lee, L. Y., Allred, J. C., and Oro, J. A.,** Characterization of bacterial growth by means of flow microfluorometry, *Science*, 198, 1175, 1977.
307. **Hatch, R. T., Wilder, C., and Cadman, T. W.,** Analysis and control of mixed cultures, *Biotechnol. Bioeng. Symp.*, 9, 25, 1979.
308. **Hatch, R. T. and Cadman, T. W.,** Computer Control of Continuous Mixed Culture Processes, presented at 172nd National Meeting of the American Chemical Society, San Francisco, September 1976.
309. **Vraná, D.,** Daughter cells as an important factor in determining the physiological state of yeast populations, *Biotechnol. Bioeng.*, 18, 297, 1976.
310. **McLaughlin, J., Bruno, C. F., and Forrest, T.,** A rapid method for detecting bacterial contamination in the presence of *penicillium* and *Streptomyces* antibiotic fermentation, *Biotechnol. Bioeng.*, 25, 1229, 1983.
311. **Hutter, K. J., Punessen, U., and Eipel, H. E.,** Flow cytometric determination of microbial contaminants, *Biotechnol. Lett.*, 1, 35, 1979.
312. **Cundell, A. M.,** Rapid counting methods for coliform bacteria, in *Advances in Applied Microbiology*, Vol. 27, Perlman, D., Ed., Academic Press, New York, 1981, 169.
313. **Humphrey, A. E.,** Present limitations to the control and understanding of a fermentation process, in Proc. of LABEX Symp. on Computer Control of Fermentation Processes, London, 1971, 1.
314. **San, K.-Y. and Stephanopoulos, G.,** Studies on on-line bioreactor identification. IV. Utilization of pH measurements for product estimation, *Biotechnol. Bioeng.*, in press.
315. **Siegell, S. D. and Gaden, E. L.,** Automatic control of dissolved oxygen levels in fermentations, *Biotechnol. Bioeng.*, 4, 345, 1962.
316. **Hill, G. A. and Robinson, C. W.,** Measurement of aerobic batch culture maximum specific growth rate and respiration coefficient using a dissolved oxygen probe, *Biotechnol. Bioeng.*, 16, 531, 1974.
317. **Taguchi, H. and Humphrey, A. E.,** Dynamic measurement of the volumetric oxygen transfer coefficient in fermentation systems, *J. Ferment. Technol.*, 44, 881, 1966.
318. **Dunn, I. J. and Dang, N. D. P.,** Oxygen transfer coefficients by the dynamic method, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 59.
319. **Blaszczyk, R., Michalski, H., Szewczyk, G., and Wieczorek, A.,** Determination of kinetic process parameters by dynamic methods taking into account the inertia of measuring instrument, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 377.
320. **Roels, J. A.,** The application of macroscopic principles to microbial metabolism, *Ann. N.Y. Acad. Sci.*, 369, 113, 1981.
321. **Roels, J. A.,** Bioengineering report: application of macroscopic principles to microbial metabolism, *Biotechnol. Bioeng.*, 22, 2457, 1980.
322. **Nyiri, L. K., Toth, G. M., and Charles, M.,** On-line measurement of gas-exchange conditions in fermentation processes, *Biotechnol. Bioeng.*, 17, 1663, 1975.
323. **Cooney, C. L., Wang, H. Y., and Wang, D. I. C.,** Computer-aided material balancing for prediction of fermentation parameters, *Biotechnol. Bioeng.*, 19, 55, 1977.
324. **Wang, H. Y., Cooney, C. L., and Wang, D. I. C.,** Computer-aided baker's yeast fermentations, *Biotechnol. Bioeng.*, 19, 69, 1977.
325. **Zabriskie, D. W., Arminger, W. B., and Humphrey, A. E.,** Applications of computers to the indirect measurement of biomass concentration and growth rate by component balancing, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 59.

326. **Zabriskie, D. W. and Humphrey, A. E.,** Real-time estimation of aerobic batch fermentation biomass concentration by component balancing, *AIChE J.*, 24, 138, 1978.
327. **Wang, H. Y., Cooney, C. L., and Wang, D. I. C.,** On-line gas analysis for material balances and control, *Biotechnol. Bioeng. Symp.*, 9, 13, 1979.
328. **Swartz, J. R. and Cooney, C. L.,** Indirect fermentation measurements as a basis for control, *Biotechnol. Bioeng. Symp.*, 9, 95, 1979.
329. **Wang, H. Y., Cooney, C. L., and Wang, D. I. C.,** Computer control of baker's yeast production, *Biotechnol. Bioeng.*, 21, 975, 1979.
330. **Harima, T. and Humphrey, A. E.,** Estimation of *Trichoderma* QM 9414 biomass and growth rate by indirect means, *Biotechnol. Bioeng.*, 22, 821, 1980.
331. **Constantinides, A. and Shao, P.,** Material balancing applied to the prediction of glutamic acid production and cell mass formation, *Ann. N.Y. Acad. Sci.*, 369, 167, 1981.
332. **Stephanopoulos, G. and San, K.-Y.,** On-line estimation of the state of biochemical reactors, in *Chemical Reaction Engineering — Boston*, ACS Symp. Ser. No. 196, Wei, J. and Georgakis, C., Eds., American Chemical Society, Washington, D.C., 1982, 155.
333. **Cooney, C. L. and Mou, D.-G.,** Application of computer monitoring and control to the penicillin fermentation, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 219.
334. **Bravard, J. P., Cordonnier, M., Kernevez, J. P., and Lebeault, J. M.,** On-line identification of parameters in a fermentation process, *Biotechnol. Bioeng.*, 21, 1239, 1979.
335. **Mou, D.-G. and Cooney, C. L.,** Growth monitoring and control through computer-aided on-line mass balancing in a fed-batch penicillin fermentation, *Biotechnol. Bioeng.*, 25, 225, 1983.
336. **Mou, D.-G. and Cooney, C. L.,** Growth monitoring and control in complex medium: a case study employing fed-batch penicillin fermentation and computer-aided on-line mass balancing, *Biotechnol. Bioeng.*, 25, 257, 1983.
337. **Swartz, J., Wang, H., Cooney, C. L., and Wang, D. I. C.,** Computer-aided yeast production, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 29.
338. **Erickson, L. E.,** Application of mass-energy balances in on-line data analysis, *Biotechnol. Bioeng. Symp.*, 9, 49, 1979.
339. **Herbert, D.,** Stoichiometric aspects of microbial growth, in *Continuous Culture 6: Applications and New Fields*, Dean, A. C. R., Ellwood, D. C., Evans, C. G. T., and Melling, J., Eds., Ellis Horwood, London, 1976.
340. **Dekkers, J. G. J., de Kok, H. E., and Roels, J. A.,** Energetics of *Saccharomyces cerevisiae* CBS 426: comparison of anaerobic and aerobic glucose limitation, *Biotechnol. Bioeng.*, 23, 1023, 1981.
341. **Ho, L.,** Process analysis and optimal design of a fermentation process based upon elemental balance equations: generalized semitheoretical equations for estimating rates of oxygen demand and heat evolution, *Biotechnol. Bioeng.*, 21, 1289, 1979.
342. **Erickson, L. E., Minkevich, I. G., and Eroshin, V. K.,** Application of mass and energy balance regularities in fermentation, *Biotechnol. Bioeng.*, 20, 1595, 1978.
343. **Patel, S. A. and Erickson, L. E.,** Estimation of heats of combustion of biomass from elemental analysis using available electron concepts, *Biotechnol. Bioeng.*, 23, 2051, 1981.
344. **Grosz, R., Stephanopoulos, G., and San, K.-Y.,** Studies on on-line bioreactor identification. III. Sensitivity problems with respiratory and heat evolution measurements, *Biotechnol. Bioeng.*, in press.
345. **Zabriskie, D. W., Humphrey, A. E., and Armiger, W. B.,** Real time estimation of fermentation biomass concentration by component balancing, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 91.
346. **Erickson, L. E.,** Applications of mass-energy balances in on-line data analysis, *Biotechnol. Bioeng. Symp.*, 9, 49, 1979.
347. **Heijnen, J. J., Roels, J. A., and Stouthamer, A. H.,** Application of balancing methods in modeling the penicillin fermentation, *Biotechnol. Bioeng.*, 21, 2175, 1979.
348. **Stephanopoulos, G. and San, K.-Y.,** State estimation for computer control of biochemical reactors, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 399.
349. **Jefferis, R. P., III, Winter, H., and Vogelmann, H.,** Digital filtering for automatic analysis of cell density and productivity, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 141.
350. **Ribot, D.,** Identification of fermentations by polynomial methods, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 125.
351. **Reuß, M., Jefferis, R. P., III, and Lehmann, J.,** Application of an on-line system of coupled computers to fermentation modelling, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 107.

352. **Lehmann, J., Reuß, M., and Jefferis, R.,** On-line estimation of parameters and state variable during fermentation in a column reactor, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 90.
353. **Seinfeld, J. H., Gavalas, G. R., and Hwang, M.,** Control of nonlinear stochastic systems, *Ind. Eng. Chem. Fundam.*, 8, 257, 1969.
354. **Seinfeld, J. H.,** Optimal stochastic control of nonlinear systems, *AIChE J.*, 16, 1016, 1970.
355. **Svrcek, W. Y., Elliott, R. F., and Zajic, J. E.,** The extended Kalman filter applied to a continuous culture model, *Biotechnol. Bioeng.*, 16, 827, 1974.
356. **Stephanopoulos, G. and San, K.-Y.,** On the continuous monitoring of time varying parameters, in *Modelling and Control of Biotechnical Processes*, Proc. 1st IFAC Workshop, Halme, A., Ed., Pergamon Press, Oxford, 1983, 195.
357. **Stephanopoulos, G. and San, K.-Y.,** Studies on on-line bioreactor identification. I. Theory, *Biotechnol. Bioeng.*, in press.
358. **Bryson, A. E. and Ho, Y.-C.,** *Applied Optimal Control*, Hemisphere, Washington, D.C., 1975.
359. **Analytic Science Corp.,** *Applied Optimal Estimation*, Gelb, A., Ed., MIT Press, Cambridge, 1974.
360. **Leung, V. P. and Padmanabhan, L.,** Improved estimation algorithms using smoothing and relinearization, *Chem. Eng. J.*, 5, 197, 1973.
361. **San, K.-Y. and Stephanopoulos, G.,** Studies on on-line bioreactor identification. II. Numerical and experimental studies, *Biotechnol. Bioeng.*, in press.
362. **Howell, J. A.,** Parameter estimation for biological waste treatment dynamic models, in *Advances in Biotechnology*, Vol. 2, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 571.
363. **Howell, J. and Jones, M. G.,** Problems in on-line parameter estimation for a structured model, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 57.
364. **Erickson, L. E., Minkevich, I. G., and Eroshin, V. K.,** Utilization of mass-energy balance regularities in the analysis of continuous-culture data, *Biotechnol. Bioeng.*, 21, 575, 1979.
365. **Erickson, L. E.,** Energetic efficiency of biomass and product formation, *Biotechnol. Bioeng.*, 21, 725, 1979.
366. **Erickson, L. E., Selga, S. E., and Viesturs, U. E.,** Application of mass and energy balance regularities to product formation, *Biotechnol. Bioeng.*, 20, 1623, 1978.
367. **Ferrer, A. and Erickson, L. E.,** Evaluation of data consistency and estimation of yield parameters in hydrocarbon fermentation, *Biotechnol. Bioeng.*, 21, 2203, 1979.
368. **Ferrer, A. and Erickson, L. E.,** Data consistency, yield, maintenance, and hysteresis in batch cultures of *Candida lipolytica* cultured on *n*-hexadecane, *Biotechnol. Bioeng.*, 22, 421, 1980.
369. **Erickson, L. E.,** Analysis of microbial growth and product formation with nitrate as nitrogen source, *Biotechnol. Bioeng.*, 22, 1929, 1980.
370. **Solomon, B. O., Erickson, L. E., and Hess, J. L.,** Application of data consistency tests and new parameter estimation methods to microbial growth on corn dust in batch culture, *Biotechnol. Bioeng.*, 23, 2333, 1981.
371. **de Kwaadsteniet, J. W., Jager, J. C., and Stouthamer, A. H.,** A quantitative description of heterotrophic growth in micro-organisms, *J. Theor. Biol.*, 57, 103, 1976.
372. **de Kok, H. and Roels, J. A.,** Method for the statistical treatment of elemental and energy balances with application to steady-state continuous-culture growth of *Sacaromyces cerevisiae* CBS 426 in the respiratory region, *Biotechnol. Bioeng.*, 22, 1097, 1980.
373. **Geurts, Th. G. E., de Kok, H., and Roels, J. A.,** A quantitative description of the growth of *Saccharomyces cerevisiae* CBS 426 on a mixed substrate of glucose and ethanol, *Biotechnol. Bioeng.*, 22, 2031, 1980.
374. **Solomon, B. O., Erickson, L. E., Hess, J. L., and Yang, S. S.,** Maximum likelihood estimation of growth yields, *Biotechnol. Bioeng.*, 24, 633, 1982.
375. **Madron, F., Veverka, V., and Vaněček, V.,** Statistical analysis of material balance of a chemical reactor, *AIChE J.*, 23, 482, 1979.
376. **Madron, F.,** Material-balance calculations of fermentation processes, *Biotechnol. Bioeng.*, 21, 1487, 1979.
377. **Romagnoli, J. A. and Stephanopoulos, G.,** Rectification of process measurement data in the presence of gross errors, *Chem. Eng. Sci.*, 36, 1849, 1981.
378. **Minkevich, I. G.,** Mass-energy balance for microbial product synthesis — biochemical and cultural aspects, *Biotechnol. Bioeng.*, 25, 1267, 1983.
379. **Grosz, R.,** private communication, 1983.
380. **Aoki, M.,** *Optimization of Stochastic Systems*, Academic Press, New York, 1967.
381. **Jones, G. L.,** Mathematical models and their use in the assessment of operational procedures for the activated sludge process, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 31.

382. **Stieber, R. W. and Gerhardt, P.,** Mathematical models and computer simulations of dialysis and non-dialysis continuous processes for ammonium-lactate fermentation, *Biotechnol. Bioeng. Symp.*, 9, 137, 1979.

383. **Blanch, H. W. and Dunn, I. J.,** Modelling and simulation in biochemical engineering, in *Advances in Biochemical Engineering*, Vol. 3, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, 1974, chap. 4.

384. **Roels, J. A.,** Kinetic models in bioengineering. Applications, prospects and problems, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 37.

385. **Roels, J. A.,** Mathematical models and the design of biochemical reactors, *J. Chem. Technol. Biotechnol.*, 32, 59, 1982.

386. **Fredrickson, A. G. and Tsuchiya, H. M.,** Microbial kinetics and dynamics, in *Chemical Reactor Theory, a Review*, Lapidus, L. and Amundson, N. R., Eds., Prentice-Hall, Englewood Cliffs, N.J., 1977, 405.

387. **Kossen, N. W. F.,** Models in bioreactor design, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 23.

388. **Moreira, A. R., Van Dedem, G., and Moo-Young, M.,** Process modeling based on biochemical mechanisms of microbial growth, *Biotechnol. Bioeng. Symp.*, 9, 179, 1979.

389. **Schügerl, K., Lubbert, A., and Lippert, J.,** Flow structures in two- and three-phase reactors, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 47.

390. **Luttmann, R., Thoma, M., Buchholz, H., and Schügerl, K.,** Model development, parameter estimation and simulation of the SCP production process in air lift tower bioreactors with external loop. I. Generalized distributed parameter model, *Comput. Chem. Eng.*, 7, 43, 1983.

391. **Luttmann, R., Thoma, M., Buchholz, H., and Schügerl, K.,** Model development, parameter estimation and simulation of the SCP production process in air lift tower bioreactors with external loop. II. Extended culture modelling and quasi steady state parameter identification, *Comput. Chem. Eng.*, 7, 51, 1983.

392. **Oura, E.,** Energetics of yeast growth under different intensities of aeration, *Biotechnol. Bioeng. Symp.*, 4, 117, 1973.

393. **Oura, E.,** Effect of aeration intensity on the biochemical composition of baker's yeast. I. Factors affecting the type of metabolism, *Biotechnol. Bioeng.*, 16, 1197, 1974.

394. **Oura, E.,** Effect of aeration intensity on the biochemical composition of baker's yeast. II. Activities of the oxidative enzymes, *Biotechnol. Bioeng.*, 16, 1213, 1974.

395. **Oura, E.,** The effect of aeration intensity on the biochemical composition of baker's yeast activities of enzymes of the glycolytic and pentose phosphate pathways, *Biotechnol. Bioeng.*, 18, 415, 1976.

396. **Rose, A. H.,** Temperature control of growth and metabolic activity, in *Fermentation Advances*, 3rd International Fermentation Symp., Perlman, D., Ed., Academic Press, New York, 1969, 157.

397. **Tanner, R. D. and Overley, J. R.,** pH control and the "shoulder" effect in parallel fermentation pathways, *Biotechnol. Bioeng.*, 16, 485, 1974.

398. **Andreyeva, L. N. and Biryukov, V. V.,** Analysis of mathematical models of the effect of pH on fermentation processes and their use for calculating optimal fermentation conditions, *Biotechnol. Bioeng. Symp.*, 4, 61, 1973.

399. **Furuya, A., Misawa, M., Nara, T., Abe, S., and Kinoshita, S.,** Metabolic controls of accumulation of amino acids and nucleotides, in *Fermentation Advances*, 3rd International Fermentation Symp., Perlman, D., Ed., Academic Press, New York, 1969, 177.

400. **Leegwater, M. P. M., Neijssel, O. M., and Tempest, D. W.,** Aspects of microbial physiology in relation to process control, *J. Chem. Tech. Biotechnol.*, 32, 92, 1982.

401. **Grenney, W. J., Bella, D. A., and Curl, H. C., Jr.,** A mathematical model of the nutrient dynamics of phytoplankton in a nitrate-limited environment, *Biotechnol. Bioeng.*, 15, 331, 1973.

402. **Boulton, R.,** Kinetic model for the control of wine fermentations, *Biotechnol. Bioeng. Symp.*, 9, 167, 1979.

403. **Rickard, P. A. D. and Hogan, C. B. J.,** Effect of the glyoxylic acid cycle on the respiratory quotient of *Saccharomyces* sp., *Biotechnol. Bioeng.*, 20, 1111, 1978.

404. **Yoshida, T. and Taguchi, H.,** The use of models in fermentation control, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 93.

405. **Chirkov, I. M., Lodshin, B. Y., and Kulakkov, A. V.,** Analysis of basic principles of fermentation process control, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 353.

406. **Shah, P. S., Erickson, L. E., Fan, L. T., and Prokop, A.,** Growth models of cultures with two liquid phases. VI. Parameter estimation and statistical analysis, *Biotechnol. Bioeng.*, 14, 533, 1972.

407. **Nihtila, M. and Virkkunen, J.,** Practical identifiability of growth and substrate consumption models, *Biotechnol. Bioeng.*, 19, 1831, 1977.

408. **Draper, N. R. and Smith, H.,** *Applied Regression Analysis*, John Wiley & Sons, New York, 1966.
409. **Meiners, M. and Rapmundt, W.,** Application of experimental methods to identify process dynamics of fermentation processes, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 385.
410. **Young, T. B. and Bungay, H. R.,** Dynamic analysis of a microbial process: a systems engineering approach, *Biotechnol. Bioeng.*, 15, 377, 1973.
411. **Alvarez, J. and Ricanõ, J.,** Modeling and optimal control of an SCP fermentation process, *Biotechnol. Bioeng. Symp.*, 9, 149, 1979.
412. **Takamatsu, T., Shioya, S., Maenaka, T., and Shiota, M.,** Mathematical model with a dummy state variable for biochemical reaction process design, in *Topics in Biochemical Engineering*, (PAChEC '77), Vol. 1, AIChE, New York, 1977, 570.
413. **Prokop, A., Votruba, J., Sobotka, M., and Panos, J.,** Yeast SCP from ethanol: measurements, modeling, and parameter estimation in a batch system, *Biotechnol. Bioeng.*, 20, 1523, 1978.
414. **Lee, S. E. and Humphrey, A. E.,** Use of continuous-culture techniques for determining the growth kinetics of a cellulolytic *Thermoactinomyces* sp., *Biotechnol. Bioeng.*, 21, 1277, 1979.
415. **Vinkler, A. and Wood, L. J.,** Multistep guaranteed cost control of linear systems with uncertain parameters, *J. Guidance Control*, 2, 449, 1978.
416. **Nesterov, B. F., Vershkov, D. S., and Sidorov, E. A.,** Control of the multicomponent gaseous medium in laboratory fermentation apparatus, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 26.
417. **Lundell, R.,** Practical implementation of basic computer control strategies for enzyme production, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 181.
418. **Jefferis, R. P., III,** Control of biochemical recovery processes, *Ann. N.Y. Acad. Sci.*, 369, 275, 1981.
419. **Nyiri, L. K.,** Computer control of mammalian cell suspension cultures, in *Cell Culture and Its Application*, Acton, R. T. and Lynn, J. D., Eds., Academic Press, New York, 1977, 161.
420. **Fawzy, A. S. and Hinton, O. R.,** Microprocessor control of fermentation process, *J. Ferment. Technol.*, 58, 61, 1980.
421. **Fawzy, A. S. and Hinton, O. R.,** Multilevel control of multistage continuous culture using a microprocessor, *Biotechnol. Bioeng.*, 23, 887, 1981.
422. **Boyle, T. J.,** Control of the quasi-steady-state in fed-batch fermentation, *Biotechnol. Bioeng. Symp.*, 9, 349, 1979.
423. **Kalogerakis, N. and Boyle, T. J.,** Experimental evaluation of a quasi-steady-state controller for yeast fermentation, *Biotechnol. Bioeng.*, 23, 921, 1981.
424. **Edwards, V. H., Ko, R. C., and Balogh, S. A.,** Dynamics and control of continuous microbial propagators subject to substrate inhibition, *Biotechnol. Bioeng.*, 14, 939, 1972.
425. **Takamatsu, T., Shioya, S., Shiota, M., and Kitabata, T.,** Application of modern control theories to a fermentation process, *Biotechnol. Bioeng. Symp.*, 9, 283, 1979.
426. **Takamatsu, T., Shioya, S., Yokoyama, K., and Ihara, D.,** State estimation and control of a biochemical reaction process, *Ann. N.Y. Acad. Sci.*, 369, 147, 1981.
427. **Hatch, R. T. and Cadman, T. W.,** Computer Control of Continuous Mixed Culture Processes, presented at 172nd National Meeting of the American Chemical Society, San Francisco, September 1976.
428. **Wilder, C. T., Cadman, T. W., and Hatch, R. T.,** Feedback control of a competitive mixed-culture system, *Biotechnol. Bioeng.*, 22, 89, 1980.
429. **Whaite, P. and Gray, P. P.,** On-line computer optimization of chemostat productivity, *Biotechnol. Bioeng.*, 19, 575, 1977.
430. **Nyiri, L. K., Toth, G. M., Krishnaswami, C. S., and Parmenter, D. V.,** On-line analysis and control of fermentation processes, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 37.
431. **Aiba, S., Nagai, S., and Nishizawa, Y.,** Fed batch culture of *Saccharomyces cerevisiae:* a perspective of computer control to enhance the productivity in baker's yeast cultivation, *Biotechnol. Bioeng.*, 18, 1001, 1976.
432. **Wochrer, W., Hampel, W., and Roehr, M.,** Ethanol- and RQ-based computer control in fed batch culture of baker's yeast, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 419.
433. **Whaite, P., Aborhey, S., Hong, E., and Rogers, P. L.,** Microprocessor control of respiratory quotient, *Biotechnol. Bioeng.*, 20, 1459, 1978.
434. **Nagai, S., Nishizawa, Y., and Yamagata, T.,** RQ control fed batch culture to enhance the productivity in baker's yeast cultivation, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 30.

435. **Yongacoglu, S., Toda, K., Prenosil, J. E., and Dunn, I. J.,** Dynamics and control of a biological phenol wastewater treatment reactor, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 365.

436. **Huang, S. Y. and Chu, W.-B.,** Automatic control for fed-batch culture of single cell protein, *Biotechnol. Bioeng.*, 23, 1491, 1981.

437. **Hopkins, T. R.,** Feed-on-demand control of fermentation by cyclic changes in dissolved oxygen tension, *Biotechnol. Bioeng.*, 23, 2137, 1981.

438. **Nishio, N., Tsuchiya, Y., Hayashi, M., and Nagai, S.,** A fed-batch culture of methanol-utilizing bacteria with pH-stat, *J. Ferment. Technol.*, 55, 151, 1977.

439. **Peringer, P. and Blachere, H. T.,** Modeling and optimal control of baker's yeast production in repeated fed-batch culture, *Biotechnol. Bioeng. Symp.*, 9, 205, 1979.

440. **Shtoffer, L. D., Biryudov, V. V., and Nikolushkina, V. M.,** Control of aeration and agitation in antibiotic fermentations, *Biotechnol. Bioeng. Symp.*, 4, 595, 1973.

441. **Shtoffer, L. D., Biryukov, V. V., and Nikolushkina, V. M.,** Control of aeration and agitation in antibiotic fermentations, *Pure Appl. Chem.*, 36, 357, 1973.

442. **Waki, T., Suga, K., and Ichikawa, K.,** Production of cellulase in fed-batch culture, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 359.

443. **Nestaas, E., and Wang, D. I. C.,** Computer control of the penicillin fermentation using the filtration probe in conjunction with a structured process model, *Biotechnol. Bioeng.*, 25, 781, 1983.

444. **Aiba, S.,** Review of process control and optimization in fermentation, *Biotechnol. Bioeng. Symp.*, 9, 269, 1979.

445. **Constantinides, A.,** Application of rigorous optimization methods to the control and operation of fermentation processes, *Ann. N.Y. Acad. Sci.*, 326, 193, 1979.

446. **Ash, S. G. and Topiwala, H. H.,** Interactions between fermentation and recovery processes, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 54.

447. **Attia, R. M., Ali, S. A., and El Gammal, S. E.,** Optimization of proteolytic enzyme fermentation, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 50.

448. **Box, M. J.,** A new method of constrained optimization and a comparison with other methods, *Comput. J.*, 8, 42, 1965.

449. **Box, G. E. P. and Wilson, K. B.,** On the experimental attainment of optimum conditions, *J. R. Stat. Soc. Ser. B*, 13, 1, 1951.

450. **Hooke, R. and Jeeves, T. A.,** "Direct search" solution of numerical and statistical problems, *J. Assn. Comp. Mach.*, 8, 212, 1961.

451. **Rosenbrock, H. H. and Storey, C.,** Optimizing I — hill-climbing methods, in *Computational Techniques for Chemical Engineers*, Pergamon Press, Oxford, 1970, 65.

452. **Rosenbrock, H. H.,** An automatic method for finding the greatest or least value of a function, *Comput. J.*, 3, 175, 1960.

453. **Powell, M. J. D.,** An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Comput. J.*, 7, 155, 1964.

454. **Spendley, W., Hext, G. R., and Himsworth, F. R.,** Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics*, 4, 441, 1962.

455. **Nelder, J. A. and Mead, R.,** A simplex method for function minimization, *Comput. J.*, 7, 308, 1965.

456. **Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mischenko, E. F.,** *The Mathematical Theory of Optimal Processes*, (English translation by Trivogoff, K. N.), Wiley-Interscience, New York, 1962.

457. **Prokop, A. and Votruba, J.,** Computer-aided scheduling of multiproduct batch fermentation plants, *Biotechnol. Bioeng.*, 21, 1689, 1979.

458. **Ohno, H., Nakanishi, E. and Takamatsu, T.,** Optimum operating mode for a class of fermentation, *Biotechnol. Bioeng.*, 20, 625, 1978.

459. **Cooney, C. L.,** Continuous culture of microorganisms: an overview and perspective, *Ann. N.Y. Acad. Sci.*, 326, 295, 1979.

460. **Schierholt, J. and Duisburg, S.-M.,** Technology and economy of two different fermentation processes for the production of citric acid, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 49.

461. **Harris, J. O., Irvine, J., Lamond, J., Seaton, J., and Watson, W.,** Modern brewing fermentation control and plant design, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 379.

462. **Kampe-Nemm, A. A. and Fedossejev, K. G.,** Optimization of a biotechnological system by research operations technique, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 42.

463. **Morari, M., Arkun, Y., and Stephanopoulos, G.**, Studies in the synthesis of control structures for chemical processes. I. Formulation of the problem, decomposition and the classification of the control tasks, analysis of the optimizing control structures, *AIChE J.*, 26, 220, 1980.
464. **Morari, M. and Stephanopoulos, G.**, Studies in the synthesis of control structures for chemical processes. II. Structural aspects and the synthesis of alternative feasible control schemes, *AIChE J.*, 26, 232, 1980.
465. **Morari, M. and Stephanopoulos, G.**, Studies in the synthesis of control structures for chemical processes. III. Measurements within the framework of state estimation in the presence of persistent unknown disturbances, *AIChE J.*, 26, 220, 1980.
466. **Arkun, Y. and Stephanopoulos, G.**, Studies in the synthesis of control structures for chemical processes. IV. Design of steady-state optimizing control structures for chemical process units, *AIChE J.*, 26, 975, 1980.
467. **Okabe, M. and Aiba, S.**, Application of the feasible decomposition and the modified complex methods in the optimization of a fermentation process, *J. Ferment. Technol.*, 52, 279, 1974.
468. **Aiba, S. and Okabe, M.**, A complementary approach to scale-up simulation and optimization of microbial processes, in *Advances in Biochemical Engineering*, Vol. 7, Ghose, T. K., Fiechter, A., and Blakebrough, N., Eds., Springer-Verlag, Berlin, Germany, 1977, 111.
469. **Okabe, M. and Aiba, S.**, Simulation and optimization of cultured-broth filtration, *J. Ferment. Technol.*, 52, 759, 1974.
470. **Okabe, M. and Aiba, S.**, Simulation and optimization of extraction in the fermentation industry — a global approach to the rationale of product recovery, *J. Ferment. Technol.*, 53, 230, 1975.
471. **Okabe, M. and Aiba, S.**, Optimization of a fermentation plant — example of antibiotic production, *J. Ferment. Technol.*, 53, 730, 1975.
472. **Okabe, M., Aiba, S., and Okada, M.**, The modified complex method as applied to an optimization of aeration and agitation in fermentation, *J. Ferment. Technol.*, 51, 594, 1973.
473. **Blachere, H. T., Peringer, P., and Corrieu, C. V.**, Optimization of fermentation plants, in *Workshop Computer Applications in Fermentation Technology*, GBF Monograph Series No. 3, Verlag Chemie, New York, 1977, 1.
474. **Constantinides, A., Spencer, J. L., and Gaden, E. L., Jr.**, Optimization of batch fermentation processes. I. Development of mathematical models for batch penicillin fermentations, *Biotechnol. Bioeng.*, 12, 803, 1970.
475. **Constantinides, A., Spencer, J. L., and Gaden, E. L., Jr.**, Optimization of batch fermentation processes. II. Optimum temperature profiles for batch penicillin fermentations, *Biotechnol. Bioeng.*, 12, 1081, 1970.
476. **Blanch, H. W. and Rogers, P. L.**, Optimal conditions for Gramicidin S. production in continuous culture, *Biotechnol. Bioeng.*, 14, 151, 1972.
477. **Middleton, A. C. and Lawrence, A. W.**, Cost optimization of activated sludge systems, *Biotechnol. Bioeng.*, 16, 807, 1974.
478. **Spitzer, D. W.**, Maximization of steady-state bacterial production in a chemostat with pH and substrate control, *Biotechnol. Bioeng.*, 18, 167, 1976.
479. **Sevely, Y. and Ribot, D.**, Numerical control and optimization of continuous fermentation processes: application to the biomass production, in *Abstr. Papers, 5th Int. Fermentation Symp.*, Dellweg, H., Ed., Verlag, Berlin, 1976, 117.
480. **Endo, I., Nagamune, T., and Inoue, I.**, Optimal strategy for batch alcoholic fermentation, *Biotechnol. Bioeng. Symp.*, 9, 321, 1979.
481. **Saguy, I.**, Utilization of the "complex method" to optimize a fermentation process, *Biotechnol. Bioeng.*, 24, 1519, 1982.
482. **Mateles, R. I.**, JERMFERM: a fermentation process development game, *Biotechnol. Bioeng.*, 20, 2011, 1978.
483. **Schröder, K.-D. and Weide, H.**, Optimization of growth conditions for yeasts, *Biotechnol. Bioeng. Symp.*, 4, 713, 1973.
484. **Votruba, J., Pilát, P., and Prokop, A.**, Application of modified Rosenbrock's method for optimization of nutrient media used in microorganism culturing, *Biotechnol. Bioeng.*, 17, 1833, 1975.
485. **Udén, A. and Hedén, G. C.**, An approach to on-line continuous culture optimization, in Proc. 1st European Conf. on Computer Process Control in Fermentation, Dijon, France, September 3 to 5, 1973.
486. **Nyeste, L., Szigeti, L., Veres, A., Pungor, E., Kurucz, I., and Holló, J.**, Automated fermentation equipment. II. Computer-fermentor system, *Biotechnol. Bioeng.*, 23, 405, 1981.
487. **Szigeti, L., Kurucz, I., Pungor, E., Nyeste, L., and Hollo, J.**, Open-loop on line optimal control of a continuous fermentation process, *Magy. Kem. Lapja.*, 34, 297, 1979.
488. **Box, G. E. P., Hunter, W. G., and Hunter, J. S.**, *Statistics for Experimenters*, John Wiley & Sons, New York, 1978.
489. **Box, G. E. P. and Draper, N. R.**, *Evolutionary Operation: a Statistical Method for Process Improvement*, John Wiley & Sons, New York, 1969.

490. **Davies, O. L.,** *The Design and Analysis of Industrial Experiments*, Oliver and Boyd, Edinburgh, 1967.

491. **Porter, M. A.,** Statistical methods and approaches for fermentation data, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 147.

492. **Chu, G. C. Y., Erickson, L. E., and Fan, L. T.,** Experimental optimization of a step aeration waste treatment process, *Biotechnol. Bioeng.*, 16, 231, 1974.

493. **Bernard, A., Cordonnier, M., and Lebeault, J. M.,** An approach to on-line optimization of biomass production in continuous processes, in *Advances in Biotechnology*, Vol. 1, 6th Int. Fermentation Symp., Moo-Young, M., Robinson, C. W., and Vezina, C., Eds., Pergamon Press, Toronto, 1981, 407.

494. **Albrecht, Ch., Pohland, D., Prause, M., and Ringpfeil, M.,** Optimal control of SCP fermentation processes, in *Computer Applications in Fermentation Technology*, 3rd Int. Conf. on Computer Applications in Fermentation Technology, Society of Chemical Industry, London, 1982, 191.

495. **Muzychenko, L. A., Mascheva, L. A., and Yakovleve, G. V.,** Algorithm for optimal control of microbiological synthesis, *Biotechnol. Bioeng. Symp.*, 4, 629, 1973.

496. **Denn, M. M.,** *Optimization by Variational Methods*, McGraw-Hill, New York, 1969.

497. **D'Ans, G., Gottlieb, D., and Kokotovic, P.,** Time optimal control for a model of bacterial growth, *J. Optimization Theory Appl.*, 7, 61, 1971.

498. **D'Ans, G., Gottlieb, D., and Kokotovic, P.,** Optimal control of bacterial growth, *Automatica*, 8, 729, 1972.

499. **D'Ans, G., Kokotovic, P., and Gottlieb, D.,** A nonlinear regulator problem for a model of biological waste treatment, *IEEE Trans. Autom. Control*, 4, 341, 1971.

500. **Takamatsu, T., Hashimoto, I., Shioya, S., Mizuhara, K., Koike, T., and Ohno, H.,** Theory and practice of optimal control in continuous fermentation process, *Automatica*, 11, 141, 1975.

501. **Ohno, H., Nakanishi, E., and Takamatsu, T.,** Optimal control of a semibatch fermentation, *Biotechnol. Bioeng.*, 18, 847, 1976.

502. **Yamané, T., Kume, T., Sada, E., and Takamatsu, T.,** A simple optimization technique for fed-batch culture, *J. Ferment. Technol.*, 55, 587, 1977.

503. **Weigand, W. A.,** Maximum cell productivity by repeated fed-batch culture for constant yield case, *Biotechnol. Bioeng.*, 23, 249, 1981.

504. **Weigand, W. A., Lim, H. C., Greagan, C. C., and Mohler, R. D.,** Optimization of a repeated fed-batch reactor for maximum cell productivity, *Biotechnol. Bioeng. Symp.*, 9, 335, 1979.

505. **Fishman, V. M. and Biryukov, V. V.,** Kinetic model of secondary metabolite production and its use in computation of optimal conditions, *Biotechnol. Bioeng. Symp.*, 4, 647, 1973.

506. **King, R. E., Aragona, J., and Constantinides, A.,** Specific optimal control of a batch fermenter, *Int. J. Control*, 20, 869, 1974.

507. **Rai, V. R. and Constantinides, A.,** Mathematical modeling and optimization of the gluconic acid fermentation, *AIChE Symp. Ser.*, 69(132), 114, 1973.

508. **Constantinides, A. and Rai, V. R.,** Application of the continuous maximum principle to fermentation processes, *Biotechnol. Bioeng. Symp.*, 4, 663, 1973.

509. **Nyeste, L., Sevella, B., Szigeti, L., Szöke, A., and Holló, J.,** Modelling and off-line optimization of batch gluconic acid fermentation, *Eur. J. Appl. Microbiol. Biotechnol.*, 10, 87, 1980.

510. **Guthke, T. and Knorre, W. A.,** Optimal substrate profile for antibiotic fermentations, *Biotechnol. Bioeng.*, 23, 2771, 1981.

511. **Ho, L. Y. and Humphrey, A. E.,** Optimal control of an enzyme reaction subject to enzyme deactivation. I. Batch process, *Biotechnol. Bioeng.*, 12, 291, 1970.

512. **San, K.-Y. and Stephanopoulos, G.,** Optimal control policy for substrate inhibited kinetics with enzyme deactivation in an isothermal CSTR, *AIChE J.*, 29, 417, 1983.

513. **San, K.-Y. and Stephanopoulos, G.,** A note on the effect of time delay on feed forward control, *Biotechnol. Bioeng.*, in press.

514. **Dairaku, K., Izumoto, E., Morikawa, H., Shioya, S., and Takamatsu, T.,** Optimal quality control of baker's yeast fed-batch culture using population dynamics, *Biotechnol. Bioeng.*, 24, 2661, 1982.

515. **Weinrich, S. D. and Lapidus, L.,** Parameter adaptation and control in a biochemical reactor, *Biotechnol. Bioeng.*, 14, 13, 1972.

APPENDIX H

A NEW APPROACH TO BIOPROCESS IDENTIFICATION AND MODELING

(The text of Appendix H consists of an article coauthored with G. N. Stephanopoulos which has appeared in *Biotechnology & Bioengineering Symposium Series*, **14**, 635-656, 1984.)

A New Approach to Bioprocess Identification and Modeling

NAM SUN WANG and GREGORY STEPHANOPOULOS
Chemical Engineering Department,
California Institute of Technology,
Pasadena, California 91125

Summary

A new approach to bioprocess identification and modeling is presented. A time-delay kernel is included in the state equations, and a generalized method of transforming the integrodifferential equations to a mathematically equivalent set of first-order ordinary differential equations is developed. The incorporation of a culture's past history in the form of a delay kernel greatly enhances the model's predictive capabilities, and the transformation permits the application of a wide range of well-established analytic techniques. The resulting model effectively combines the simplicity of an unstructured lumped model with the power of a complex structured model. Keywords: time delay; kernel; modeling; adaptive on-line state estimation; commensalism; integrodifferential equations.

I. INTRODUCTION

It has been pointed out [1] that two important problems in the optimal design and operation of a biological reactor are the lack of reliable biological sensors and the lack of simple mathematical models with satisfactory predictive capability. The sensor inadequacy is especially acute in the areas of continuous measurement of cell mass and substrate/product concentrations, which are among the most fundamental state variables in nearly all fermentation systems. The relatively poor state of instrumentation means that the current measurements are discrete in time and frequently contain a high level of noise which must be filtered out before they are to be used to control a bioreactor.

As far as the existing models are concerned, they are either inadequate during transient operation (lumped-parameter models), very complicated and time consuming for control and optimization purposes (single-cell models) or contain a large number of directly indeterminable parameters (structured models) for any practical application. Despite significant modeling efforts, simple, descriptive, and easy to construct models are not yet available.

One of the main purposes of advance measurement and modeling capabilities should be the satisfactory control of biological processes. This control action should be viewed more as a mechanism to safeguard the process against various

Biotechnology and Bioengineering Symp. No. 14 (1984)
 CCC 0572-6565/84/010635-22$04.00

types of disturbances rather than as a mean of improving performance. To be sure, there are situations in which the performance of a process can be dramatically improved with the proper control, but these cases are less likely to be found in bioprocesses aiming at the volume production of fuels and chemicals.

Perhaps the best way to describe the type of control problems that may arise in a biological process and the essential features of the associated control structure in terms of measurement and modeling requirements is by means of an example especially appropriate for the theme of this symposium. This example concerns the simultaneous saccharification and fermentation process of cellulosic and hemicellulosic biomass by a mixed culture of *Clostridium thermocellulum* (CT_1) and *Clostridium thermosaccharolyticum* (CT_2). Shown in Figure 1 is a schematic of the basic interactions of this system which has been proposed for the direct conversion of biomass to ethanol [2]. According to this scheme, both types of cellulosic and hemicellulosic biomass are first hydrolyzed by CT_1 to form six- and five-carbon sugars, respectively. The same microorganism can further convert the intermediate six-carbon sugars into ethanol as the final product. For the complete conversion of biomass material into ethanol, the second organism CT_2 is introduced to convert the five-carbon sugars into ethanol. This mixed-culture system is an example of commensalism; namely, that interaction in which one population (the commensal population) depends on the growth of the host population but not vice versa. Initial studies on this system [2] showed that the growth of the host (CT_1) is uninhibited by the concentration of the substrate, but the growth of the commensal microorganism (CT_2) exhibits substrate inhibition, as shown schematically in Figure 2(a) depicting the μ (specific growth rate) versus s (limiting substrate concentration) curves for the two populations. The productivity as a function of the dilution rate in a continuous fermentor employing this mixed culture is also shown. It is desirable to operate the reactor at a dilution rate corresponding to the maximum productivity, which is generally very close to the washout of the second population.

The dynamics of commensal mixed populations growing in a continuous biochemical reactor has been analyzed [3], and it is found that the phase plane diagram in the vicinity of the maximum productivity is quite sensitive to the value of the dilution rate, D as shown in Figures 2(b) and 2(c).

For $D > \mu_{2max}$, where μ_{2max} is the maximum specific growth rate of the commensal population, the commensal population is washed out, and the only stable steady state is the propagation of the host organism. For $D < \mu_{2max}$ two coex-

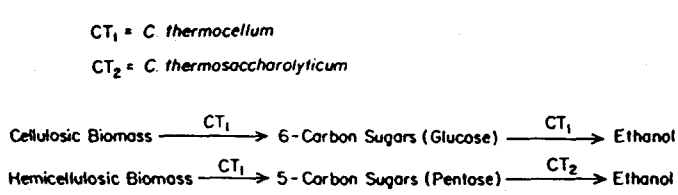


Fig. 1. Conversion of biomass material into ethanol by the mixed commensal populations of *C. thermocellum* and *C. thermosaccharolyticum*.

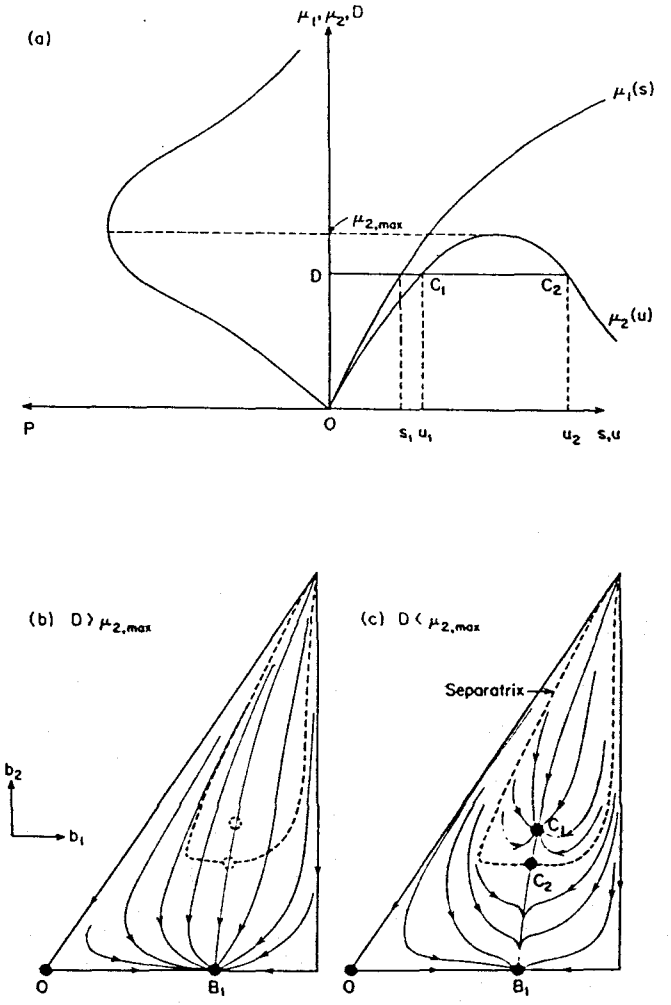


Fig. 2. (a) Productivity of a process that involves a mixed culture of commensal populations as a function of the dilution rate. (b) Schematic of the phase plane trajectories for the case described in (a) when $D < \mu_{2max}$ [3]. (c) Schematic of the phase plane trajectories for $D > \mu_{2max}$.

istence steady states are possible; one is a stable coexistence steady state (C_1) while the other is an unstable saddle point (C_2) on the separatrix. These two steady states are very close to each other for dilution rates close to the value of maximum productivity. Under normal conditions, operation at the stable coexistence steady state is desired. However, various perturbations such as a sudden increase in D or a decrease in μ_{2max}, caused by, among other things, a perturbation in the pH or temperature, can cause the state of the culture to move toward the partial washout steady state indicated by point B_1. If such perturbations remain undetected for an appreciable period of time and if appropriate corrective control

action is not implemented quickly, the state of the system may cross the original separatrix. At this point, even the best regulation of D or μ_{2max} to the set point can only restore the phase diagram to that of Figure 2(c). However, for an initial composition of the mixed culture below the separatrix, this control will still fail to reverse the attraction of the state toward B_1. An effective control scheme of this system would have to determine first the current state of the reactor, establish the dynamics of the system, and, then, control the dilution rate in such a way that the separatrix passes below the current state. The dilution rate is then gradually restored to the previous value as the system approaches the coexistence steady state C_1.

The above example shows that for a successful control, one needs an efficient on-line state and parameter estimation scheme to detect the perturbation early enough to prevent the state from crossing the separatrix, if possible. In addition, a descriptive dynamic model is required to assess the effects of a possible control action and guide the state back to the set point of operation. In the following sections, a systematic methodology for the on-line estimation of key bioreactor variables and culture parameters will be briefly presented. Subsequently, a new approach to bioprocess identification and modeling will be outlined. The proposed approach considers the effect on rates and yields of not only the present state of the system but also the previous history through the concept of a kernel integral. The set of the resulting integrodifferential equations is then shown to be equivalent to a set of first-order ordinary differential equations representing a generalized structured model. These simple ordinary differential equations then can be relatively easily manipulated with the well-developed mathematical techniques to yield insightful information on the dynamics of the system, including the analysis of the stability of steady states, etc. Furthermore, size-reduction techniques are outlined which can lead to a low-dimensional, directly observable model while preserving at the same time the biological significance of various parameters.

II. PARAMETER AND STATE ESTIMATION

The basic feature of the parameter and state estimation scheme as proposed by Stephanopoulos and San [4] is to represent the biological conversion of substrate to cell mass and product by a chemical reaction:

$$a\,C_xH_yO_z + b\,O_2 + c\,NH_3 \longrightarrow C_\alpha H_\beta O_\gamma B_\delta + d\,H_2O + e\,CO_2 + f\,C_{\alpha'}H_{\beta'}O_{\gamma'}N_{\delta'} \quad (1)$$

where $C_xH_yO_z$, $C_\alpha H_\beta O_\gamma N_\delta$, and $C_{\alpha'}H_{\beta'}O_{\gamma'}N_{\delta'}$ are the chemical formulas for the substrate, cell biomass, and product, respectively; and a, b, c, d, e, and f are the six unknown stoichiometric coefficients for the reaction. The principle of elemental balances for C, H, O, and N gives four equations, which can be combined with the on-line measurement of the respiratory quotient (e/b, obtained from continuous gas phase O_2 and CO_2 concentration measurements) and the

rate of ammonium hydroxide addition for pH control to solve for the six stoichiometric coefficients. The various yields can be easily obtained from these stoichiometric coefficients.

It should be pointed out that the above procedure yields only the stoichiometric coefficients (yields) of Eq. (1) and the *total* rates of growth (R), product formation, etc. However, as indicated earlier, the key bioreactor variables are the various concentrations and the *specific* rates of growth, product formation, etc. A direct integration of the governing differential equations based on material balancing cannot yield accurate estimates of the above variables, for no provision is made for the presence of any noises that are always present in both the measurements and the process itself and that may affect significantly the accuracy of the so obtained variables.

In order to eliminate such noises, an adaptive Kalman–Bucy filter is applied to the measurements to estimate continuously the biomass, substrate, and product concentrations as well as various yields and the specific growth rate of the culture. The essence of the Kalman filter is as follows. The dynamics of a nonlinear system and the measurements can be expressed in the general form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u}) + \zeta(t) \tag{2}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \xi(t) \tag{3}$$

where $\mathbf{x}$ is the state vector of a dynamic system, $\mathbf{u}$ is the nonrandom input vector, $\zeta(t)$ is the random disturbance, and $\xi(t)$ is the random noise in the measurement $\mathbf{y}$. The linearized Kalman estimate $\hat{\mathbf{x}}$ of the true state $\mathbf{x}$ is described by the following set of vector and matrix filtering differential equations:

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{f}(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{P}\mathbf{h}_{\mathbf{x}}^{T}(\hat{\mathbf{x}})\mathbf{S}^{-1}(\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}})) \tag{4}$$

$$\frac{d\mathbf{P}}{dt} = \mathbf{f}_{\mathbf{x}}(\hat{\mathbf{x}},\mathbf{u})\mathbf{P} + \mathbf{P}\mathbf{f}_{\mathbf{x}}^{T}(\hat{\mathbf{x}},\mathbf{u}) + \mathbf{Q} - \mathbf{P}\mathbf{h}_{\mathbf{x}}^{T}(\hat{\mathbf{x}})\mathbf{S}^{-1}\mathbf{h}_{\mathbf{x}}(\hat{\mathbf{x}})\mathbf{P} \tag{5}$$

where $\mathbf{P}$ is the symmetric covariance matrix of the estimation error, and $\mathbf{Q}$ and $\mathbf{S}$ are the positive definite matrices which are measures of the intensities of the noise processes ζ and ξ, respectively.

The general state equation (2) takes the following form when applied to the systems of *Saccharomyces cerevisiae* growing on glucose-limited medium in a chemostat and fermenting glucose to ethanol [5]:

$$\frac{dx}{dt} = \mu x - Dx \tag{6}$$

$$\frac{ds}{dt} = D(s_f - s) - \frac{1}{Y_s}\mu x \tag{7}$$

$$\frac{dp}{dt} = \frac{1}{Y_p}\mu x - Dp \tag{8}$$

where x, s, and p are the biomass, glucose, and ethanol concentrations, respectively, and s_f is the glucose concentration in the feed stream. In addition, unknown parameters such as the specific growth rate, μ, and cell yields Y_s and Y_p may also be estimated simultaneously by including them in the state vector and by treating the noise associated with the fast changing variable as colored, as implemented below:

$$\frac{d\mu}{dt} = c + \zeta_1(t) \tag{9}$$

$$\frac{dc}{dt} = -\frac{c}{\tau} + \zeta_1(t) \tag{10}$$

$$\frac{dY_s}{dt} = \zeta_2(t) \tag{11}$$

$$\frac{dY_p}{dt} = \zeta_3(t) \tag{12}$$

where c is the colored noise correction variable, and τ is the time constant associated with c, which can be set equal to D^{-1}. Similarly, the measurement Eq. (3) in this case becomes:

$$R = \mu x + \xi_1(t) \tag{13}$$

$$Y_{s,ms} = Y_s + \xi_2(t) \tag{14}$$

$$Y_{p,ms} = Y_p + \xi_3(t) \tag{15}$$

The adaptive nature of the estimation-filtering algorithm is because a varying intensity is used for the noise ζ_i depending on the value of the residue between measured variables and predictions for the same variables based on the available state estimates. If the residues are smaller than what can be attributed to the measurement noise and the uncertainty of the state estimates, then the variance of the noise is set equal to zero, which is equivalent to assuming that no change occurred during two consecutive measurements. In the opposite case the filter opens in order to accept the new measurements and adjusts the parameter values appropriately.

The above methodology was applied to the continuous fermentation of *S. cerevisiae*, and Figure 3 shows some representative results of the capabilities of the method [5]. These figures reveal that smooth and reliable estimates on biomass, glucose, and ethanol concentrations are achievable. The estimate of the specific growth rate obtained by the adaptive Kalman filter is especially smooth compared to that obtained by an ordinary averaging method (RC filter). Additional experimental results and numerical simulation have further verified the ability of this methodology to provide accurate, on-line state estimates [5]. Furthermore, the sensitivity of the method has been investigated extensively [6] so that it can be claimed that the proposed methodology is reliable and can meet the requirements of a potential estimation-control structure for biological reactors.

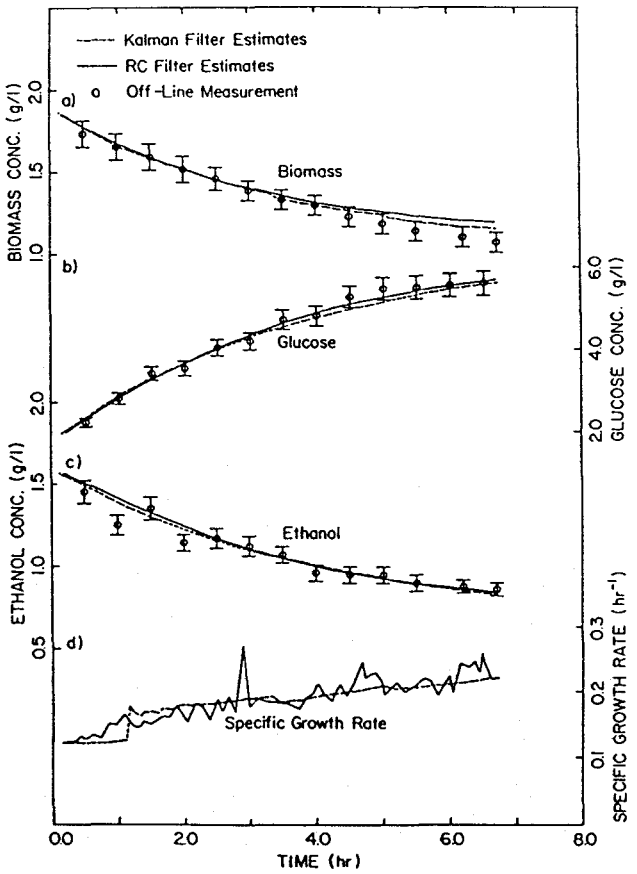


Fig. 3. On-line estimates and off-line measurements in a dilution rate step-up experiment of continuous glucose-limited cultivation of *S. cerevisiae* with ethanol formation. Shift is at 0 hr [5]. (a) Cell biomass concentration; (b) glucose concentration; (c) ethanol concentration; (d) specific growth rate.

III. NEW MODELING APPROACH

Mathematical models are needed for control purposes and bioreactor design. They are the condensed version of our knowledge about a system, and their sophistication can vary widely. A useful model should be properly balanced with respect to its mathematical complexity and its ability to capture the essential features for the intended purpose. It should also be simple enough to permit direct determination of its key parameters by performing feasible experimental procedures. The validity of a complex model is especially questionable when it contains a large number of parameters whose values cannot be experimentally evaluated. The success of models in engineering has always depended on the

valid use of approximations and assumptions in reducing the complexity of the real world to simple and manageable mathematical abstraction, and biochemical engineering is no exception in this respect.

In the following section, a modeling approach is introduced which combines the simplicity of an unstructured model with the power of a complex structured model. The essence of the approach is the inclusion of a time-delay kernel in the equation describing the dynamics of a bioreactor. As an introductory example, consider the familiar case of a continuous bioreactor operation modeled by a lumped-parameter two-state-variable model, namely:

$$\frac{dx}{dt} = -Dx + \mu(s)x \tag{16}$$

$$\frac{ds}{dt} = D(s_f - s) - \frac{1}{Y_s}\mu(s) \tag{17}$$

where we assume that the specific growth rate, μ, of biomass, x, is a function of the limiting substrate, s. The above model, and for that matter almost all other models presently in use, states that the behavior of the biomass–substrate system depends only on the present state, and there is no provision for the past history of the microorganism. It has been recognized for a long time, however, that the observed response of a cell population at a certain time instant is the composite result of various biological processes that were initiated at different time instants in the past as a response to the instantaneous environmental conditions prevailing at each particular time. These various processes result in a present overall specific growth rate than can be described with the introduction of a time-delay kernel, $k(t,h)$, in the specific growth rate:

$$\frac{dx}{dt} = -Dx + \left(\int_{-\infty}^{t} \mu[s(h)]k(t,h)dh\right)x \tag{18}$$

$$\frac{ds}{dt} = D(s_f - s) - \frac{1}{Y_s}\left(\int_{-\infty}^{t} \mu[s(h)]k(t,h)dh\right)x \tag{19}$$

The idea of a variable's dependence on its past history has been in existence for quite some time [7–9]. In ecological studies, the interaction of prey–predator has been described by Volterra models, which include a kernel associated with one of the states of the system. As shown later, our handling of the kernel is more general in the sense that the shape of the kernel is not restricted. For a linearized time-invariant system, k no longer depends on t and the integration variables h separately but on the difference $t - h$.

$$\frac{dx}{dt} = -Dx + \left(\int_{-\infty}^{t} \mu[s(h)]k(t - h)dh\right)x \tag{20}$$

$$\frac{ds}{dt} = D(s_f - s) - \frac{1}{Y_s}\left(\int_{-\infty}^{t} \mu[s(h)]k(t - h)dh\right)x \tag{21}$$

Nondimensionalization can be carried out to simplify the above equations without any loss of generality. If time is scaled with reference to D^{-1} and concentrations are scaled with reference to s_f, then we have:

$$\frac{dx}{dt} = \left(-1 + \int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right)x \tag{22}$$

$$\frac{ds}{dt} = 1 - s - \frac{1}{Y_s}\left(\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh\right)x \tag{23}$$

The kernel $k(t)$ is usually referred to as the impulse response function and can be interpreted as a weighing factor as shown schematically in Figure 4. Since it can be generally assumed that future states have no effect on the present, $k(t)$ can be implicitly set to zero for $t < 0$. Note that, strictly speaking, $k(t)$ is not a time-delay probability distribution function and $k(t) < 0$, that is, negative weighing, is possible. The μ in the integrand of Eqs. (22) and (23) is the specific growth rate that would have been realized if the system operated at a steady state characterized by the corresponding value of s for a prolonged period of time; it is the true specific growth rate in the absence of time-delay effects. The presently observed apparent value of the specific growth rate is given by the integral of Eqs. (22) or (23) and can be conceptualized as a string of impulses each of which is felt by the system over a period of time according to the impulse transfer function. The main questions, of course, are how can one determine the appropriate kernel form and what is the biological significance of the latter. These two points will be discussed later.

Various possibilities exist for the function form of $k(t)$, see Figure 5. One can set $k(t)$ to be a delta function, $\delta(t)$, meaning that both the future and the past have absolutely no weight on the specific growth rate and that the present instant carries all the weight. The integral $\int_{-\infty}^{t} \mu[s(h)]k(t-h)dh$ reduces to $\mu[s(t)]$ in this case, and Eqs. (22) and (23) reduce to the conventional unstructured model of Eqs. (16) and (17).

Another possibility is to assume that there is a fixed time lag in the response of the system, that is, $k(t) = \delta(t - \tau)$, meaning that the specific growth rate depends on the substrate concentration at a discrete time instant τ units before the present. The state equations in this case are reduced to:

$$\frac{dx}{dt} = \{-1 + \mu[s(t-\tau)]\}x \tag{24}$$

$$\frac{ds}{dt} = 1 - s - \frac{1}{Y_s}\mu[s(t-\tau)]x \tag{25}$$

Analysis of this case can be performed by using the theories of differential–difference equations [10]. The relatively simple system of Eqs. (24) and (25) can be successfully analyzed, but, because the mathematical theories of differential–difference equations are not as well developed as ordinary differential

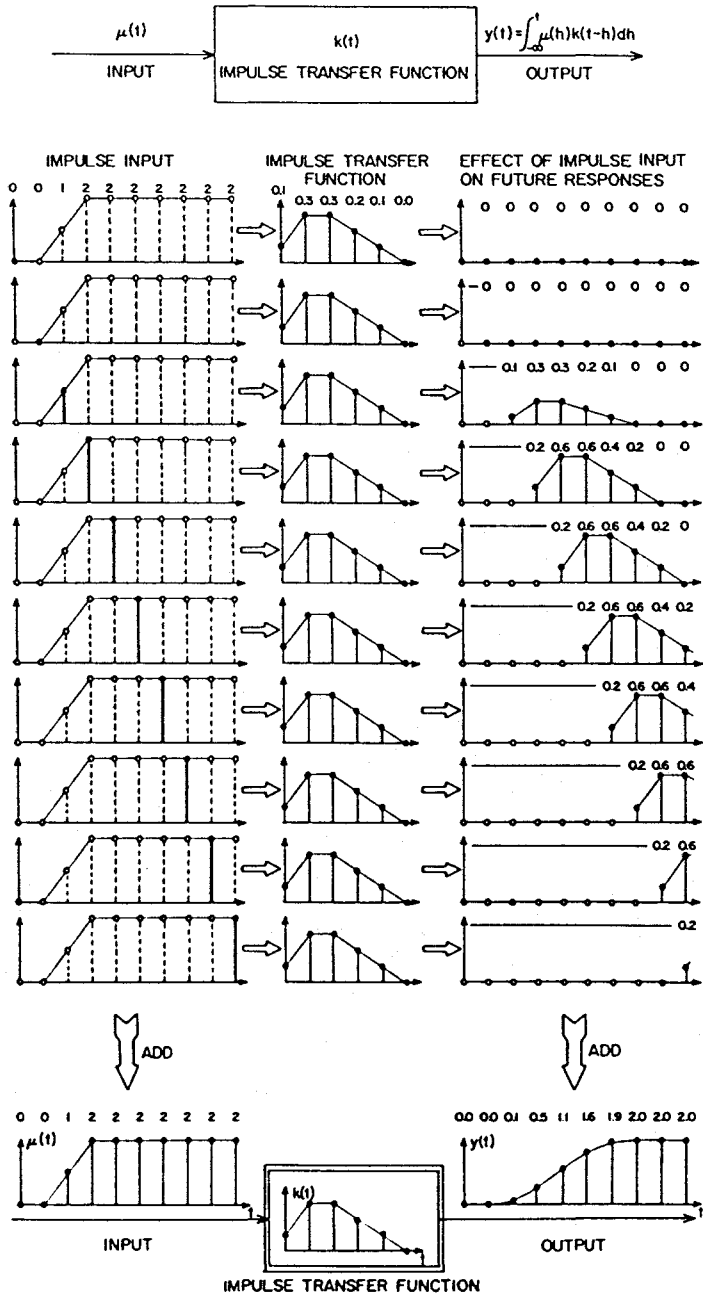


Fig. 4. Interpretation of a time-invariant kernel integral which relates the input to the output of a linear system.

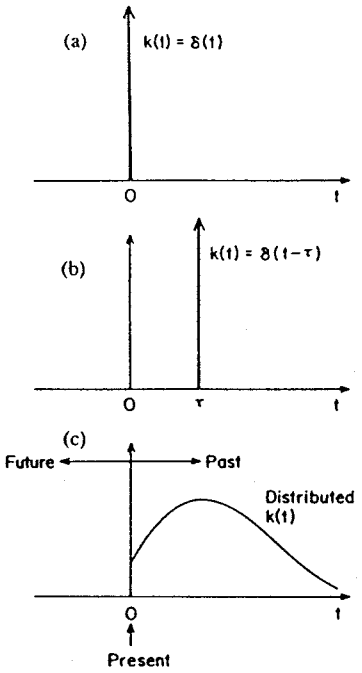


Fig. 5. Three frequently used functional forms of $k(t)$: (a) delta function without time delay; (b) delta function with a discrete time delay τ; (c) general distributed time delay. Note the directions of past and future are the reverse of the conventional time plots.

equations, some problems may arise in the integration and general analysis of this type of differential–difference equations especially in slightly more complicated systems.

A more general approach is to express an arbitrary function $k(t)$ in terms of a series of base functions which permit the transformation of the integrodifferential equations into a set of simple first-order equations. This is accomplished by approximating an analytical function $k(t)$ as a summation of exponential distribution functions of order m or less:

$$k(t) = a_0k_0(t) + a_1k_1(t) + a_2k_2(t) + \cdots + a_mk_m(t) \tag{26}$$

where the general expression for the nth exponential distribution function is

$$k_n(t) = \begin{cases} \dfrac{T^{-1}}{n!}\left(\dfrac{t}{T}\right)^n e^{-t/T} & \text{for} \quad t \geq 0 \\ 0 & \text{for} \quad t < 0 \end{cases} \tag{27}$$

The first two exponential distribution functions are sometimes used in ecological studies and they have special names.

$$n = 0 \qquad k_0 = T^{-1}e^{-t/T} \qquad \text{weak generic delay} \tag{28}$$

$$n = 1 \qquad k_1 = T^{-2}te^{-t/T} \qquad \text{strong generic delay} \tag{29}$$

Some of the properties of the exponential distribution functions are shown in Figure 6, and the first few of these exponential distribution functions are shown in Figure 7. Note that if these functions are normalized with respect to the average delay, $(n + 1)T$, then one can see that the peak at the average delay becomes higher and narrower as n increases (see Fig. 8). It can be shown that as $n \rightarrow \infty$, $k_n(t) \rightarrow \delta(t - \tau)$, where τ is the average delay. In this limiting case, the state equations are again reduced to Eqs. (24) and (25).

At close inspection, the nth-order exponential distribution function is identical to the residence time distribution function of a system of n-CSTRs in series in the modeling of a chemical reactor. Accordingly, if $k(t)$ is expressed as the sum of m exponential functions, the observed specific growth rate at time t, expressed as

$$y(t) \equiv \int_{-\infty}^{t} \mu[s(h)]k(t - h)dh$$

will be the weighed sum of m integrals

$$\int_{-\infty}^{t} \mu[s(h)]k_j(t - h)dh \qquad j = 1,2, \ldots, m$$

that is,

$$\begin{aligned} y(t) &= \sum_{j=0}^{m} a_j \left(\int_{-\infty}^{t} \mu[s(h)]k_j(t - h)dh \right) \\ &= \int_{-\infty}^{t} \mu[s(h)] \left(\sum_{j=0}^{m} a_j k_j(t - h) \right) dh \\ &= \int_{-\infty}^{t} \mu[s(h)]k(t - h)dh \end{aligned} \tag{30}$$

The weighing factors a_j and the delay time constant T are chosen in such a way as to fit the observed transient of the specific system in a shift-up or shift-down experiment. A small value of m usually gives a very satisfactory fit.

Quite significantly, we are not bounded by the limited functional shapes of each individual exponential distribution function. By expressing the kernel as a linear combination of these base functions, any sufficiently smooth continuously differentiable function can be represented if a sufficiently large number of base functions are used. This is because the approach is essentially the same as expanding the function $e^{t/T}k(t)$ by a Taylor's series. Theoretically, m could be extended to ∞, but two or three terms should be sufficient under most circumstances in practice. For example, a linear combination of $k_0(t)$ and $k_1(t)$ results in

$$k(t) = (a_0T^{-1} + a_1T^{-2}t)e^{-t/T} \tag{31}$$

where $a_0 + a_1 = 1$ so that the kernel is normalized to unity. Some of the shapes

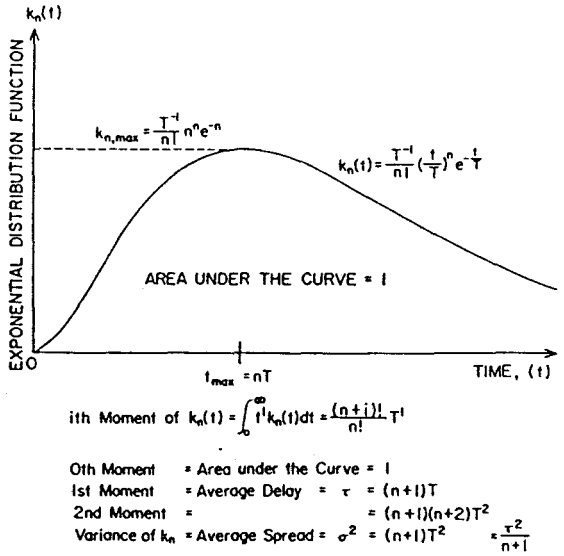


Fig. 6. Some properties of the exponential distribution functions.

of $k(t)$ generated by a combination of these two base functions are shown in Figure 9.

The reason for choosing exponential distribution functions as the base functions is that they permit easy and elegant transformation of a set of integrodifferential equations into a set of simple ordinary differential equations. These exponential distribution functions possess the property that each and every one of them is

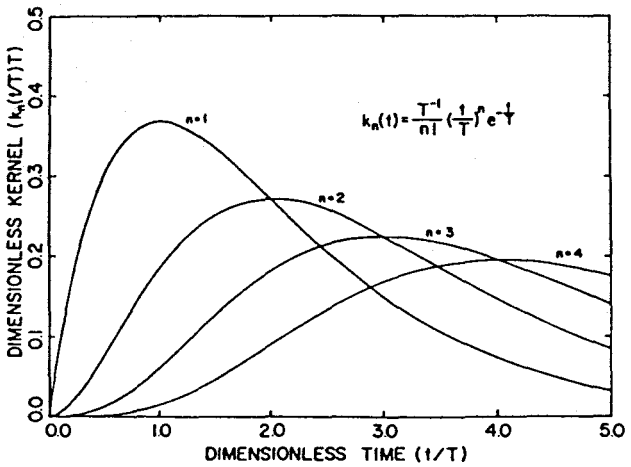


Fig. 7. Exponential distribution function of order n.

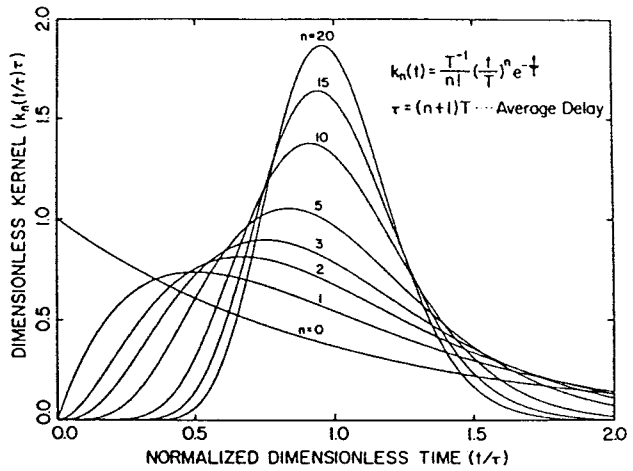


Fig. 8. Exponential distribution function of order n normalized with respect to the average delay.

the solution to the following differential equation:

$$\sum_{i=0}^{n+1} T^i \binom{n+1}{i} \frac{d^i k_n(t)}{dt^i} = 0 \tag{32}$$

with initial conditions:

$$\begin{cases} \dfrac{d^i k_n(0)}{dt^i} = 0 & i = 0,1,2,\ldots,n-1 \\ \dfrac{d^n k_n(0)}{dt^n} = T^{-(n+1)} \end{cases}$$

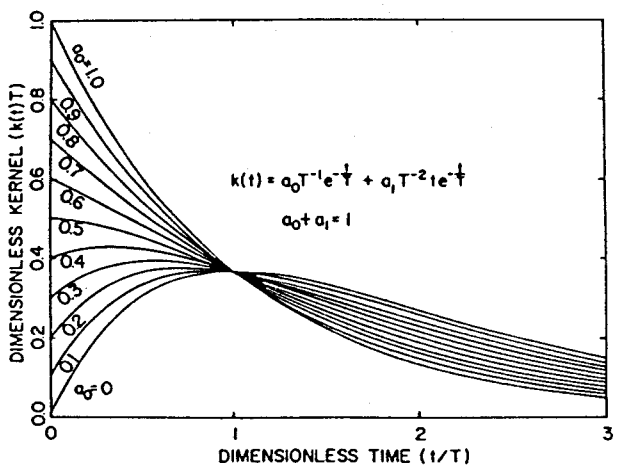


Fig. 9. Linear combination of the zero-order exponential distribution function, $k_0(t)$, and the first-order exponential distribution function, $k_1(t)$; $k(t) = a_0 k_0(t) + a_1 k_1(t)$, $a_0 + a_1 = 1$.

For example, $k_0(t) = T^{-1}e^{-t/T}$ satisfies

$$T\frac{dk_0}{dt} + k_0 = 0 \tag{33}$$

with initial condition:

$$k_0(0) = T^{-1}$$

Similarly, $k_1(t) = T^{-2}te^{-t/T}$ satisfies

$$T^2\frac{d^2k_1}{dt^2} + 2T\frac{dk_1}{dt} + k_1 = 0 \tag{34}$$

with initial conditions:

$$k_1(0) = 0$$

$$\frac{dk_1(0)}{dt} = T^{-2}$$

The above properties of the exponential distribution functions can be used to eliminate the kernel from the integrodifferential Eqs. (22) and (23) and convert them into a larger, but mathematically identical, set of first-order ordinary differential equations.

If we treat the integral containing the kernel as a new function $y_n(t)$,

$$y_n(t) \equiv \int_{-\infty}^{t} \mu(h)k_n(t - h)dh \tag{35}$$

then differentiating $y_n(t)$ with respect to t, $n + 1$ times with the help of Liebnitz's rule yields:

$$\sum_{i=0}^{n+1} T^i\binom{n+1}{i}\frac{d^iy_n(t)}{dt^i} = \mu(t) \tag{36}$$

The set of the resulting differential equations is one order higher than the kernel originally contained inside the integral. This higher-order differential equation can be easily transformed into a set of first-order differential equations through some well-known canonical transformations. Thus, for a simple nth-order kernel $k_n(t)$, the integral is transformed to the following set of equations:

$$\begin{aligned}
\frac{dy}{dt} &= z_1 \\
\frac{dz_1}{dt} &= z_2 \\
&\vdots \\
\frac{dz_{n-1}}{dt} &= z_n \\
\frac{dz_n}{dt} &= T^{-(n+1)}\left(-\sum_{i=0}^{n} T^i\binom{n+1}{i}z_i + \mu(t)\right)
\end{aligned} \tag{37}$$

Because of the linear properties of the differential and integration operators, a linear combination of more than one base functions of $k_n(t)$ will leave the approach unchanged. Thus if a first-order kernel has the form:

$$k(t) = (a_0T^{-1} + a_1T^{-2}t)e^{-t/T} \qquad (a_0 + a_1 = 1) \tag{38}$$

Then,

$$y(t) = \int_{-\infty}^{t} \mu[s(h)]k(t - h)dh$$

can be converted to a second-order ordinary differential equation:

$$T^2\frac{d^2y(t)}{dt^2} + 2T\frac{dy(t)}{dt} + y(t) = \mu(t) + a_0T\frac{d\mu(t)}{dt} \tag{39}$$

which can be further transformed to a mathematically equivalent set of two first-order ordinary differential equations.

$$\frac{dy(t)}{dt} = z \tag{40}$$

$$\frac{dz(t)}{dt} = -2T^{-1}z - T^{-2}y + T^{-2}\mu(t) + a_0T^{-1}\frac{d\mu(t)}{dt} \tag{41}$$

For example, with the kernel of Eq. (38), the system dynamic equations (22) and (23) are now:

$$\frac{dx}{dt} = (y - 1)x \tag{42}$$

$$\frac{ds}{dt} = 1 - s - \frac{1}{Y_s}yx \tag{43}$$

$$\frac{dy}{dt} = z \tag{44}$$

$$\begin{aligned}\frac{dz}{dt} &= -2T^{-1}z - T^{-2}y + T^{-2}\mu[s(t)] + a_0T^{-1}\frac{d\mu}{ds}\frac{ds}{dt} \\ &= -2T^{-1}z - T^{-2}y + T^{-2}\mu(s) \\ &\quad + a_0T^{-1}\frac{d\mu(s)}{ds} - a_0T^{-1}\frac{d\mu(s)}{ds}s - a_0T^{-1}\frac{1}{Y_s}\frac{d\mu(s)}{ds}yx\end{aligned} \tag{45}$$

Since, as it was mentioned, a first-order kernel is usually sufficient in describing bioreactor dynamics, the dependence of the specific growth rate (and other similar culture parameters) on the past history of the culture can thus be described with only two additional differential equations. This increase in the dimensionality of the system is a small price to pay considering the significantly enhanced predictive capabilities of the model.

The dynamics of a chemostat culture in the presence of time-delay kernels

has been analyzed with the use of linearized stability analysis and bifurcation theories. The full spectrum of dynamic behavior, including damped oscillations (when a zero-order kernel is included) and sustained oscillations (when a first-order kernel is included) can be predicted. The inclusion of kernels in other variables such as Y_s and x can also be analyzed in a similar manner. Furthermore, product formation, although not considered in this paper, can also be similarly studied. More detailed and complete results on the effect of time delay on the stability, classical process control consideration, and optimal control formulations will be the subject of forthcoming publications.

The experimental determination of the kernel has also been investigated for various transient situations. Shown in Figures 10(a) and 10(b) are the computer-simulated responses of a biochemical reactor described by Eqs. (20) and (21). From the noisy transient data of $\mu(t)$ and $y(t)$ when the dilution rate is shifted

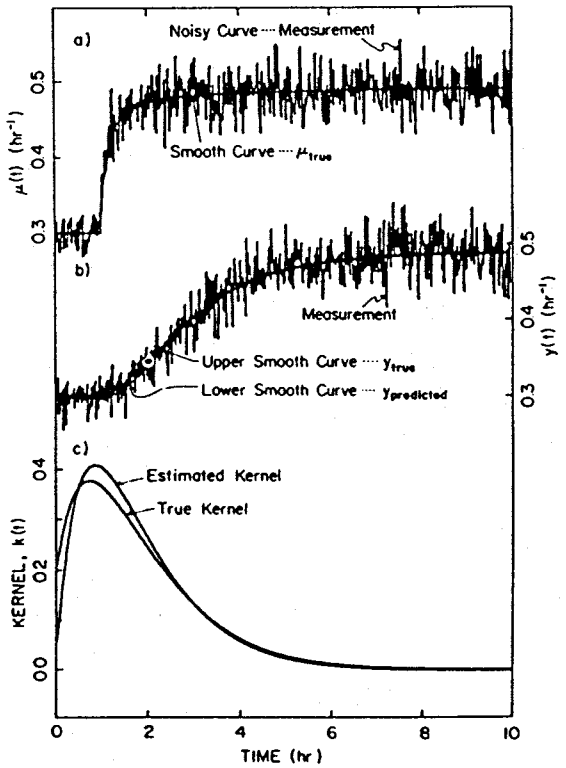


Fig. 10. (a) Simulated input (i.e., the specific growth rate in the absence of time-delay effects) as a function of time in a continuously operated bioreactor described by the state Eqs. (20) and (21) after a shift-up in the dilution rate from 0.3 to 0.7 h^{-1}. [Parameters used: $\mu = 0.5\, s/(0.1 + s)$; $s_f = 5.0$; $Y_s = 0.5$; noise level in measurement = 5%.] (b) Simulated output (i.e., the observed specific growth rate containing time-delay effects) as a function of time. [Upper smooth curve: the true value of $y(t)$; lower smooth curve: the calculated value of $y(t)$ based on the estimated kernel function of (c).] (c) True and estimated shapes of the kernel.

up from 0.3 to 0.7 h^{-1}, the kernel was reconstructed by minimizing the mean-square deviation of the $y(t)$ predicted by the kernel away from the observed $y(t)$. The resulting kernel is shown in Figure 10(c). Here, $\mu(t)$ is assumed to be the true specific growth rate in the absence of the effects of time delay. Given $s(t)$, this true specific growth rate $\mu(t)$ is obtained, in actuality, from a μ vs s curve constructed from a series of steady-state experiments, in which the time-delay effects are eliminated. For the purpose of this simulation, the μ vs s curve is assumed to follow the Monod model; however, it need not be so. Since the frequency response function or the pulse response function can be considered merely as another representation of the impulse response function (i.e., the kernel), sinusoidal or pulse methods can also be utilized to determine experimentally the shape of the kernel. The above example represents the worst case of estimating the impulse response function from a pulse experiment. Much better agreement between the true kernel and the estimated one can be achieved if an impulse can be applied to the system; the agreement is also considerably better if the noise level is decreased. (Of course, the reconstructed kernel coincides with the true one in the absence of noise.)

The above simulation study suggests that the first step in the experimental determination of the kernel is to construct a μ vs s curve through a series of steady-state runs. During a transient experiment in which the dilution rate or the feed substrate concentration is shifted up or down, the substrate concentration can be continuously estimated as a function of t, as shown in the previous sections. Furthermore, by referring to the μ vs s curve, $\mu[s(t)]$ can be generated continuously as well. The estimation scheme presented earlier can also be used to provide a continuous estimate of the instantaneous specific growth rate $y(t)$, and, finally, the kernel is generated.

Currently, work is under way in order to determine experimentally the shape of the kernel for a continuous culture of *S. cerevisiae*. Shift-up, shift-down, and sinusoidal perturbation experiments of the dilution rate, the substrate feed concentration, the pH, or the temperature are being performed. The on-line measurements with the aid of the parameter and state estimation algorithms described in the previous section will be used to determine the shape of the kernel function, and this new approach to bioprocess identification and modeling will be tested in terms of the model's capability in predicting the microbial behavior, including the more general and revealing behavior, such as the occurrence of sustained oscillation, under different conditions.

The difference between a complex structured model and a simple unstructured model is analogous to that between statistical and classical thermodynamics. Whereas a structured model tries to explain the observed phenomena through a large set of differential equations in terms of the more fundamental variables such as the concentrations of various intermediates; unstructured models are usually composed of those variables that can be physically "seen" or "felt" more readily and are, thus, more comprehensible to human minds. The proposed modeling approach herein attempts to retain the general form of an unstructured model so as to facilitate simple physical interpretation of the variables by such

familiar terms as the specific growth rate. At the same time, this modeling approach attempts to incorporate only those metabolic intermediates that are important to the dynamics of the system and to reduce the order of a complicated structured model through the analysis of eigenvalue–eigenvector of a linearlized system. How this can be accomplished is briefly outlined below.

In general, a dynamic system (including a structured model) can be described by a set of first-order differential equations:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},t) \tag{46}$$

where $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the input to the system as in Eq. (2). For a system linear in the state variables, the above equation can be written

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{g}(t) \tag{47}$$

The fundamental-matrix solution to the above differential equation is expressed by the following Lagrange formula:

$$\mathbf{x}(t) = \int_{-\infty}^{t} \mathbf{K}(t,h)\mathbf{g}(h)dh \tag{48}$$

If the linearization matrix $\mathbf{A}(t)$ is constant, then this solution further reduces to:

$$\mathbf{x}(t) = \int_{-\infty}^{t} \mathbf{K}(t-h)\mathbf{g}(h)dh \tag{49}$$

where K is the fundamental matrix of Eq. (46). Thus, the appearance of a kernel in Eqs. (18)–(21) is spontaneous; it arises mathematically during the process of solving a set of differential equations. The eigenvalue and eigenvector of the matrix $\mathbf{A}(t)$ can be analyzed to simplify and to reduce the dimension of the system by retaining only the first few most important modes and eliminating the remaining nonsignificant modes.

If the unstructured part of the system (i.e., biomass, substrate, product, etc.) are included in the state variable $\mathbf{x}(t)$, then the state variable can be grouped according to those that appear in the unstructured model $[\mathbf{x}_1(t)]$ and those that are contained only in the structured model $[\mathbf{x}_2(t)]$.

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_2(t) \\ \mathbf{x}_2(t) \end{bmatrix} \tag{50}$$

The linearization matrix $\mathbf{A}(t)$ and the nonhomogeneous forcing function $\mathbf{g}(t)$ can be partitioned similarly:

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{A}_{11}(t) & \mathbf{A}_{12}(t) \\ \mathbf{A}_{21}(t) & \mathbf{A}_{22}(t) \end{bmatrix} \tag{51}$$

$$\mathbf{g}(t) = \begin{bmatrix} \mathbf{g}_1(t) \\ \mathbf{g}_2(t) \end{bmatrix} \tag{52}$$

With this partition, Eq. (47) becomes

$$\frac{d\mathbf{x}_1(t)}{dt} = \mathbf{A}_{11}(t)\mathbf{x}_1(t) + \mathbf{A}_{12}(t)\mathbf{x}_2(t) + \mathbf{g}_1(t) \tag{53}$$

$$\frac{d\mathbf{x}_2(t)}{dt} = \mathbf{A}_{21}(t)\mathbf{x}_1(t) + \mathbf{A}_{22}(t)\mathbf{x}_2(t) + \mathbf{g}_2(t)$$

$$= \mathbf{A}_{22}(t)\mathbf{x}_2(t) + \tilde{\mathbf{g}}(t) \tag{54}$$

Thus, the unstructured model's equivalent of the structured model described by Eq. (46) is

$$\frac{d\mathbf{x}_1(t)}{dt} = \mathbf{f}_1(\mathbf{x}_1,\mathbf{x}_2,\mathbf{u},t) \tag{55}$$

where $\mathbf{x}_2$ is the delay kernel integral defined by:

$$\mathbf{x}_2 = \int_{-\infty}^{t} \mathbf{K}_{22}(t,h)\tilde{\mathbf{g}}(h)dh \tag{56}$$

where $\mathbf{K}_{22}(r,h)$ is the fundamental matrix to $\mathbf{A}_{22}(t)$ of Eq. (54). As can be seen from the preceding equations, the time-delay kernel arises quite naturally as a consequence of reducing a larger set of dynamic equations in a structured model by a smaller set of dynamic equations in an unstructured model.

IV. DISCUSSION

Microbial behavior depends not only on the present state of the environment but on past histories as well. This is the main reason for the inadequacy of the simple set of Eqs. (16) and (17). The dependence of a culture on its past history is manifested in the presence of a lag phase in the beginning of a batch cultivation. It is also present during the transients of continuous fermentors resulting from, among others, a shift-up of nutrient concentration. For example, Figure 3 shows that although glucose concentration was suddenly increasing at 0 h, the observed apparent specific growth rate did not start to increase until 1.1 h later. Such a lag has often been explained in terms of the need to synthesize the necessary pools of enzymes and intermediates before the rate of substrate utilization is adjusted to the changed conditions. The importance and the presence of time lag have been recognized for many years, and in this paper we have attempted to offer a simple mathematical means by which the idea of time delay can be incorporated into the existing models without drastically increasing the complexity of the models. Furthermore, by expanding the delay kernel in a series of exponential distribution functions, the integrodifferential equations can be easily reduced to a set of first-order ordinary differential equations for which the mathematical theories are well developed and various established techniques are available to analyze them.

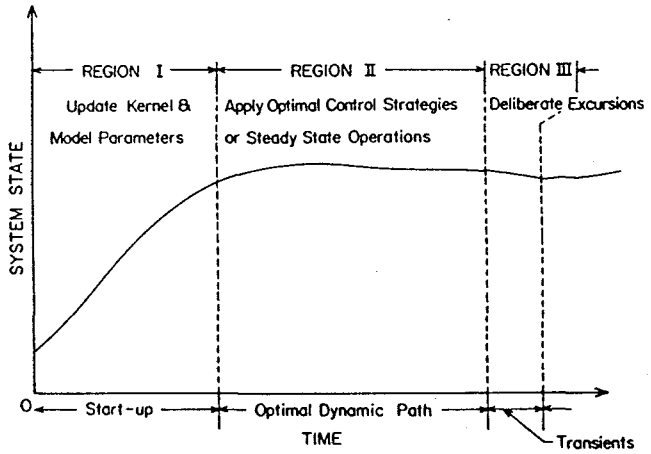


Fig. 11. Use of transient data for the determination of kernel and model parameters during the start-up of a bioreactor and the subsequent utilization of the model in control and optimization.

Much benefit can be derived from the recognition of time delay. It is a well-known fact that time delay can cause, among other undesirable problems, serious instability difficulties if it is neglected in a control strategy. Furthermore, an optimal control scheme may not be truly optimal if time delay is not properly considered. Figure 11 shows a hypothetical run in a bioreactor. The data collected during the short transient period after the start up can be used to update the shape of the kernel and other model parameters. Based on the updated model and model parameters and objective functions, an optimal path can be calculated by an on-line computer. Occasionally, deliberate excursions can be introduced to update the kernel and model parameters if they are suspected of gradual changes during a long steady-state run. A way in which a simple but powerful model such as the one proposed herein and the state and parameter estimation scheme discussed in the previous sections can be used is in the combined forward and feedback control of a bioreactor. Shown schematically in Figure 12 is an

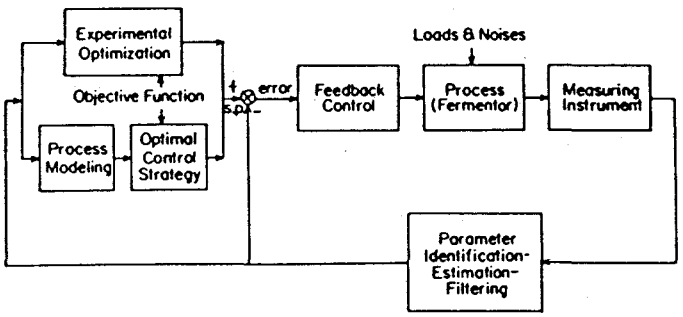


Fig. 12. Block diagram of the measurement–estimation–modeling–optimization–control configuration [11].

interactive estimation-control optimization scheme in which the on-line measurement on a bioreactor is passed through an estimation-filter block to get rid of the noise and to yield a set of on-line estimates for the state variables and growth parameters. These estimates are used as the basis for feedback control as well as for on-line process modeling. The biochemical process is continuously modeled, new values of the model parameters are estimated, and the biological model itself, including the shape of the kernel, is constantly updated. This can be accomplished by tracking the control history and comparing the deviation of the actual state away from the predicted values. Although such an ideal scheme does not exist presently, the state and parameter estimation and the new approach to modeling proposed herein are steps toward the realization of such a scheme.

A partial financial support for this work provided by ARCO's Young Faculty Investigator Award is gratefully acknowledged.

References

[1] C. L. Cooney, *Biotechnol. Bioeng. Symp.*, **9**, 1 (1979).

[2] D. I. C. Wang, I. Biocic, H. Y. Fang, and S. D. Wang, in *Proceedings of the 3rd Annual Biomass Systems Conference* (SERI, Golden, CO, 1979). pp. 61–67.

[3] G. Stephanopoulos, *Biotechnol. Bioeng.*, **23**, 2243 (1981).

[4] G. Stephanopoulos and K.-Y. San, "Studies on On-Line Bioreactor Identification. I. Theory," *Biotechnol. Bioeng.*, **26**, 1176 (1984).

[5] K.-Y. San and G. Stephanopoulos, "Studies on On-Line Bioreactor Identification. II. Numerical and Experimental Studies," *Biotechnol. Bioeng.*, **26**, 1189 (1984).

[6] R. Grosz, G. Stephanopoulos, and K.-Y. San, "Studies on On-Line Bioreactor Identification. III. Sensitivity Problems with Respiratory and Heat Evolution Measurements," *Biotechnol. Bioeng.*, **26**, 1198 (1984).

[7] R. M. May, *Stability and Complexity in Model Ecosystems* (Princeton U. P., Princeton, NJ, 1973).

[8] J. M. Cushing, *Integrodifferential Equations and Delay Models in Population Dynamics* (Springer-Verlag, Berlin, 1977).

[9] N. MacDonald, "Time Delays in Chemostat Models," in *Population Dynamics* (CRC Press, Boca Raton, FL, 1982), Chap. 2.

[10] R. Bellman and K. L. Cooke, *Differential-Difference Equations* (Academic, New York, 1963).

[11] N. S. Wang and G. Stephanopoulos, *CRC Critical Rev. Biotechnol.*, **2**(1), 1 (1984).