# STUDY OF THE STALL-SPIN PHENOMENA USING ANALYSIS AND INTERACTIVE 3-D GRAPHICS

Thesis by

Pascale C. Dubois

In Partial Fulfillment of the Requirements

for the Degree of

Aeronautical Engineer

June 1986

(submitted in December 1985)

California Institute of Technology

Pasadena, California

## ACKNOWLEDGEMENTS

I would like to thank Professor Fred E.C. Culick for suggesting the topic treated in this thesis and for his constant encouragement and guidance. My gratitude also goes to the members of my committee, Professor H. Liepmann and Professor A. Roshko, for their advice and concern and to Professor Antonsson for his help with the communications between the host computer and the graphic device. Financial support during the course of this research was provided by the California Institute of Technology. Special thanks go to my friends without whom this work would never have been accomplished. Finally, I would like to express my most sincere thanks to my friend, Janet Campagna, for editing this thesis.

# ABSTRACT

The purpose of this study is to gain a better understanding of the nonlinear stall-spin phenomenon through numerical analysis and interactive 3-D graphics.

The linear aerodynamic range was thoroughly examined for the NAVION, a light aviation aircraft. Nonlinear aerodynamic behavior was modeled by adding nonlinearities to the lift, pitching and rolling moments. The results of this analysis are promising; however, a more sophisticated model is needed to fully simulate the stall-spin phenomenon.

A graphic tool is described which allows the user to interact with the simulation process. This gives the user a "feel" for the dynamics of aircraft and effectively displays the characteristic features of the dynamic model.

## NOMENCLATURE

b.....Wing reference span

c.....Wing mean aerodynamic chord

$C_D$.....Dimensionless drag coefficient

CG.....Center of gravity

$C_L$.....Dimensionless lift coefficient

$C_l$.....Dimensionless rolling moment coefficient

$C_m$.....Dimensionless pitching moment coefficient

$C_n$.....Dimensionless yawing moment coefficient

$C_y$.....Dimensionless side force coefficient

D.....Drag force

$F_a$.....Aerodynamic and propulsive force

g.....Acceleration due to gravity

$G_f$.....Gravity force

h.....Altitude

$I_x, I_y, I_z$..Moments of inertia referred to body axis

$I_{xz}$.....Product of inertia referred to body axis

L.....Rolling moment about the x-body axis due to aerodynamic torque (positive right wing down)

L.....Lift

m.....Mass

M.....Pitching moment about the y-body axis due to aerodynamic torque (positive nose up)

N.....Yawing moment about the z-body axis due to aerodynamic torque ( positive nose right)

p.....Roll rate, angular velocity about x-body axis
(positive right wing down)

q.....pitch rate, angular velocity about y-body axis
(positive nose up)

$\bar{q}$.....Dynamic pressure, $\frac{1}{2} \rho V^2$

r.....Yaw rate, angular velocity about z-body axis
(positive nose right)

S.....Reference wing area

u.....Forward velocity along x-body axis

$u_o$.....Steady-state forward velocity

v.....Velocity along the y-body axis

V.....Magnitude of the velocity vector

$V_o$.....Reference velocity: $2 m g = \rho S V_o^2$

w.....Velocity along the z-body axis

$w_o$.....Steady-state downward velocity

W.....Weight of the aircraft

X,Y,Z.....Inertial axis (Z pointing downward)

x,y,z.....Body axis (x pointing forward, z downward)

$\alpha$.....Angle of attack

$\alpha_o$.....Steady-state (trim) angle of attack

$\alpha_S$.....Stall angle of attack

$\beta$.....Sideslip angle

$\gamma$.....Flight path angle

$\delta_a$.....Aileron control surface deflection (positive
for positive rolling moment)

$\delta_e$.....Elevator control surface deflection
( positive for nose-down pitching moment)

$\delta_r$.....Rudder control surface deflection

( positive for nose-left yawing moment)

Note positive for negative yawing moment.

$\theta$.....Pitch angle, positive nose up

$\theta_o$.....Steady-state pitch angle

$\rho$.....Mass density of air

$\phi$.....Roll angle, positive right wing down

$\psi$.....Yaw angle, positive for right nose

## NONDIMENSIONAL DERIVATIVE DEFINITIONS

LONGITUDINAL AXIS

$$C_L = \frac{L}{q\,S} \quad , \quad \text{positive up}$$

$$C_D = \frac{D}{q\,S} \quad , \quad \text{positive aft}$$

$$C_m = \frac{M}{q\,S\,c}$$

$$C_{L\alpha} = \frac{\partial C_L}{\partial \alpha}$$

$$C_{L\delta} = \frac{\partial C_L}{\partial \delta}$$

$$C_{D\alpha} = \frac{\partial C_D}{\partial \alpha}$$

$$C_{D\delta} = \frac{\partial C_D}{\partial \delta}$$

$$C_{m\alpha} = \frac{\partial C_m}{\partial \alpha}$$

$$C_{m\delta} = \frac{\partial C_m}{\partial \delta}$$

$$C_{mq} = \frac{2\,V_{T_e}}{c}\,\frac{\partial C_m}{\partial q}$$

LATERAL AXIS

$$C_y = \frac{Y}{q\,S}$$

$$C_l = \frac{L}{q\,S\,b}$$

$$C_n = \frac{N}{q\,S\,b}$$

$$C_{y\beta} = \frac{\partial C_y}{\partial \beta}$$

$$C_{l\beta} = \frac{\partial C_l}{\partial \beta}$$

$$C_{n\beta} = \frac{\partial C_n}{\partial \beta}$$

$$C_{y\delta} = \frac{\partial C_y}{\partial \delta}$$

$$C_{l\delta} = \frac{\partial C_l}{\partial \delta}$$

$$C_{n\delta} = \frac{\partial C_n}{\partial \delta}$$

$$C_{lp} = \frac{2\,V_{T_\varrho}}{b}\,\frac{\partial C_l}{\partial p}$$

$$C_{np} = \frac{2\,V_{T_\varrho}}{b}\,\frac{\partial C_n}{\partial p}$$

$$C_{lr} = \frac{2\,V_{T_\varrho}}{b}\,\frac{\partial C_l}{\partial r}$$

$$C_{nr} = \frac{2\,V_{T_\varrho}}{b}\,\frac{\partial C_n}{\partial r}$$

## SIGN CONVENTION

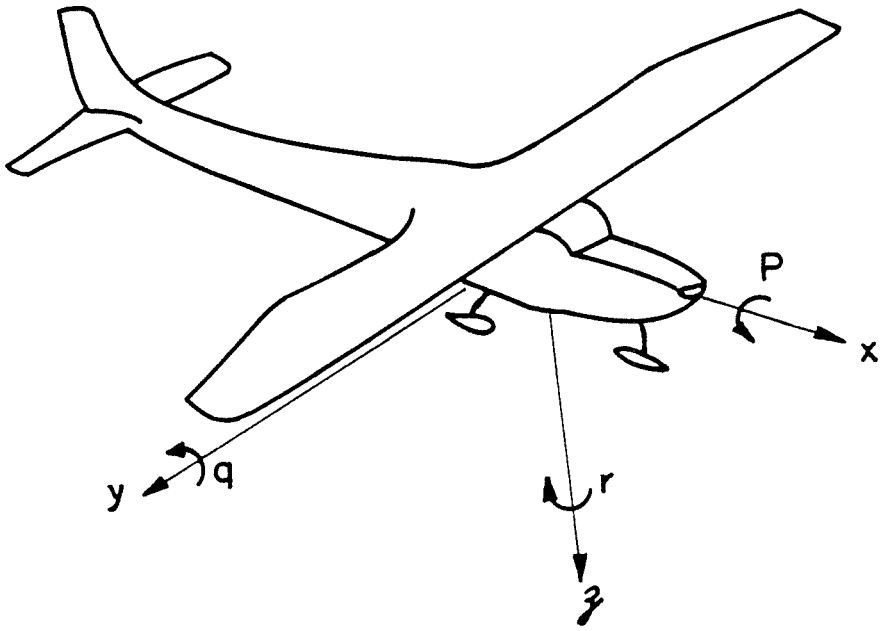


TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER I

## INTRODUCTION

The potential hazard associated with accidental stalling and spinning of aircraft has received different attention depending on the type of aircraft. For a military aircraft, the high angle of attack range is part of the maneuver domain, but for a general aviation aircraft the high angle of attack range is certainly not a part of normal operation. However, the safety concern, for all types of aircraft, motivates interest in this area due to the high fatality rate associated with such accidents.

In an effort to improve the safety record, which is of prime concern for general aviation aircraft, several controversial issues have arisen, including pilot training; e.g., should spin recovery be part of pilot testing and aircraft certification? Should an aircraft be able to recover from a fully developed spin?

Although a lot of work has been done in this area, the stall-spin phenomenon is still poorly understood and seems to be dependent on many parameters [1], [2]. In contrast, the theory of flight dynamics in the low angle of attack range is well developed, and in most cases the calculations are sufficient to give confidence in a specific design.

In the late 70's a significant research effort concentrated on that area [3], [4]. An extensive experimental program was launched, which included several standard testing methods ranging from easy to complex. Wind tunnel tests are usually the first to be performed. In conjunction with water channel flow visualizations, they provide an understanding of the general structure of the flow, as well as knowledge of the static stability of the airplane under study. The second stage is usually dynamic testing (forced oscillations)[5]. Combined with data obtained with a rotary balance, this set of experiments produces the information necessary for an analytical prediction method. However, the most reliable source of information on stall-spin characteristics, prior to actual flight tests, which are done last, is obtained by testing dynamically scaled airplane models. These models may be dropped from a helicopter or an aircraft , powered and radio-controlled, or flown in a spin tunnel. The most noticeable fruits of this work are the observation of the high nonlinearity of the phenomenon and the strong coupling between the different modes of motion. The inherent nonlinearities of stall

behavior prevent the generalization of tests from one configuration to another, and even a minor change in the geometry can induce some drastic effects , e.g., pitch-up moment at stall instead of pitch-down moment. No characteristic trends have been displayed by the studies other than the high costs of the experiments and the high sensitivity to many factors.

The decreasing cost of numerical studies relative to experimental studies can only strengthen the interest in analytical tools. Two main approaches are taken in the literature to study this problem numerically.

In the first approach [6], [7], the forces and moments of the airplane are computed using a table lookup method which yields coefficients based on test data (rotary balance and dynamic testings). The results are reliable, being in close agreement with experiments, and the computations are not complex.

The second, and by far the most ambitious approach , aims for a thorough analysis through the use of lifting surface theory [8]. Inputs to the program are typically geometry specifications and a 2-D lift curve. This method [9],[10] is complex and requires extensive computation. It is still in a development phase and the assumptions made are very restrictive. However, an obvious advantage is that an airplane need not actually be built before

using this method. Therefore, it can be used in the design phase. It should be emphasized that the reliability of this analytical tool has not been determined.

# PRESENTATION OF THE SUBJECT

The purpose of this study is to gain a better understanding of the stall-spin phenomenon and its associated nonlinear behavior , by using analysis and interactive 3-D graphics. The originality of this work resides in its use of graphics. Computational output is usually displayed by using plots of the several dependent variables as functions of time. However, with plots, even a slightly complex motion may become obscure and difficult to visualize. An example is given in Figure (1.a): The fuselage of the airplane is, in fact, decribing a cone while the airplane is rolling around its x body axis.

The graphic output designed for this study displays a plane "flying" in real time, some flight instrumentation and control input devices. Output files are generated for further study and/or playback . This arrangement allows the user to see the motion as it is computed and to influence it through the controls. The interaction feature is very important in testing a numerical model of the aerodynamics and to quickly point out the interesting characteristics of the flight dynamics associated with it. The code developed here is intended to be more a learning device than a simulator and no special efforts were made to have the model respond

as a specific airplane. The underlying idea was inspired by linear aerodynamics for which computations can be made easier by omitting those parameters which had little or no influence. Exploring the relative importance of the parameters through this interactive tool is the main goal of this work.

# CHAPTER II

## EQUATIONS OF MOTION

The motion of an aircraft is a 6-degree of freedom problem and therefore can be fully described by a set of six nonlinear coupled differential equations of the second order, representing the translational and rotational accelerations of the airplane in a body-fixed coordinate system [11].

$$F_x = m \left( \frac{du}{dt} + q\, w - r\, v \right);$$

$$F_y = m \left( \frac{dv}{dt} + r\, u - p\, w \right);$$

$$F_z = m \left( \frac{dw}{dt} + p\, v - q\, u \right);$$

(2.1)

$$N_x = I_x \frac{dp}{dt} + I_{xz} \frac{dr}{dt} + (I_z - I_y)\, q\, r + I_{xz}\, p\, q\ ;$$

$$N_y = I_x \frac{dq}{dt} + (I_x - I_z)\, r\, p + I_{xz}\, p\, q;$$

$$N_z = I_{xz} \frac{dp}{dt} + I_z \frac{dr}{dt} + (I_y - I_x) p\, q - I_{xz}\, q\, r.$$

In most airplanes, the y plane is a plane of symmetry; therefore, $I_{xy} = I_{yx} = 0$ .

The last two terms of each of the equations are the kinematic coupling terms due to the rotation of the axis of the aircraft. These terms represent the inertial nonlinearities in the system. Other sources of nonlinearities are the aerodynamic forces and moments which depend on angle of attack, angle of sideslip, velocity and rotational rates. In order to minimize the complexity of the equations (expressions of aerodynamic forces and moments), another set of variables was chosen. The equations of motion were transformed to $(u,\beta,\alpha)$ form. (Appendix I)

This body-fixed reference system is convenient for describing the aerodynamic forces and moments. However, a representation of the gravity field through Euler angles is then also necessary [12]. The Euler angle representation is as follows (see figure (2.a)). To transform the inertial axis into body axis, the inertial system is first rotated with respect to its Z-axis with an angle $\psi$, yaw angle, then by an angle $\theta$, pitch angle, with respect to the new y-axis and finally, by an angle $\phi$, roll angle with respect to the x-body axis. The transformation matrix is consequently:

$$R(\psi,\theta,\phi) = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi\cos\phi+\cos\psi\sin\theta\sin\phi & \cos\psi\cos\phi+\sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi+\cos\psi\cos\phi\sin\theta & -\cos\psi\sin\phi+\sin\psi\cos\phi\sin\theta & \cos\theta\cos\phi \end{bmatrix}.$$

The order in which the rotations are performed is important.

Any vector $\bar{V}$ expressed in the inertial reference system can be resolved in the body axis system :

$$\begin{Bmatrix} V_x \\ V_y \\ V_z \end{Bmatrix} = R(\psi,\theta,\phi) \begin{Bmatrix} V_X \\ V_Y \\ V_Z \end{Bmatrix} \tag{2.2}$$

This representation is valid and unique for $-\frac{\pi}{2} < \theta \leq \frac{\pi}{2}$, $0 < \psi \leq \pi$ and $0 < \phi \leq \pi$ .

The gravity force is along the Z-inertial axis, so

$$G_F = (m\ g\ \cos\theta\ \sin\phi\ ,\ -m\ g\ \sin\theta\ ,\ m\ g\ \cos\theta\ \cos\phi\ ), \tag{2.3}$$

in body axes.

The equations to be solved are :

$$\dot{u} = \frac{F_x}{m} - q\ u\ \tan\alpha + r\ u\ \frac{tg\beta}{\cos\alpha},$$

$$\dot{\alpha} = \frac{\cos\alpha}{m\ u}\left[F_z\cos\alpha - F_x\sin\alpha\right] - p\ tg\beta\cos\alpha + q - \sin\alpha tg\beta\ r,$$

$$\dot{\beta} = \frac{\cos\alpha\cos\beta}{m\ u}\left[\cos\beta\ F_y - \sin\beta\cos\alpha\ F_x - \sin\beta\sin\alpha\ F_z\right] - r\ \cos\alpha + p\ \sin\alpha,$$

$$N_x = I_x\ \frac{dp}{dt} + I_{xz}\ \frac{dr}{dt} + (I_z - I_y)\ q\ r + I_{xz}\ p\ q\ ,$$

(2.4)

$$N_y = I_x\ \frac{dq}{dt} + (I_x - I_z)\ r\ p + I_{xz}\ p\ q,$$

$$N_z = I_{xz}\ \frac{dp}{dt} + I_z\ \frac{dr}{dt} + (I_y - I_x)p\ q - I_{xz}\ q\ r,$$

$$\dot{\phi} = p + tg\theta\ (q\ \sin\phi + r\ \cos\phi),$$

$$\dot{\theta} = q \cos\phi - r \sin\phi,$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r,$$

with: $F_x = -m\, g \sin\theta - D \sin\alpha \cos\beta + L \sin\alpha + T,$

$$F_y = m\, g \cos\theta \sin\phi - D \sin\beta, \qquad (2.5)$$

$$F_z = m\, g \cos\theta \cos\phi - D \sin\alpha \cos\beta - L \cos\alpha.$$

Note here that $\dot{\theta} = q$, $\dot{\psi} = r$, $\dot{\phi} = p$ is true only in the linear range. The equations (2.5) are a complete representation of the mechanics of the system.

They have several singularities. Some of them, at $\beta = \pm 90°$, $u = 0$ and $\alpha = \pm 90°$, were ignored because of their unusual occurrence. The singularities at $\theta = \pm 90°$ could not be discarded as they can occur in maneuvers which are of interest to our study, i.e., looping and terminal phase of a spin. This singularity in the pitch angle is not in the physics of the problem but is introduced by the Euler representation: at $\theta = \pm 90°$, both roll and yaw angle are referenced with the same axis (the Z-inertial axis is then aligned with the x-body axis) so that the system can detect only $\phi - \psi$. This can also be seen in the transformation matrix $R(\phi, \theta = 90°, \psi)$ for $\theta = 90°$. $\phi$ and $\psi$ cannot be determined individually; thus, the equations have to be singular.

$$R(\phi,\theta=90°,\Psi) = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\phi-\psi) & \cos(\phi-\psi) & 0 \\ \cos(\phi-\psi) & -\sin(\phi-\psi) & 0 \end{bmatrix} \qquad (2.6)$$

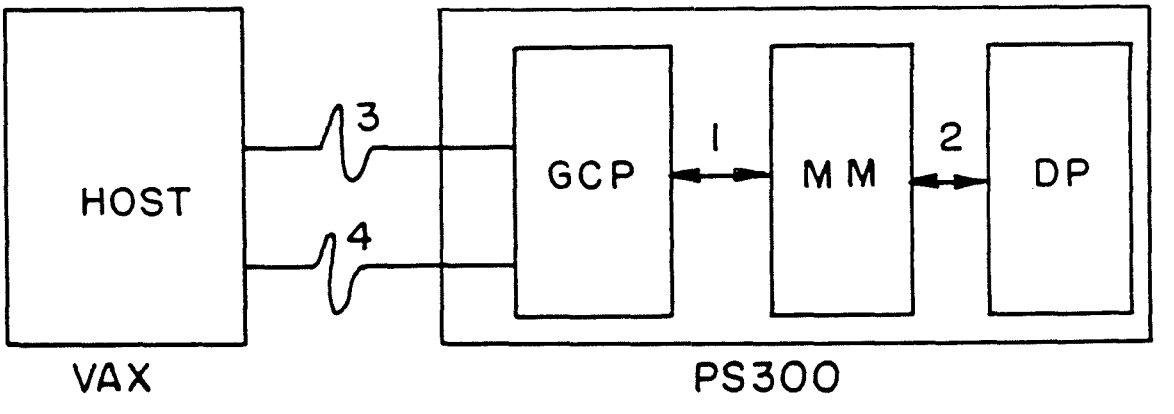
An analysis around these singularities was performed and is presented in Appendix II. However,because of a variable time step feature, which checks the value of the error, no further modifications of the program are necessary. The differential equation system solver, MODDEQ [13], sucessfully handles those singularities. This software subroutine uses a Runge-Kutta-Gill scheme featuring automatic control of truncation error and variable step size. The code is reasonnably fast ( 1000 time steps require about 15 CPU s). Results are presented and discussed in Chapter IV and Chapter V.

# CHAPTER III

## INTERACTIVE DISPLAY

### 3.1 General description of the PS300

The Evans and Sutherland PS300 is a graphic device which, through a Motorola 6800 microprocessor, can handle all graphic transformations independently from the host computer, in this case, a VAX-VMS; the interface is then best suited for infrequent communications of small amounts of data. The PS300 display rate is about 60 Hertz for a structure of up to 50,000 vectors; thus, the animation of the picture is smooth. A more technical description follows.



GCP : Graphics Control Processor

*controls communications with the host

*processes commands and creates data structures in the Mass Memory(MM)

*performs memory management.

MM : Mass Memory of 1 Megabyte

DP : Display Processor

*generates a picture on the screen.

Internal communications

*Interface 1 is a 16 bit-path.

*Interface 2 is an 8 bit-path.

External communications

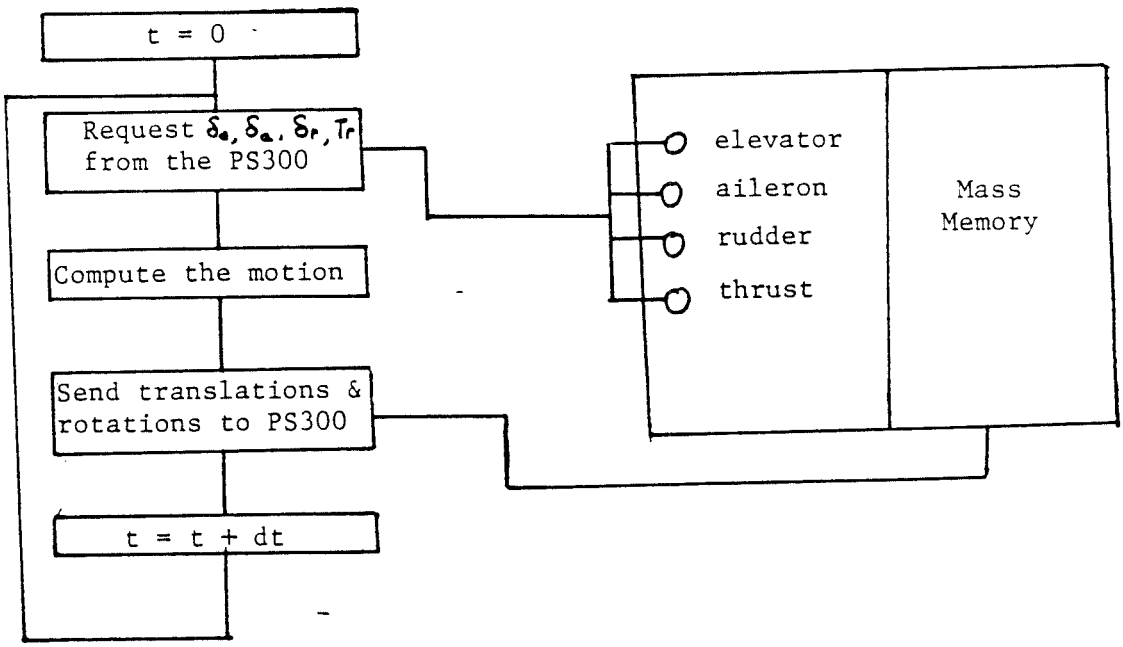
*Interface 3 is an asynchroneous line of 19.2 K baud.

*Interface 4 is a 16-bit parallel direct memory access 1 mbyte/s

## 3.2 Operation

First an ASCII file is downloaded through the serial interface to the PS300. This file will create all the display structure. For example, it will include a list of vectors representing the airplane as well as a function network. The function network establishes connections between transformations, a rotation, for example, and an object to be transformed (Appendix III). When all the display

structures are in the PS300 memory, a FORTRAN program is run on the VAX, which determines the airplane motion by solving the equations. The translations and rotations are then sent down to the nodes of the function network in the PS300 memory, and information such as control surface deflections and thrust setting are requested from the graphic device where they can be set interactivally by turning the control dials. The flowchart below summarizes the organization of the program. At each step of time, the VAX requests the elevator, the rudder, the aileron deflections, as well as the thrust setting from the PS300 using fast subroutines developped at CALTECH by Professor Antonsson; the program calculates then the new position in space and velocities of the airplane and sends translations, rotations and values to the PS300 functions network.



All the communications between the VAX and the PS300 are through the parallel interface. The VAX program writes directly into the PS300 memory, bypassing the graphics control processor that usually deals with communications with the host. The GCP can then process the display commands faster. Figure (3.a) shows the physical graphic setup. Figure (3.b) represents a looping maneuver through a series of three pictures. On Figure (3.c), a typical display is represented. The biplane is an EVANS and SUTHERLAND design (it was in a demonstration program). On the top left, the control surfaces panel is showing the deflections of the surfaces. On the bottom left , the angle of attack indicator is next to the speed indicator. The attitude instrument on the right shows the pitch angle (distance from the horizontal line), the sideslip angle (distance from the vertical line), and the roll angle (angle between the horizontal line and the symbolized airplane). The number on the bottom far left is the iteration number, giving to the display a time reference.

# CHAPTER IV

## LINEAR AERODYNAMICS

### 4.1 Linear aerodynamic model

The first step of this study is to reproduce the well-known motion of an airplane [11] corresponding to small angles of attack . In this range of angle of attack, the aerodynamic is linear; i.e., the aerodynamic coefficients are expressed as linear functions of the different angles and velocities. The expansions of these coefficients are chosen as follows:

$$
\begin{aligned}
C_L &= C_{L_o} + C_{L\alpha}\,(\alpha - \alpha_o) + C_{L\delta e}\,\delta_e, \\
C_D &= C_{D_o} + C_{D\alpha}\,(\alpha - \alpha_o), \\
C_Y &= C_{Y\beta}\,\beta + C_{Y\delta r}\,\delta_r, \\
C_l &= C_{l\beta}\,\beta + C_{lp}\,p + C_{lr}\,r + C_{l\delta r}\,\delta_r + C_{l\delta a}\,\delta_a, \\
C_m &= C_{m\alpha}\,(\alpha - \alpha_o) + C_{mq}\,q + C_{m\delta e}\,\delta_e,
\end{aligned}
\qquad (4.1)
$$

$$C_n = C_{n\beta}\,\beta + C_{np}\,p + C_{nr}\,r + C_{n\delta r}\,\delta_r + C_{n\delta a}\,\delta_a .$$

The derivatives are assumed constant. The value of these derivatives for the NAVION and the CHEROKEE 180 are shown in Appendix IV. This representation is very close to reality in the small angle of attack range. When the angle of attack is close to $\alpha_S$, the stall angle of attack, $C_{L\alpha}$ changes drastically and this model is no longer valid .

## 4.2 Longitudinal motion

In the equations of motion described in Chapter II , the longitudinal mode can be uncoupled from the lateral modes; i.e., a longitudinal motion stays longitudinal as long as no lateral perturbation occurs [14].

$$\begin{aligned} F_x &= m\left[\frac{du}{dt} + q\,w\right] \\ F_z &= m\left[\frac{dw}{dt} - q\,u\right] \\ N &= I_y\,\frac{dq}{dt}, \end{aligned} \tag{4.2}$$

where: $q = \frac{d\theta}{dt}$ and $\bar{F} = \bar{F}_a + \bar{G}_F$ .

These equations are the longitudinal equations of motion. No assumptions have yet been made concerning the mechanics of the system, but since the aerodynamic model limits the problem to small

angles of attack , the equations of motion can be linearized. First, they are nondimensionalized; then each variable is expressed as the sum of an equilibrium value and a perturbation part . Finally, the new expressions are introduced in the equations where the second-order terms are neglected (see Appendix V for this derivation). The resulting linear system is:

$$
\begin{aligned}
2\,\frac{du}{dt} &= C_{xu}\,u - C_{L_o}\cos\theta_o\,\theta \\
2\,\frac{d(\alpha-\theta)}{dt} &= C_{zu}\,u - C_{L_o}\sin\theta_o\,\theta \qquad (4.3) \\
\frac{2}{\mu}\left(\frac{k_y}{c}\right)^2\frac{d^2}{dt^2}\theta &= C_m,
\end{aligned}
$$

where: $\mu = \dfrac{m}{\rho S c}$,

$$\left(\frac{k_y}{c}\right)^2 = \frac{I_y}{mc^2}.$$

This motion can be studied analytically . Two characteristic motions are found: the phugoid mode, which is a long period, typically 20 seconds, lightly damped oscillation of the pitch angle and the velocity; and the short period, which is a rapid, heavily damped oscillation acting mostly on the angle of attack. When the short period is excited, an oscillation dies out in about 2 periods, typically 2 seconds. Consequently, this mode is difficult to display. In Figure (4.a) the short period mode is noticeable during the first few seconds, after which the phugoid mode takes over. However, crude estimates of its period and its damping are 1.4

seconds and -2.6, respectively. The period of the phugoid displayed in figure (4.b) is about 30.8 seconds and the damping -0.0155. Another example of a phugoid mode is given Figure 4.c, where the airplane describes a looping before the long period oscillation. The analytical values (Appendix VII) are, for the NAVION [15]:

Phugoid: T=30.1 seconds and damping=-0.0149

Short period: T=0.99 seconds and damping=-2.13.

The results from the numerical analysis are close to those of the theoretical.

## 4.3 Complete motion

The full set of equations (2.4) is then used in order to obtain the lateral modes of motion. The numerical model used to get the results described in this paragraph includes the complete nonlinearized , 6 degrees-of-freedom equations of motion as well as the linearized aerodynamic model discussed in paragraph (4.1). The program was first sucessfully tested in the case of longitudinal motion, and similar results to those in paragraph 4.2 were found.

The lateral motion is characterized by three modes, the roll subsidence, the spiral mode and the dutch roll. The roll subsidence is a stable mode, predominantly a heavily damped rolling motion. The spiral mode consists mostly of a heading change with small roll and sideslip angles. The dutch roll mode is a damped oscillation

where the three angles $\phi$, $\beta$, $\psi$ have approximately the same magnitudes. Typically, the time it takes to damp to half-amplitude is approximately one period. In the dutch roll mode, as the airplane yaws to the right, it slips to the left and rolls to the right; then the motion reverses; i.e, it yaws to the left, slips to the right and rolls to the left in a continuous process.

For the NAVION, both roll subsidence and spiral mode are stable [11]. The roll mode is very heavily damped ($\sigma$=-8.435). Consequently, both of these modes are almost impossible to notice, whereas the dutch roll has a strong influence and is easily characterized (see Figure 4.d). According to this plot, the period of the dutch roll is about 2.75 seconds and its damping is $\sigma$=-0.422. By linearizing the equations around an equilibrium position and by looking for a nontrivial solution of the form $\exp(\lambda t)$, a period of 2.69 seconds and a damping of -0.46 can be found analytically [11,15]. These results are, again, very close.

For the Cherokee 180, the roll subsidence is also damped but the spiral mode is a diverging mode [11] and the characteristic change of heading, $\psi$, is shown in Figure (4.e).

This finishes a complete analysis of the linear aerodynamic case; the agreement of the numerical and analytical results validates the mechanics of the model.

# CHAPTER V

# NONLINEAR AERODYNAMICS

## 5.1 Nonlinear aerodynamic model

Near stall, the general picture of the flow becomes more complex with cells of separation, stalled surfaces and vortex arrangement [11]. These phenomena interact in a manner which makes theoretical analysis difficult and the aerodynamic behavior highly nonlinear. A few degrees of change in the angle of attack can trigger drastic changes in the aerodynamic coefficients of the aircraft.

Autorotation phenomenon, an interesting behavior associated with the nonlinearities, was first explained by Glauert [16] as follows: For an aircraft flying at an angle of attack $\alpha_0$, a roll rate p can be damped or amplified depending upon the initial angle of attack $\alpha_0$. The angle of attack on the right wing, for a positive rolling moment, increases as the angle of attack of the left wing decreases. If the initial angle of attack is below stall, then the lift curve slope is positive and the difference in local angle of

attack creates a damping moment; i.e., the lift increases on the right wing and decreases on the left wing. If $\alpha_o$ is in the range of the negative lift curve slope, as the lift decreases on the right wing and increases on the left wing, the resulting positive aerodynamic moment amplifies the rolling motion (see Figure 5.a).

This can explain how the slightest nonsymmetrical perturbation encountered near stall transforms a symmetrical departure into a spin departure; i.e, the nonsymmetry is amplified by the same mechanism. However, experiments show that the autorotation leads to a steady roll rate [17] [20]. The experimental setup used to get these results restrains the degrees of freedom of the plane to a rotation. In the case of the actual airplane, the strong coupling between the degrees of freedom can be one of the reasons a steady autorotation rate is not reached.Since the spin entry appears to be related to the autorotation phenomenon, an aerodynamic model designed to produce such behavior should have the same instabilities, i.e., a negative lift curve slope and uncoupled wings, allowing the rolling amplification. A simple model gives insight into the driving mechanisms involved. The analysis done with the first model computes the angle of attack on each wing from the average angle of attack and the rolling rate (see Figure 5.a):

$$\alpha_{rw} = \text{Arctan} \left( \tan\alpha + \frac{p\ b}{4\ V} \right) \qquad (5.1)$$

$$\alpha_{lw} = \text{Arctan}\left( \tan\alpha - \frac{p\,b}{4\,V} \right).$$

The lift and rolling moment are then computed :

$$\begin{aligned}
&\text{If } -0.2 \leq \alpha < 0.2 \text{ , then } C_L = 0.406 + 4.44\,(\alpha-\alpha_o) + 0.2\,\delta_e \text{ ;}\\
&\text{if } 0.2 \leq \alpha < 0.362, \text{ then } C_L = (8\alpha-2.32)^3 - 8\alpha + 3.26 + 0.2\delta_e \text{ ;}\\
&\text{if } 0.362 \leq \alpha \text{ , then } C_L = 0.56 + 0.2\,\delta_e; \qquad (5.2)\\
&\text{if } -0.28 \leq \alpha < -0.2, \text{ then } C_L = (8\alpha+2.32)^3 - 8\alpha - 2.46 + 0.2\,\delta_e \text{ ;}\\
&\text{if } \alpha < -0.28, \text{ then } C_L = -0.219 + 0.2\,\delta_e \text{ .}
\end{aligned}$$

$$C_l = -0.074\,\beta + 0.5\,(C_{Lrw} - C_{Llw}) + 0.01\,r + 0.01\,\delta_r + 0.13\,\delta_a \text{ ,}$$

and finally, $$L = \bar{q}\,\frac{S}{2}\,(C_{Lrw} + C_{Llw}).$$

Figure (5.b) represents the lift curve.

As long as the system is experiencing a longitudinal motion, the behavior of the airplane is coherent. Some interesting features of this regime are reviewed in the next paragraph. When the angle of attack is greater than $\alpha_D$, i.e., deep-stall range, the airplane is very stable and responds accurately to the controls. On the contrary, a small lateral perturbation occurring in the negative lift-curve range ($\alpha_S < \alpha < \alpha_D$) is drastically amplified and induces some violent variations in the angles and velocities (see Figure 5.c). The solutions seem to be reaching a chaotic stage and the

airplane displayed on the interactive graphics device goes through erratic maneuvers.

## 5.2 Longitudinal motion

The 6-degree of freedom system has some strong instabilities that the longitudinal system does not have. The longitudinal motion computed with this model has some interesting features studied in further detail with a simplified 3-degree-of-freedom system including only the longitudinal variables. The system to be solved is then $4^{th}$ order. It is interesting to compare the linearized longitudinal nondimensional equations (2.4) to this set of nonlinear longitudinal nondimensional equations. (Appendix VI contains the derivation. )

$$\dot{u} = -\sin\theta - \frac{C_D}{\cos\alpha} u^2 + \frac{\sin\alpha}{\cos^2\alpha} C_L u^2 + T' - \dot{\theta} u \tan\alpha;$$

$$\dot{\alpha} = \frac{\cos(\theta-\alpha)\cos\alpha}{u} - \frac{u}{\cos^2\alpha} - T' \cos\alpha \sin\alpha + \dot{\theta} \qquad (5.3)$$

$$\theta = A C_m u^2,$$

where $A = \frac{M c V_0^2}{I_y g}$ and $\tilde{T} = \frac{T}{M g}$ and $C_L$ , $C_m$ as defined in paragraph (5.1).

A typical solution of these equations is shown in Figure (5.d), a plot of the angle of attack variation versus time . The angle of attack diverges first and then reaches a stable oscillatory motion. This behavior is called a limit cycle; A representation in phase-space is given in Figure (5.e), where the angle of attack is plotted versus the pitch angle. For a positive or null lift slope at equilibrium ($\dot{\alpha} = \dot{\theta} = \dot{u} = 0$) the motion is stable; i.e., a trajectory starting in the neighborhood of the equilibrium point will eventually spiral in to this point, but if the starting point is too far from the equilibrium point, the trajectory will reach a limit cycle. Figure (5.f) shows a typical phase-space plot for an equilibrium angle of attack of 12°. The equilibrium point and the unstable limit cycle (dashed line) as well as the stable limit cycle are represented here. When the trajectory starts inside the first curve (unstable limit cycle), it converges to the equilibrium point. When it starts outside this curve it tends to reach the stable limit cycle (the second curve) and stays there. The sequence of stable equilibrium point, unstable limit cycle and stable limit cycle is a typical configuration in a nonlinear dynamical system [18]. It should be noted that for $\alpha$ on the negative lift slope, no unstable limit cycle can be found; i.e, it does not matter how close to the equilibrium point the trajectory starts; it always goes back to the limit cycle.

Another major observation is that, depending on the control path, the results can be very different even if the same final control setting is reached. In Figure (5.g), two elevator setting histories are displayed as well as the associated angle of attack variations. In the plots on the left, a small positive elevator deflection provides, at first, the system with an excess in forward speed. Then the elevator is set to its final negative deflection; the airplane goes through a series of loopings and seems to reach a steady state in this looping mode. On the contrary, when the elevator is set directly to its final negative value (the same as before), the airplane encounters an oscillation in angle of attack which is amplified until it reaches a steady state.

Prior to pursuing this course of study , it is necessary to address the question: " Is this phenomenon relevant to the stall behavior or is it a parasite solution occurring because of an inaccurate model?". No definitive answer can be given. This limit cycle phenomenon, while possibly part of the driving mechanism of stall-spin departure, may not be observed in real tests because the strong lateral instabilities such as autorotation cause a spin departure. However, the model is very simple and consequently may include some major flaws. In particular, the straight $C_m$ curve is an oversimplification that may have led to inaccurate results. Therefore, more complex models have been tested, and these results are presented below.

## 5.3 Other nonlinear aerodynamic models

The negative lift slope range is of prime importance to the stall phenomenon; i.e, it triggers the strong lateral instabilities responsible for spin departure. A variable lift curve as represented in Figure (5.h) has been incorporated in the previous aerodynamic model. The parameter A is the slope of the curve in the stall range. Surprisingly, the results show little sensitivity to this parameter. The limit cycle behavior is not changed. The period of the oscillations changes very slowly.

The biggest flaw of the model is then the linear variation of the pitching moment versus $\alpha$. Some airplanes may experience a pitch-up moment at stall. The loss of efficiency in the controls is also usual because the control surfaces stall when the angle of attack is too high. Both effects are integrated into the model through the pitching moment shown in Figure (5.j). In this figure, the solid line represents the pitching moment coefficient when the elevator setting is zero. The new feature here is mainly the change of slope of this curve around stall. The slope is negative in the linear range and in the deep-stall range but positive for the stall-development range ( $\alpha_S < \alpha < \alpha_D$ ). The dotted line represents the pitching moment coefficient for an elevator setting of -0.05 radian. For large angles of attack, the dotted line lies on top of the solid line because, at these angles of attack, the elevator surface is stalled and has lost its efficiency. In the linear

range, the distance between the two lines is constant as predicted by the linear model (see Chapter IV). In the stall-development range, the efficiency of the control surface is decreasing exponentially. With this model, the airplane experiences a stall from which recovery is difficult (the elevator loses almost all its efficiency). The solution still has limit cycles, but the results look quite different. Introduction into the system of some further damping by the means of $C_{L_{\dot{\alpha}}}$ creates drastic changes; i.e., the limit cycles disappear. As the longitudinal equations of motion are back to a more stable behavior, we can try the 6-degree-of-freedom model.

This last nonlinear aerodynamic model was tried on the complete 6-degree of freedom system. The results are promising. At stall, a step-rudder input triggers a spin, but this motion is quickly damped and the airplane reaches a straight flight path after about half a turn. This behavior shows a lack in the aerodynamic model , more specifically, in the yawing moment. No nonlinearities are included in its expression . The steady developed spin requires a difficult balance between aerodynamic and kinematic moments [1], and an accurate description of these through the numerical method is necessary.

# CONCLUSIONS

This study has shown a high sensitivity of the results to the chosen aerodynamic model, e.g, damping through $C_{L_\alpha}$., etc. This fact seems to imply that a small change in the model might produce a completely different solution structure, and the validity of the idea that a simple aerodynamic model can give insight in the full phenomenon could be questioned. Furthermore, the stall-spin behavior and more specifically the steady developed spin involve a precise balance between kinematic and aerodynamic moments [1]. The kinematics of the system are exactly reproduced in the numerical analysis, but the aerodynamic model remains crude. Its major flaw is the linear yawing moment coefficient. Changing this part of the model is certainly the next logical step in reproducing stall-spin departure and subsequent motion.

The numerical analysis developed in this study does not yet provide a satisfactory simulation of the phenomenon, but much progress has been made in this direction. The coding is complete and the aerodynamic model can be changed without changing the main program. The communication problem between the graphic device and the VAX host computer is solved, and a fast information exchange rate is now possible through the parallel interface. The graphic tool is complete and provides the user with a rapid "feel" of the

characteristic features of the computed motion as well as its validity. This is difficult to do by just using plots. This tool was heavily used throughout the study. The pictures 3.a, 3.b, 3.c are the only output from the PS300 shown in this thesis, not because they are the only relevant things from the graphic display, but because only a movie could reproduce accurately the help gained through graphics and unfortunately a movie cannot be inserted into a book.

## REFERENCES

1. James S. Bowman Jr. : Summary of Spin Technology as Related to Light General Aviation Airplanes. NASA TN D-6575 [1971]

2. NASA : 1980 General Aviation Stall-spin Workshop. [1980]

3. J. R. Chambers and S. Grafton : Aerodynamics Characteristics of Airplanes at High Angles of Attack. NASA TM 740097. [1977]

4. Joseph R. Chambers : Overview of Stall/Spin technology. AIAA paper 80-1580. [1980]

5. K.J. Orlick-Ruckemann : On Aerodynamic Coupling Between Lateral and Longitudinal Degrees of Freedom AIAA 15$^{th}$ Aerospace Sciences Meeting.[1977]

6. Joseph R. Chambers, William P. Gilbert : Results of piloted Simulator Studies of Fighter Aircraft at High Angles of Attack AGARD CP-235. [1978]

7. D. Johnston, D. Mitchell, T. Myers : Investigation of High Angle of Attack Maneuver Limiting Factors. TR AFWAL-tr-80-3141. [1980]

8. W. Bollay : A Nonlinear Wing Theory and its Application to Rectangular Wings of Small Aspect Ratio. Z. angew. Math. Mech. Bd.19 Nr.1. [1939]

9. E. S. Levinsky : Theory of Wing Span Loading Instabilities near stall. AGARD CP-204. [1976]

10. M.A.Hreha and F.H.Lutze : A Dynamic Model for Aircraft

Poststall Departure.
Journal of Aircraft, Vol.21,No.1. [1983]

11. Barnes W. McCormick: Aerodynamics, Aeronautics, and Flight Mechanics. New York, Wiley,[1979].

12. James Wertz: Spacecraft Attitude Determination and Control. Amsterdam, Reidel,[1980].

13. Kiko Matsumoto: MODDEQ/ Differential Equation Solver. Mathematical Subroutine Library, California Institute of Technology. [1984]

14. Perkins and Hage: Aircraft Performance Stability and Control. New York, Willey. [1949]

15. G. Teper: Aircraft Stability and Control Data. NASA STI TR 176-1. [1976]

16. H.Glauert: The Rotation of an Airfoil about a Fixed Axis. Advisory Committee for Aeronautics, Rep. Memo. No. 595. [1919]

17. Montgomery Knight : Wind Tunnel Tests on Autorotation and "Flat Spin." NACA Report No. 273. [1927]

18. Guckenheimer and Holmes: Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields. Springer, Verlag, New-York. [1983]

20. H. Lugt : Autorotation. Ann. Rev. of Fluid Mech. 15:123-47. [1983]

# APPENDIX I

## The $(u,\alpha,\beta)$ Form of the Equations of Motion

We already know that the equations which express the linear velocities are:

$$F_x = m \left[ \frac{du}{dt} + q\,w - r\,v \right]$$

$$F_y = m \left[ \frac{dv}{dt} + r\,u - p\,w \right] \qquad \text{(I-1)}$$

$$F_z = m \left[ \frac{dw}{dt} + p\,v - q\,u \right].$$

If V is the magnitude of the velocity vector, $\alpha$ the angle of attack and $\beta$ the sideslip angle, then:

$$u = V \cos\alpha \cos\beta$$

$$v = V \sin\beta \qquad \text{(I-2)}$$

$$w = V \sin\alpha \cos\beta.$$

Therefore:

$$v = \frac{\tan\beta}{\cos\alpha}\, u$$

$$w = \tan\alpha\, u.$$

This representation reveals a singularity at $\alpha = \pm\frac{\pi}{2}$ which will not be considered because light aviation aircraft usually do not reach such extreme angles of attack.

We can then write:

$$\frac{dy}{dt} = \frac{\tan\beta}{\cos\alpha}\frac{du}{dt} + \frac{\dot{\beta}}{\cos\alpha\cos^2\beta}u + \dot{\alpha}\frac{\tan\beta\tan\alpha}{\cos\alpha}u$$

$$\frac{dw}{dt} = \tan\alpha\frac{du}{dt} + \frac{\dot{\alpha}}{\cos^2\alpha}u. \qquad (I\text{-}4)$$

Combining those equations with the above (I-1)equations we get :

$$\dot{u} = \frac{F_x}{m} - q\,u\,\frac{\tan\beta}{\cos\alpha}$$

$$\dot{\alpha} = \frac{\cos\alpha}{m\,u}\left[F_z\cos\alpha - F_x\sin\alpha\right] - \tan\beta\cos\alpha\,p + q - \sin\alpha\tan\beta\,r$$

$$\dot{\beta} = \frac{\cos\beta\cos\alpha}{m\,u}\left[\cos\beta\,F_y - \sin\beta\cos\alpha\,F_x - \sin\beta\sin\alpha\,F_z\right] - \cos\alpha\,r + \sin\alpha\,p.$$

# APPENDIX II

## Study of the Singularity in Pitch Angle

The equations which govern the variation of Euler angles are the following:

$$\begin{aligned} \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\phi} &= p + \tan\theta\,(\sin\phi\, q + r\cos\phi) \\ \dot{\psi} &= \frac{\sin\phi}{\cos\theta}\, q + \frac{\cos\phi}{\cos\theta}\, r, \end{aligned} \tag{II-1}$$

where $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$, $0 \leq \psi \leq \pi$ and $0 \leq \phi \leq \pi$.

The equations have several singularities. In order to gain a better understanding of them, we must first remember that they were artificially introduced by the choice of Euler representation and then we must analyze the behavior of the equations around these singularities. Let us consider a pure longitudinal motion. We have $p = r = 0$, and the equations become:

$$\begin{aligned} \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\phi} &= \tan\theta\sin\phi\, q \end{aligned} \tag{II-2}$$

$$\psi = \frac{\sin\phi}{\cos\theta}\, q.$$

The equilibrium position for $\phi$ or $\psi$ is $\phi = 0$ or $\frac{\pi}{2}$, i.e., points where $\dot{\phi} = 0$ and $\dot{\psi} = 0$ . When $\phi \simeq 0$, $\sin\phi$ behaves like $\phi$ and we get :

$$\dot{\phi} = q \tan\theta\, \phi.$$

If both q and $\theta$ were constants, then $\phi \approx \exp(q \tan\theta\, t)$ for small enough $\phi$, and $\phi$ would be a stable equilibrium position as long as $\theta < 0$; i.e., $\tan\theta < 0$ and unstable for $\theta > 0$. When $\theta$ approaches $\frac{\pi}{2}$, a small perturbation around $\phi = 0$ would have an infinite exponential growth, i.e., $\tan\theta \to \infty$ . In this case, the variation of q and $\theta$ can be considered slow in comparison to $\phi$. $\phi$ is expected to jump close to the stable equilibrium position, $\phi = \pi$, when $\theta$ is positive .

This, in fact, is the result when the program is run. When $\phi$ is exactly zero, then the pitch angle goes through the value $\frac{\pi}{2}$. If $\phi$ takes on a small nonzero value, the pitch angle remains less than $\frac{\pi}{2}$, and the roll and yaw angles jump by approximatively $\pi$. Figure (II-a) presents some typical results.

# APPENDIX III

## Presentation of the Function Network

Before any program is run on the VAX, a file containing PS300 commands is downloaded to the PS300. This file defines, in the PS300 memory, objects and transformations. As an example, all the PS300 commands affecting the speed indicator are pointed out in the command file.

The Vax is sending data directly into the PS300 memory to speed up the communication process. In order to avoid that a same buffer be accessed by both the PS300 (reading process) and the VAX (writing process), a double buffered structure is adopted. For example , when the VAX writes into the nodes called K***, the PS300 GCP traverses the nodes L*** and the process reverses; i.e., the VAX writes into the nodes L*** and the PS300 reads the nodes K***.

The PS300 command file containing all the functions network is shown below:

```
{-------------------------------------------------------------}
{ This is the speed indicator.                                }
{-------------------------------------------------------------}
{  First, we need to design the instrument and the graduations }
{-------------------------------------------------------------}
        line:=vector_list n=2
              -1,0 -0.75,0;
        line100:=rotate in z -25.7 then line101;
        line200:=rotate in z -51.43 then line;
        line300:=rotate in z -77.14 then line301;
        line400:=rotate in z -102.86 then line401;
        line500:=rotate in z -128.57 then line;
        line600:=rotate in z -154.28 then line601;
        line700:=rotate in z -180    then line;


        tch1:=begin_structure
           character size 0.02 ;
           translate by -1.2,-0.05;
           set char screen_ ;
              character '100';
              end_structure;
        line101:=instance of line,tch1;

        tch3:=begin_structure
```

```
          character size 0.02 ;
          translate by -1.1,-0.2;
          set char screen_ ;
             character '300';
             end_structure;
       line301:=instance of line,tch3;

       tch4:=begin_structure
          character size 0.02 ;
          translate by -1.075,0;
          set char screen_ ;
             character '400';
             end_structure;
       line401:=instance of line,tch4;

       tch6:=begin_structure
          character size 0.02 ;
          translate by -1.075,0 ;
          set char screen_ ;
     character '600';
             end_structure;
       line601:=instance of line,tch6;
{-------------------------------------------------------------------}
{ we create the needle                                              }
{-------------------------------------------------------------------}


       linem:=begin_structure
              scale by 0.2;
              vector_list n=2
              0,0 -1,0;
              end_structure;

{-------------------------------------------------------------------}
{  The nodes receiving information from the vax are then created    }
{   The double buffered feature can be noticed                      }
{-------------------------------------------------------------------}
{ The vax sends a rotation angle to the node called K100 or L100    }
{-------------------------------------------------------------------}


       k100:=rotate in z 0 then linem;
       l100:=rotate in z 0 then linem;

       velol:=instance of
 line,line100,line200,line300,line400,line500,line600,line700;
       velos:=scale by 0.2 then velol;

       k101:=instance of velos,k100;
       l101:=instance of velos,l100;
```

```
k102:=translate by 0,-0.95  then k101;
l102:=translate by 0,-0.95  then l101;


{-------------------------------------------------------------------}
{ the commands concerning the speed indicator are complete          }
{                                                                   }
{-------------------------------------------------------------------}

K000:=SET CONDITIONAL_BIT 1 ON APPLIED TO ABC;

ABC:=BEGIN_STRUCTURE
IF CONDITIONAL_BIT 1 ON THEN K021;
IF CONDITIONAL_BIT 1 OFF THEN L021;
END_STRUCTURE;



K001:=ROTATE IN Y 0 THEN k002;
L001:=ROTATE IN Y 0 THEN l002;

K002:=ROTATE IN Z 0 THEN K003;
L002:=ROTATE IN Z 0 THEN L003;

K003:=ROTATE IN X 0 THEN fu3;
L003:=ROTATE IN X 0 THEN fu3;

K031:=TRANSLATE BY -1,0.3,0 THEN K001;
L031:=TRANSLATE BY -1,0.3,0 THEN L001;

k008:=instance of k004,k007,k205;
l008:=instance of l004,l007,l205;

K005:=characters 'wait';
L005:=characters 'wait';

K006:=scale by 0.02 then K005;
L006:=scale by 0.02 then L005;

K007:=translate by 0.9,-0.9,0 then K006;
L007:=translate by 0.9,-0.9,0 then L006;

k004:=begin_structure
    a:=field_of_view 30
      front boundary = 2
      back boundary = 4;
    c:=set intensity on 0.75:1;
      instance of k031;
```

```
          end_structure;
        1004:=begin_s
            a:=field_of_view 30
              front boundary = 2
              back boundary = 4;
            c:=set intensity on 0.75:1;
              instance of 1031;
             end_s;
        DISPLAY K000;

{-------------------------------------------------------------------}
{ NOW WE DEAL WITH THE CONTROL SURFACES                               }
{-------------------------------------------------------------------}


        pri:=f:print;
        prif:=f:print;


        send 'reset cs' to <1>flabel1;
        send 'reset th' to <1>flabel2;

        strin:=characters -15,0 'AILERON  =';
        stri2:=characters -15,0 'ELEVATOR =';
        stri3:=characters -15,0 'RUDDER   =';
        stri4:=characters -15,0 'THRUST   =';

    {THIS ONE IS THE AILERON DEFLECTION}
        roll:=characters  '0';

    {THIS ONE IS THE ELEVATOR DEFLECTION}
        cora:=characters  '0';

    {THIS ONE IS THE RUDDER DEFLECTION}
        ampl:=characters  '0';

        thr:=characters '315';
        st2:=translate by 0,3,0 then str2;
        st3:=translate by 0,6,0 then str3;
        st4:=translate by 0,9,0 then str4;

        str2:=instance of stri2,cora;
        str3:=instance of stri3,ampl;
                str1:=instance of strin,roll;
        str4:=instance of stri4,thr;
        string:=instance of str1,st2,st3,st4;
        strinb:=scale by 0.02 then string;
        strinc:=trans by -0.5,0.75,0 then strinb;
        display strinc;
```

```
variable rudder,aileron,elevon,ITERATION,thrust;

rot:=f:dxrotate;
roy:=f:dxrotate;
rou:=f:dxrotate;
roth:=f:dxrotate;
send 10 to <3>rot;
send 10 to <3>roy;
send 10 to <3>rou;
send 50 to <3>roth;

        printr:=f:print;
        printy:=f:print;
        printp:=f:print;
        printth:=f:print;
COCA1:=F:CONCATENATEC;
COCA2:=F:CONCATENATEC;
COCA3:=F:CONCATENATEC;
coca4:=f:concatenatec;
        limr:=f:limit;
        lima:=f:limit;
        lime:=f:limit;
        limth:=f:limit;
send -15.0 to <3>limr;
send -15.0 to <3>lima;
send -15.0 to <3>lime;
send 0 to <3>limth;

send 15.00 to <2>limr;
send 15.00 to <2>lima;
send 15.00 to <2>lime;
send 800 to <2>limth;


pri:=f:print;

send 0.0 to <1>rudder;
send 0.0 to <1>aileron;
send 0.0 to <1>elevon;
send 315.0 to <1>thrust;
SEND 0 TO <1>ITERATION;
conn dials<1>:<1>rot;
conn dials<2>:<1>roy;
conn dials<3>:<1>rou;
conn dials<4>:<1>roth;
conn rot<2>:<1>limr;
conn limr<1>:<1>rudder;
conn roy<2>:<1>lima;
conn lima<1>:<1>aileron;
```

```
conn rou<2>:<1>lime;
conn lime<1>:<1>elevon;
conn roth<2>:<1>limth;
conn limth<1>:<1>thrust;


 conn printr<1>:<1>COCA3;
conn printp<1>:<1>COCA2;
conn printy<1>:<1>COCA1;
conn printth<1>:<1>coca4;

SEND '       ' TO <2>COCA1;
SEND '       ' TO <2>COCA2;
SEND '       ' TO <2>COCA3;
SEND '       ' TO <2>COCA4;

CONN COCA1<1>:<1>AMPL;
CONN COCA2<1>:<1>CORA;
CONN COCA3<1>:<1>ROLL;
conn coca4<1>:<1>thr;

conn limr<1>:<1>printy;
conn lima<1>:<1>printr;
conn lime<1>:<1>printp;
conn limth<1>:<1>printth;
send 'AILERON' to <1>dlabel2;
send 'ELEVator'to <1>dlabel3;
send 'RUDDER' to <1>dlabel1;
send 'thrust' to <1>dlabel4;



brou:=f:broutec;
eqc:=f:eqc;
broth:=f:broutec;
eqct:=f:eqc;
conn fkeys<1>:<1>eqc;
conn fkeys<1>:<1>eqct;
send fix(2) to <2>eqct;
send fix(1) to <2>eqc;
send 0.00 to <2>brou;
conn eqc<1>:<1>brou;
send 315.0 to <2>broth;
conn eqct<1>:<1>broth;
conn brou<1>:<1>roth;
conn broth<1>:<2>roth;
send 315 to <2>roth;
conn brou<1>:<1>rou;
```

```
        conn brou<1>:<2>rou;
        conn brou<1>:<2>rot;
        conn brou<1>:<1>rot;
        conn brou<1>:<2>roy;
        conn brou<1>:<1>roy;

{-------------------------------------------------------------------}
{       NOW we are downloading the flight instruments               }
{-------------------------------------------------------------------}
        scr:=vector_list n=10
                    -0.1,0.2 -0.2,0.1 -0.2,-0.1 -0.1,-.2 0.1,-.2
 .2,-0.1
                     .2,0.1 0.1,.2 -0.1,.2;
        gage:=vector_list itemized n=2
                   p .2,0 l -.2,0
                   p 0,.2 l 0,-.2;
        temoin:=vector_list n=4
                  0.04,0 0.004,0 0,0.01 -0.004,0 -0.04,0;

        k020:=translate by 0,0,0 then k010;
        l020:=translate by 0,0,0 then l010;

        k022:=translate by 0.75,-0.75,0 then k023;
        l022:=translate by 0.75,-0.75,0 then l023;

        k023:=instance of scr,gage,k020;
        l023:=instance of scr,gage,l020;

        k021:=instance of k022,k008,k102;
        l021:=instance of l022,l008,l102;

        K010:=ROTATE IN Z 0 THEN TEMOIN;
        L010:=ROTATE IN Z 0 THEN TEMOIN;



{-------------------------------------------------------------------}
{    ANGLE OF ATTACK INDICATOR                                      }
{-------------------------------------------------------------------}
        at_lin:=vector_list itemized n=6
                p 0,-1 l 0,1
                p -0.1,0.2 l 0.1,0.2
                p -0.05,0.362 l 0.05,0.362
                p -0.05,0    l 0.05,0
                p -0.1,-0.26 l 0.1,-0.26;
        at_rw:=vector_list n=1 0.5,0 0,0;
        at_lw:=vector_list n=1 -0.5,0 0,0 ;
        k206:=translate by 0,0,0 then at_rw;
```

```
        l206:=translate by 0,0,0  then at_rw;

        k207:=translate by 0,0 then at_lw;
        l207:=translate by 0,0 then at_lw;

        k205:=begin_structure
                translate by -0.8,-0.5,0;
                scale by 0.2;
                instance of k207,k206,at_lin;
          end_structure;
        l205:=begin_structure
                translate by -0.8,-0.5,0;
                scale by 0.2;
                instance of l207,l206,at_lin;
          end_structure;
{-------------------------------------------------------------------}
{      BIPLANE                                                      }
{-------------------------------------------------------------------}

PLANE := BEGIN_S
        VECTOR_LIST  BLOCK  ITEMIZED  N=157
                P  0.08936,-0.07446, 0.0
                L  0.08936,-0.08191, 0.07446
                L -0.05957,-0.08191, 0.07446
                L -0.05957,-0.07446, 0.0
                L -0.05957,-0.08191,-0.07446
                L  0.08936,-0.08191,-0.07446
                L  0.08936,-0.07446, 0.0
                L  0.08936,-0.08936, 0.14893
                L -0.05957,-0.08936, 0.14893
                L -0.05957,-0.07446, 0.0
                L -0.05957,-0.08936,-0.14893
                L  0.08936,-0.08936,-0.14893
                L  0.08936,-0.07446, 0.0
                L  0.08936,-0.09680, 0.22339
                L -0.05957,-0.09680, 0.22339
                L -0.05957,-0.07446, 0.0
                L -0.05957,-0.09680,-0.22339
                L  0.08936,-0.09680,-0.22339
                L  0.08936,-0.07446, 0.0
                L  0.08936,-0.10425, 0.29785
                L -0.05957,-0.10425, 0.29785
                L -0.05957,-0.07446, 0.0
                L -0.05957,-0.10425,-0.29785
                L  0.08936,-0.10425,-0.29785
                L  0.08936,-0.07446, 0.0
                L  0.08936,-0.11169, 0.37231
                L -0.05957,-0.11169, 0.37231
                L -0.05957,-0.11169,-0.37231
                L  0.08936,-0.11169,-0.37231
```

```
L  0.08936,-0.07446, 0.0
L  0.08936,-0.11914, 0.44678
L -0.05957,-0.11914, 0.44678
L -0.05957,-0.07446, 0.0
L -0.05957,-0.11914,-0.44678
L  0.08936,-0.11914,-0.44678
L  0.08936,-0.07446, 0.0
L  0.05957,-0.07446, 0.0
L  0.01489, 0.05212, 0.28296
L -0.02979,-0.10425, 0.29785
L -0.07446, 0.05212, 0.28296
L -0.02979,-0.07446, 0.0
L -0.07446, 0.05212,-0.28296
L -0.02979,-0.10425,-0.29785
L  0.01489, 0.04468,-0.28296
L  0.05957,-0.07446, 0.0
P -0.10425, 0.07446, 0.05957
L  0.04468, 0.07446, 0.05957
L  0.04468, 0.07446,-0.05957
L -0.10425, 0.07446,-0.05957
L -0.10425, 0.07446, 0.05957
L -0.10425, 0.06702, 0.13403
L  0.04468, 0.06702, 0.13403
L  0.04468, 0.07446, 0.05957
L  0.04468, 0.07446,-0.05957
L  0.04468, 0.06702,-0.13403
L -0.10425, 0.06702,-0.13403
L -0.10425, 0.07446,-0.05957
L -0.10425, 0.07446, 0.05957
L -0.10425, 0.05957, 0.20850
L  0.04468, 0.05957, 0.20850
L  0.04468, 0.07446, 0.05957
L  0.04468, 0.07446,-0.05957
L  0.04468, 0.05957,-0.20850
L -0.10425, 0.05957,-0.20850
L -0.10425, 0.07446,-0.05957
L -0.10425, 0.07446, 0.05957
L -0.10425, 0.05212, 0.28296
L  0.04468, 0.05212, 0.28296
L  0.04468, 0.07446, 0.05957
L  0.04468, 0.07446,-0.05957
L  0.04468, 0.05212,-0.28296
L -0.10425, 0.05212,-0.28296
L -0.10425, 0.07446,-0.05957
L -0.10425, 0.07446, 0.05957
L -0.10425, 0.04468, 0.35742
L  0.04468, 0.04468, 0.35742
L  0.04468, 0.07446, 0.05957
L  0.04468, 0.07446,-0.05957
L  0.04468, 0.04468,-0.35742
```

```
L -0.10425, 0.04468,-0.35742
L -0.10425, 0.07446,-0.05957
L -0.10425, 0.07446, 0.05957
L -0.10425, 0.03723, 0.43188
L  0.04468, 0.03723, 0.43188
L  0.04468, 0.07446, 0.05957
L  0.04468, 0.07446,-0.05957
L  0.04468, 0.03723,-0.43188
L -0.10425, 0.03723,-0.43188
L -0.10425, 0.07446,-0.05957
P  0.23828,-0.02979, 0.02979
L -0.13403,-0.02979, 0.02979
L -0.46167,-0.02979, 0.0
L -0.13403,-0.02979,-0.02979
L -0.05957,-0.02979,-0.02979
L -0.05957,-0.07446, 0.0
L -0.05957,-0.02979, 0.02979
L -0.05957,-0.02979,-0.02979
L  0.08936,-0.02979,-0.02979
L  0.08936,-0.07446, 0.0
L  0.08936,-0.02979, 0.02979
L  0.08936,-0.02979,-0.02979
L  0.23828,-0.02979,-0.02979
L  0.23828,-0.02979, 0.02979
L  0.23828, 0.0, 0.02979
L  0.08936, 0.01489, 0.04468
L -0.13403, 0.01489, 0.04468
L -0.46167, 0.0, 0.0
L -0.13403, 0.01489,-0.04468
L  0.08936, 0.01489,-0.04468
L  0.23828, 0.0,-0.02979
L  0.23828,-0.02979,-0.02979
L  0.23828, 0.04468,-0.02979
L  0.08936, 0.07446,-0.02979
L  0.04468, 0.07446,-0.02979
L  0.06702, 0.14893,-0.06702
L  0.08936, 0.07446,-0.02979
L -0.13403, 0.07446,-0.02979
L -0.46167, 0.02979, 0.0
L -0.13403, 0.07446, 0.02979
L  0.08936, 0.07446, 0.02979
L  0.06702, 0.14893, 0.06702
L  0.04468, 0.07446, 0.02979
L  0.08936, 0.07446, 0.02979
L  0.23828, 0.04468, 0.02979
L  0.23828,-0.02979, 0.02979
P -0.13403,-0.04468, 0.0
L -0.13403,-0.02979, 0.02979
L -0.13403,-0.02979,-0.02979
L -0.13403,-0.04468, 0.0
```

```
                L -0.40210,-0.02979, 0.0
                L -0.43188,-0.13403, 0.0
                L -0.46167,-0.13403, 0.0
                L -0.46167, 0.02979, 0.0
                L -0.49890, 0.02979, 0.0
                L -0.49890,-0.13403, 0.0
                L -0.46167,-0.13403, 0.0
                P -0.46167, 0.0,-0.14893
                L -0.46167, 0.0, 0.14893
                L -0.43188, 0.0, 0.14893
                L -0.40210, 0.0, 0.0
                L -0.43188, 0.0,-0.14893
                L -0.49890, 0.0,-0.14893
                L -0.49890, 0.0,-0.01489
                L -0.46167, 0.0, 0.0
                L -0.49890, 0.0, 0.01489
                L -0.49890, 0.0, 0.14893
                L -0.46167, 0.0, 0.14893
                P -0.07446, 0.07446,-0.02979
                L -0.02979,-0.10425,-0.29785
                L  0.01489, 0.05212,-0.28296
                L  0.05957,-0.10425,-0.29785
                L  0.01489, 0.07446,-0.02979
                L  0.01489, 0.07446, 0.02979
                L  0.05957,-0.10425, 0.29785
                L  0.01489, 0.05212, 0.28296
                L -0.02979,-0.10425, 0.29785
                L -0.07446, 0.07446, 0.02979 ;
        INST PLANE_1;
END_S;
PLANE_1 := ROTATE IN X 0. THEN PLANE_2;
PLANE_2 := VECTOR_LIST BLOCK ITEMIZED N=8
                P  0.23828, 0.0, 0.0
                L  0.26807, 0.0, 0.0
                L  0.27551, 0.01489, 0.09680
                L  0.26062,-0.01489, 0.09680
                L  0.23828, 0.0, 0.0
                L  0.26062, 0.01489,-0.09680
                L  0.27551,-0.01489,-0.09680
                L  0.26807, 0.0, 0.0 ;

fu3:=begin_structure
       rotate in x 180;
       scale by 0.4;
     instance of plane;
    end_structure;

PLANE_3 := F:CLFRAMES;
PLANE_4 := F:MODC;
PLANE_5 := F:XROTATE;
```

```
CONNECT PLANE_5<1>:<1>PLANE_1;
CONNECT PLANE_4<1>:<1>PLANE_5;
CONNECT PLANE_3<2>:<1>PLANE_4;
CONNECT PLANE_3<2>:<5>PLANE_3;
SEND FIX(360.) TO <2>PLANE_4;
SEND FIX(2.) TO <1>PLANE_3;
SEND FIX(1.) TO <2>PLANE_3;
SEND FALSE TO <3>PLANE_3;
SEND FIX(-34.) TO <4>PLANE_3;
SEND FIX(0.) TO <5>PLANE_3;
```

# Appendix IV

## Linear Aerodynamic Coefficients for the NAVION and the CHEROKEE 180

### NAVION [15]

| | | | | | |
|---|---|---|---|---|---|
| S ($ft^2$) | 180 | $C_{L_o}$ | =0.406 | $C_{l\beta}$ | =-0.074 |
| b (ft) | 33.4 | $C_{L\alpha}$ | =4.44 | $C_{lp}$ | =-0.410 |
| c (ft) | 5.7 | $C_{L\delta e}$ | =0.355 | $C_{lr}$ | =0.107 |
| W (lb) | 2,750 | $C_{D_o}$ | =0.05 | $C_{l\delta a}$ | =0.1342 |
| m (slugs) | 85.4 | $C_{D\alpha}$ | =0.33 | $C_{l\delta r}$ | =0.0118 |
| $I_x$ (slug-$ft^2$) | 1,048 | $C_{mq}$ | =-9.96 | $C_{n\beta}$ | =0.0701 |
| $I_y$ (slug-$ft^2$) | 3,000 | $C_{m\delta e}$ | =-1.74 | $C_{np}$ | =-0.0575 |
| $I_z$ (slug-$ft^2$) | 3,530 | $C_{y\beta}$ | =-0.564 | $C_{nr}$ | =-0.125 |
| $I_{xz}$ (slug-$ft^2$) | 0 | $C_{y\delta a}$ | =0 | $C_{n\delta a}$ | =-0.0035 |
| $V_{t_o}$ (ft/s) | 176 | $C_{y\delta r}$ | =0.157 | $C_{n\delta r}$ | =-0.0717 |
| $\alpha_o$ (deg) | 0.6 | $C_{m\alpha}$ | =-0.683 | | |

CHEROKEE 180 [11]

| | | | | | |
|---|---|---|---|---|---|
| S ($ft^2$) | 160 | $C_{L_o}$ | =0.543 | $C_{l\beta}$ | =-0.099 |
| b (ft) | 25.77 | $C_{L\alpha}$ | =4.68 | $C_{lp}$ | =-0.429 |
| c (ft) | 5.25 | $C_{L\delta e}$ | =0.934 | $C_{lr}$ | =0.198 |
| W (lb) | 2,400 | $C_{D_o}$ | =0.06 | $C_{l\delta a}$ | =0.0531 |
| m (slugs) | 74.5 | $C_{D\alpha}$ | =0.44 | $C_{l\delta r}$ | =0.0105 |
| $I_x$ (slug-$ft^2$) | 170 | $C_{mq}$ | =-7.42 | $C_{n\beta}$ | =0.0672 |
| $I_y$ (slug-$ft^2$) | 1,249 | $C_{m\delta e}$ | =-2.40 | $C_{np}$ | =-0.0905 |
| $I_z$ (slug-$ft^2$) | 1,312 | $C_{y\beta}$ | =-0.396 | $C_{nr}$ | =-0.0873 |
| $I_{xz}$ (slug-$ft^2$) | 0 | $C_{y\delta a}$ | =0 | $C_{n\delta a}$ | = |
| $V_{t_o}$ (ft/s) | 164 | $C_{y\delta r}$ | =0.117 | $C_{n\delta r}$ | =-0.0509 |
| $\alpha_o$ (deg) | 0.11 | $C_{m\alpha}$ | =-0.741 | | |

# APPENDIX V

## Linearized Longitudinal Equations

From the general longitudinal equations of motion :

$$F_x = m \left[ \frac{du}{dt} + q\, w \right],$$

$$F_y = m \left[ \frac{dw}{dt} - q\, u \right],$$

$$M = I_y \frac{dq}{dt},$$

a linearization around a trim position is done . Each term is expressed as the sum of an equilibrium value and a perturbation part. The equilibrium values are noted $(\ )_o$ as the varying parts are noted $(\ )'$.

$$u = u_o\ (1+ u')$$

$$w = w'\ u_o \qquad\qquad w_o=0$$

$$F_x = \bar{q}\, S\ (\ C_{x_o} + C'_x + C_{xg_o} + C'_{xg}).$$

Because the linearization is done around an equilibrium position, $C_{x_o} + C_{xg_o} = 0$.

$F_z = \bar{q}\, S\ (C_z + C_{zg})$, where $C_x$ and $C_z$ are the nondimensional forms of $F_{ax}$ and $f_{az}$, and $C_{xg}$ ,$C_{zg}$ are the dimensionless forms of $G_{fx}$ and

$G_{fz}$. $t' = \frac{t}{\tau}$, where $\tau = \frac{m\, u_o}{2\, q\, S}$. Note that the $(\ )'$ terms are dimensionless. This gives the nondimensional longitudinal linearized equations of motion.

$$\frac{du'}{dt'} = C_x + C_{xg}$$

$$\frac{d(\alpha-\theta)}{dt'} = \frac{1}{2}\left( C_z - C_{zg}\right)$$

Estimates of the derivatives.

$$F_a = \begin{Bmatrix} T - D\cos\alpha + \sin\alpha \\ 0 \\ -L\cos\alpha - D\sin\alpha \end{Bmatrix},$$

where T is the thrust, L is the lift and D the drag.

For $\alpha=0$ we get: $$F_a = \begin{Bmatrix} T - D \\ 0 \\ -L \end{Bmatrix}.$$

Therefore, for a propeller aircraft where T u =const, we have:

$$C_x = C_{xu}\, u + C_{x\alpha}\, \alpha, \qquad \text{where } C_{xu} = -3\, C_D \text{ and } C_{x\alpha} = C_L - C_{D\alpha}.$$

The gravity field is:

$$G_f = \begin{Bmatrix} -m\, g \sin\theta \\ 0 \\ -m\, g \cos\theta \end{Bmatrix}.$$

Therefore:

$$G'_f = \begin{Bmatrix} m\ g\ \cos\theta_o\ \theta' \\ 0 \\ -m\ g\ \sin\theta_o\ \theta' \end{Bmatrix} .$$

With $\alpha$ small, we can assume that $-L_o = m\ g$;

therefore, $C_{xg} = -C_{L_o}\ \cos\theta_o\ \theta'$ and $C_{zg} = -C_{L_o}\ \sin\theta_o\ \theta'$.

# APPENDIX VI

## Derivation of the Nondimensional Longitudinal Equations.

The equations of motion (2.4) are simplified to get the dimensional longitudinal equations by letting $\beta = \phi = \psi = p = r = 0$.

$$\frac{du}{dt} = \frac{F_x}{m} - q\,u\,\tan\alpha \;;$$

$$\frac{d\alpha}{dt} = \frac{\cos\alpha}{m\,u}\left[F_z\cos\alpha - F_x\sin\alpha\right] + q \;;$$

$$M = I_x\frac{dq}{dt} \;;$$

$$\frac{d\theta}{dt} = q \;,$$

where $F_x = -m\,g\sin\theta - D\cos\alpha + L\sin\alpha + T$ and

$$F_z = m\,g\cos\theta - D\sin\alpha - L\cos\alpha.$$

$$\frac{du}{dt} = -g\sin\theta - D\,\frac{\cos\alpha}{m} + L\,\frac{\sin\alpha}{m} + \frac{T}{m} - q\,u\tan\alpha;$$

$$\frac{d\alpha}{dt} = \frac{\cos\alpha}{u}\,g\cos(\theta-\alpha) - \frac{L\cos\alpha}{m\,u} - T\,\frac{\cos\alpha\,\sin a}{m\,u} + \frac{d\theta}{dt};$$

$$\frac{d^2\theta}{dt^2} = \frac{M}{I_y}$$

In order to nondimensionalize these equations, some reference velocity and time have to be chosen.

The reference velocity, $V_o$, is defined by : $m\,g = \frac{1}{2}\,\rho\,S\,V_o^2$.

The reference time, $t_o$, is defined by : $t_o = \frac{V_o}{g}$;

therefore, the reference length has to be: $l_o = \frac{V_o^2}{g}$.

Let $u^* = \frac{u}{V_o}$ , $t^* = \frac{t}{t_o}$ and $T^* = \frac{T}{m\,g}$; writing again the above equations we get:

$$\dot{u}^* = -\sin\theta - \frac{C_D}{\cos\alpha}\,u^{*2} + \frac{\sin\alpha}{\cos^2\alpha}\,C_L\,u^{*2} + T^* - \dot{\theta}\,u^*\tan\alpha;$$

$$\dot{\alpha} = \frac{\cos(\theta-\alpha)\,\cos\alpha}{u^*} - \frac{u^*}{\cos^2\alpha} - T^*\cos\alpha\,\sin\alpha + \dot{\theta}$$

$$\theta = A\,C_m\,u^{*2},$$

where $A = \frac{m\,c\,V_o^2}{I_y\,g}$ .

Dropping the * we recognize equation (5.3).

# APPENDIX VII

## Approximation to Phugoid and Short Period Modes

PHUGOID MODE.

Assume that $\dot{\alpha} = 0$ and ignore rotational energy and damping in pitch. Equation (4.2) becomes then:

$$C_{xu}\, u - 2\, \dot{u} - C_L\, \theta = 0$$

$$C_{zu}\, u + 2\, \dot{\theta} = 0.$$

Looking for a nontrivial solution of the form $\exp(\lambda\, t)$, we find that $\lambda$ has to satisfy the following equation.

$$4\, \lambda^2 - 2\, C_{xu}\, \lambda - C_L\, C_{zu} = 0$$

The solutions in $\lambda$ are:

$$\lambda = \frac{C_{xu}}{4} \pm \frac{i}{2}\sqrt{(-C_L\, C_{zu}) - \frac{C_{xu}^2}{4}}$$

where $C_{xu} = -3\, C_D$ and $C_{zu} = -2\, C_D$.

Note that in the preceding equations everything is dimensionless. To get the damping and the frequency in hertz we need to divide $\lambda$ by $\tau$, the time scale. We get: $\sigma = -\frac{3\, C_D\, g}{4\, C_L\, V}$ $\omega = \sqrt{2}\, \frac{g}{V}$

SHORT PERIOD MODE

It is mostly a rapid incidence adjustment. Assume a constant forward speed; i.e., $\dot{u} = 0$. This implies clearly that $\theta = \alpha$. Therefore, the angular momentum equation gives:

$$2K \frac{d^2\theta}{dt^2} = C_{m\alpha}\,\theta + C_{m\theta}\,\dot{\theta}.$$

Looking for a nontrivial solution in $\lambda$ as before gives:

$$\lambda = -\frac{1}{4K} C_{m\theta} \pm i \sqrt{-\frac{C_{m\alpha}}{2K} - \left(\frac{C_{m\theta}}{4K}\right)^2}.$$

# APPENDIX VIII

## Program Listing

```
        INCLUDE 'PROCONST.FOR/NOLIST'
        COMMON /PSBUF/JCT,JBUF(1000)
        INTEGER*2 JCT,IRATE
        LOGICAL*1 JBUF,CHAR(8)
        DATA JCT/1000/

        integer*4 IRETRY,ICHAN,IVALI1,IVALI2,IVALI3,ivali4,i
        INTEGER*2 IADRLO1,IADRLO2,IADRLO3,IADRHI1,IADRHI2,IADRHI3
        integer*2 iadrlo4,iadrhi4,iwrite
        real*4 X(3),x2(3),atr,atl,t
        REAL Y(9),DETE,TE,yd,ydo,u(3),ui(3),a(3,3),xfov,angle
        REAL YDOT(9),RUD,AIL,ELE,ite,dins(3),thr,clal,x3
        CHARACTER*80 DEVSPEC,label,string
        COMMON rud, ele,ail,thr,clal

c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$


        EXTERNAL cherokee
        EXTERNAL FUlila
        OPEN (UNIT=07,FILE='DEF.IN',STATUS='OLD')
        OPEN (UNIT=23,FILE='command.in',STATUS='OLD')
        OPEN (UNIT=22,FILE='input.in',STATUS='OLD')
        read(23,*)string
c If idata=0, no output files are generated, if idata=1 then output
 files
c are created. For example attac.out will contain the angle of
 attack
c versus time variations.

        read(23,*)idata
        if (idata.eq.1) then
        call open
        end if


        x2(3)=2
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
c  If the motion is not interactive then the user has decided which
```

```
c  motion to run. According to this choice, the program must run
 different
c  sequences.
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
        READ(23,*)iinter
        IF (IINTER.EQ.1) THEN
        OPEN (UNIT=02,FILE='PLAYBACK.IN',STATUS='NEW')
        END IF
        IF (IINTER.EQ.10) THEN
        OPEN (UNIT=02,FILE='playback.in',STATUS='old')
        END IF

        IF (IINTER.EQ.15) THEN
        OPEN (UNIT=02,FILE='looppb.in',STATUS='old')
        END IF


        READ(23,*)idisplay
        read(22,*)string
        READ(22,*)DETE
        write(*,*)'DETE=',dete
        READ(22,*)ITER
        read(22,*)devspec

        READ(22,*)Y(1)
        READ(22,*)Y(2)
        READ(22,*)Y(3)
        READ(22,*)Y(4)
        READ(22,*)Y(5)
        READ(22,*)Y(6)
        READ(22,*)Y(7)
        READ(22,*)Y(8)
        read(22,*)y(9)

        read(22,*)devspec
        read(22,*)clal
        read(22,*)ail
        read(22,*)ele
        read(22,*)rud

        thr=315.00
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
        TE=0
        K=1
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
c If the program is to be run interactivaly, then the VAX and the
 PS300
c are linked to allow fast communication rates.
```

```
        if (iinter.eq.1) then
        CALL PS3_PHY_ATTACH(ICHAN)

c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
        IRETRY=10
c In order to achieve a fast communication rate, the GCP is bypassed
c and the VAX program will request directly from a location in the
 PS300
c memory a value needed in the computation. For that, the VAX has to
 know
c the location in memory of some variables, i.e., rudder deflection.
c Those subroutines were developped at CALTECH by Doctor Antonsson
        label='rudder'
        CALL PS3_PHY_GET_ADDR(label,IRETRY,ICHAN,IADRHI1,IADRLO1)
        label='aileron'
        CALL PS3_PHY_GET_ADDR(label,IRETRY,ICHAN,IADRHI2,IADRLO2)
        label='elevon'
        CALL PS3_PHY_GET_ADDR(label,IRETRY,ICHAN,IADRHI3,IADRLO3)
        label='thrust'
        CALL PS3_PHY_GET_ADDR(label,IRETRY,ICHAN,IADRHI4,IADRLO4)
        end if


C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
c Then to send and display information on the screen, some PS300
 subroutines
c were used .
        if (idisplay.eq.1) then
        CALL PSINIT('K',IRATE)
        end if
        ifnum=10
        if (iinter.eq.1) then
        write(2,10)t,ail,ele,rud,thr
        end if
c-------------------------------------------------------------------
----
        DO 20 I=1,ITER
c-------------------------------------------------------------------
----

        if (i.ge.ifnum) then
        ele=0
        ail=0
        rud=0
c   Now that we have the address, it is fast to get a value from
c the PS300 memory. Those subbroutines were also developped by
 Doctor Antonsson.
        if (iinter.eq.1) then
       CALL
 PS3_PHY_READ_VAR(ICHAN,IADRHI1,IADRLO1,IRETRY,RUD,IVALI1)
```

```
      CALL
 PS3_PHY_READ_VAR(ICHAN,IADRHI2,IADRLO2,IRETRY,AIL,IVALI2)
      CALL
 PS3_PHY_READ_VAR(ICHAN,IADRHI3,IADRLO3,IRETRY,ELE,IVALI3)
      CALL
 PS3_PHY_READ_VAR(ICHAN,IADRHI4,IADRLO4,IRETRY,thr,IVALI4)
       RUD=RUD*3.1416/180
       AIL=AIL*3.1416/180
       ELE=ELE*3.1416/180
       write(2,10)te,ail,ele,rud,thr
       end if
       if (iinter.eq.10.or.iinter.eq.15) then
       read(02,10)t,ail,ele,rud,thr
       end if
10     format(5(x,e12.5))
       ifnum=ifnum+10

       end if

       if (iinter.ne.14) then
       CALL MODDEQ(FUlila,K,9,TE,Y,YDOT,DETE,1.E-4)
       goto 01
       end if
       CALL MODDEQ(cherokee,K,9,TE,Y,YDOT,DETE,1.E-4)
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$
c      COMPUTATION OF THE SPACE POSITION OF THE AIRPLANE
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$
 01    if (idisplay.eq.1) then

       YD=-Y(4)+TAN(Y(8))*(Y(5)*SIN(Y(7))+Y(6)*COS(Y(7)))
       YDO=Y(5)*COS(Y(7))-Y(6)*SIN(Y(7))
       u(1)=y(1)
       u(2)=y(1)*tan(y(3))/cos(y(2))
       u(3)=y(1)*tan(y(2))
       a(1,1)=cos(y(9))*cos(y(8))
       a(1,2)=-sin(y(9))*cos(y(7))+cos(y(9))*sin(y(8))*sin(y(7))
       a(1,3)=sin(y(9))*sin(y(7))+cos(y(9))*sin(y(8))*cos(y(7))
       a(2,1)=sin(y(9))*cos(y(8))
       a(2,2)=cos(y(9))*cos(y(7))+sin(y(9))*sin(y(8))*sin(y(7))
       a(2,3)=-cos(y(9))*sin(y(7))+sin(y(9))*sin(y(8))*cos(y(7))
       a(3,1)=-sin(y(8))
       a(3,2)=cos(y(8))*sin(y(7))
       a(3,3)=cos(y(8))*cos(y(7))


      do 30 j=1,3
      ui(j)=a(j,1)*u(1)+a(j,2)*u(2)+a(j,3)*u(3)
      x(j)=x(j)+ui(j)*dete/2000
```

```
 30     CONTINUE
        x3=x3+ui(3)*dete/2000
        end if
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$
C               WRITE THE RESULTS
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$
        atr=atan(tan(y(2))+y(4)*8.35/(Y(1)/(COS(Y(2))*COS(Y(3)))))
        atl=atan(tan(y(2))-y(4)*8.35/(Y(1)/(COS(Y(2))*COS(Y(3)))))
        if (idata.eq.1) then
        call write(y,te,x)
        end if


c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$
C          SEND DOWN ALL THE INFORMATION TO THE PS300
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$
        if (idisplay.eq.1) then
        x2(1)=x(1)
        x2(2)=-x(3)
        x2(3)=-x(2)
      if (x2(3).gt.4) then
         x2(3)=1
         end if
      if (x2(3).lt.1) then
         x2(3)=4
         end if
      xfov=x2(3)*0.2588

      if (x2(1).gt.xfov) then
         x2(1)=-xfov
         x2(2)=-x2(2)/2
         end if

      if (x2(1).lt.-xfov) then
         x2(1)=xfov
         x2(2)=-x2(2)/2
         end if

      if (x2(2).gt.xfov) then
         x2(2)=-xfov
         x2(1)=-x2(1)/2
         end if

      if (x2(2).lt.-xfov) then
         x2(2)=xfov
         x2(1)=-x2(1)/2
```

```
        end if

       x(1)=x2(1)
       x(2)=-x2(3)
       x(3)=-x2(2)

       dins(1)=y(3)/3
      if (dins(1).ge.0.08) then
       dins(1)=0.08
       end if

       if (dins(1).le.-0.08) then
       dins(1)=-0.08
       end if
       dins(2)=(mod(y(8)+3.1416,2*3.1416)-3.1416)/5
       if (dins(2).ge.0.19) then
       dins(2)=0.19
       end if

       if (dins(2).le.-0.19) then
       dins(2)=-0.19
       end if

       dins(3)=0

       encode(4,126,char)i
126    format(i4)
       call tran(206,0,atr,0)
       call tran(207,0,atl,0)
       CALL TRAN(031,X2(1),X2(2),X2(3))
       CALL ROT(001,Y(9),2)
       CALL ROT(002,Y(8),3)
       CALL ROT(003,-Y(7),1)
       CALL TEXT(005,4,CHAR)
       call tran(020,dins(1),dins(2),dins(3))
       CALL ROT(010,Y(7),3)
       call rot(100,-y(1)*3.1416/700,3)
       call rot(133,x3*3.1416,3)

              CALL NUFRAM


       end if

  20   CONTINUE
  11   FORMAT(E20.5)
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$
       if (iinter.eq.1) then
       CALL PS3_PHY_detACH(ICHAN)
```

```
      end if
      END



      subroutine open

      OPEN (UNIT=11,FILE='VELO.OUT',STATUS='NEW')
      OPEN (UNIT=12,FILE='ATTAC.OUT',STATUS='NEW')
      OPEN (UNIT=13,FILE='beta.OUT',STATUS='NEW')
      OPEN (UNIT=14,FILE='rora.OUT',STATUS='NEW')
      OPEN (UNIT=15,FILE='PIra.OUT',STATUS='NEW')
      OPEN (UNIT=16,FILE='yara.OUT',STATUS='NEW')
      OPEN (UNIT=17,FILE='phi.OUT',STATUS='NEW')
      OPEN (UNIT=18,FILE='teta.OUT',STATUS='NEW')
      OPEN (UNIT=19,FILE='psi.OUT',STATUS='NEW')
      OPEN (UNIT=20,FILE='verti.out',STATUS='new')

c     WRITE(11,*)'TITLE VELOCITY VERSUS TIME'
c     WRITE(12,*)'TITLE ANGLE OF ATTACK'
c     WRITE(13,*)'TITLE ANGLE OF sideslip'
c     WRITE(14,*)'TITLE roll rate'
c     WRITE(15,*)'TITLE PITCH RATE'
c     WRITE(16,*)'TITLE yaw rate'
c     WRITE(17,*)'TITLE phi'
c     WRITE(19,*)'TITLE psi'
c     WRITE(18,*)'TITLE teta'
c     WRITE(22,*)'TITLE vertical trajectory'

      return
      end


      subroutine write(y,te,x)
      real y(9),te,x(3)

      WRITE (11,10)TE,Y(1)
      WRITE (12,10)TE,Y(2)
      WRITE (13,10)TE,Y(3)
      WRITE (14,10)TE,Y(4)
      WRITE (15,10)TE,Y(5)
      WRITE (16,10)TE,Y(6)
      WRITE (17,10)TE,Y(7)
      WRITE (19,10)TE,Y(9)
      WRITE (18,10)TE,Y(8)
c      write (20,10)te,x(3)
 10   FORMAT(2(2X,E12.5))
```

```
        return
        end



C THIS SUBROUTINE DETERMINES THE AERODYNAMIC OF THE PROBLEM
C*************************************************************************
***
        SUBROUTINE FUlila(NN,TE,Y,YDOT)

        REAL CL,CD,CM,CY,CLL,CN,Q,L,D,SF,M,N,LL,FX,FY,FZ
        REAL Y(9),YDOT(9),TE,A,B,MU
        real RUD,AIL,ELE,thr,clal,effi
        INTEGER NN
        common rud,ele,ail,thr,clal
C Y(1) IS THE FORWARD VELOCITY
C Y(2) IS THE ANGLE OF ATTACK
C Y(3) IS THE SIDE SLIP ANGLE
C Y(4) IS THE ROLL RATE
C Y(5) IS THE PITCH RATE
C Y(6) IS THE YAW RATE
C Y(7) IS THE ROLL ANGLE, PHI
C Y(8) IS THE PITCH ANGLE, TETA
C Y(9) IS THE YAW ANGLE, PSI
        effi=1

C CL IS THE LIFT COEFFICIENT
C HERE FOR EXAMPLE CL ALPHA IS EQUAL TO 4.44


        CL=2*(0.203+2.22*(y(2)-0.0105)+0.1*ELE)
        IF (y(2).GT.0.2.AND.y(2).LE.0.22) THEN
        CL=1.247+0.2*effi*ele
        END IF
        IF (y(2).GT.0.22.and.y(2).lt.(0.22+0.7323/clal)) then
        CL=1.247-clal*(y(2)-0.22)
        END IF
        if (Y(2).gt.(0.22+0.697/clal)) then
        cl=0.55+0.2*effi*ele
        end if
        IF (y(2).LT.-0.2.AND.y(2).GE.-0.22) THEN
        CL=-0.4529
```

```
        END IF
        IF (y(2).lT.-0.22.and.y(2).ge.(-0.22-0.2673/clal)) THEN
        CL=-0.4529-clal*(y(2)+0.22)
        END IF
        if (y(2).lt.(-0.22-0.2429/clal)) then
        cl=-0.21
        end if
c$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$
C CD IS THE DRAG COEFFICIENT

 333     CD=0.05 + 0.33*(Y(2)-0.00105)

C CY IS THE SIDE FORCE COEFFICIENT

        CY=-0.564*Y(3) + 0.157*RUD

C CLL IS THE ROLL MOMENT COEFFICIENT
        CLL=-0.074*Y(3)-0.039*y(4)+0.0101*Y(6)+0.0118*RUD+
     1  0.1342*AIL


C CM IS THE PITCHING MOMENT COEFFICIENT
        CM=-0.683*(Y(2)-0.0105)-0.161*Y(5)-1.77662*ELE

C CN IS THE YAWING MOMENT COEFFICIENT

 CN=0.0701*Y(3)-0.00546*Y(4)-0.0118*Y(6)-0.0717*RUD-0.00346*AIL
C NOW THAT WE HAVE ALL THE COEFFICIENTS, LET'S COMPUTE THE FORCES.
        Q=0.5*0.002378*(Y(1)/(COS(Y(2))*COS(Y(3))))**2
C THE LIFT:
        L=Q*184*CL
C THE DRAG:
        D=Q*184*CD
C THE SIDE FORCE:
        SF=Q*184*CY
C THE YAWING MOMENT;
        N=Q*CN*184*33.4
C THE ROLLING MOMENT:
        LL=Q*CLL*184*33.4
C THE FORCE ALONG THE X-BODY AXIS:
      FX=-2750*SIN(Y(8))-D*COS(Y(2))*COS(Y(3))+L*SIN(Y(2))+thr
C THE FORCE ALONG THE Y-BODY AXIS:
        FY=2750*COS(Y(8))*SIN(Y(7))-D*SIN(Y(3))
C THE FORCE ALONG THE Z-BODY AXIS:
        FZ=2750*COS(Y(7))*COS(Y(8))
 -D*SIN(Y(2))*COS(Y(3))-L*COS(Y(2))
```

```
C NOW, WE WRITE THE EQUATIONS OF MOTION AS DESCRIBED IN APPENDIX.

        YDOT(1)=Y(6)*Y(1)*TAN(Y(3))/COS(Y(2))-Y(5)*Y(1)*TAN(Y(2))
     1          + FX/85.4

C HERE THE PITCHING MOMENT IS COMPUTED
        M=Q*184*CM*5.7


 YDOT(2)=-Y(6)*SIN(Y(2))*TAN(Y(3))+Y(5)-Y(4)*TAN(Y(3))*COS(Y(2))
     1          +COS(Y(2))/(Y(1)*85.4)*(-FX*SIN(Y(2))+FZ*COS(Y(2)))

      YDOT(3)=
 cos(y(2))*cos(y(3))*(cos(y(3))*fy-sin(y(3))*(cos(y(2))
     1  *fx+ sin(y(2))*fz))/85.4/y(1)-y(6)*cos(y(2))+y(4)*sin(y(2))
      A=(3000-3530)*Y(5)*Y(6)+LL
      B=(1048-3000)*Y(4)*Y(5)+N
        YDOT(4)=A/1048
        YDOT(5)=M/3000-(1048-3530)*Y(6)*Y(4)/3000
        YDOT(6)=B/3530

        YDOT(7)=Y(4)+TAN(Y(8))*(Y(5)*SIN(Y(7))+Y(6)*COS(Y(7)))
        YDOT(8)=Y(5)*COS(Y(7))-Y(6)*SIN(Y(7))

        ydot(9)=y(5)*sin(y(7))/cos(y(8))+y(6)*cos(y(7))/cos(y(8))
        RETURN
        END
```

# FIGURES

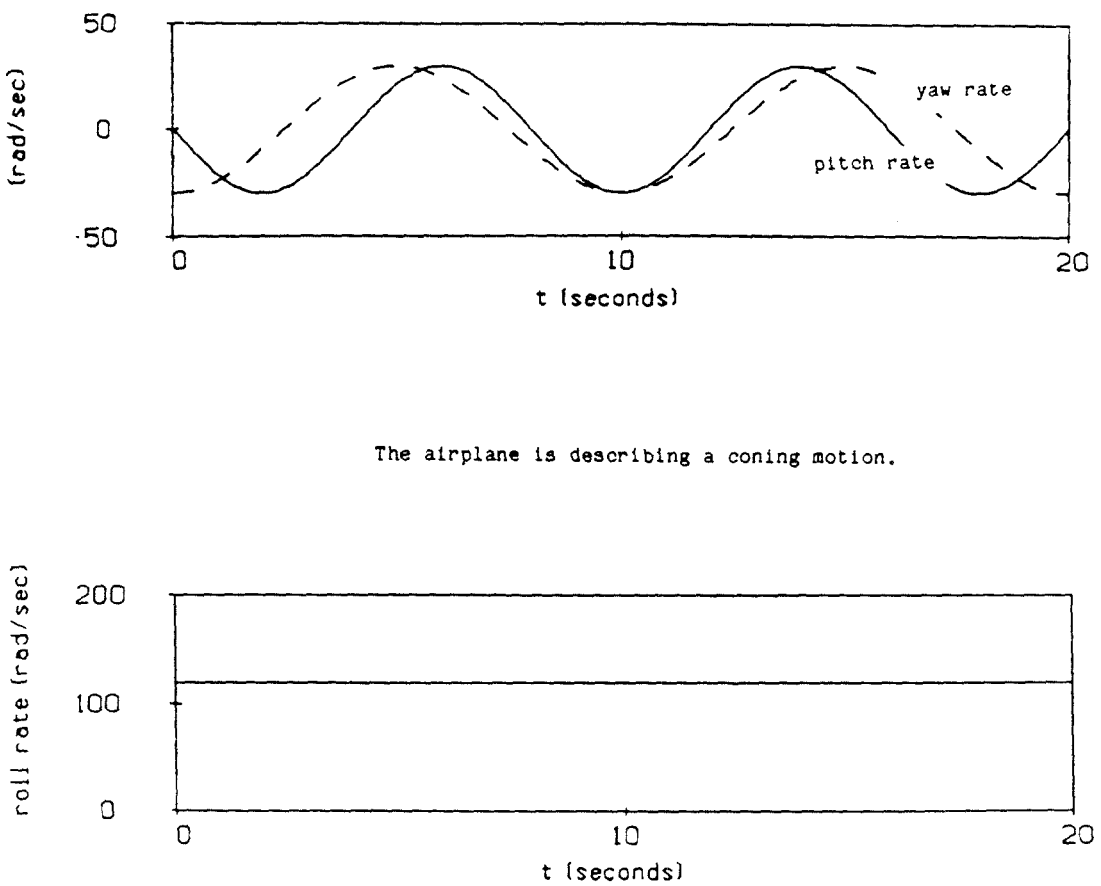



FIG.1.a AN EXAMPLE OF MOTION REPRESENTATION

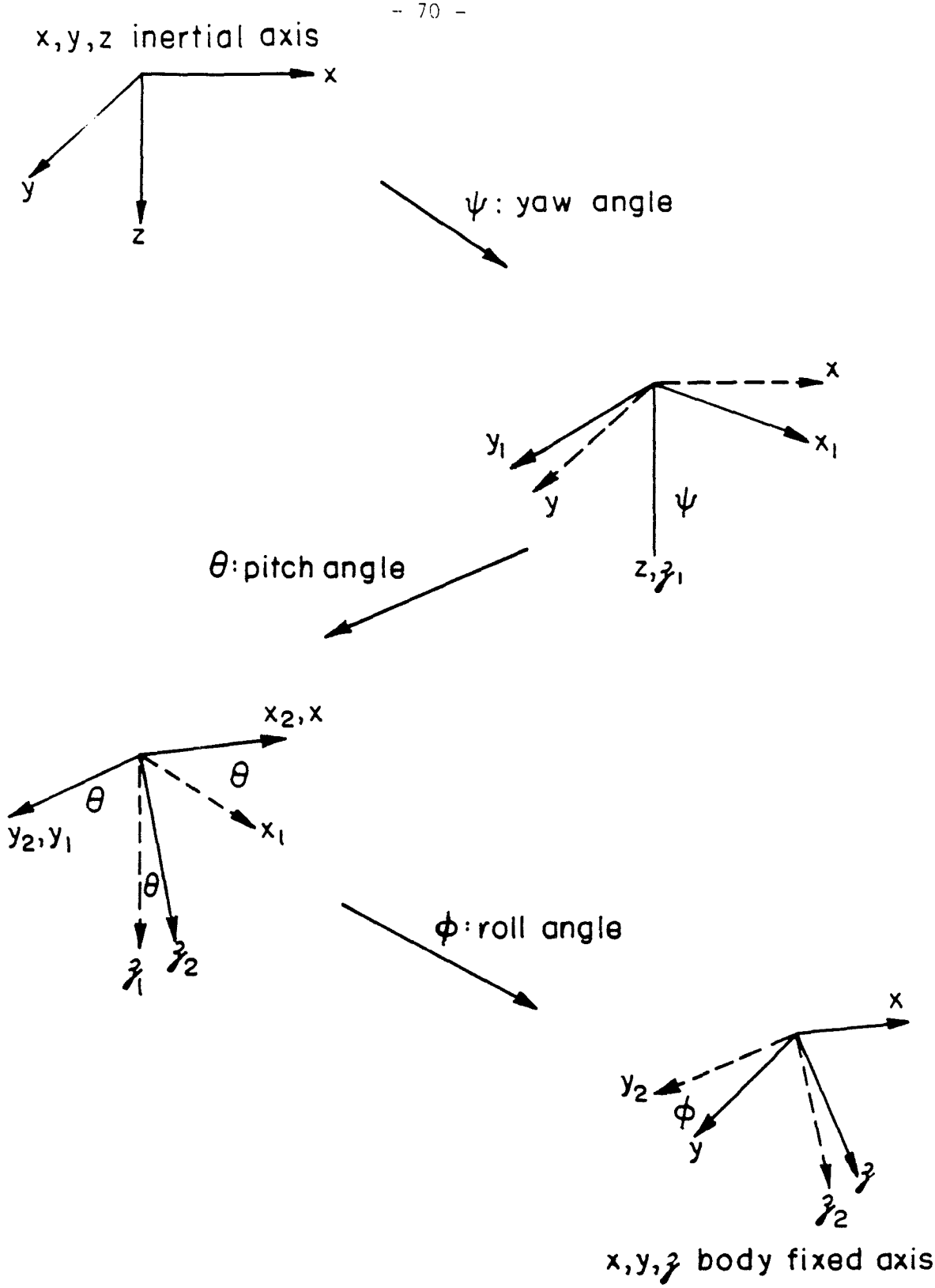
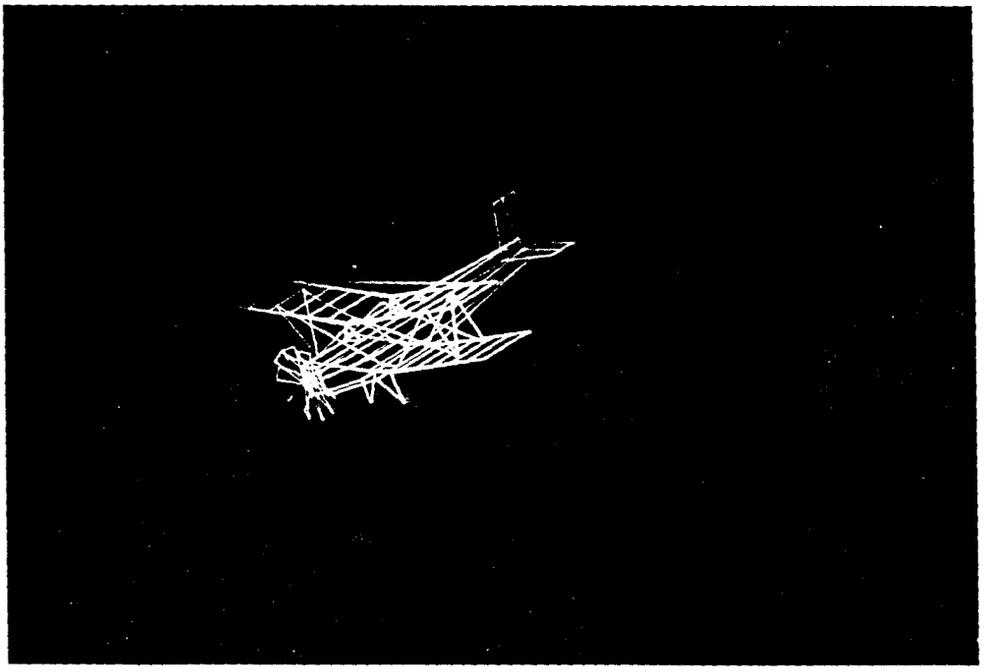



FIG. 2a EULER ANGLE REPRESENTATION



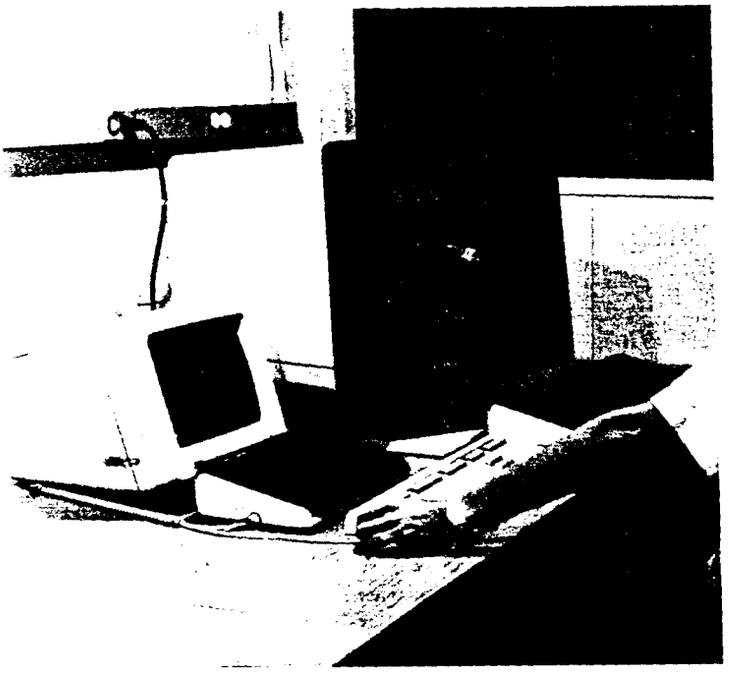


FIG.3.a. GRAPHIC SETUP

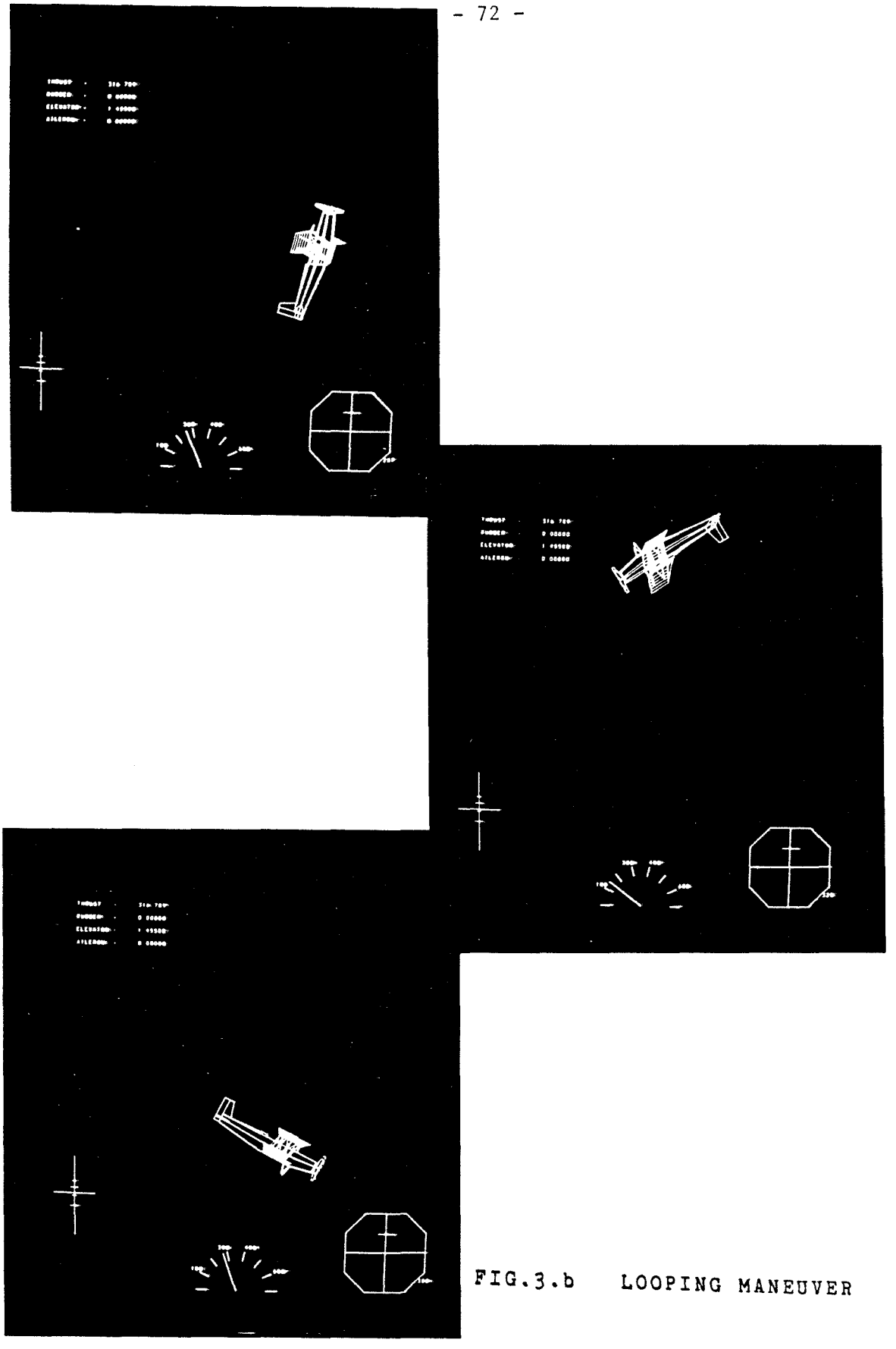
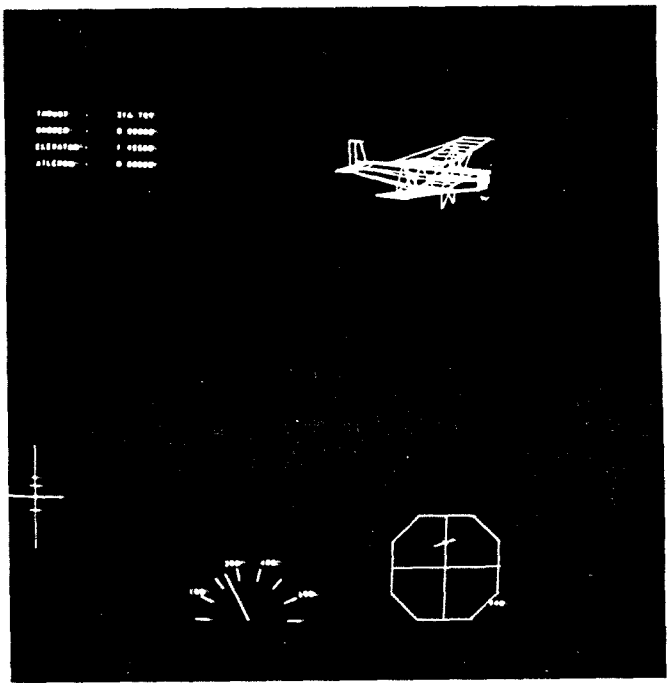
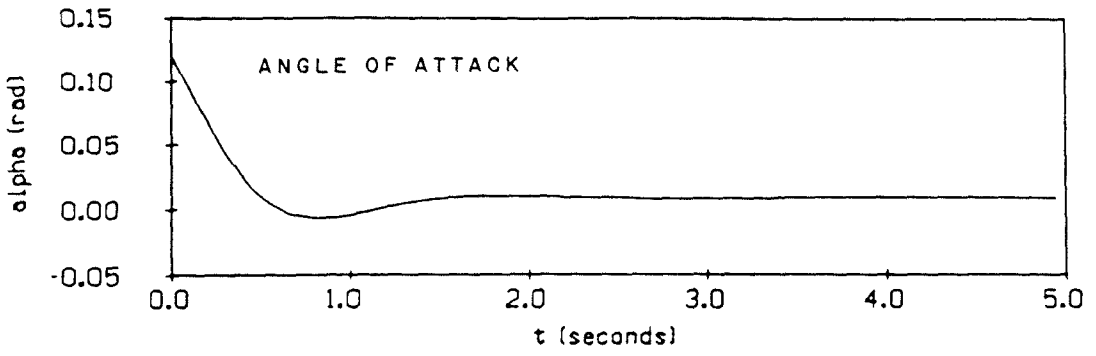


FIG.3.b LOOPING MANEUVER



On the top left, the control panel indicating the elevator, rudder, aileron deflections as well as the thrust setting. On the bottom left, the angle of attack indicator. On the bottom center a speed indicator in feet per second. On the bottom right an attitude indicator . The symbolic airplane position displays the roll angle (angle with the horizontal line), the pitch angle (distance from the horizontal line) and the sideslip angle (distance from the vertical line).

FIG.3.c DISPLAY ORGANIZATION



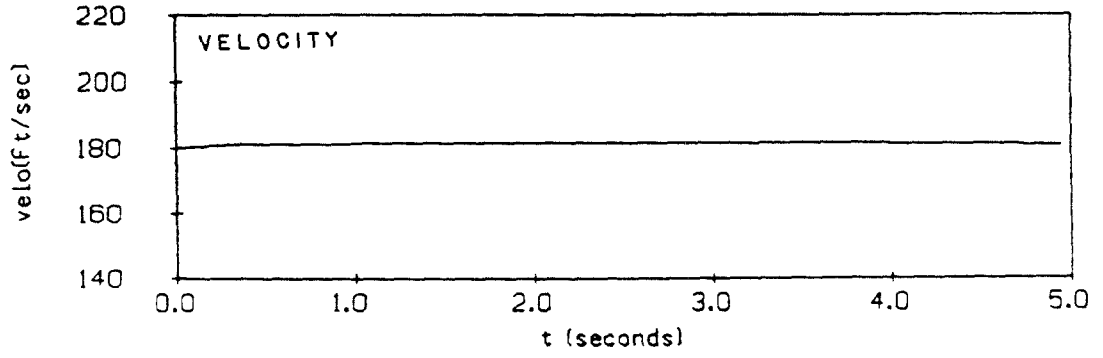
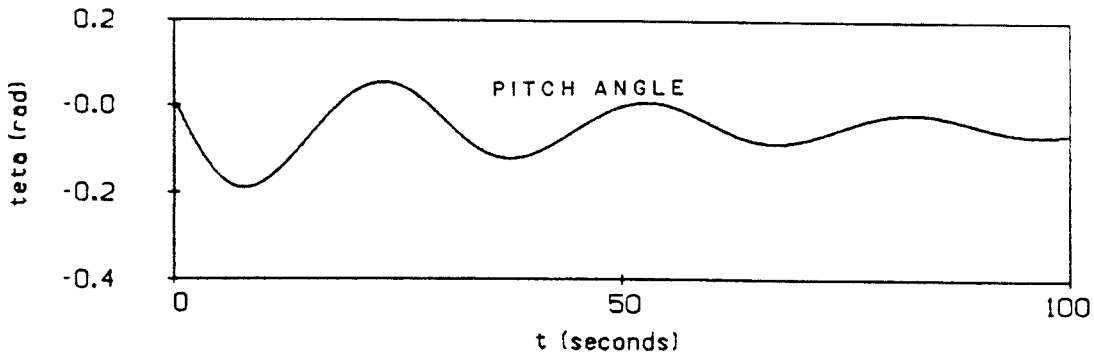




FIG.4.a SHORT PERIOD



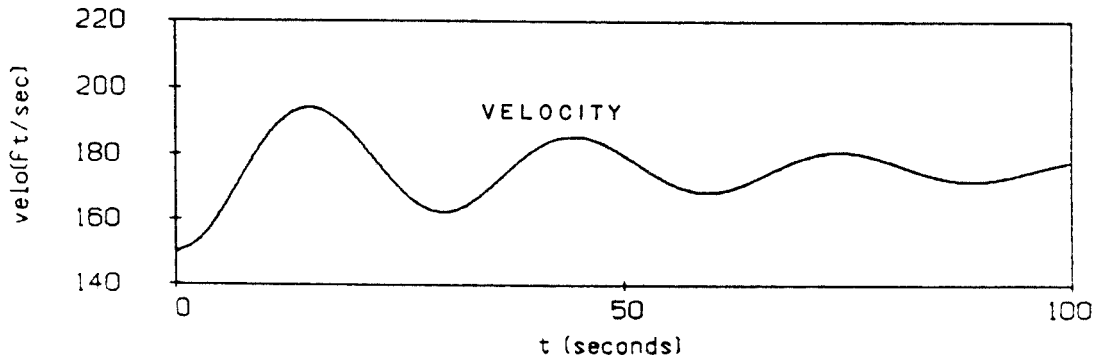




FIG.4.b PHUGOID MODE

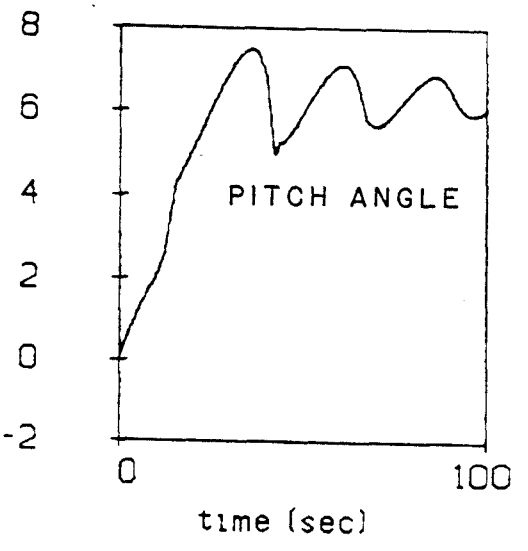



FIG.4.c LOOPING AND PHUGOID

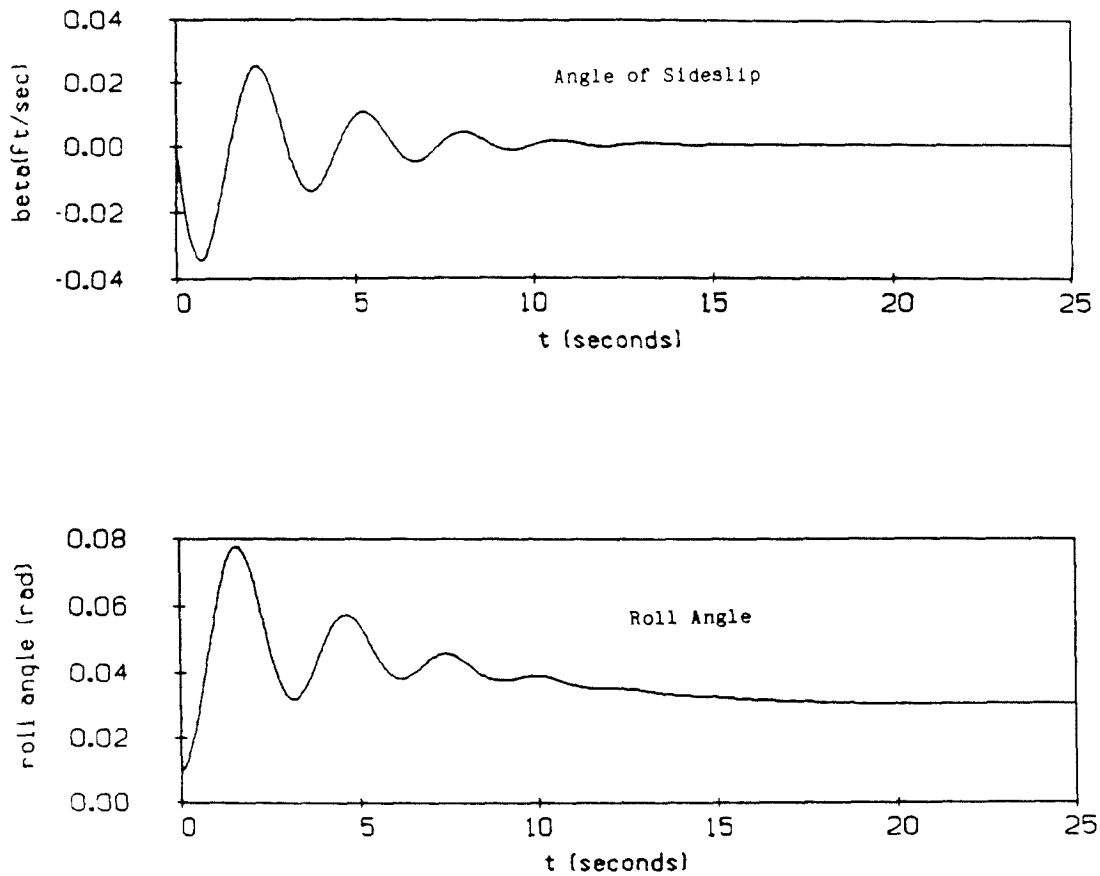



FIG.4.d DUTCH ROLL MODE

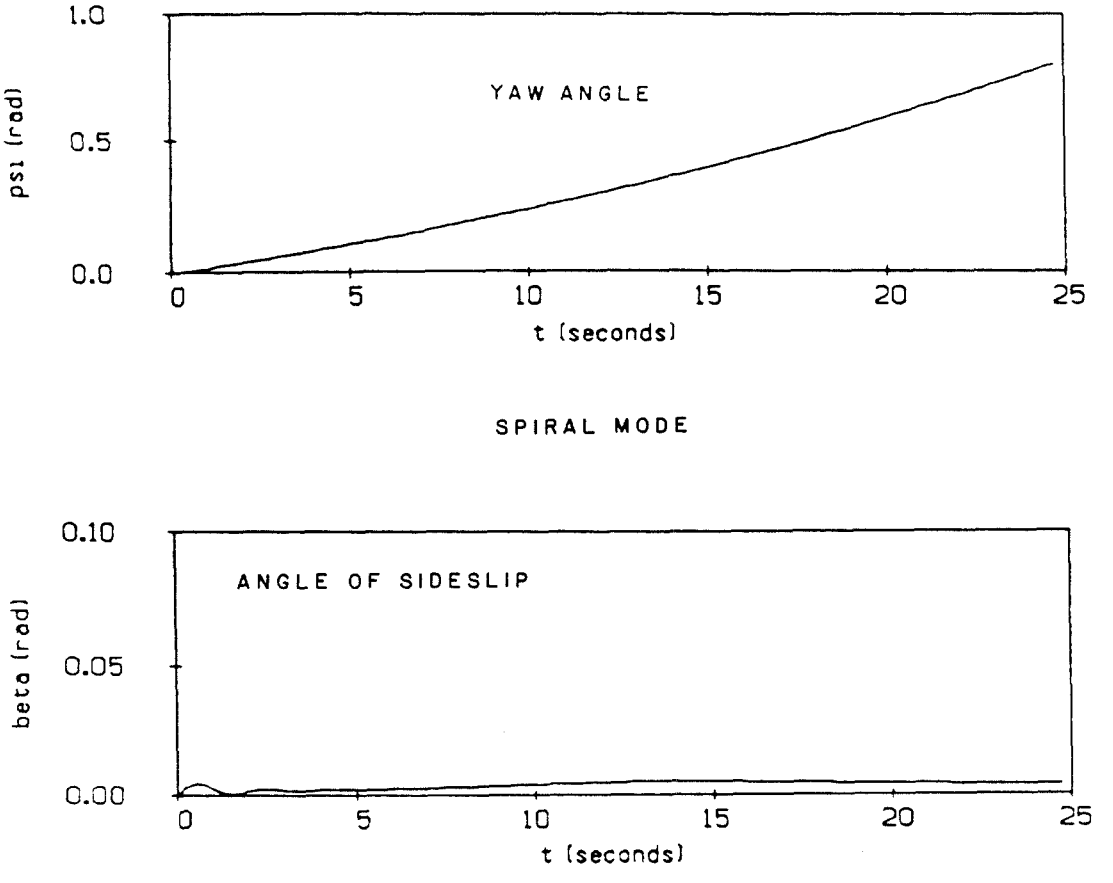



FIG.4.e SPIRAL MODE

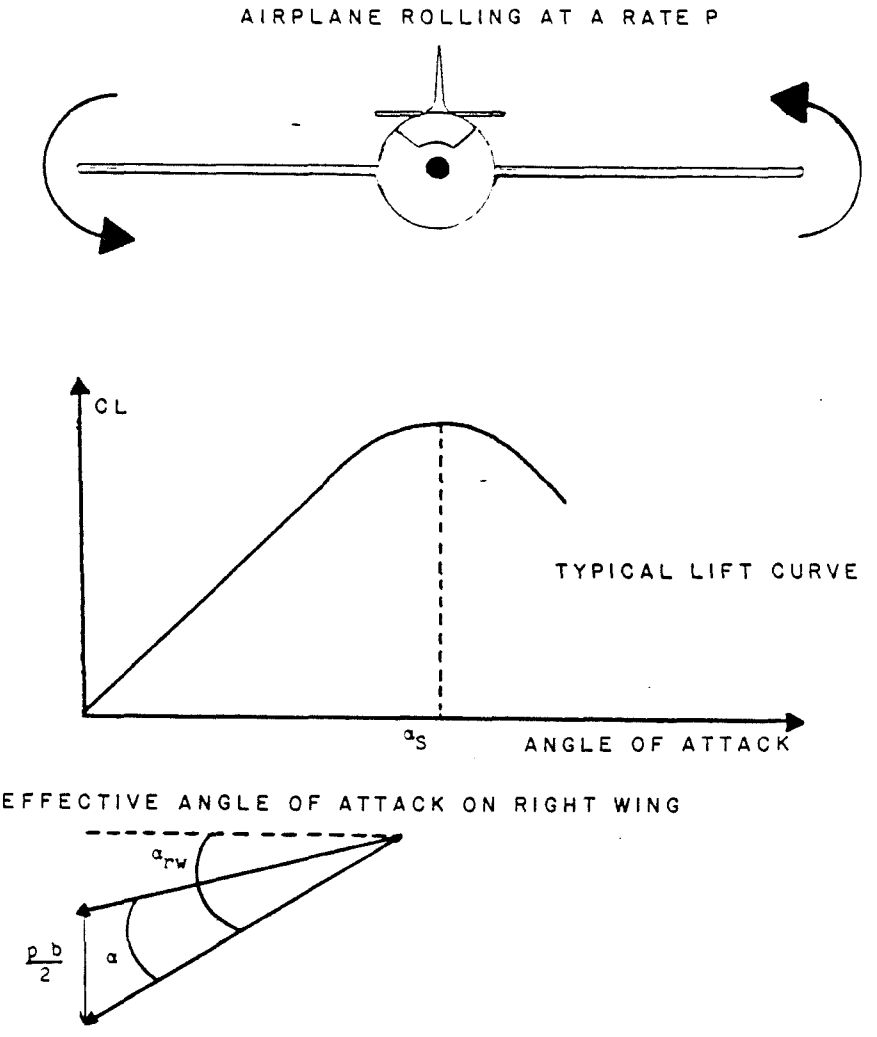



FIG.5.a AUTOROTATION

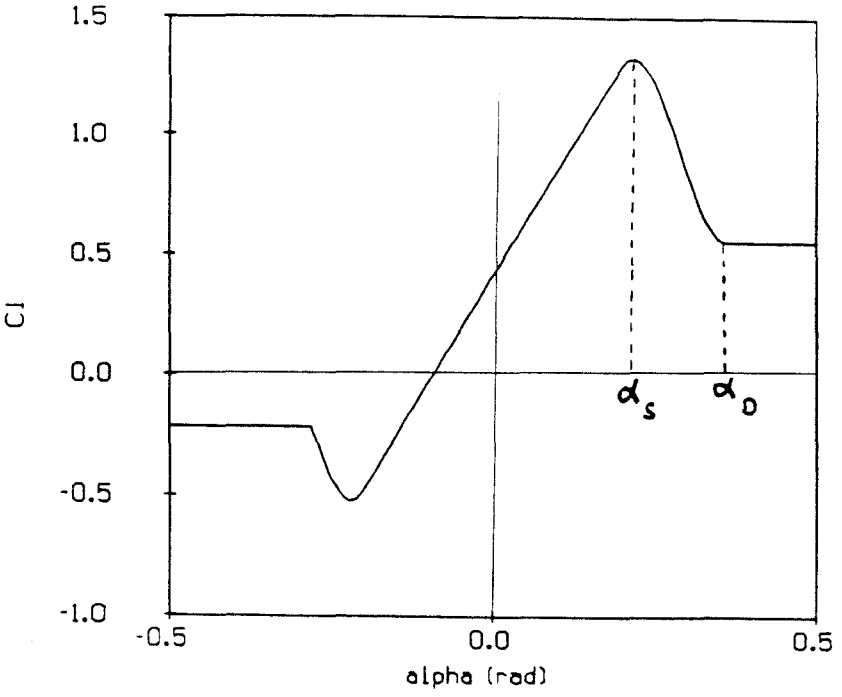



FIG.5.b NONLINEAR LIFT CURVE

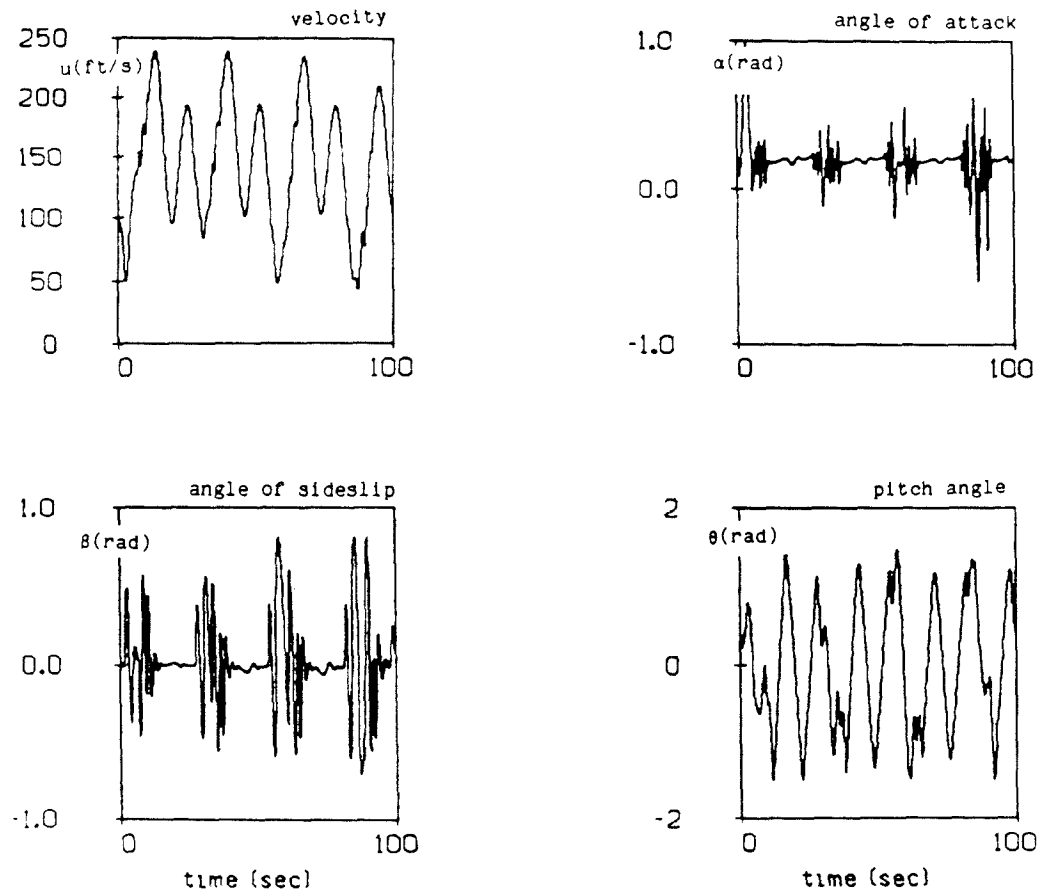



FIG.5.c ERRATIC MOTION

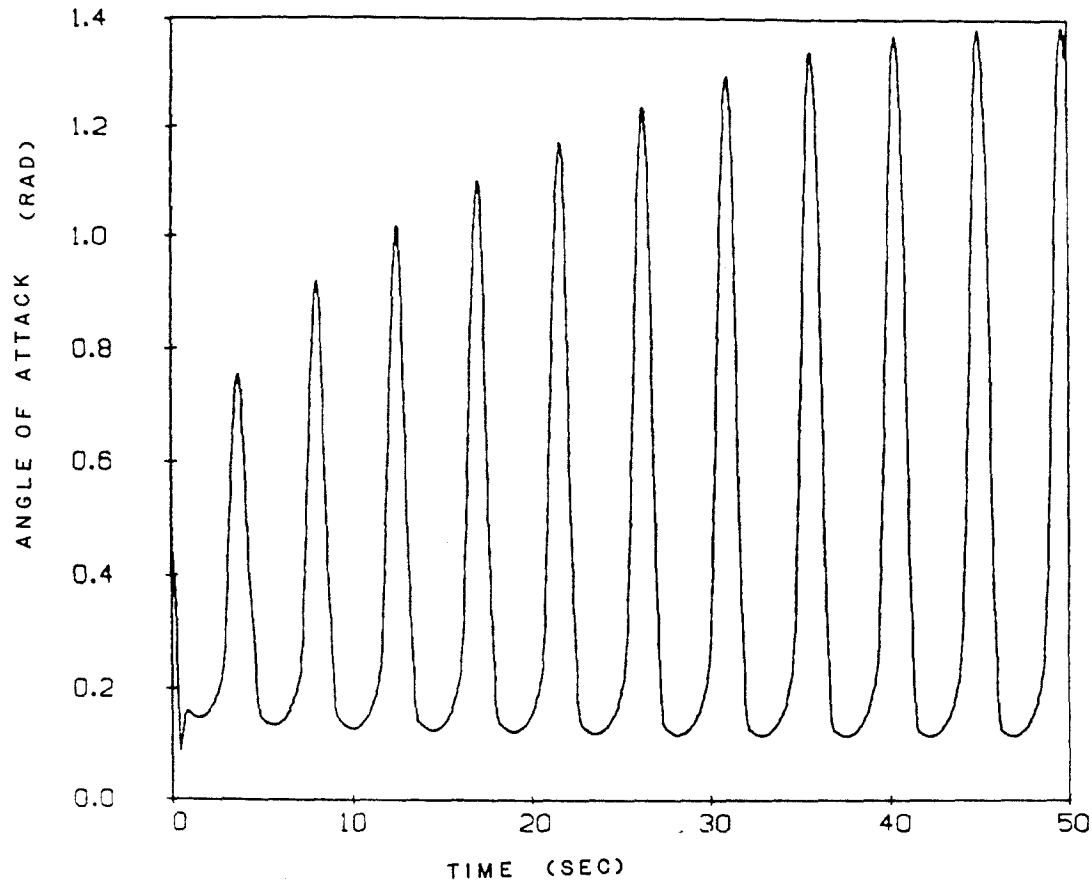



FIG.5.d NONLINEAR ANGLE OF ATTACK RESPONSE

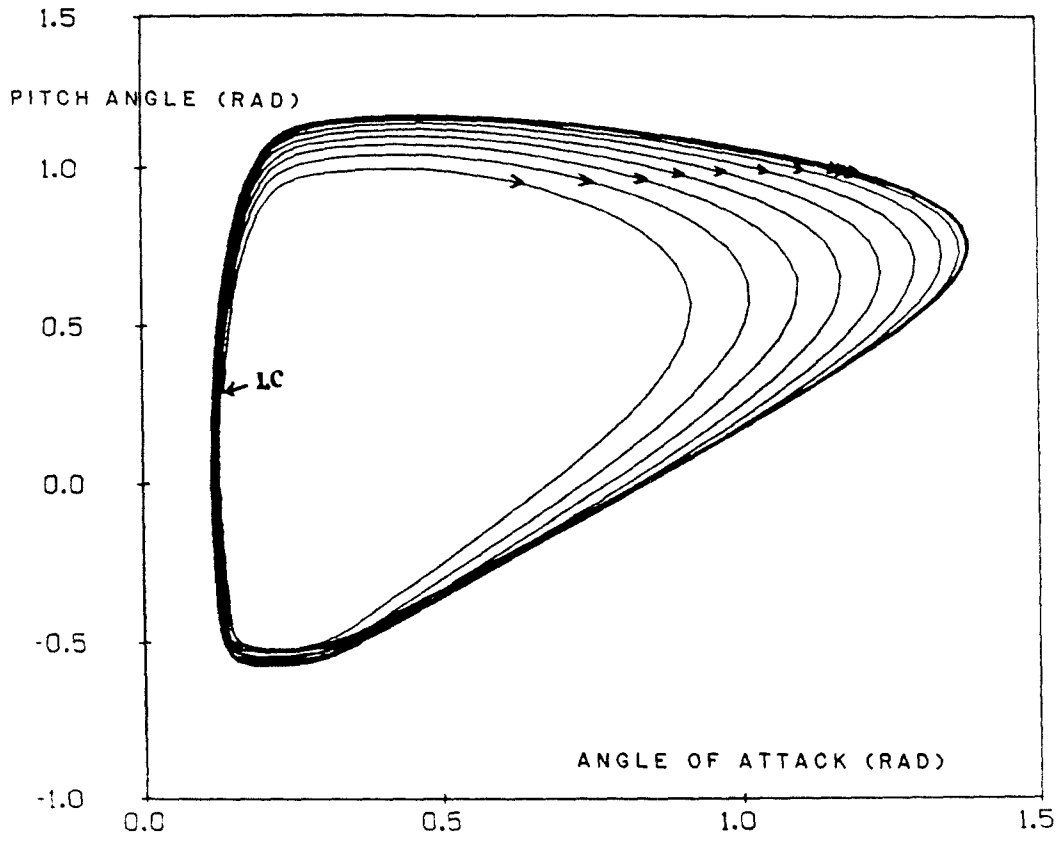



FIG.5.e PHASE-SPACE REPRESENTATION OF LIMIT CYCLE

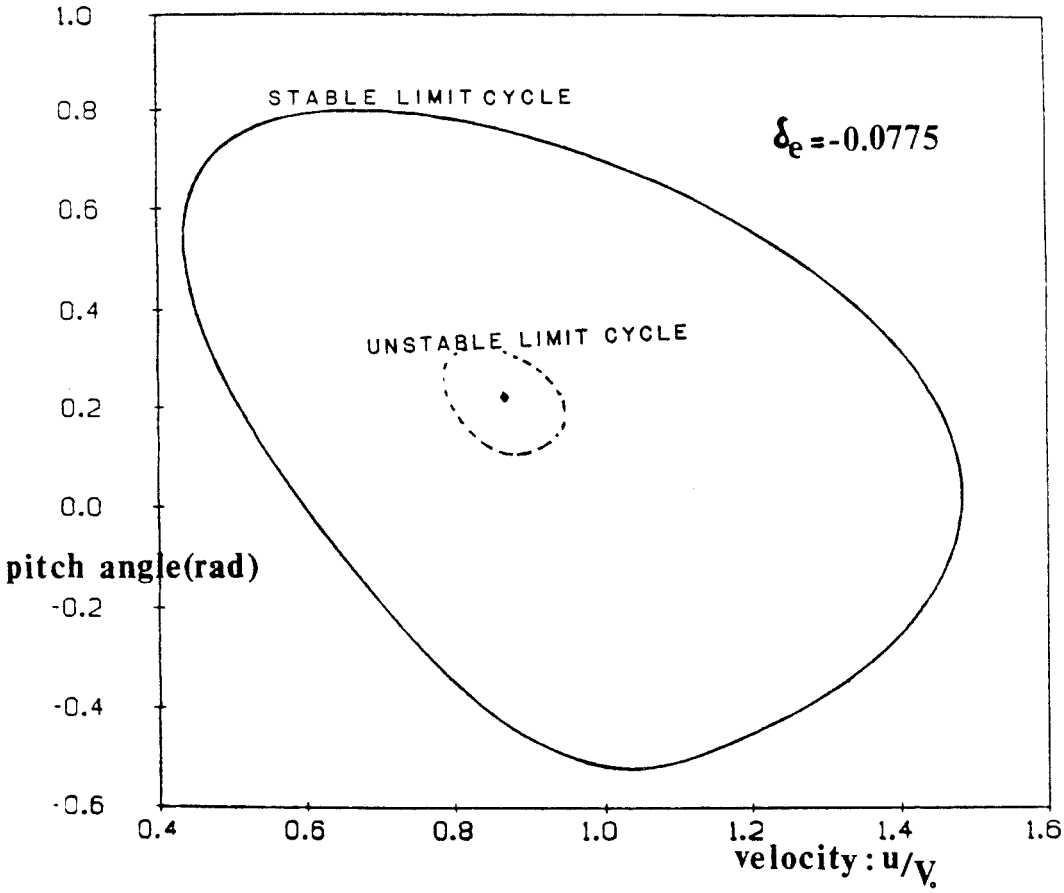
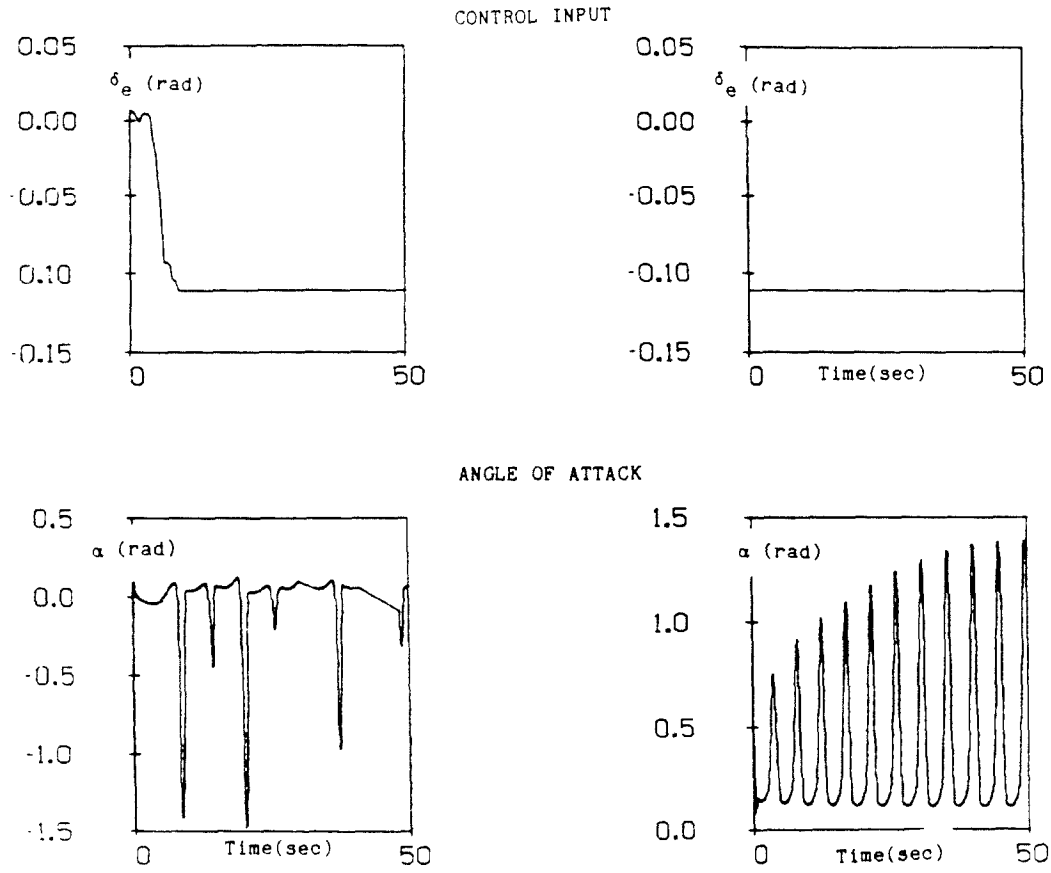



FIG.5.f TYPICAL PHASE-SPACE CONFIGURATION




TWO DIFFERENT HISTORY IN CONTROL SETTING BUT SAME INITIAL CONDITIONS

FIG.5.g SENSITIVITY TO CONTROL HISTORY

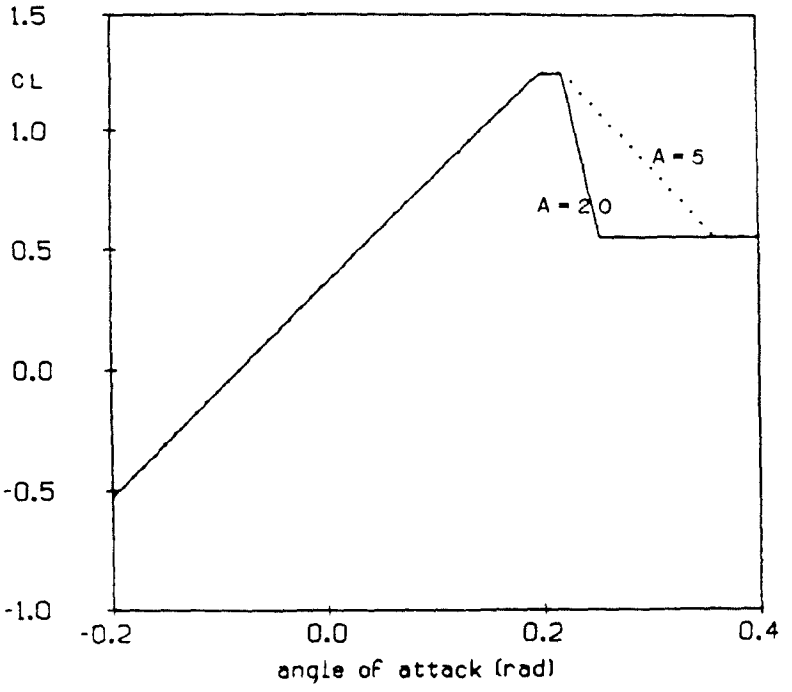



FIG.5.i VARIABLE NONLINEAR LIFT CURVE

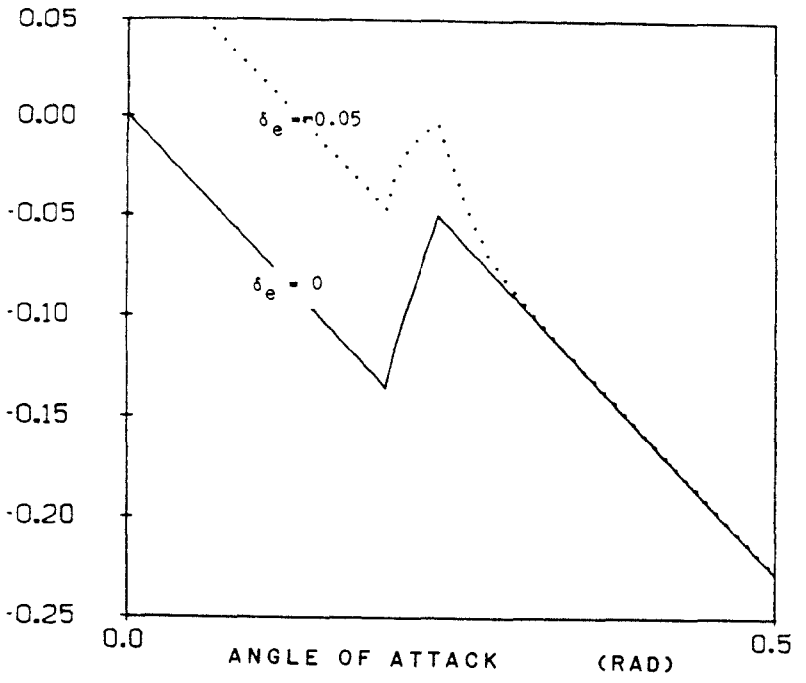
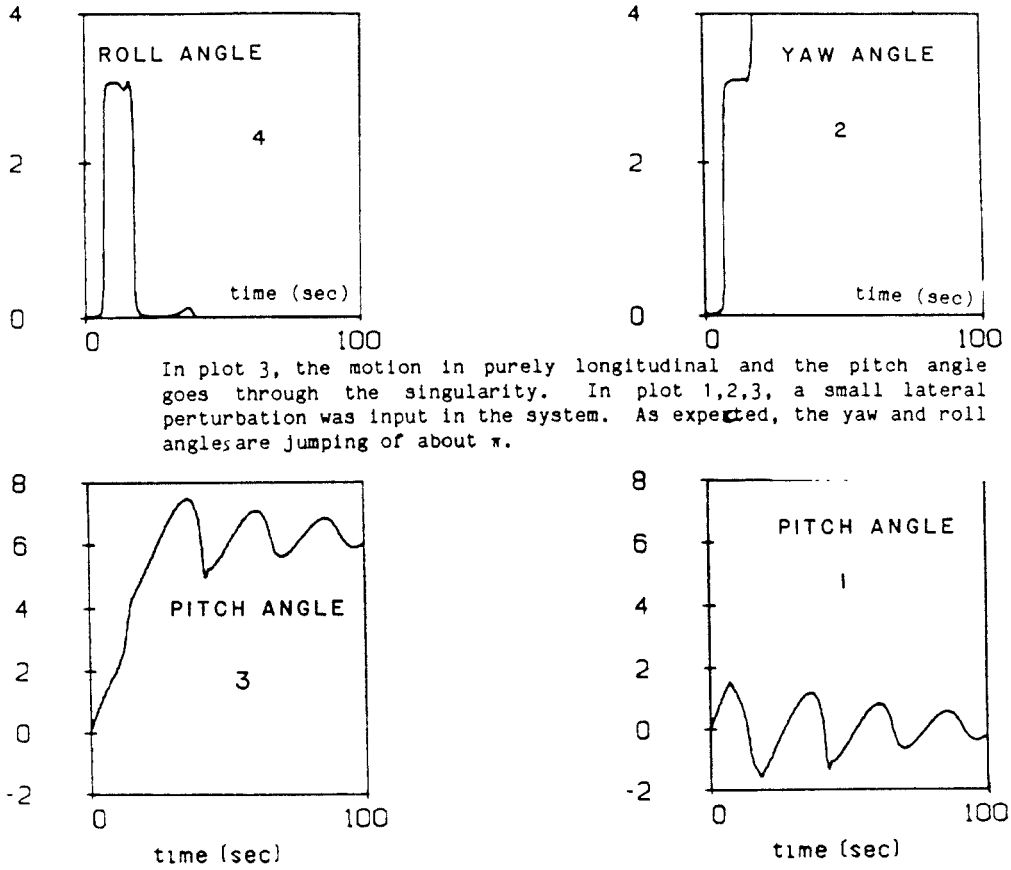



FIG.5.j NONLINEAR ROLLING MOMENT COEFFICIENT




In plot 3, the motion in purely longitudinal and the pitch angle goes through the singularity. In plot 1,2,3, a small lateral perturbation was input in the system. As expected, the yaw and roll angles are jumping of about $\pi$.

FIG.II.a STUDY OF THE SINGULARITY IN PITCH