Resource Optimization for Networked Estimator with Guaranteed Estimation Quality

Thesis by

Ling Shi

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2009

(Defended August 12, 2008)

Acknowledgements

Five years ago, I came to Pasadena for the first time, and everything was new and exciting to me. Now I am about to leave for a new stage of life. The past five years has been one of the best times in my life, and I truly enjoyed every single day I spent here, not just for the great facilities, but more importantly for the many great people I have met and worked with.

My most sincere thanks and gratitude go to my advisor, Prof. Richard Murray. I remember how he motivated and taught me to do research from the very beginning and encouraged me on my first paper at Caltech. He has always been a source of inspiration and a source of instant help. Thank you for all your support over the past few years, for which I am truly and always grateful.

Prof. Karl H. Johansson is a great mentor. I enjoyed very much the collaboration with him when he spent his sabbatical here two years ago. The series of work that I did with him during that time lay out the foundation of my thesis. Thank you for your guidance, support, and for your kind hosting of my visit in Nov. 2007.

The conversation with Prof. John Doyle in summer 2006 inspired me to look at multi-sensors instead of single-sensor networked control applications, and focus on the interaction between network topology and estimator and controller design.

Prof. Mani Chandy and Prof. Joel Burdick are kindly willing to serve as my thesis committee members beside Prof. Richard Murray, Prof. Karl H. Johansson, and Prof. John Doyle. I would like to thank you all for your precious time.

Prof. Bruno Sinopoli is both a great friend as well as a great collaborator. His pioneering work on Kalman filtering over lossy network greatly inspired me to explore this exciting research field. Thank you for all your support and your kind host for my visit at CMU. Prof. Lihua Xie is also a great friend as well as a great collaborator. I enjoyed working with him when he visited Caltech this year.

I enjoyed being a TA for Prof. Jerrold Marsden, and I learned many effective teaching skills which were very useful for the CDS 270-4 class I taught this spring term. I also enjoyed many conversations with Prof. Leonard Schulman, whose inspiration and insight often gave me new ideas and let me view problems from a different angle.

My former advisors, Prof. Zexiang Li and Li Qiu at the Hong Kong University of Science and Technology continue to encourage and support me even after I graduated in 2002. I would like to thank them for their time and help over these years. Prof. John Koo is a great friend, and I would like to thank him for his support and help over the years too.

It has been my great pleasure to spend the previous five years with the CDS family, especially with the Murray group, who make me feel I am at home. I very much enjoyed the weekly meetings with Michael Epstein, my most frequent collaborator in the first few years, whose encouragement and proficient computer skills often made our work very easy. Michael is also a great friend, and he is always ready to offer his help whenever needed; "Ma Shang, Ma Shang," he says. The collaboration with Ahbishek Tiwari at my second year opened the door to a series of work on networked control systems. I spent the first year with Dennice Gayme, Shaunak Sen, and Dmitriy Kogan taking the same set of courses. I got a lot of help from Xin Liu, Lun Li, Zhipu Jin, Jiantao Wang, Lijun Chen, and Dr. Wang Sang Koon during my first few years at CDS and Caltech. For all other CDS family members: I would like to thank you all and you have made my stay at Caltech so enjoyable.

I am grateful for the Chinese community here. In particular, I would like to thank all Braun residents: Kevin Ko, Xiaojie Gao, Changlin Pang, Chengzhong Zhang, Xin Guo, Zhaoyan Zhu, and Hao Jiang. I enjoyed my first two years here staying with them.

I would like to thank a few friends: Guanfeng Liu, Jijie Xu, Jian Cao, Yu Chen (for all your encouragement and support), Agostino Capponi (I enjoyed the collaboration with you), Nawaf Bou-Rabee (I enjoyed working with you for CDS 202), Stephen Backer (I enjoyed the apple pie that you prepared), Alex Backer (I enjoyed working with you as my intern), Jing Liu and Wei Liang (I enjoyed the delicious food prepared by your parents), Chunhui Gu (I enjoyed the many conversations with you), and Wen Yang (I enjoyed the many meals and conversations with you).

I would also like to thank some Caltech staff: Gloria Bain, Charmaine Boyd, Cierina Marks, Wendy McKay, and Peggy Conney at CDS, Jim Endrizzi at the ISP, Jonie Watanabe

Tsuji at the career center, and Alice M. Sogomonian at the health center, who made my stay at Caltech particularly easy.

I enjoyed the Bible study every Saturday at the Brother House. I would like to thank Peter Tsui, Andy Pearce, Zhongtian Lin, Jing Wu, and their families, Zhuo Li, Zhaoyu Zhang, and other brothers and sisters. Thank you all for making my life so wonderful and happy.

Last but not least, I would like to thank my parents, my sister and her family, and my wife and her family for their strong support.

I would like to dedicate this thesis to my parents for their unconditional and forever love.

Abstract

Advances in fabrication, modern sensor and communication technologies, and computer architecture have enabled a variety of new networked sensing and control applications. However, many difficulties are inherent with these systems, for example, the constrained communication and computation capabilities, and limited energy resources, which are frequently seen in a wireless sensor network. As a consequence, the networks typically induce many new issues such as limited bandwidth, packet loss, and delay. Estimation and control over such networks thus require new design paradigms beyond traditional sampled-data control, as the aforementioned constraints undoubtedly affect system performance or even stability. In this thesis work, I consider the problem of state estimation over networks. As communication, computation, and energy are scarce resources in such networks, I focus on optimizing the use of them. When the state estimation is carried out over a sensor network, I consider the problem of minimizing the sensor energy usage and maximizing the network lifetime. When the state estimation is carried out over a packet-delaying network, I consider the problem of minimizing the buffer length at the remote state estimator. In each scenario, a certain desired level of estimation quality is guaranteed.

Contents

Acknowledgements **iii**

Abstract **vi**

1 Introduction **1**

1.1 Background 1

1.2 Related Work 3

1.3 Summary of Contributions and Overview of Thesis 5

2 Preliminaries and Definitions **8**

2.1 Definitions 8

2.2 Kalman Filter and Modified Kalman Filter 9

2.3 Math Preliminaries 13

3 Kalman Filtering over Multihop Sensor Trees **16**

3.1 Introduction 16

3.2 Problem Set-up 17

3.3 Optimal Estimation Over a Sensor Tree 19

3.4 Example 21

3.5 Some Useful Inequalities 23

4 Minimizing Sensor Energy **26**

4.1 Introduction 26

4.2 Problem Set-up 27

4.3 Minimizing Sensor Energy 29

4.4 Examples 38

5 Maximizing Network Lifetime **43**

5.1 Introduction 43

5.2 Problem Set-up 44

5.3 Constructing and Scheduling Sensor Trees 47

5.4 Examples 50

6 Minimizing Buffer Length **56**

6.1 Introduction 56

6.2 Problem Setup 57

6.3 Sensor without Computation Capability 59

6.4 Sensor with Computation Capability 69

6.5 Examples 73

7 Conclusions and Future Work **78**

7.1 Concluding Remarks 78

7.2 Future Directions 79

Bibliography **81**

Chapter 1

Introduction

1.1 Background

Advances in fabrication, modern sensor and communication technologies, and computer architecture have enabled a variety of new networked sensing and control applications. In many of these applications, there is an economic incentive towards using off-the-shelf sensors and standardized communication solutions. A consequence of this is that the individual hardware components might be of relatively low quality and that communication resources are quite limited.

Networked sensing and control applications are found in a growing number of areas, including automobiles, autonomous vehicles, environmental monitoring, industrial automation, power distribution, space exploration, surveillance, and transportation. For example, Alice is an autonomous vehicle that was developed at California Institute of Technology for the 2005 DARPA Grand Challenge [11]. The sensors mounted on Alice include an inertial measurement unit (IMU), global positioning system (GPS), velocity and measurement range sensors, and stereo vision. To allow the vehicle to autonomously navigate through its environment, sensor data are fused to provide Alice an estimate of its own state and of the environment around it. The heterogeneous set of sensors is connected with the computation platform through an Ethernet local area network providing an architecture for networked estimation and control.

Wireless sensor networks constitute another important class of networked sensing and control applications, which have attracted much attention in the past few years. Sensor networks have a wide range of applications, including environment and habitat monitoring, health care, home and office automation, and traffic control [13]. This area of research

brings together researchers from computer science, communication, control, etc. [12]. A typical wireless sensor network consists of a large number of sensor nodes and some base stations [18]. Sensor nodes are usually battery powered and have limited computation capabilities. They interact with the physical world and collect information of interest, e.g., temperature, humidity, pressure, air density, etc. Depending on the routing protocol as well as the available resources (network bandwidth, node energy, etc.), the collected data are transmitted to their final destination, usually a fusion center, at appropriate times. For example, the Pursuer-Evader game (Figure 1.1) carried out at UC Berkeley [47] consists of hundreds of tiny sensor nodes which are capable of measuring the state of incoming evaders. The measured information is sent back to a computational unit via multi-hop communication paths, and corresponding control laws are computed and sent to the pursuers.

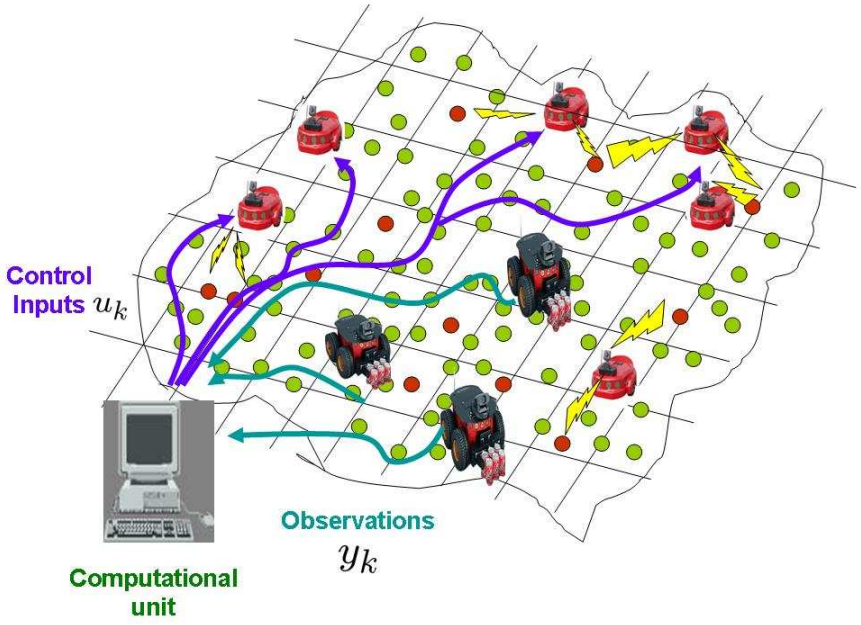

Figure 1.1: Pursuer Evader Game. Photo Courtesy: UC Berkeley

Although tremendous progress has been made in the past few years in making sensor network an enabling technology, many challenging problems remain to be solved, e.g., network topology control and routing, collaborative signal collection and information processing, synchronization, etc. Estimation and control over such resource-constrained networks thus require new design paradigms beyond traditional sampled-data control. For example, consider the problem of state estimation over such a network using a Kalman

filter. The Kalman filter [24] is a well-established methodology for model-based fusion of sensor data [2, 15, 16, 23] that has played a central role in systems theory and has found wide applications in many fields such as control, signal processing, and communications. In the standard Kalman filter, it is assumed that sensor data are transmitted along perfect communication channels and are available to the estimator instantaneously, and no interaction between communication and control is considered. This underlying assumption breaks when networks, especially wireless networks, are used in sensing and control systems for transmitting data from sensors to controller and/or from controller to actuator.

Many difficulties are inherent in these networked sensing and control systems, for example, constrained communication and computation capabilities, and limited energy resources, which are frequently seen in a wireless sensor network [13]. Communication between network nodes is limited, particularly, if nodes are located physically far way from each other. It takes time to transfer information from one node to another. As a consequence, the networks typically induce many new issues such as limited bandwidth, packet loss, and delay. These constraints affect system performance or even stability, and cannot be neglected when designing estimation and control algorithms; this has inspired a lot of research in control with communication constraints.

The rapid developments of networked sensing and control technologies enable drastic change of the architecture and embedded intelligence in these systems. The theory and design tools for these systems are not fully developed, but there is a lot of current research, some of which is described next.

1.2 Related Work

The problem of state estimation and stabilization of a linear time invariant (LTI) system over a digital communication channel that has a finite bandwidth capacity was introduced by Wong and Brockett [53,54] and further pursued by others (e.g., [7,33,38,49]). Sinopoli et al. [45] discussed how packet loss can affect state estimation. They showed there exists a certain threshold of the packet loss rate above which the state estimation error diverges in the expected sense, i.e., the expected value of the error covariance matrix becomes unbounded as time goes to infinity. They also provided lower and upper bounds of the threshold value. Following the spirit of [45], Liu and Goldsmith [29] extended the idea to the case

where there are multiple sensors and the packets arriving from different sensors are dropped independently. They provided similar bounds on the packet loss rate for a stable estimate, again in the expected sense. Packet losses are characterized as a Markov chain and some sufficient and necessary stability conditions under the notion of peak covariance stability are given in [20,55]. The drawback of using mean covariance matrix as a stability measure is that it may conceal the fact that events with arbitrarily low probability may make the mean value diverge. Different from [20, 45, 55], the stability of the Kalman filter was investigated via a probabilistic approach in [41].

One way to deal with the problem of asynchronous generation of sensor data is to use event-triggered control instead of conventional time-triggered control [3, 25]. How to efficiently encode control information for event-triggered control over communication channels with severe bandwidth limitations was discussed in [4]. A scheme based on multi-description coding for lossy networks, but limited to the estimation, was considered by Jin et al. [21]. A compensation scheme in the controller for the variations on the transport layer that such routing protocols give rise to was presented by Witrant et al. [52]. A robust control approach to control over multi-hop networks was discussed in [37].

Kalman filtering under certain information constraints, such as decentralized implementation, has been extensively studied [44]. Implementations for which the computations are distributed among network nodes were considered by Alriksson and Rantzer [1]. The interaction between Kalman filtering and how data is routed on a network seems to be less studied. Routing of data packets in networks is typically done based on the distance to the receiver node [5]. Some recent work addresses how to couple data routing with the sensing task using information theoretic measures [22]. A heuristic algorithm for event detection and actuator coordination was proposed by Ngai et al. [35]. For control over wireless sensor networks, the experienced delays and packet losses are important parameters. Randomized routing protocols that give probabilistic guarantees on delay and loss were proposed in [6, 27].

The problem of Kalman filtering for systems with delayed measurements is not new and has been studied even before the emergence of networked control [39, 57]. It is well known that discrete-time systems with constant or known time-varying bounded measurement delays may be handled by state augmentation in conjunction with the standard Kalman filtering or by the reorganized innovation approach [59–61]. Although sensor data are usually

time-stamped and thus transmission delays are known to the filter, the delays in networked systems are random in nature. Thus, the state augmentation and the reorganized innovation approaches are generally not applicable.

For the problem of randomly delayed measurements, Ray et al. [39] presented a modification of the conventional minimum variance state estimator to accommodate the effects of the random arrival of measurements, whereas a suboptimal filter in the least-mean-square sense is given in [57]. In [30], a recursive minimum variance state estimator was presented for linear discrete-time partially observed systems where the observations are transmitted by communication channels with randomly independent delays. Using covariance information, recursive least-squares linear estimators for signals with random delays were studied in [34]. Furthermore, the filtering problems with random delays and missing measurements have been investigated in [40, 48, 51] via the linear matrix inequality and the Riccati equation approaches, respectively.

More related work can be found in a recent survey of networked control systems in [17].

1.3 Summary of Contributions and Overview of Thesis

The main contribution of this thesis work is to tackle the aforementioned networked control problems by optimizing the limited resources of those networks while guaranteeing a certain level of desired estimation quality. In particular, I consider minimizing the sensor energy usage, maximizing the network lifetime, and minimizing the buffer length, with each corresponding to a class of networked sensing and control applications. The scenarios and algorithms investigated are:

- *(Chapter 3)* Optimal estimation algorithm over a sensor tree is presented with a closed-form expression on the steady-state error covariance.
- *(Chapter 4)* A sensor tree reconfiguration algorithm is presented to minimize the sensor energy usage.
- *(Chapter 5)* A sensor tree construction and scheduling algorithm is presented to maximize the sensor network lifetime.
- *(Chapter 6)* Kalman filtering over a packet-delaying network is considered, and the minimum buffer length at the remote estimator is determined.

For all the scenarios considered in the thesis, a certain level of estimation quality is guaranteed.

After a short introduction to Kalman filtering and modified Kalman filtering in Chapter 2, we first present the optimal estimation algorithm when the sensor communications are represented by a tree in Chapter 3.

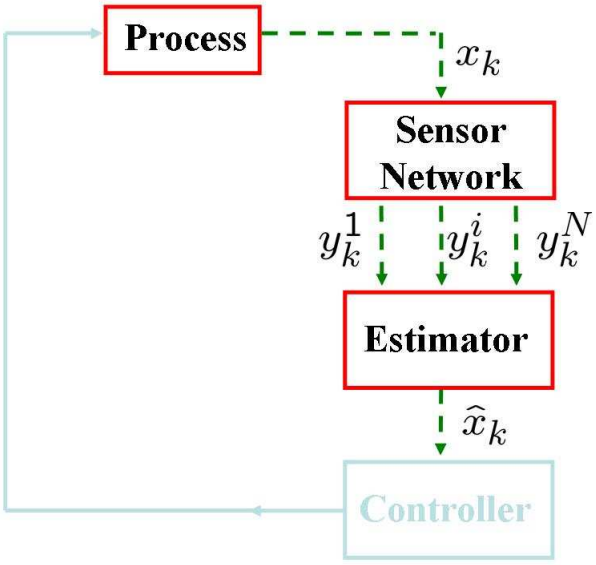

Figure 1.2: State Estimation over a Wireless Sensor Network

Then we consider the sensor energy minimization problem in Chapter 4, where sensor measurement data are sent to a fusion center over a multi-hop wireless network (Figure 1.2). The quality of the state estimate depends not only on the sensor quality but also on the communication delay, i.e., the number of hops a sensor measurement needs to take until it reaches the fusion center. Many short hops take longer time than few long hops. On the other hand, few long hops require larger transmission power since the required transmission power grows rapidly with the distance between the wireless nodes. Hence, there is a trade-off between the state estimation quality and the overall energy cost. The proposed solution is to optimize the network path for the sensor data such that the overall transmission energy is minimized, but guarantees a specified level of estimation quality. The resulting local sensor topology has the structure of a tree for which the fusion center is the root. When the process is given by a linear system, the optimal estimator is given by a chain of Kalman filters due to the communication delays.

We further consider the network lifetime maximization problem in Chapter 5, where we first motivate that by minimizing the overall energy cost is not sufficient to maximize the network lifetime. Therefore we propose a sensor tree construction and scheduling algorithm that maximizes the network lifetime.

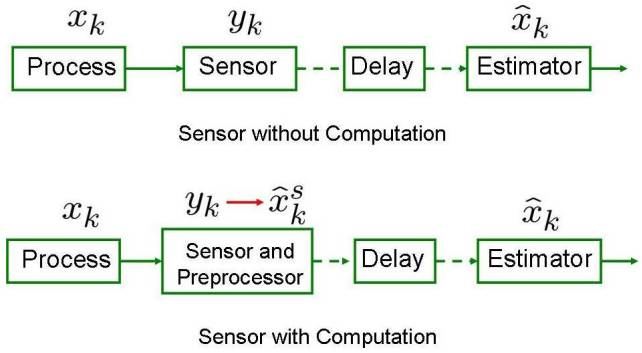

Figure 1.3: State Estimation over a Packet Delaying Network

In Chapter 6, we study the performance of a Kalman filter under random measurement delay (Figure 1.3). The probability distribution of the delay is assumed to be given and we give a complete characterization of filter performance via a probabilistic approach. Due to the limited computation capability of the filtering center and also in consideration of the fact that a late-arriving measurement related to the system state in the far past may not contribute much to the improvement of the accuracy of the current estimate, it is practically important to determine a proper buffer length for measurement data within which a measurement will be used to update the current state and beyond which the data will be discarded. The buffer provides a tradeoff between the filter performance and computational load. In this thesis, for a given buffer length, we give lower and upper bounds for the probability at which the filtering error covariance is within a prescribed bound, i.e., $\mathbf{Pr}[P_k \leq M]$ for some given M. The upper and lower bounds can be easily evaluated by the probability distribution of the delay and the system dynamics. An approach for determining the minimum buffer length for a required performance in probability is given and an evaluation on the number of expected filter updates is provided. Both the cases of sensor with and without computation capability for filter updates are considered.

Finally, in Chapter 7, concluding remarks are given, and we discuss some directions of future work along the line of this thesis.

Chapter 2

Preliminaries and Definitions

In this chapter, some definitions are provided which are used frequently in remaining chapters. A brief introduction to the Kalman filter and the modified Kalman filter is also included, upon which the main results of later chapters rely.

2.1 Definitions

The following definitions are frequently used throughout later chapters. $\mathbb{Z}^+$ denotes the set of nonnegative integers. $\mathbb{R}^n$ is the real n-dimensional vector space. $\mathbb{R}^{n\times n}$ is the set of n by n real matrices. $\mathbb{S}_+^n$ is the set of n by n positive semidefinite matrices. When $X \in \mathbb{S}_+^n$, we simply write $X \geq 0$; when X is positive definite, we write $X > 0$.

We are frequently dealing with systems with parameters (A, C, Q, R), where $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times m}$ are the system and sensor measurement matrices, $Q \in \mathbb{S}_+^n$ and $R \in \mathbb{S}_+^m$ with $R > 0$ are the process and measurement noise covariance matrices respectively, e.g., in Eqn (2.7) and (2.8). We define the function $h_{[A,Q]} : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$h_{[A,Q]}(X) \triangleq AXA' + Q. \tag{2.1}$$

As we shall see shortly, applying h to the previous error covariance matrix corresponds to the time update of the standard Kalman filter. Similarly, we define the function $g_{[A,C,Q,R]} : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$g_{[A,C,Q,R]}(X) \triangleq AXA' + Q - AXC'[CXC' + R]^{-1}CXA', \tag{2.2}$$

and the function $\tilde{g}_{[C,R]} : \mathbb{S}_+^n \to \mathbb{S}_+^n$ as

$$\tilde{g}_{[C,R]}(X) \triangleq X - XC'[CXC' + R]^{-1}CX. \tag{2.3}$$

Then g and $\tilde{g}$ correspond to the measurement update for the a priori and a posteriori error covariance matrices respectively in the standard Kalman filter. We simply write $h_{[A,Q]}$ as h, $g_{[A,C,Q,R]}$ as g or g_C, and $\tilde{g}_{[A,C,Q,R]}$ as $\tilde{g}$ or $\tilde{g}_C$ when there is no confusion on the underlying parameters $[A, C, Q, R]$. It is easy to see that

$$g = h \circ \tilde{g}. \tag{2.4}$$

We denote $\lambda_i(A)$ as the i-th eigenvalue of A and $\rho(A)$ as spectral radius of A, i.e., $\rho(A) = \max_i |\lambda_i(A)|$. We say A is stable if $\rho(A) < 1$, and A is unstable if A is not stable. For functions $f, f_1, f_2 : \mathbb{S}_+^n \to \mathbb{S}_+^n$, $f_1 \circ f_2$ is defined as

$$f_1 \circ f_2(X) \triangleq f_1\big(f_2(X)\big), \tag{2.5}$$

and f^t is defined as

$$f^t(X) \triangleq \underbrace{f \circ f \circ \cdots \circ f}_{t \text{ times}}(X). \tag{2.6}$$

For a random variable X, we write its expectation value as $\mathbb{E}[X]$ and its conditional probability given another random variable Y as $\mathbf{Pr}[X|Y]$.

2.2 Kalman Filter and Modified Kalman Filter

Kalman Filter Preliminaries

Consider the following linear discrete time system:

$$x_k = Ax_{k-1} + w_{k-1}, \tag{2.7}$$

$$y_k = C_k x_k + v_k. \tag{2.8}$$

In the above equations, $x_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^m$ is the observation vector, $w_{k-1} \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are zero-mean white Gaussian random vectors with $\mathbb{E}[w_k {w_j}'] =$

$\delta_{kj}Q \geq 0$, $\mathbb{E}[v_k v_j{}'] = \delta_{kj}R > 0$, $\mathbb{E}[w_k v_j{}'] = 0\ \forall j,k$, where $\delta_{kj} = 0$ if $k \neq j$ and $\delta_{kj} = 1$ otherwise. The Kalman filter in its most general form can assume time-varying A and Q. The special form we look at here suffices for deriving the optimal estimation algorithms in later chapters.

Assume a linear estimator receives y_k and computes the optimal state estimate at each time k. Define the following terms at the estimator:

$$\begin{aligned}
\hat{x}_k^- &\triangleq \mathbb{E}[x_k|\text{all measurements up to } k-1], \\
\hat{x}_k &\triangleq \mathbb{E}[x_k|\text{all measurements up to } k], \\
P_k^- &\triangleq \mathbb{E}[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)'], \\
P_k &\triangleq \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)'], \\
P^* &\triangleq \lim_{k\to\infty} P_k^-, \text{if the limit exists}, \\
\overline{P} &\triangleq \lim_{k\to\infty} P_k, \text{if the limit exists}.
\end{aligned}$$

It is well known that $\hat{x}_k$ and P_k can be computed as

$$(\hat{x}_k, P_k) = \mathbf{KF}(\hat{x}_{k-1}, P_{k-1}, y_k, C_k, R_k),$$

where **KF** denotes the Kalman filter given by the following update equations:

$$\begin{aligned}
\hat{x}_k^- &= A\hat{x}_{k-1}, && (2.9)\\
P_k^- &= AP_{k-1}A' + Q, && (2.10)\\
K_k &= P_k^- C_k'[C_k P_k^- C_k' + R_k]^{-1}, && (2.11)\\
\hat{x}_k &= A\hat{x}_{k-1} + K_k(y_k - C_k A\hat{x}_{k-1}), && (2.12)\\
P_k &= (I - K_k C_k)P_k^-. && (2.13)
\end{aligned}$$

It can be shown that P_k^- and P_k evolve as

$$P_k^- = g_{[C_{k-1},R_{k-1}]}(P_{k-1}^-), \quad (2.14)$$

$$P_k = \tilde{g}_{[C_k,R_k]}(P_k^-). \quad (2.15)$$

When parameters C_k and R_k are not time-varying, i.e., $C_k = C$ and $R_k = R$, we have the following lemma regarding the properties of the steady state error covariances.

Lemma 2.1 *When $C_k = C, R_k = R$, the pair $(A, \sqrt{Q})$ is stabilizable and (A, C) is observable, P^* and $\overline{P}$ exist and satisfy the following equations:*

$$P^* = g(P^*), \tag{2.16}$$

$$\overline{P} = \tilde{g}(P^*), \tag{2.17}$$

$$\overline{P} = \tilde{g} \circ h(\overline{P}). \tag{2.18}$$

Proof: By standard Kalman filtering analysis, if $(A, \sqrt{Q})$ is stabilizable and (A, C) is observable, then Eqn (2.14) converges to a unique value for any initial condition $P_0 \geq 0$. Therefore Eqn (2.16) and (2.17) simply follow from Eqn (2.14) and (2.15) by letting $k \to \infty$. Eqn (2.18) holds as

$$\tilde{g} \circ h(\overline{P}) = \tilde{g} \circ h\big(\tilde{g}(P^*)\big) = \tilde{g} \circ g(P^*) = \tilde{g}(P^*) = \overline{P}.$$

■

Modified Kalman Filter

In many networked control applications, the measurement packet y_k is sent via an unreliable communication network, e.g., y_k can be dropped by the network possibly due to network traffic, channel fading, etc. In this case, the optimal linear estimator is known to be given by a modified Kalman filter (**MKF**) [45].

Let γ_k be the indicator functor for y_k, which is defined as follows.

$$\gamma_k = \begin{cases} 1, \text{if } y_k \text{ is received at time } k, \\ 0, \text{otherwise.} \end{cases}$$

We write $(\hat{x}_k, P_k)$ in compact form as

$$(\hat{x}_k, P_k) = \mathbf{MKF}(\hat{x}_{k-1}, P_{k-1}, \gamma_k y_k),$$

which represents the following set of equations:

$$\begin{cases} \hat{x}_k^- &= A\hat{x}_{k-1}, \\ P_k^- &= AP_{k-1}A' + Q, \\ K_k &= P_k^- C'[CP_k^- C' + R]^{-1}, \\ \hat{x}_k &= A\hat{x}_{k-1} + \gamma_k K_k(y_k - CA\hat{x}_{k-1}), \\ P_k &= (I - \gamma_k K_k C)P_k^-. \end{cases}$$

Notice that if $\gamma_k = 1$ for all k, then **MKF** reduces to the standard Kalman filter, i.e., Eqn (2.9)–(2.13). When $\gamma_k = 0$, it is easy to show that

$$P_k = P_k^- = h(P_{k-1}).$$

Properties of h and g Functions

Many useful properties of the h and g functions defined earlier in this chapter are presented below.

Lemma 2.2 *For any $X, Y \in \mathbb{S}_+^n$, and $X \leq Y$,*

1. $h(X) \leq h(Y)$.
2. $g(X) \leq g(Y)$.
3. $\tilde{g}(X) \leq \tilde{g}(Y)$.
4. $\tilde{g}(X) \leq X$.
5. $g(X) \leq h(X)$.

Proof: $h(X) \leq h(Y)$ holds as $h(X)$ is affine in X. The proof for $g(X) \leq g(Y)$ can be found in Lemma 1-c in [45]. As $\tilde{g}$ is a special form of g by setting $A = I$ and $Q = 0$, we immediately obtain $\tilde{g}(X) \leq \tilde{g}(Y)$. Next we have

$$\tilde{g}(X) = X - XC'[CXC' + R]^{-1}CX \leq X,$$

and

$$g(X) = h(X) - AXC'[CXC' + R]^{-1}CXA' \leq h(X).$$

■

When the measurement matrix C is invertible, the function g exhibits a very nice property. When we apply g to any $X \geq 0$, we have a bounded value. The following lemma gives this bound.

Lemma 2.3 *Assume C^{-1} exists and let $\overline{M} = C^{-1}RC^{-1'}$. Then for any $X \geq 0, \tilde{g}(X) \leq \overline{M}$.*

Proof: For any $t > 0$, we have

$$\begin{aligned} \tilde{g}(t\overline{M}) &= \frac{t}{t+1}\overline{M} \\ &\leq \overline{M}. \end{aligned}$$

For all $X \geq 0$, since $\overline{M} > 0$, it is clear that there exists $t_1 > 0$ such that $t_1\overline{M} > X$. Therefore

$$\tilde{g}(X) \leq \tilde{g}(t_1\overline{M}) \leq \overline{M}.$$

■

Recall that $\overline{P}$ is the steady state error covariance. Suppose the Kalman filter enters the steady state, so that $P_k = \overline{P}$. When a sensor measurement packet is lost at time t, only time update is performed, i.e., $P_t = h(\overline{P})$. Intuitively, we shall get a larger estimation error. The following lemma verifies this intuition.

Lemma 2.4 $\overline{P} \leq h(\overline{P})$.

Proof:

$$h(\overline{P}) = h \circ \tilde{g}(P^*) = g(P^*) = P^* \geq \tilde{g}(P^*) = \overline{P},$$

where the first and the last equality are from Eqn (2.17), and the third equality is from Eqn (2.16). The inequality is due to Lemma 2.2. ■

2.3 Math Preliminaries

The following three lemmas are well known, and they are stated without proofs. Those lemmas will be used to establish the properties of many algorithms introduced in the remaining

chapters. The first lemma is the Matrix Inversion Lemma, which provides an alternative formula for computing the inverse of a matrix.

Lemma 2.5 (Matrix Inversion Lemma) *Let $X > 0$. If $X = B^{-1} + CD^{-1}C'$, then the inverse of X can be written as*

$$X^{-1} = B - BC(D + C'BC)^{-1}C'B.$$

The second lemma is the Schur Complement lemma. It provides a set of equivalent relationships for a positive definite matrix M.

Lemma 2.6 (Schur Complement) *Let*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Then the following three conditions are equivalent to each other.

1. $M > 0$.
2. $A > 0$ *and* $S_A \triangleq D - CA^{-1}B > 0$.
3. $D > 0$ *and* $S_D \triangleq A - BD^{-1}C > 0$.

The last one is the Block Matrix Inversion lemma, which, as its name suggests, inverts a block matrix using the Schur complement of the matrix.

Lemma 2.7 (Block Matrix Inversion) *Let*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} > 0.$$

Then M^{-1} can be computed as

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix},$$

or it can be computed as

$$M^{-1} = \begin{bmatrix} S_D^{-1} & -S_D^{-1}BD^{-1} \\ -D^{-1}CS_D^{-1} & D^{-1} + D^{-1}CS_D^{-1}BD^{-1} \end{bmatrix}.$$

Chapter 3

Kalman Filtering over Multihop Sensor Trees

3.1 Introduction

In this chapter, we consider the problem of state estimation over a sensor network. When the sensor communications are represented by a tree, the optimal estimator is shown to be given by a chain of Kalman filters due to the communication delays.

The problem of Kalman filtering for systems with delayed measurements is not new and has been studied even before the emergence of networked control [39, 57]. It is well known that discrete-time systems with constant or known time-varying bounded measurement delays may be handled by state augmentation in conjunction with the standard Kalman filtering or by the reorganized innovation approach [59–61].

The optimal estimation scheme that we propose in this chapter is computationally efficiently, and it gives insight into how each sensor (with different delays) contributes to the overall estimation accuracy. Hence it enables us to construct minimum energy sensor tree in Chapter 4. As we show in Chapter 6, this scheme also enables us to borrow tools developed for Kalman filtering over packet dropping networks to analyze Kalman filtering over packet delaying networks.

The rest of this chapter is organized as follows. We first introduce the mathematical problem in Section 3.2. Then we present the optimal estimation algorithm with a closed-form expression on the steady-state error covariance in Section 3.3. An example is provided in Section 3.4 to demonstrate the optimal estimation algorithm. Some useful inequalities are presented in the last section.

3.2 Problem Set-up

Consider the problem of state estimation over a wireless sensor network (Figure 3.1). The process dynamics is described by

$$x_k = Ax_{k-1} + w_{k-1}. \tag{3.1}$$

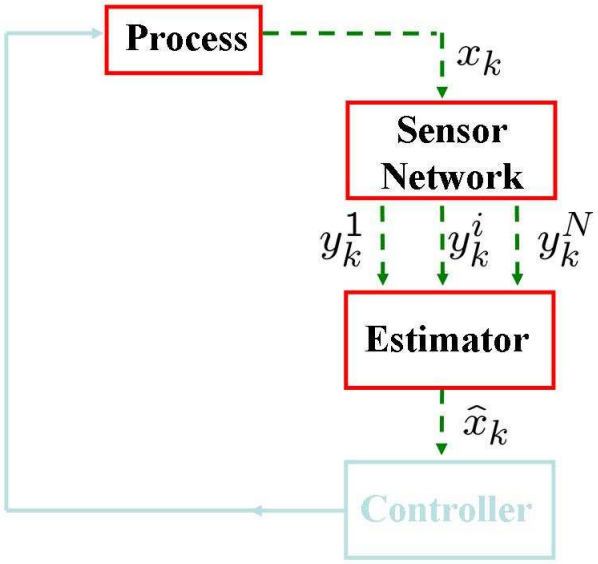

Figure 3.1: State Estimation Using a Wireless Sensor Network

A wireless sensor network consisting of N sensors, i.e., $\{S_1, \cdots, S_N\}$, is used to measure the state. When S_i takes a measurement of the state in Eqn (3.1), it returns

$$y_k^i = H_i x_k + v_k^i, i = 1, \cdots, N. \tag{3.2}$$

In Eqn (3.1) and (3.2), $x_k \in \mathbb{R}^n$ is the state vector, $y_k^i \in \mathbb{R}^{m_i}$ is the observation vector for S_i, $w_{k-1} \in \mathbb{R}^n$ and $v_k^i \in \mathbb{R}^{m_i}$ are zero-mean white Gaussian random vectors with $\mathbb{E}[w_k w_j{}'] = \delta_{kj} Q \geq 0$, $\mathbb{E}[v_k^i v_t^{i\prime}] = \delta_{kt} \Pi_i > 0$, $\mathbb{E}[v_k^i v_t^{j\prime}] = 0 \; \forall t, k$ and $i \neq j$, $\mathbb{E}[w_k v_t^{i\prime}] = 0 \; \forall i, t, k$. We assume that $(A, \sqrt{Q})$ is controllable, and (A, C_{all}) is observable, where $C_{\text{all}} = [H_1; \cdots; H_N]$, i.e., the joint measurement matrix of all sensors.

Each sensor can potentially communicate via a single-hop connection with a subset of all the sensors by adjusting its transmission power. Let us introduce a fusion center which

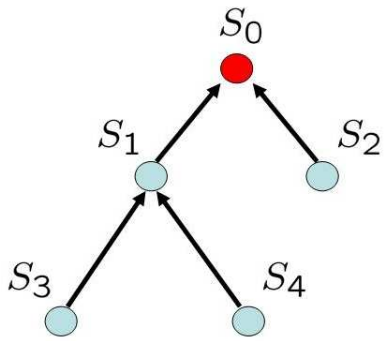

Figure 3.2: An Example of a Sensor Tree

we denote as S_0, and consider a tree T with root S_0 (see Figure 3.2). We suppose that there is a non-zero single-hop communication delay, which is smaller than the sampling time of the process. All sensors are synchronized in time, so the data packet transmitted from S_i to S_0 is delayed one sample when compared with the parent node of S_i. We also assume that S_i aggregates the previous time data packets from all its child nodes with its current time measurement into a single data packet. Therefore only one data packet is sent from S_i to its parent node at each time k.

Let us define the following state estimate and other quantities at S_0 for a given T:

$$\begin{aligned}
\hat{x}_k^-(T) &\triangleq \mathbb{E}[x_k|\text{all measurements up to } k-1], \\
\hat{x}_k(T) &\triangleq \mathbb{E}[x_k|\text{all measurements up to } k], \\
P_k^-(T) &\triangleq \mathbb{E}[(x_k - \hat{x}_k^-(T))(x_k - \hat{x}_k^-(T))'], \\
P_k(T) &\triangleq \mathbb{E}[(x_k - \hat{x}_k(T))(x_k - \hat{x}_k(T))'], \\
P_\infty^-(T) &\triangleq \lim_{k\to\infty} P_k^-(T), \text{if the limit exists}, \\
P_\infty(T) &\triangleq \lim_{k\to\infty} P_k(T), \text{if the limit exists}.
\end{aligned}$$

We drop the dependence on T, i.e., we write $\hat{x}_k^-(T)$ as $\hat{x}_k^-$, etc., if there is no confusion on the underlying T. In this chapter, we are interested in computing $\hat{x}_k$ and P_k for a given T.

3.3 Optimal Estimation Over a Sensor Tree

Assume T has depth D. Define $\mathcal{Y}_k^{k-i+1}$ as the set of all measurements available at the fusion center for time $k-i+1$ at time $k, i = 1, \cdots, D$. For the tree example in Figure 3.2, at time k, the fusion center has

$$\begin{aligned} \mathcal{Y}_k^k &= \{y_k^1, y_k^2\}, \\ \mathcal{Y}_k^{k-1} &= \{y_{k-1}^1, y_{k-1}^2, y_{k-1}^3, y_{k-1}^4\}. \end{aligned}$$

We immediately notice that $\mathcal{Y}_{k-i}^{k-i} \subset \mathcal{Y}_k^{k-i}$, i.e., more measurements for time $k-i$ are collected at k compared with at time $k-i$. For example, $\mathcal{Y}_{k-1}^{k-1} = \{y_{k-1}^1, y_{k-1}^2\}$ are the only available measurements at time $k-1$. However at time k, the available measurements for time $k-1$ changes to $\mathcal{Y}_k^{k-1}$. Hence we can obtain a better estimate of x_{k-1} at time k than at time $k-1$. This inspires us to recompute the optimal estimate of the previous states and use them as input to generate the current estimate. That is the basic idea contained in Theorem 3.1, where we recompute the optimal estimate of $x_{k-D+1}, \cdots, x_{k-1}$ at time k and then make use of the updated estimates to compute the current estimate $\hat{x}_k$. Figure 3.3 shows the overall estimation scheme at time k.

Let S_{i_j} be the node that is j hops away from S_0. Define

$$\begin{aligned} \Gamma_j &\triangleq [H_{1_j}; H_{2_j}; \cdots], \quad j = 1, \cdots, D \\ C_i &\triangleq [\Gamma_1; \cdots; \Gamma_i], \quad i = 1, \cdots, D \\ \Upsilon_j &\triangleq \text{diag}\{\Pi_{1_j}, \Pi_{2_j}, \cdots\}, \quad j = 1, \cdots, D \\ R_i &\triangleq \text{diag}\{\Upsilon_1, \cdots, \Upsilon_i\}, \quad i = 1, \cdots, D. \end{aligned}$$

Intuitively, Γ_j is the joint measurement matrix and Υ_j is the joint noise covariance from all sensors that are j hops from the fusion center. C_i is the joint measurement matrix, and R_i is the joint noise covariance from all sensors that are j or less than j hops from the fusion center. With these definitions, the following theorem presents the optimal estimation algorithm over a sensor tree.

Theorem 3.1 *Consider a sensor tree T with depth D.*

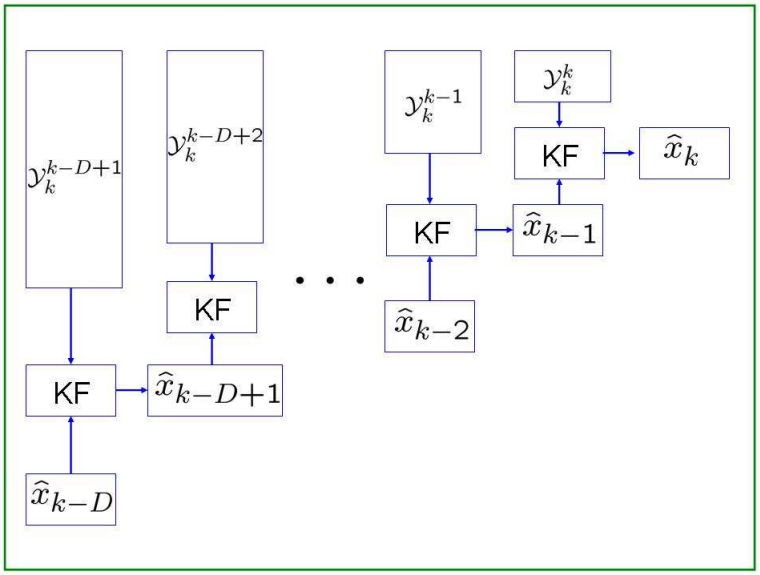

Figure 3.3: Kalman Filter Iterations at Time k

1. $\hat{x}_k$ and P_k can be computed from D Kalman filters as

$$\begin{aligned}
(\hat{x}_{k-D+1}, P_{k-D+1}) &= \boldsymbol{KF}(\hat{x}_{k-D}, P_{k-D}, \mathcal{Y}_k^{k-D+1}, C_D, R_D) \\
&\vdots \\
(\hat{x}_{k-1}, P_{k-1}) &= \boldsymbol{KF}(\hat{x}_{k-2}, P_{k-2}, \mathcal{Y}_k^{k-1}, C_2, R_2) \\
(\hat{x}_k, P_k) &= \boldsymbol{KF}(\hat{x}_{k-1}, P_{k-1}, \mathcal{Y}_k^{k}, C_1, R_1).
\end{aligned}$$

2. P_∞^- and P_∞ satisfy

$$P_\infty^- = g_{C_2} \circ \cdots \circ g_{C_{D-1}}(P^*), \tag{3.3}$$

$$P_\infty = \tilde{g}_{C_1} \circ g_{C_2} \circ \cdots \circ g_{C_{D-1}}(P^*), \tag{3.4}$$

where P^* is the unique solution to $g_{C_D}(P^*) = P^*$.

Proof: 1) We know that the estimate $\hat{x}_k$ is generated from the estimate of $\hat{x}_{k-1}$ together with all the available measurements at time k through a traditional Kalman filter. Similarly, the estimate $\hat{x}_{k-1}$ is generated from the estimate of $\hat{x}_{k-2}$ together with all the available measurements for time $k-1$ at time k, etc. This recursion for D steps corresponds to the

D Kalman filters stated in the theorem.

2) Follows directly from Eqn (2.14) and (2.15). ■

Remark 3.2 The estimation algorithm presented in Theorem 3.1 readily extends to a general graph that represents the sensor communications. The fusion center only needs to keep track of the measurement data up to previous time $k - D + 1$. Thus in a distributed setting, every node acts as a fusion center and the system robustness (against sensor failure) is increased.

3.4 Example

We consider an integrator chain in this section. The discrete time system dynamics is given by Eqn (3.1) with

$$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}.$$

and with process noise covariance $Q = 0.3I$. There are two sensors available. The measurement equations are given by

$$\begin{aligned} y_k^1 &= [\ 0 \quad 1\ ]x_k + v_k^1 = H_1 x_k + v_k^1, \\ y_k^2 &= [\ 1 \quad 0\ ]x_k + v_k^2 = H_2 x_k + v_k^2, \end{aligned}$$

with covariances $\Pi_1 = 0.25$ and $\Pi_2 = 0.5$. Consider the following two sensor topologies (Figure. 3.4).

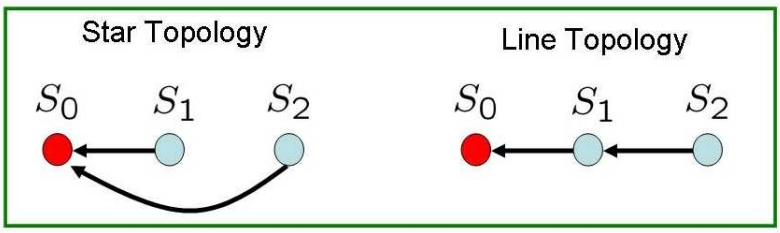

Figure 3.4: Integrator Chain Example

The first one is the star topology, i.e., the two sensors communicate with the fusion center directly, which corresponds to the centralized Kalman filter. The second one is a line topology (a special tree), and the measurement data from sensor two to the fusion center get delayed by one step. For this example, it is easy to calculate that

$$P^* = \begin{bmatrix} 0.1838 & 0.0103 \\ 0.0103 & 0.1822 \end{bmatrix},$$

which is the unique solution to $P^* = g_{[H_1;H_2]}(P^*)$. As a result, for the star topology,

$$P_\infty(\text{star}) = \tilde{g}_{[H_1;H_2]}(P^*) = \begin{bmatrix} 0.0825 & 0.0021 \\ 0.0021 & 0.0822 \end{bmatrix},$$

with $\text{Tr}\big(P_\infty(\text{star})\big) = 0.1647$. For the line topology,

$$P_\infty(\text{line}) = \tilde{g}_{[H_1]}(P^*) = \begin{bmatrix} 0.1835 & 0.0047 \\ 0.0047 & 0.0823 \end{bmatrix},$$

with $\text{Tr}\big(P_\infty(\text{line})\big) = 0.2658$.

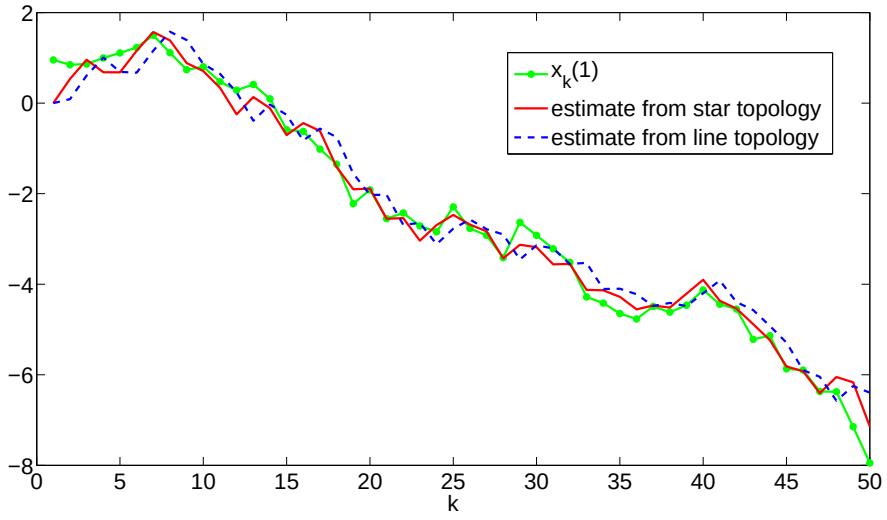

Figure 3.5: True State and its Estimates

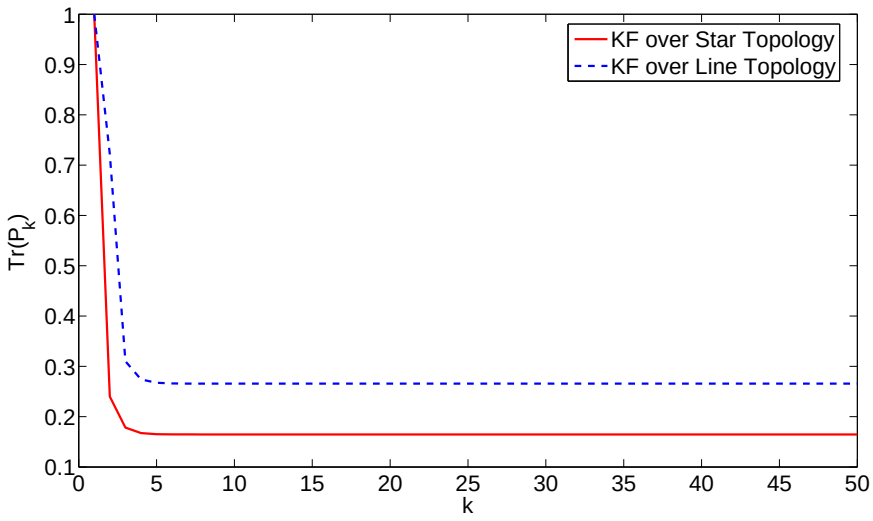

Figure 3.6: Error Covariances

We plot the first component of the true state and its estimates based on the two sensor topologies in Figure 3.5. We also plot the corresponding error covariance in Figure. 3.6. As those figures demonstrate, the simulations agree well with the theory developed.

3.5 Some Useful Inequalities

We conclude this chapter by presenting some useful inequalities that will be used in the next chapter.

Lemma 3.3 *Assume* $1 \leq i \leq j \leq D$ *and* $P \in \mathbb{S}_+^n$. *Then*

$$C_i'[C_iPC_i' + R_i]^{-1}C_i \leq C_j'[C_jPC_j' + R_j]^{-1}C_j. \tag{3.5}$$

Proof: We first prove for $i = 1$ and $j = 2$. When $i = 1$ and $j = 2$, we can write Eqn (3.5) as

$$\begin{aligned} & \Gamma_1'[\Gamma_1 P \Gamma_1' + \Upsilon_1]^{-1}\Gamma_1 \\ \leq & \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}' \left[\begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} P \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}' + R_2 \right]^{-1} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} \\ = & \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}' \begin{bmatrix} B & M \\ M' & G \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} \end{aligned}$$

where $B = \Gamma_1 P \Gamma_1' + \Upsilon_1$, $G = \Gamma_2 P \Gamma_2' + \Upsilon_2$, and $M = \Gamma_1 P \Gamma_2'$. Since $B > 0, G > 0$, and

$$\begin{bmatrix} B & M \\ M' & G \end{bmatrix} > 0,$$

from Lemma 2.6, the Schur complement

$$S_B \triangleq B - MG^{-1}M' > 0.$$

From Lemma 2.7, we obtain

$$\begin{aligned} & \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}' \begin{bmatrix} B & M \\ M' & G \end{bmatrix}^{-1} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} \\ = & \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}' \begin{bmatrix} X_1 & -B^{-1}MS_B^{-1} \\ -S_B^{-1}M'B^{-1} & S_B^{-1} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} \\ = & \Gamma_1' B^{-1} \Gamma_1 + X_2 X_2' \\ \geq & \Gamma_1' B^{-1} \Gamma_1. \end{aligned}$$

where $X_1 = B^{-1} + B^{-1}MS_B^{-1}M'B^{-1}$ and $X_2 = \Gamma_1' B^{-1} M S_B^{-\frac{1}{2}} - \Gamma_2' S_B^{-\frac{1}{2}}$. Having proved the case $i = 1, j = 2$, the general case easily follows if we write $\Gamma_1 := C_i$ and $\Gamma_2 := C_j \setminus C_i$. ■

Corollary 3.4 *For all $i = 1, \cdots, n-1$, and all $X \geq 0$,*

$$g_{C_{i+1}}(X) \leq g_{C_i}(X).$$

Corollary 3.5 *For all* $i = 1, \cdots, n-1$*, and all* $X \geq 0$*,*

$$\tilde{g}_{C_{i+1}}(X) \leq \tilde{g}_{C_i}(X).$$

We can interpret Corollary 3.4 and 3.5 in the following sense. For an estimator, the more information it has (i.e., more sensors), and/or the less delay the measurement data arrive, the more accurate it can predict the process state. We will use these inequalities to guide us looking for the minimum energy sensor tree in the next chapter.

Chapter 4

Minimizing Sensor Energy

4.1 Introduction

Given a tree T that represents the sensor communications with the fusion center, we have seen in Chapter 3 how the optimal state estimate $\hat{x}_k$ can be computed at the fusion center, and we have derived a closed form of the steady-state error covariance in Eqn (3.4).

As stated in Chapter 1, the communication between the sensor nodes is limited, particularly if nodes are located physically far way from each other. It takes time to transfer information from one node to another. Most nodes are battery powered, and hence to extend the life time of such nodes, data are communicated over a multi-hop wireless network, instead of a single-hop network. The quality of the state estimate at the fusion center thus depends not only on the sensor quality but also on the communication delay, i.e., the number of hops the sensor measurement data need to take until they reach the fusion center. Many short hops take longer time than a few long hops. On the other hand, a few long hops require larger transmission power since the required transmission grows rapidly with the distance between the wireless nodes. Hence, there is a trade-off between the state estimation quality and the overall energy cost. In this chapter, we consider the energy minimization subject to performance constraint.

Sensor network energy minimization is typically done via efficient MAC protocol design [32], or via efficient scheduling of the sensor states [28, 50]. Yu et al. [58] proposed a scalable topology and energy management scheme in wireless sensor networks. The interaction between Kalman filtering and how data is routed on a network seems to be less studied. Routing of data packets in networks are typically done based on the distance to

the receiver node [5]. Some recent work addresses how to couple data routing with the sensing task using information theoretic measures [22].

The solution we propose is to optimize the network path for the sensor data via the *Tree Reconfiguration Algorithm* such that the overall transmission energy is minimized, but guarantees a certain level of estimation quality. The resulting local sensor topology has the structure of a tree for which the fusion center is the root. In case sensor node failure happens, or new sensors join, or existing sensors leave to serve other applications, the tree can be reformed dynamically, which increases robustness of the overall system.

There are several potential application areas of the work presented in this chapter, including building automation, environmental monitoring, industrial automation, etc.

The rest of this chapter is organized as follows. After the mathematical framework is set up in Section 4.2, we state the energy minimization problem in Section 4.3, and propose the *Tree Reconfiguration Algorithm* to minimize the energy usage of the sensors. The algorithm is presented in detail with a performance analysis. Examples are provided in the end to illustrate the algorithms developed.

4.2 Problem Set-up

Consider the problem of state estimation over a wireless sensor network (Figure 5.1).

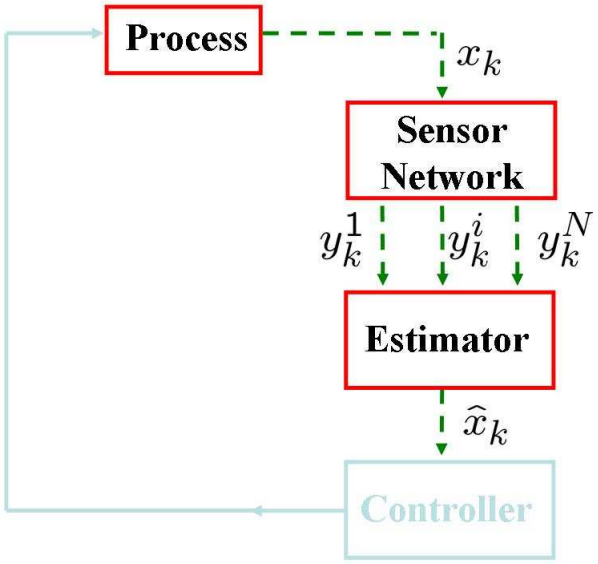

Figure 4.1: State Estimation Using a Wireless Sensor Network

Plant and Sensor Models

The process dynamics and sensor measurement equations are the same as in Eqn (3.1) and (3.2), i.e.,

$$x_k = Ax_{k-1} + w_{k-1},$$
$$y_k^i = H_i x_k + v_k^i, i = 1, \cdots, N.$$

For a tree T that represents the sensor communications with the fusion center (S_0), we have defined the estimation quantities $[\hat{x}_k(T), P_k(T), \text{ etc.}]$ in Chapter 3.2. For the remaining of this chapter, Node(T) means all the nodes of T, which is a subset of all sensors $\{S_1, \cdots, S_N\}$; $\text{Fam}_T(S_i)$ is the subtree of T that is rooted at S_i; $\text{Par}_T(S_i)$ is the parent node of S_i in T; Edge(T) is the edges of T, i.e.,

$$\text{Edge}(T) \triangleq \big\{(S_i, S_j) : S_i \in \text{Node}(T), S_j = \text{Par}_T(S_i)\big\}.$$

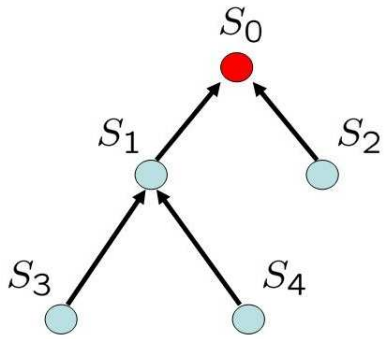

Figure 4.2: Sensor Tree Example Revisited

Example 4.1 For the tree in Figure 4.2, we have that $\text{Node}(T) = \{S_1, S_2, S_3, S_4\}$, $\text{Fam}_T(S_1) = \{S_3, S_4\}$, $\text{Par}_T(S_3) = S_1$ and $\text{Edge}(T) = \{(S_3, S_1), (S_4, S_1), (S_1, S_0), (S_2, S_0)\}$.

Sometimes we write $S_i \in T$ to mean $S_i \in \text{Node}(T)$. For all notations, we drop the subscript T when the considered tree follows from the context.

Sensor Energy Cost Model

We assume that the sensor nodes are battery powered. Sensors spend energy in many ways, i.e., packet transmission and reception, idle listening, computing, etc. [13]. By using appropriate MAC protocol, e.g., the TDMA protocol, packet transmission and reception dominate the total energy usage. Given a tree T that represents the sensor communications with S_0, define $e^i_{tx}(T)$ as the energy cost for S_i sending a measurement packet to its parent node and $e^i_{rx}(T)$ as the energy cost for S_i receiving a measurement packet from one of its children, and we write $e^i(T) = e^i_{tx}(T) + e^i_{tx}(T)$ as the total energy cost for S_i in T. The transmission power $e^i_{tx}(T)$ typically grows rapidly with the distance to the receiver. An estimate of e^i_{tx} can be be computed based on the wireless technology. A common model is that if the distance between S_i and $\mathrm{Par}(S_i)$ is d_i, then $e^i_{tx} = \beta_i + \alpha_i(d_i)^{n_i}$, where β_i represents the static part of the energy cost and $\alpha_i(d_i)^{n_i}$ the dynamic part. The path-loss exponent n_i is typically between 2 and 6. The receiving energy $e^i_{rx}(T)$ is about the same for each sensor, therefore without loss of generality, we write $e^i_{rx}(T) = e_{rx}$. Then the total energy cost of T per time is given by

$$e(T) = \sum_{S_i \in T} e^i_{tx}(T) + |T| e_{rx}, \tag{4.1}$$

where $|T|$ denotes the number of sensor nodes in T.

4.3 Minimizing Sensor Energy

Problem of Interest

Since the sensors operate on batteries, it is natural to let the network operate at an energy level that is as low as possible. Thus we are interested in the following problem:

Problem 4.2 *How should the tree T be established such that the total network energy cost is minimum yet the network provides a guaranteed level of estimation quality? i.e.,*

$$\min_{T \in \mathcal{T}_{all}} e(T)$$

subject to

$$P_\infty(T) \le P_{desired}$$

where $\mathcal{T}_{all}$ is the set of all sensor trees, and $P_{desired} \in \mathbb{S}_+^n$ is given.

Solving the previous problem via exhaustive search is impractical, as $|\mathcal{T}_{\text{all}}|$ grows even faster than exponentially as a function of N. Thus we consider the following *Tree Reconfiguration Algorithm* to approximate the optimal solution.

Tree Reconfiguration Algorithm

In this section, we present the *Tree Reconfiguration Algorithm* (Figure 4.3) which solves Problem 4.2 efficiently, but at the price of losing optimality in some cases.

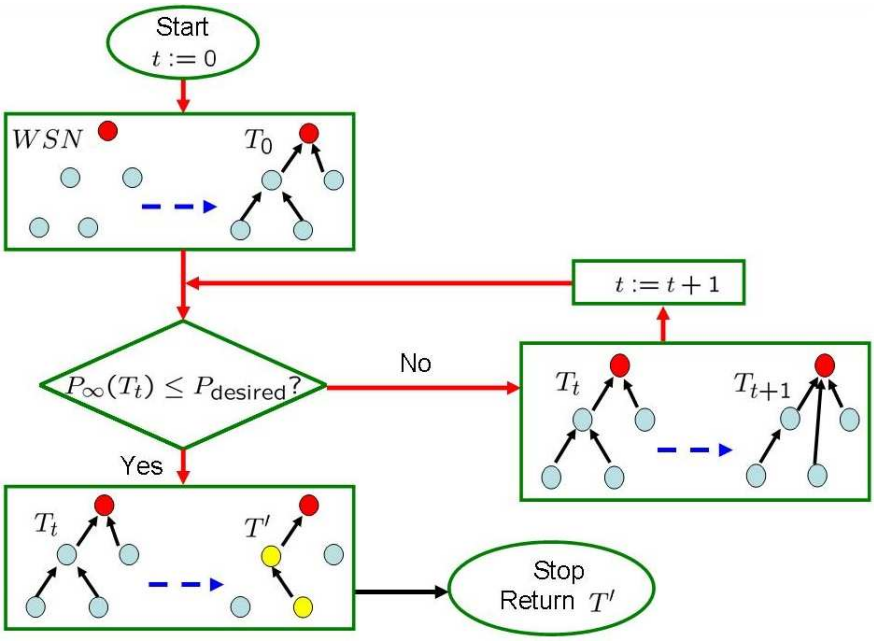

Figure 4.3: Tree Reconfiguration Algorithm

The *Tree Reconfiguration Algorithm* (Figure 4.3) consists of three subroutines. The first subroutine is called by executing the *Tree Initialization Algorithm* to form an initial tree T_0 (the top rectangular block). Depending on whether T_0 provides the required estimation quality, two other subroutines are called by executing the *Switching Tree Topology Algorithm* (the middle-right rectangular block) and the *Minimum Energy Subtree Algorithm* (the bottom rectangular block), respectively. These algorithms are presented in detail next.

Tree Initialization Algorithm

Let T_0 denote the tree which represents the initial connection of the sensors with S_0. T_0 is constructed via the *Tree Initialization Algorithm* presented graphically in Figure 4.4.

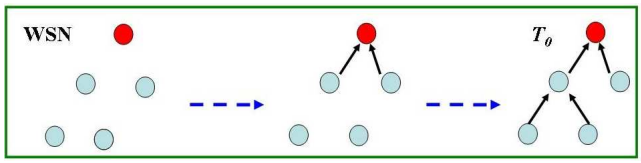

Figure 4.4: Tree Initialization Algorithm: Intuitive Idea

The idea is that S_0 first establishes direct connections with its neighbor sensors using minimum transmission power level Δe. After that, its neighbor sensors establish further connections with their own neighbor sensors also using minimum transmission power level Δe. This process continues until a tree of depth D is formed. Let $S(t)$ be the sensors added to T_0 at or before step t, $\Delta S(t)$ be the set of sensors added at step t, and $V_{\Delta e}(S_i)$ be the set of sensors that are reachable by S_i using Δe energy. The *Tree Initialization Algorithm* is presented in its flow diagram form in Figure 4.5 with

$$V_{\Delta e}(\Sigma) \triangleq \bigcup_{S_i \in \Sigma} V_{\Delta E}(S_i).$$

Switching Tree Topology Algorithm

For a given tree T_t, if $P_\infty(T_t) \nleqslant P_{\text{desired}}$, the tree needs to be adjusted in a way that the estimation quality is improved. The *Switching Tree Topology Algorithm* provides such a way (Figure 4.6).

We define $\pi(T_t, S_i)$ as the operation that S_i breaks connection with $\text{Par}(S_i)$ and connects directly to S_0, i.e., $\text{Node}\big(\pi(T_t, S_i)\big) = \text{Node}(T_t)$ and

$$\text{Edge}\big(\pi(T_t, S_i)\big) = \text{Edge}(T_t) \bigcup \{S_i, S_0\} \setminus \{S_i, \text{Par}_{T_t}(S_i)\}.$$

Further define $S_{2\text{hop}} \triangleq \{S_i \in T_t : \text{Par}(\text{Par}(S_i)) = S_0\}$. The algorithm is then given as follows.

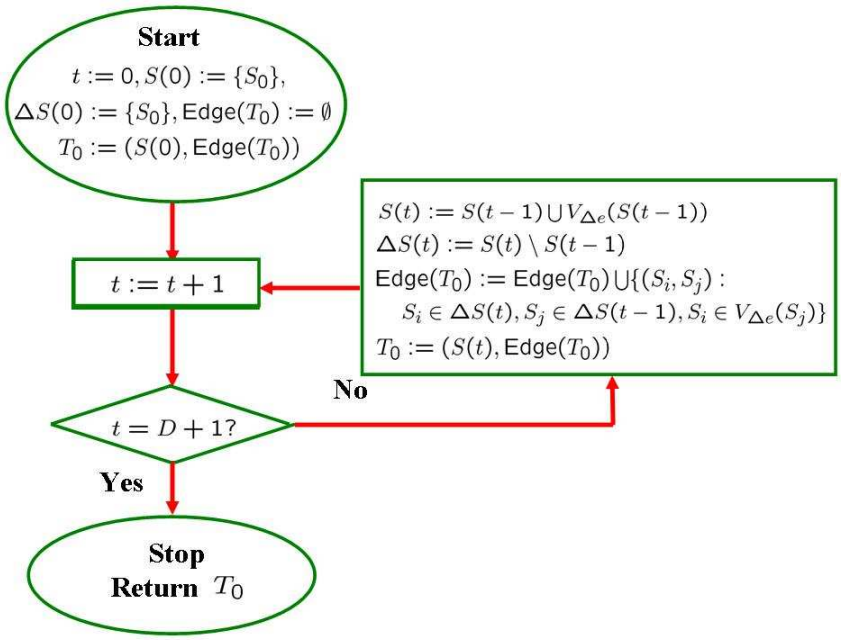

Figure 4.5: Tree Initialization Algorithm: Flow Diagram

Algorithm 1 Switching Tree Topology Algorithm

Init: T_t.

Compute

$$S_p = \arg \min_{S_i \in S_{2\text{hop}}} P_\infty(\pi(T_t, S_i)).$$

Return $T_{t+1} := \pi(T_t, S_p)$.

Minimum Energy Subtree Algorithm

For a given tree T with $P_\infty(T) \leq P_{\text{desired}}$, the *Minimum Energy Subtree Algorithm* finds the subtree T' rooted at S_0 with the property that $P_\infty(T') \leq P_{\text{desired}}$, and $e(T') \leq e(\tilde{T})$ for any subtree $\tilde{T}$ of T rooted at S_0. The idea is that all possible subtrees $\tilde{T}$ rooted at S_0 and

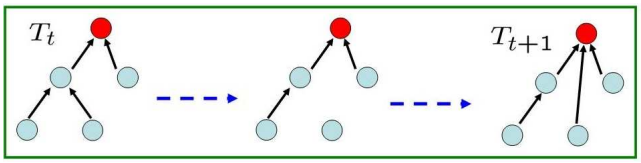

Figure 4.6: Switching Tree Topology Algorithm: Intuitive Idea

satisfying

$$P_\infty(\tilde{T}) \leq P_{\text{desired}}$$

are found in an efficient way utilizing the structure of T. Then the subtree T' which has the least overall energy cost is returned. The details are as follows.

To make the presentation clear and easy to follow, we divide the algorithm into several key steps and provide an example to illustrate each step. Before introducing the algorithm, let us define

$$\begin{aligned} S(i_1 i_2 \cdots i_p) &\triangleq \{S_{i_1}, S_{i_2}, \cdots S_{i_p}\}, \\ \Omega(i_1 i_2 \cdots i_p) &\triangleq T \setminus S(i_1 i_2 \cdots i_p), \end{aligned}$$

where it is assumed $i_1 \leq i_2 \leq \cdots \leq i_p$. We consider the following example to demonstrate the algorithm.

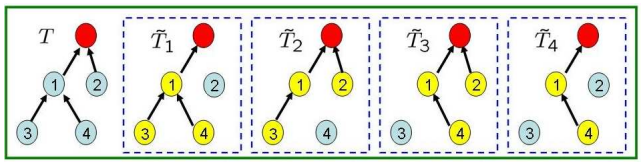

Figure 4.7: Tree T and Some Subtree $\tilde{T}$s

Example 4.3 Consider the tree T with four sensor nodes in Figure 4.7. Assume the following:

1) T provides enough estimation quality, i.e., $P_\infty(T) \leq P_{\text{desired}}$.

2) No single sensor provides enough estimation quality, i.e.,

$$P_\infty(S(i)) \nleq P_{\text{desired}}, i = 1, 2, 3, 4.$$

3) Among the two sensor pairs, only $\{S_1, S_4\}$ can provide enough estimation quality, i.e.,

$$P_\infty(S(ij)) \leq P_{\text{desired}} \text{ iff } \{i, j\} = \{1, 4\}.$$

4) Any three sensors except $\{S_2, S_3, S_4\}$ can provide enough estimation quality, i.e.,

$$P_\infty(\Omega(i)) \leq P_{\text{desired}}, i = 2, 3, 4.$$

5) The energy cost of single hop communication is Δe.

By the above assumptions, it is easy to see that the minimum energy subtree T' is given by $\tilde{T}_4$ with $e(\tilde{T}_4) = 2\Delta e$.

Let us examine the case when we take T as an input to the *Minimum Energy Subtree Algorithm* which consists of the following key steps.

Step 1

- Init: T
- $l := 0, \mathcal{D}_l := \{S_{i_p} \in T : P_\infty(\Omega(i_p)) \leq P_{\text{desired}}\}$.

In this step, $\mathcal{D}_0$ holds all single-sensor nodes without which the remaining sensors still satisfy the accuracy requirement. Therefore in Example 4.3, $\mathcal{D}_0 = \{S_2, S_3, S_4\}$.

Step 2

- $l := l + 1, \mathcal{D}_l := \mathcal{D}_{l-1}$
- $\forall\ S_{i_p} \in \mathcal{D}_{l-1}$ with $P_\infty(\Omega(i_p)) \leq P_{\text{desired}}$
  - $\forall\ q > p$ and $S_{i_q} \notin \text{Fam}(S_{i_p})$,
    if $P_\infty(\Omega(i_p i_q)) \leq P_{\text{desired}}$, $\mathcal{D}_l := \mathcal{D}_l \bigcup S(i_p i_q)$.

In this step, $\mathcal{D}_1$ holds all single-sensor or two-sensor pairs without which the remaining sensors still satisfy the accuracy requirement. The third line of step 2 eliminates the redundancy in listing the subtrees as $S(i_p i_q) = S(i_q i_p)$, and if S_{i_p} is removed from a tree, so is $\text{Fam}(S_{i_p})$. Therefore in Example 4.3, $\mathcal{D}_1 = \{S_2, S_3, S_4, S(23)\}$.

Step 3

- $l := l + 1, \mathcal{D}_l := \mathcal{D}_{l-1}$
- $\forall\, S(i_p i_q) \in \mathcal{D}_{l-1}$ with $P_\infty(\Omega(i_p i_q)) \le P_{\text{desired}}$
  - $\forall\, o > q$ and $S_{i_o} \notin (\text{Fam}(S_{i_p}) \bigcup \text{Fam}(S_{i_q}))$,
    if $P_\infty(\Omega(i_p i_q i_o)) \le P_{\text{desired}}$,
    $\mathcal{D}_l := \mathcal{D}_l \bigcup S(i_p i_q i_o)$.

Similar to step 3, $\mathcal{D}_2$ holds all single-sensor, two-sensor pairs or three-sensors without which the remaining sensors still satisfy the accuracy requirement. The algorithm continues in this way until $\mathcal{D}_r = \mathcal{D}_{r-1}$ at some step $r \le D$.

Step r + 1

- Return $T' = \arg\min_{\Omega(\cdot)\in\mathcal{D}} e(\Omega(\cdot))$

In Example 4.3, $\mathcal{D}_2 = \{S_2, S_3, S_4, S(23)\} = \mathcal{D}_1$. Hence the algorithm stops and returns $T' = \Omega(23) = S(14) = \tilde{T}_4$ with $P_\infty(T') \le P_{desired}$ and $e(T') = 2\Delta e$.

Performance Analysis of the Algorithms

The performance of the *Tree Reconfiguration Algorithm* is summarized in the following theorem.

Theorem 4.4 *(1) Given a tree T_t, the Switching Tree Topology Algorithm returns $T_{t+1} \in \mathcal{T}_{all}$ such that*

$$P_\infty(T_{t+1}) \le P_\infty(T_t)\ .$$

(2) Given a tree T with $P_\infty(T) \le P_{desired}$, the Minimum Energy Subtree Algorithm returns $T' \subset T$ rooted at S_0 such that

$$P_\infty(T') \le P_{desired} \text{ and } e(T') \le e(\tilde{T})$$

for any other $\tilde{T} \subset T$ that is rooted at S_0.

(3) If $\exists\, T \in \mathcal{T}_{all}$ such that $P_\infty(T) \le P_{desired}$, then the output T' from the Tree Reconfiguration Algorithm satisfies $P_\infty(T') \le P_{desired}$.

Proof: (1) We provide the proof for the line topology (Figure 4.8). It is straightforward to extend the proof for a general tree. For T_t, $P_\infty(T_t)$ is given by Eqn (3.4) as

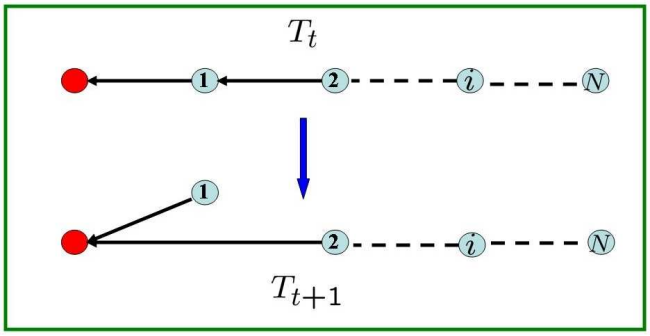

Figure 4.8: Switching Tree Topology

$$P_\infty(T_t) = \tilde{g}_{C_1} \circ g_{C_2} \circ g_{C_3} \circ \cdots \circ g_{C_{D-1}}(P^*),$$

where $P^* \ge 0$ is the unique solution to

$$g_{C_D}(P^*) = P^*.$$

For T_{t+1}, $P_\infty(T_{t+1})$ is given by

$$\begin{aligned} P_\infty(T_{t+1}) &= \tilde{g}_{C_2} \circ g_{C_3} \circ g_{C_4} \circ \cdots \circ g_{C_{D-1}}(P^*) \\ &= \tilde{g}_{C_2} \circ g_{C_3} \circ g_{C_4} \circ \cdots \circ g_{C_{D-1}} \circ g_{C_D}(P^*) \\ &\le \tilde{g}_{C_1} \circ g_{C_2} \circ g_{C_3} \circ \cdots \circ g_{C_{D-2}} \circ g_{C_{D-1}}(P^*) \\ &= P_\infty(T_t) \end{aligned}$$

where the inequality is from Corollary 3.4 and 3.5.

(2) Suppose $T^* = (S^*, \text{Edge}(T^*))$ is the subtree that has the least energy cost with $P_\infty(T^*) \le P_{\text{desired}}$. Let $\Delta S = S \setminus S^* = \{S_{i_1}, S_{i_2}, \cdots, S_{i_m}\}$ with $i_1 \le i_2 \le \cdots \le i_m$. Then

$\Delta S \subset \mathcal{D}_r$, as $P_\infty(T^*) \leq P_{\text{desired}}$. We also have $S(i_1 i_2) \in \mathcal{D}_r$ as

$$P_\infty(T \setminus S(i_1 i_2)) \leq P_\infty(T^*) \leq P_{\text{desired}}.$$

Similarly, $S(i_1 i_2 \cdots i_m) \in \mathcal{D}_r$ and so $T^* = T \setminus S(i_1 i_2 \cdots i_m)$ is returned by the *Minimum Energy Subtree Algorithm* as we assume T^* is the subtree that has the least energy cost.

(3) Let $T^\star$ denote the star tree, i.e., all sensors communicate with S_0 directly. It is easy to verify that $P_\infty(T^\star) \leq P_\infty(T)$ for all $T \in \mathcal{T}_{\text{all}}$. If there exists T such that $P_\infty(T) \leq P_{\text{desired}}$, we must also have $P_\infty(T^\star) \leq P_\infty(T)$. Suppose at t_1, $P_\infty(T_{t_1}) \leq P_{\text{desired}}$, then it is clear that $P_\infty(T') \leq P_{\text{desired}}$. Otherwise, the *Tree Reconfiguration Algorithm* continues until direct connections between all sensors with S_0 are established, in which case $P_\infty(T_t) = P_\infty(T^\star) \leq P_{\text{desired}}$. Hence $P_\infty(T') \leq P_{\text{desired}}$. ■

A TDMA Scheduling Scheme

In this section, we present a TDMA scheduling scheme for practical implementation of the Tree Reconfiguration Algorithm, which is similar to the scheduling phase in [10]. Assume each discrete time is divided into some equal length time slots, e.g., in wireless HART protocol [19], there are 100 time slots per second. The advantage of TDMA scheme is that sensors only need to communicate during its own time slot and hence save energy by avoiding the idle listening. The scheme consists of the following two phases.

1. The fusion center calculates the $(S_i, \text{Par}(S_i))$ pair according to the Tree Reconfiguration Algorithm. It then assigns one time slot to each $(S_i, \text{Par}(S_i))$ pair, i.e., S_i will communicate with $\text{Par}(S_i)$ at the specified time slot.

2. At the beginning of every M times, the fusion center broadcasts the communication schedule to all sensors.

Notice that the fusion center broadcasts every M times for several reasons. Firstly, synchronization is a very difficult in general wireless sensor networks due to the drift of each individual clock, etc. This scheme solves this problem as long as M is sufficiently small, as all sensors can then synchronize their clocks with the fusion center at the beginning of each M cycles. Secondly, existing sensors might cease working possibly due to running out of battery, hardware failure, etc. In these cases, the fusion center can calculate a new

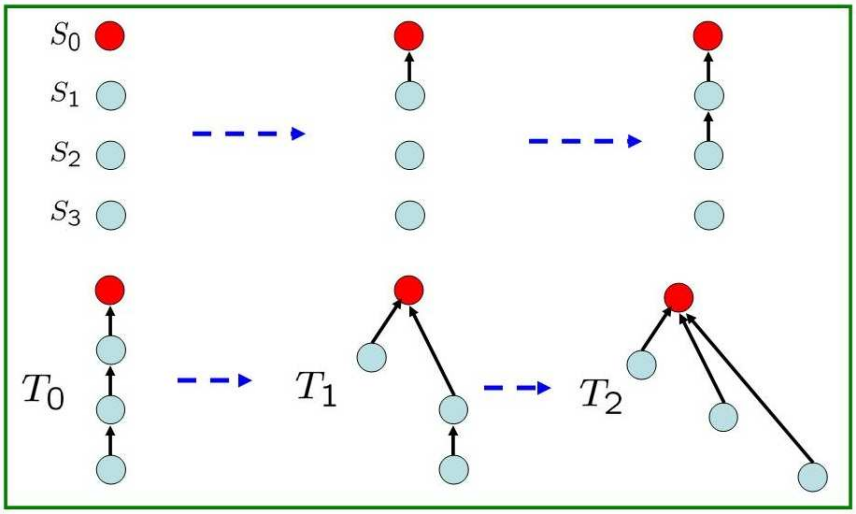

Figure 4.9: Different Trees Formed by the Tree Reconfiguration Algorithm

schedule based on the remaining sensors and broadcasts the new schedule at the beginning of the next cycle. Similarly, new sensors might join the network for the same or different applications. The fusion center can again calculate and broadcast the new schedule.

4.4 Examples

Three sensors are available to measure the state of a process (see Figure 4.9). Assume that if S_i is connected to $S_{i-1}, i = 1, 2, 3$, the energy of communication is Δe; if S_i is connected to $S_{i-2}, i = 2, 3$, the energy is $4\Delta e$ and if S_3 is connected to S_0, the energy is $8\Delta e$. Without loss of generality, for the remaining of the examples, we only calculate the total transmitting energy. We consider two scenarios in this section.

Tree Reconfiguration with Time-Varying Disturbances

The discrete time system dynamics is given by Eqn (3.1) with

$$A = \begin{bmatrix} 1 & 0.1 & 0.05 & 0.0002 \\ 0 & 1 & 0.1 & 0.05 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The measurement equations for the three sensors are given by

$$\begin{aligned} y_k^1 &= [\ 1 \quad 0 \quad 0 \quad 0\ ]x_k + v_k^1, \\ y_k^2 &= [\ 0 \quad 1 \quad 0 \quad 0\ ]x_k + v_k^2, \\ y_k^3 &= [\ 0 \quad 0 \quad 1 \quad 0\ ]x_k + v_k^3, \end{aligned}$$

with $\Pi_1 = 0.25, \Pi_2 = 0.5$, and $\Pi_3 = 0.5$. Assume S_i is i hops away from S_0 (Figure 4.9).

The dynamics in the simulation include control input as well, i.e., $x_k = Ax_{k-1} + Bu_k + w_{k-1}$, which is computed via the standard LQG control procedures. In this example $B = [0\ 0\ 0\ 1]$.

Suppose it is required that $\mathrm{Tr}(P_\infty) \leq 10$ for this system. Notice that we can simply replace $P_\infty \leq P_{\text{desired}}$ with $\mathrm{Tr}(P_\infty) \leq \mathrm{Tr}(P_{\text{desired}})$ in all previous algorithms. Initially, assume $Q_k = Q_0 \triangleq 0.2I$ for all $k \leq k_1 = 200$. After T_0 is set up, S_0 computes $\mathrm{Tr}(P_\infty(T_0)) = 4.1297 < 10$. Thus it starts to run the *Minimum Energy Subtree Algorithm* to find out T'. In this case $T' = T_0 \setminus S_3$ with $\mathrm{Tr}(P_\infty(T')) = 9.6411$ and $e(T') = 2\Delta e$.

We model the disturbance to the plant as changes to Q_k. Suppose at time $k_1 + 1$, Q_k changes to $4Q_0$ and will last for 100 time steps. We assume the changes in Q_k are known to S_0. In the actual implementation, we can estimate the value of Q_k using various available schemes (e.g., see [31]). In this case, $T_0 \setminus S_3$ no longer provides enough accuracy as $\mathrm{Tr}\big(P_\infty(T')\big)$ changes to 34.9300. Thus S_0 executes the *Tree Reconfiguration Algorithm* again to find the desired tree. Now only the star topology T_2, with $\mathrm{Tr}(P_\infty(T_2)) = 9.6369$, provides enough accuracy. The price to pay for reconfiguring to T_2 is that $e(T_2) = 13\Delta e$. Figure 4.11 shows how the different tree locations change in the energy-error diagram for these scenarios. Later when Q_k changes back to Q_0 at $k_2 = 300$, T_2 is reconfigured to $T_0 \setminus S_3$ correspondingly.

Figure 4.10 shows the evolution of the fourth component of x_k and the estimation error e_k without and with the tree reconfiguration. As we can see from the lower half of the figures, the state and the estimation error remain almost the same after the tree reconfiguration, while if the tree is kept the same, there is a big fluctuation in the state and the estimation error during the times when Q_k changes to a higher value.

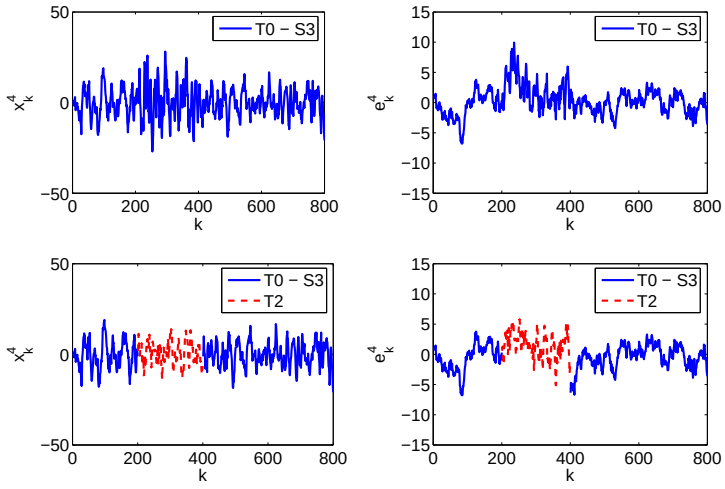

Figure 4.10: x_k^4 and e_k^4 without/with Tree Reconfiguration

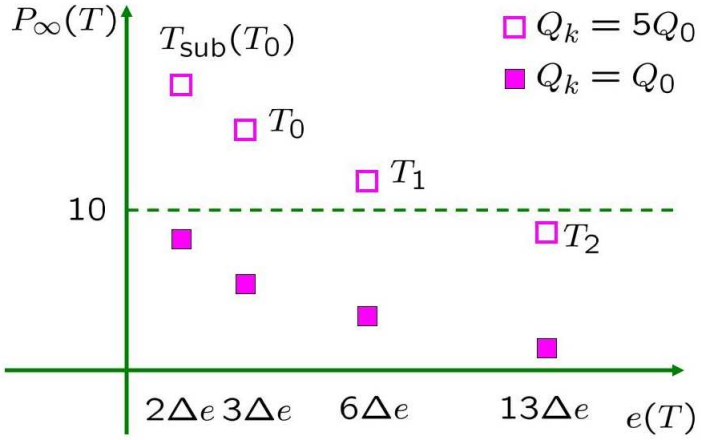

Figure 4.11: Changes of Tree Locations in Energy-Error Diagram

Tree Reconfiguration with Time-Varying Performance Requirement

In this example, the dynamics of the process and sensor measurement equations are as follows:

$$\begin{aligned} x_k &= 0.9x_{k-1} + w_{k-1}, \\ y_k^1 &= x_k + v_k^1, \\ y_k^2 &= x_k + v_k^2, \\ y_k^3 &= x_k + v_k^3, \end{aligned}$$

with $Q = 1, \Pi_1 = 1.5, \Pi_2 = 1$, and $\Pi_3 = 0.5$. Suppose the following performance requirement is received by the fusion center:

$$\begin{aligned} P_\infty &\leq 0.75, 1 \leq k \leq 100, \\ P_\infty &\leq 0.25, 101 \leq k \leq 200, \\ P_\infty &\leq 1.0, 201 \leq k \leq 300, \\ P_\infty &\leq 0.75, 301 \leq k \leq 500. \end{aligned}$$

Then the fusion center can find the corresponding minimum energy tree that fulfills the performance requirement. The intuitive idea is presented in Figure 4.12.

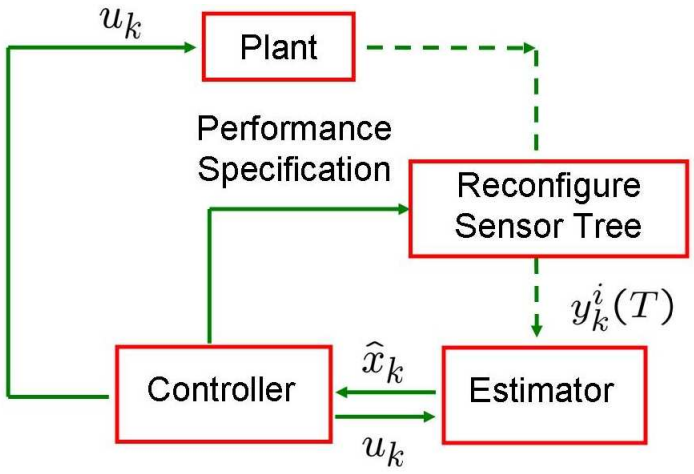

Figure 4.12: Tree Reconfiguration with Time-Varying Performance Requirement

Figure 4.13 shows the simulation result when the fusion center uses the same tree ($T0 \setminus S_3$) all the time, and when it reconfigures the trees according to the performance requirement. It is easy to see that when $101 \leq k \leq 200$, the total energy usage increases from $2\Delta e$ to $13\Delta e$. However, the error becomes much smaller; when $201 \leq k \leq 300$, the total energy usage reduces to just Δe. Although in this case the error becomes much larger, the performance specification is still satisfied.

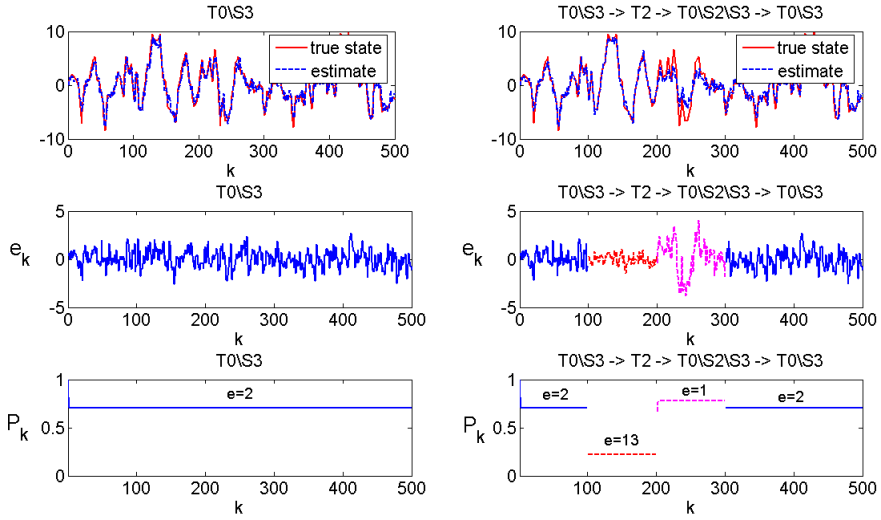

Figure 4.13: State and Error Evolution without/with Tree Reconfiguration

Chapter 5

Maximizing Network Lifetime

5.1 Introduction

In the previous chapter, we study the problem of minimizing sensor energy usage when considering estimation over a sensor network. In this chapter, we study the lifetime maximization problem. Sensor network lifetime maximization problem has been a hot area of research over the the past few years, as one of the critical constraints of such networks is limited energy resources available. Xue and Ganz [56] showed that the lifetime of the sensor networks is influenced by transmission schemes, network density and transceiver parameters with different constraints on network mobility, position awareness and maximum transmission range. Chamam and Pierre [8] proposed a sensor scheduling scheme to optimally put sensors in active or inactive modes. A sensor transmitting scheduling was suggested by Chen et al. [9]. Lai et al. [26] proposed a scheme to divide the deployed sensors into disjoint subsets of sensors such that each subset can complete the mission, and then maximized the number of such disjoint subsets. Similar approaches can be found in [36] where the sensors are partitioned into groups which are successively scheduled to be active for sensing and delivering data.

The solution we propose is to construct a set of sensor trees which all provide the guaranteed estimation quality and then schedule those sensor trees appropriately to maximize the lifetime of the whole network.

The rest of this chapter is organized as follows. After introducing a motivating example, the mathematical framework is set up in Section 5.2. Then the sensor tree construction and scheduling algorithm is introduced in Section 5.3, followed by an example to demonstrate the algorithm.

5.2 Problem Set-up

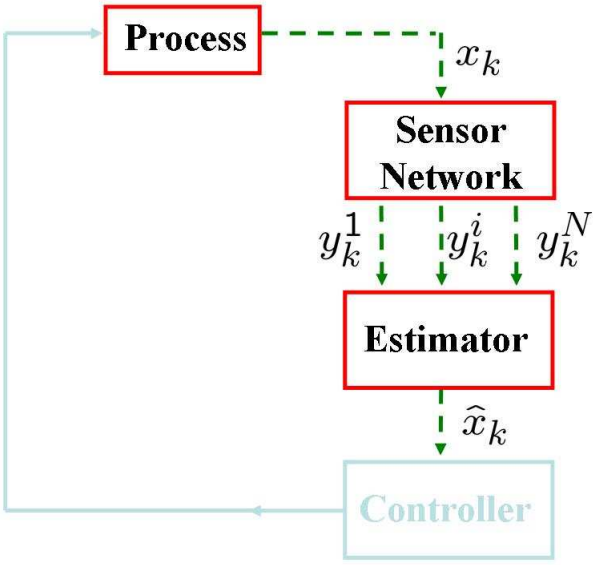

Figure 5.1: State Estimation Using a Wireless Sensor Network

Again consider the problem of state estimation over a wireless sensor network (Figure 5.1). The process dynamics, sensor measurement equations and sensor energy models are the same as in Section 4.2, i.e.,

$$x_k = Ax_{k-1} + w_{k-1},$$
$$y_k^i = H_i x_k + v_k^i, i = 1, \cdots, N,$$

and the total energy cost of a tree T is

$$e(T) = \sum_{S_i \in T} e_{tx}^i(T) + |T| e_{rx}.$$

A Motivating Example

Example 5.1 Consider the following network with $N = 2$. Assume both T_1 and T_2 in Figure 5.2 satisfy

$$P_\infty(T_i) \leq P_{\text{desired}}, i = 1, 2.$$

Further assume that

$$P_\infty(S_i) \nleqslant P_{\text{desired}}, i = 1, 2.$$

Let e_{ij} be the total energy cost for S_i in $T_j, i, j = 1, 2$, and let $\mathcal{E}_i$ be the initial energy for S_i. Consider the following parameters.

$$E = [e_{ij}] = \begin{bmatrix} 10 & 1 \\ 1 & 10 \end{bmatrix}, \mathcal{E}_1 = \mathcal{E}_2 = 1000.$$

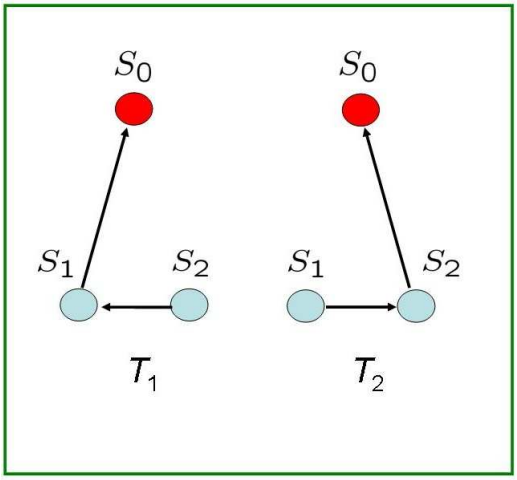

Figure 5.2: Network with Two Sensors

If we measure the lifetime of the network (denoted as L) as the first time that a sensor dies due to running out of battery, then it is easy to see that $L = 100$ when the *Tree Reconfiguration Algorithm* is executed, as T_1 is the only tree used.

It turns out that we can increase L by mixing the use of T_1 and T_2. Let $0 \leq \alpha \leq 1$ denote the portion of times that T_1 is used, we can show that if $0 < \alpha < 1$, then $L > 100$. It is also easy to verify that L attains its maximum value at 181 when $\alpha = 0.5$.

From this example, we see that simply minimizing the total energy usage of the sensors may not maximize the network lifetime, which is the focus of this chapter.

Problem of Interest

Let Θ be the set of all scheduling policies on $\mathcal{T}_{\text{all}}$, i.e.,

$$\Theta \triangleq \{\theta : \mathbb{Z}^+ \to \mathcal{T}_{\text{all}}\}.$$

Let $\theta(k) = T_k^\theta \in \mathcal{T}_{\text{all}}$. As $\theta(k)$ determines the energy consumed by S_i at k, given an initial energy level $\mathcal{E}_i$ for S_i, L can be calculated using θ. We are interested in finding a scheduling policy θ such that L is maximized. We also require that the estimation quality at the fusion center reaches certain desired level. In mathematical form, we are interested in solving the following *network lifetime maximization problem*

$$\begin{aligned} & \max_{\theta \in \Theta} L(\theta) \\ \text{subject to} & \\ & P_\infty(T_k^\theta) \leq P_{\text{desired}}. \end{aligned}$$

Let $\mathcal{T}_{d-\text{depth}}$ denote the set of trees which have depth d. Then it is easy to see that

$$\begin{aligned} |\mathcal{T}_{\text{all}}| &= \sum_{d=1}^{N} |\mathcal{T}_{d-\text{depth}}| \\ &> |\mathcal{T}_{2-\text{depth}}| \\ &= \sum_{j=1}^{N-1} \left(\begin{array}{c} N \\ j \end{array} \right) j^{N-j}. \end{aligned}$$

For example, when $N = 10$, $|\mathcal{T}_{2-\text{depth}}| \approx 2.24 \times 10^6$, therefore it is computationally intractable to consider all trees in $\mathcal{T}_{\text{all}}$ and all scheduling policies in Θ when N is large. We thus propose the following two steps to approximate the optimal solution to the network lifetime maximization problem.

Step 1: Construct an appropriate $\mathcal{T} \subset \mathcal{T}_{\text{all}}$.
Step 2: Schedule $\mathcal{T}$ appropriately.

Step 1 finds a set of trees which satisfy the estimation quality constraint as well as provide different energy costs for each S_i. Step 2 provides the optimal scheduling θ^* by

restricting θ on $\mathcal{T}$. As $\mathcal{T}$ is only a subset of $\mathcal{T}_{\text{all}}$, the resulting θ^* may not be globally optimal in general. In the next few sections, we give details for the above two steps.

5.3 Constructing and Scheduling Sensor Trees

Step 1: Construct Sensor Trees

The proposed Tree Construction Algorithm consists of three main subroutines which are the Random Initialization Algorithm, the Topology Improvement Algorithm and the Tree Reconfiguration Algorithm. The overall algorithm is presented in Figure 5.3.

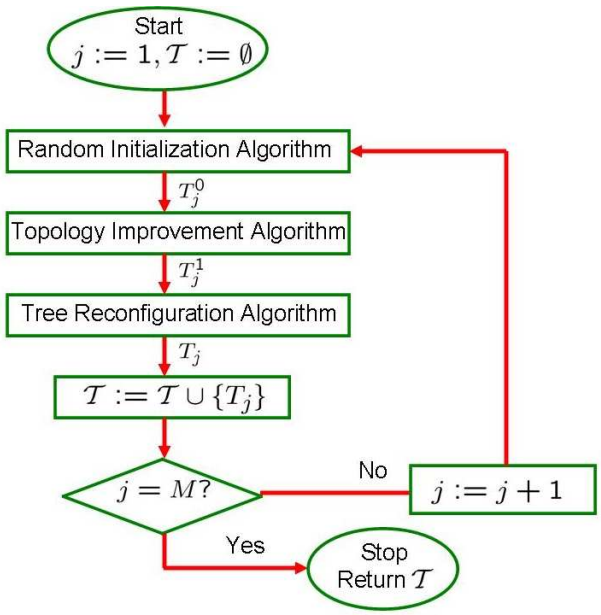

Figure 5.3: Tree Construction Algorithm

Random Initialization Algorithm

Define the following quantities:

$$\begin{aligned} \mathcal{S}_{j-hop} &\triangleq \{S_i : S_i \text{ is } j-\text{hop away from } S_0\}, \\ \mathcal{S}^c(T) &\triangleq \{S_i : S_i \text{ is not in } T\}. \end{aligned}$$

The intuitive idea of the Random Initialization Algorithm is that $\mathcal{S}_{j-hop}, j = 1, \cdots, D$ are randomly determined in sequence until all S_is are included in the tree.

Algorithm 2 RANDOM INITIALIZATION ALGORITHM

$D := 0$
$T := \{S_0, \emptyset\}$
$\forall j\ \mathcal{S}_{j-hop} := \emptyset$
$\mathcal{S}^c = \{S_1, \cdots, S_N\}$
while $(\mathcal{S}^c \neq \emptyset)$ **do**
  $D := D + 1$
  Pick n_D from $(1, |\mathcal{S}^c|)$ uniformly randomly.
  $l := 1$
  while $(l \leq n_D)$ **do**
    Pick any $S_p \in \mathcal{S}^c$ and any $S_q \in \mathcal{S}_{(D-1)-hop}$ uniformly randomly.
    Connect S_p to S_q.
    $\mathcal{S}^c := \mathcal{S}^c \setminus \{S_p\}$
    $T := T \cup \{S_p, (S_p, S_q)\}$
    $\mathcal{S}_{D-hop} := \mathcal{S}_{D-hop} \cup \{S_p\}$
    $l := l + 1$
  end while
end while

After the execution of the Random Initialization Algorithm, an initial tree of depth D is constructed with $|\mathcal{S}_{j-hop}| = n_j, j = 1, \cdots, D$, and $\sum_{j=1}^{D} n_j = N$.

Remark 5.2 If $n_1 = N$, then the algorithm returns $T^\star$, i.e., all sensor nodes connect to S_0 directly.

Topology Improvement Algorithm

Since the previous algorithm randomly constructs the initial tree, some sensor communication paths may be established inefficiently, i.e., some sensors use more energy yet need more hops to communicate with S_0. The Topology Improvement Algorithm aims to remove this inefficiency.

When S_i is connected to S_p, we define τ_{ip} as the number of hops between S_i and S_0, and e_{ip} as the transmission energy cost of S_i. Similarly we define τ_{i0} and e_{i0} for S_i in the initial tree constructed by the Random Initialization Algorithm.

We consider modifying the path of S_i in the initial tree, where $S_i \in \mathcal{S}_{j-hop}, j \geq 2$, only if there exists S_p in the same tree and $S_p \in \mathcal{S}_{j-hop}, j \leq \tau_{i0} - 1$ such that either $e_{ip} < e_{i0}$ or $e_{ip} = e_{i0}$ and $\tau_{ip} < \tau_{i0}$. In these cases, S_i is connected to S_p. The first condition corresponds to reducing the energy cost of S_i yet not making the hops between S_i and S_0 larger; the second condition corresponds to making the hops between S_i and S_0 smaller

yet not increasing its energy cost. Define F_i as the indicator function for S_i, where $F_i = 1$ means that S_i has already been examined for possible improvement and $F_i = 0$ otherwise. The full algorithm is presented below.

Algorithm 3 TOPOLOGY IMPROVEMENT ALGORITHM

```
∀i F_i := 0
∀S_i ∈ S_{j−hop}, j ≤ 1, F_i := 1
while ∃F_i = 0 do
    F_i := 1
    Σ := {S_p : S_p ∈ S_{j−hop}, j ≤ τ_{i0} − 1, e_{ip} ≤ e_{i0}}
    if Σ ≠ ∅ then
        τ_{iq} := min{τ_{ip} : S_p ∈ Σ}
        if e_{iq} < e_{i0} or (e_{iq} = e_{i0} and τ_{iq} < τ_{i0}) then
            reconnect S_i to S_q
            update S_{j−hop}, j ≤ τ_{i0}
        end if
    end if
end while
```

Notice that F_i is set to be 1 for all $S_i \in \mathcal{S}_{j-hop}, j \leq 1$, as for those sensor nodes that are one hop away from S_0, no improvement can be made that further reduces the energy cost (and maintains the same hop numbers) or reduces the hop numbers.

At this step, we have constructed a set of M randomized initial trees. We then use them as input to the Tree Reconfiguration Algorithm presented in Section 4.3 to make sure that each tree provides the desired estimation quality.

Step 2: Schedule Sensor Trees

The Random Initialization Algorithm and Topology Improvement Algorithm aim to create a set of sensor trees $\mathcal{T}$ with different energy cost of individual sensors (due to the randomness). The Tree Reconfiguration Algorithm changes the resulting sensor trees and guarantees that for all $T_j \in \mathcal{T}$,

$$P_\infty(T_j) \leq P_{\text{desired}}.$$

Denote e_{ij} as the total energy cost (transmission and receiving energy) for S_i in T_j, and $t_j(\theta)$ as the time that T_j is used for a policy θ, so the network lifetime $L(\theta)$ is given by

$$L(\theta) = \sum_{j=1}^{M} t_j(\theta).$$

Therefore the network lifetime maximization problem can be written equivalently as

$$\begin{aligned} & \max_{t_j} \sum_{j=1}^{M} t_j \\ \text{subject to} \quad & \\ & \sum_{j=1}^{m} t_j e_{ij} \leq \Pi_i, i = 1, \cdots, N \\ & t_j \geq t_{\min}, j = 1, \cdots, M \end{aligned}$$

where $t_j \geq t_{\min}$ is added to make sure the estimation process enters steady state after some transient period. This problem can be solved efficiently via linear programming, as both the objective function and constraints are linear functions of the variables.

5.4 Examples

In this section, we provide an example to demonstrate the theory and algorithms developed so far. We start by describing the process and sensor models.

Process and Sensor Models

We consider the process in Eqn (3.1) with

$$A = \begin{bmatrix} 1 & 0.1 & 0.05 & 0.0002 \\ 0 & 1 & 0.1 & 0.05 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $Q = 0.1I$. There are three sensors available. The measurement equations are given by

$$\begin{aligned} y_k^1 &= [\ 1 \quad 0 \quad 0 \quad 0\ ]x_k + v_k^1, \\ y_k^2 &= [\ 0 \quad 1 \quad 0 \quad 0\ ]x_k + v_k^2, \\ y_k^3 &= [\ 1 \quad 0 \quad 1 \quad 0\ ]x_k + v_k^3, \end{aligned}$$

with $\Pi_1 = 0.5, \Pi_2 = 0.25$, and $\Pi_3 = 0.1$. Assume the sensors are placed in a line (Figure 5.4) with relative distance

$$d_{10} = 2, d_{21} = 1, d_{32} = 1,$$

where d_{pq} is the distance between S_p and S_q.

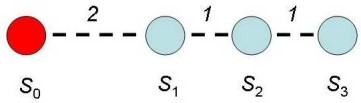

Figure 5.4: Initial Sensor Topology

Let $e_{tx}(S_p, S_q)$ be the energy cost for S_p transmitting a packet to S_q and $e_{rx}(S_p, S_q)$ be the energy cost for S_q receiving such a packet from S_p. We use the following simplified energy model

$$e_{rx}(S_p, S_q) = 1, e_{tx}(S_p, S_q) = d_{pq}^2, \forall 1 \leq p, q \leq 3, p \neq q.$$

Assume the initial energy $\mathcal{E}_i$ available at S_i is known and given by

$$\mathcal{E}_1 = \mathcal{E}_2 = \mathcal{E}_3 = 2000.$$

Tree Construction Algorithm

Let the performance specification at the fusion center be

$$\mathrm{Tr}(P_\infty(T_k)) \leq 2.5 \; \forall \; k.$$

We pick $M = 4$ for this example. Figure (5.5)–(5.8) demonstrate the use of the Tree Construction Algorithm. As a result,

$$\mathcal{T} = \{T_1, T_2, T_3, T_4\}$$

is returned with

$$\mathrm{Tr}(P_\infty(T_1)) = 1.7076, \mathrm{Tr}(P_\infty(T_2)) = 1.8038,$$
$$\mathrm{Tr}(P_\infty(T_3)) = 2.4744, \mathrm{Tr}(P_\infty(T_4)) = 1.6437,$$

and energy cost

$$E = [e_{ij}] = \begin{bmatrix} 4 & 10 & 1 \\ 1 & 11 & 1 \\ 0 & 9 & 16 \\ 5 & 1 & 16 \end{bmatrix}.$$

Notice that during the construction of T_1, only the Switching Tree Topology Algorithm modifies the input tree. For T_2 and T_4, only the Topology Improvement Algorithm modifies the input tree. For T_3, both the Topology Improvement Algorithm and the Minimum Energy Subtree Algorithm modify the input tree.

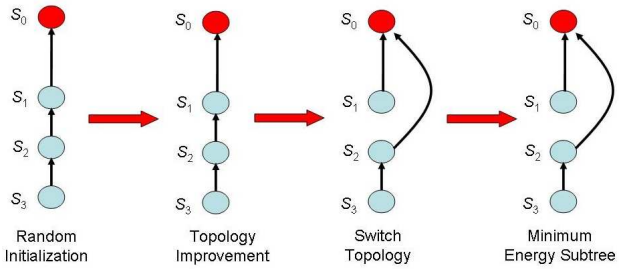

Figure 5.5: Tree 1

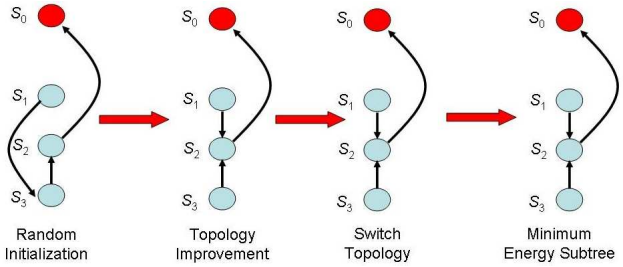

Figure 5.6: Tree 2

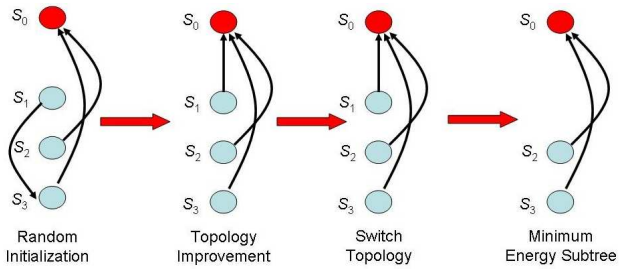

Figure 5.7: Tree 3

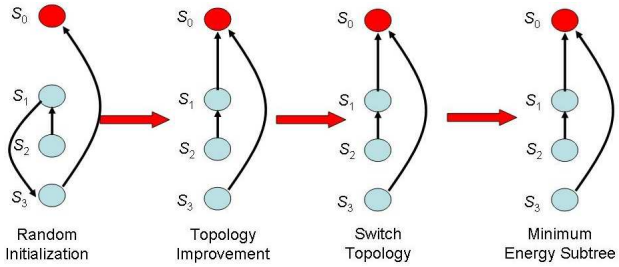

Figure 5.8: Tree 4

Tree Scheduling Algorithm

Using the notations in Section 5.3, let t_j be the time that T_j will be used. Assume $t_{\min} = 20$. In order to maximize the lifetime of the network, we solve the following scheduling problem:

$$\max_{t_j} \sum_{j=1}^{4} t_j$$

subject to

$$\sum_{j=1}^{4} t_j e_{ij} \leq 2000, i = 1, 2, 3$$
$$t_j \geq 20, j = 1, 2, 3, 4.$$

Define the following quantities

$$\begin{aligned} z &\triangleq [t_1 \ t_2 \ t_3 \ t_4]', \\ f &\triangleq [-1 \ -1 \ -1 \ -1]', \\ \Phi &\triangleq \begin{bmatrix} E' \\ -I \end{bmatrix}, \\ b &\triangleq [2000 \ 2000 \ 2000 \ -20 \ -20 \ -20 \ -20 \ ]'. \end{aligned}$$

We then write the above maximization problem in compact linear programming (LP) form as

$$\min_z f'z$$

subject to

$$\Phi z \leq b$$

which can be easily solved via standard LP toolbox. The following optimal value of z is obtained as a result[1]:

$$z^* = [150\ 20\ 20\ 94],$$

and the maximum network lifetime L^* is given by

$$L^* = -f'z^* = 284.$$

Let $L(j)$ denote the lifetime of the network when T_j is always used. It is easy to compute that

$$L(1) = 200, L(2) = 181, L(3) = 222, L(4) = 125,$$

and hence the network lifetime is indeed enhanced by the Tree Scheduling Algorithm.

True and Computed Error Process

In the previous section, we calculate the optimal scheduling policy for $\mathcal{T}$. We then run a Monte Carlo simulation based on the optimal time that T_j is used. Figure 5.9 shows the average P_k and $||e_k||^2$ as a result of 1000 runs. The performance specification is clearly satisfied.

[1]Since t_j can only be an integer, we replace z_i^* as the largest integer κ_i such that $\kappa_i \leq z_i^*$.

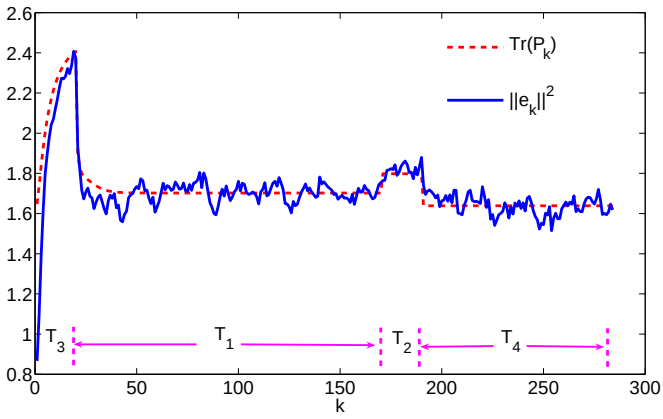

Figure 5.9: P_k and $||e_k||^2$

Chapter 6

Minimizing Buffer Length

6.1 Introduction

The Kalman filter has played an important role in system theory and has found wide applications in many fields, such as control, signal processing, and communications. In the standard Kalman filter, it is assumed that sensor data are transmitted along perfect communication channels and are available to the estimator either instantaneously or with some fixed delays, and no interaction between communication and control is considered. This abstraction has been adopted until recently when networks, especially wireless networks, are used in sensing and control systems for transmitting data from sensors to controller and/or from controller to actuator. While having many advantages such as low cost and flexibility, networks also induce many new issues due to their limited capabilities and uncertainties such as limited bandwidth, packet loss, and delay. On the other hand, in wireless sensor networks, sensor nodes also have limited computation capability in addition to their limitations in communications. These constraints undoubtedly affect system performance or even stability and cannot be neglected when designing estimation and control algorithms, which has partially inspired this thesis work. We devote this chapter to the study of Kalman filtering over a packet delaying network.

The problem of Kalman filtering for systems with delayed measurements is not new and has been studied even before the emergence of networked control [39, 57]. For the problem of randomly delayed measurements, Ray et al. [39] presented a modification of the conventional minimum variance state estimator to accommodate the effects of the random arrival of measurements whereas a suboptimal filter in the least mean square sense is given in [57]. The filtering problems with random delays and missing measurements have been investi-

gated in [40, 48, 51] via the linear matrix inequality and the Riccati equation approaches, respectively. We study the problem from a probabilistic angle, which is different than most existing approaches.

The probability distribution of the delay is assumed to be given and we aim to give a complete characterization of filter performance. Due to the limited computation capability of the filtering center, and also in consideration of the fact that a late-arriving measurement related to the system state in the far past may not contribute much to the improvement of the accuracy of the current estimate, it is practically important to determine a proper buffer length for measurement data within which a measurement will be used to update the current state and beyond which the data will be discarded.

The buffer provides a tradeoff between performance and computational load. In this chapter, for a given buffer length, we give lower and upper bounds for the probability at which the filtering error covariance is within a prescribed bound, i.e., $\mathbf{Pr}[P_k \leq M]$ for some given M. The upper and lower bounds can be easily evaluated by the probability distribution of the delay and the system dynamics. An approach for determining the minimum buffer length for a required performance in probability is given, and an evaluation on the number of expected filter updates is provided. Both the cases of sensor with and without necessary computation capability for filter updates are considered.

6.2 Problem Setup

System Model

We consider the networked control systems shown in Figure 6.1, where a sensor measures the current state of a process and sends the measurement data (or preprocesses the data and sends its local estimate of the state) via a packet delaying network to a remote estimator.

The process dynamics and sensor measurement equation are given as follows:

$$x_k = Ax_{k-1} + w_{k-1}, \tag{6.1}$$

$$y_k = Cx_k + v_k. \tag{6.2}$$

In the above equations, $x_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^m$ is the observation vector, $w_{k-1} \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are zero-mean white Gaussian random vectors with $\mathbb{E}[w_k {w_j}'] =$

$\delta_{kj}Q \geq 0$, $\mathbb{E}[v_k v_j{}'] = \delta_{kj}R > 0$, $\mathbb{E}[w_k v_j{}'] = 0$ $\forall j, k$. The pair (A, C) is assumed to be observable and $(A, \sqrt{Q})$ is controllable.

Depending on its computational capability, the sensor can either send y_k or preprocess y_k and send $\hat{x}_k^s$ to the remote estimator, where $\hat{x}_k^s$ is defined at the sensor as

$$\hat{x}_k^s \triangleq \mathbb{E}[x_k | y_1, \cdots, y_k].$$

The two cases correspond to the two scenarios in Figure 6.1, i.e., sensor without/with computation capability.

Network Delay Model

After taking a measurement at time k, the sensor sends y_k (or $\hat{x}_k^s$) to a remote estimator for computing the state estimate. We assume that the measurement data packets from the sensor are to be sent across a packet delaying network, with negligible quantization effects, to the estimator. Each y_k (or $\hat{x}_k^s$) is delayed by d_k times, where d_k is a random variable described by a probability mass function f, i.e.,

$$f(j) = \mathbf{Pr}[d_k = j], j = 0, 1, \cdots . \tag{6.3}$$

For simplicity, we assume d_{k_1} and d_{k_2} are independent if $k_1 \neq k_2$. Notice that the i.i.d packet drop with drop rate $1 - \gamma$ considered in the literature can be treated as a special case here, i.e.,

$$f(0) = \gamma, f(\infty) = 1 - \gamma, f(j) = 0, 1 \leq j < \infty.$$

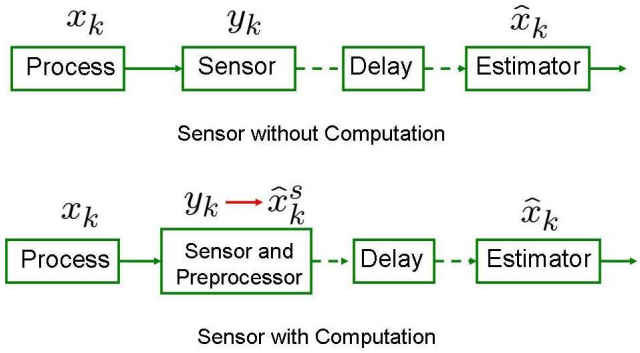

Figure 6.1: System Block Diagram

Thus the theory developed in the paper includes the packet drop analysis as well, e.g., partial result in [14, 41, 42] can be considered as a special case here.

Problems of Interest

Define the following state estimate and other quantities at the remote estimator:

$$\begin{aligned}
\hat{x}_k^- &\triangleq \mathbb{E}[x_k|\text{all data packets up to } k-1], \\
\hat{x}_k &\triangleq \mathbb{E}[x_k|\text{all data packets up to } k], \\
P_k^- &\triangleq \mathbb{E}[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)'], \\
P_k &\triangleq \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)'].
\end{aligned}$$

Assume the estimator discards any data y_k (or $\hat{x}_k^s$) that are delayed by D times or more. Given the system and the network delay models in Eqn (6.1)–(6.3), we are interested in the following problems:

1. How should $\hat{x}_k$ be computed?
2. What is the relationship between P_k and D?
3. For a given $M \geq 0$ and $\epsilon \in [0, 1]$, what is the minimum D such that

$$\mathbf{Pr}[P_k \leq M|D] \geq 1 - \epsilon.$$

In the rest of the chapter, we will provide solutions to the above three problems for each of the two scenarios in Figure 6.1.

6.3 Sensor without Computation Capability

In this section, we consider the first scenario in Figure 6.1, i.e., the sensor has no computation and sends y_k to the remote estimator. Assume C is full rank and initially assume C^{-1} exists. The general C case will be considered in Section 6.4.

Let γ_t^k be the indicator functor for y_t at time $k, t \leq k$, which is defined as follows:

$$\gamma_t^k = \begin{cases} 1, y_t \text{ received at time } k, \\ 0, \text{otherwise.} \end{cases}$$

Further define $\gamma_{k-i} \triangleq \sum_{j=0}^{i} \gamma_{k-i}^{k-j}$, i.e., γ_{k-i} indicates whether y_{k-i} is received by the estimator at or before k.

Optimal Estimation with Delayed Measurements

As y_{k-i} may arrive at time k due to the delays introduced by the network, we can improve the estimation quality by recalculating $\hat{x}_{k-i}$ utilizing the new available measurement y_{k-i} (similar to the algorithm presented in Theorem 3.1). Once $\hat{x}_{k-i}$ is updated, we can update $\hat{x}_{k-i+1}$ in a similar fashion. This is the basic idea contained in the flow diagram in Figure 6.2. Theorem 6.1 summarizes the main estimation result.[1]

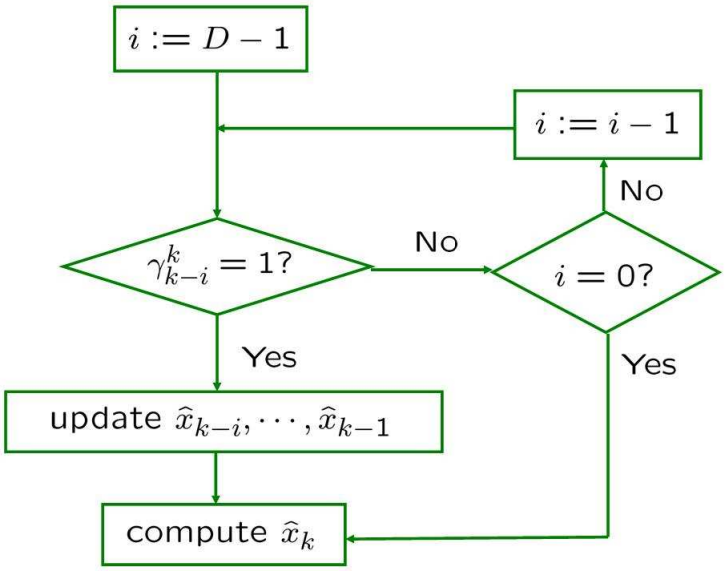

Figure 6.2: Optimal Estimation: Sensor without Computation Capability

[1] If A^{-1} exists, one may also use the out-of-sequence-measurement algorithm to include the delayed packets for estimating $\hat{x}_k$, i.e., by rewriting $y_{k-i} = CA^{-i}x_k + \tilde{v}_k$. By stacking all the measurements together, only one **MKF** is needed at each time.

Theorem 6.1 *Let $y_{k-i}, i \in [0, D-1]$ be the oldest measurement received by the estimator at time k. Then $\hat{x}_k$ is computed by $i+1$ modified Kalman filters as*

$$\begin{aligned}
(\hat{x}_{k-i}, P_{k-i}) &= \boldsymbol{MKF}(\hat{x}_{k-i-1}, P_{k-i-1}, y_{k-i}) \\
(\hat{x}_{k-i+1}, P_{k-i+1}) &= \boldsymbol{MKF}(\hat{x}_{k-i}, P_{k-i}, \gamma_{k-i+1} y_{k-i+1}) \\
&\vdots \\
(\hat{x}_{k-1}, P_{k-1}) &= \boldsymbol{MKF}(\hat{x}_{k-2}, P_{k-2}, \gamma_{k-1} y_{k-1}) \\
(\hat{x}_k, P_k) &= \boldsymbol{MKF}(\hat{x}_{k-1}, P_{k-1}, \gamma_k y_k).
\end{aligned}$$

*Furthermore, $\hat{D}$, the average number of **MKF** used at each time k is given by*

$$\hat{D} = \prod_{i=1}^{D-1} \big(1 - f(i)\big) + \sum_{j=2}^{D} \prod_{i=j}^{D-1} \big(1 - f(i)\big) f(j-1) j + f(D-1) D. \tag{6.4}$$

Proof: We know that the estimate $\hat{x}_k$ is generated from the estimate of $\hat{x}_{k-1}$ together with $\gamma_k y_k$ at time k through a modified Kalman filter. Similarly, the estimate $\hat{x}_{k-1}$ is generated from the estimate of $\hat{x}_{k-2}$ together with $\gamma_{k-1} y_{k-1}$ at time k through a modified Kalman filter, etc. This recursion for $i+1$ steps corresponds to the $i+1$ modified Kalman filters stated in the theorem. Let $\hat{D}_k$ be the number of **MKF** used at each time k. Notice that $1 \le \hat{D}_k \le D$. Thus

$$\hat{D} = \sum_{j=1}^{D} j \mathbf{Pr}[\hat{D}_k = j].$$

Consider $\mathbf{Pr}[\hat{D}_k = 1]$. Since $\hat{D}_k = 1$ iff $\gamma_{k-i}^k = 0$ for all $1 \le i \le D-1$, we have

$$\mathbf{Pr}[\hat{D}_k = 1] = \mathbf{Pr}[\gamma_{k-i}^k = 0, 1 \le i \le D-1].$$

As $\mathbf{Pr}[\gamma_{k-i}^k = 0] = 1 - f(i)$, we obtain

$$\mathbf{Pr}[\hat{D}_k = 1] = \prod_{i=1}^{D-1} \big(1 - f(i)\big).$$

Similarly, for $2 \le j \le D-1$, we have

$$\mathbf{Pr}[\hat{D}_k = j] = \prod_{i=j}^{D-1} \big(1 - f(i)\big) f(j-1)$$

and when $j = D$,

$$\mathbf{Pr}[\hat{D}_k = j] = f(D-1).$$

Therefore we obtain $\hat{D}$ in Eqn (6.4). ■

Remark 6.2 Notice that in the first **MKF**, $\gamma_{k-i}^k = 1$ and hence $\gamma_{k-i} = 1$. As a result, we simply write $\gamma_{k-i} y_{k-i} = y_{k-i}$.

Lower and Upper Bounds of $\mathbf{Pr}[P_k \leq M|D]$

Since d_k is random and described by the probability mass function f, γ_{k-i}^k $(i = 0, \cdots, D-1)$ is also random. As a consequence, P_k computed as in Theorem 6.1 is a random variable. Define $\hat{\gamma}_i(D)$ as

$$\hat{\gamma}_i(D) \triangleq \begin{cases} \sum_{j=0}^{i} f(j), \text{if } 0 \leq i < D, \\ \sum_{j=0}^{D-1} f(j), \text{if } i \geq D. \end{cases}$$

Recall that γ_{k-i} indicates whether y_{k-i} is received by the estimator at or before k, so it is easy to verify that

$$\mathbf{Pr}[\gamma_{k-i} = 1|D] = \hat{\gamma}_i(D). \tag{6.5}$$

Define $\overline{M} \triangleq C^{-1}RC^{-1'}$. Then we have the following result that shows the relationship between P_k and $\overline{M}$.

Lemma 6.3 *For any $k \geq 1$, if $\gamma_k = 1$, then $P_k \leq \overline{M}$.*

Proof: As $\gamma_k = 1$, we have $P_k = \tilde{g} \circ h(P_{k-1}) \leq \overline{M}$, where the inequality is from Lemma 2.3. ■

Remark 6.4 We can also interpret Lemma 6.3 as follows. One way to obtain an estimate $\tilde{x}_k$ when $\gamma_k = 1$ is simply by inverting the measurement, i.e., $\tilde{x}_k = C^{-1}y_k$. Therefore

$$\tilde{e}_k = C^{-1}v_k \text{ and } \tilde{P}_k = \mathbb{E}[\tilde{e}_k \tilde{e}_k^{'}] = C^{-1}RC^{-1'} = \overline{M}.$$

Since Kalman filter is optimal among the set of all linear filters, we conclude that $P_k \leq \tilde{P}_k = \overline{M}$.

We defined P^* and $\overline{P}$ earlier in Chapter 2 as the a priori and a posteriori steady state error covariance when the network is perfect, i.e., the estimator has y_k at each time k, and we have shown in Lemma 2.1 that P^* and $\overline{P}$ satisfy

$$\begin{aligned} P^* &= g(P^*), \\ \overline{P} &= \tilde{g}(P^*), \\ \overline{P} &= \tilde{g} \circ h(\overline{P}). \end{aligned}$$

For $M \geq \overline{M}$, define $k_1(M)$ and $k_2(M)$ as follows:

$$k_1(M) \triangleq \min\{t \geq 1 : h^t(\overline{M}) \not\leq M\}, \tag{6.6}$$

$$k_2(M) \triangleq \min\{t \geq 1 : h^t(\overline{P}) \not\leq M\}. \tag{6.7}$$

We sometimes write $k_i(M)$ as $k_i, i = 1, 2$ for simplicity. The following lemma shows the relationship between $\overline{P}$ and $\overline{M}$ as well as k_1 and k_2.

Lemma 6.5 *(1) $\overline{P} \leq \overline{M}$; (2) $k_1 \leq k_2$ whenever either k_i is finite, $i = 1, 2$.*

Proof: (1) $\overline{P} = \tilde{g}(P^*) \leq \overline{M}$ where the inequality is from Lemma 2.3. (2) Without loss of generality, we assume k_2 is finite. If k_1 is finite, and $k_1 > k_2$, then according to their definitions, we must have

$$M \geq h^{k_1-1}(\overline{M}) \geq h^{k_1-1}(\overline{P}) \geq h^{k_2}(\overline{P}),$$

which violates the definition of k_2. Notice that we use the property that h is nondecreasing as well as $h(\overline{P}) \geq \overline{P}$ from Lemma 2.2 and 2.4. Similarly we can show that k_1 cannot be infinite. Therefore we must have $k_1 \leq k_2$. ■

The following lemma will be used to establish the lower bound of $\mathbf{Pr}[P_k \leq M|D]$.

Lemma 6.6 *Assume $P_0 \geq \overline{P}$. Then for all $k \geq 0$,*

$$P_k \geq \overline{P}.$$

Proof: Since **MKF** is used at each time k,

$$P_k = \hat{f}_k^k \circ \hat{f}_{k-1}^k \circ \cdots \circ \hat{f}_1^k(P_0) \geq \overline{P},$$

where $\hat{f}_{k-i}^k = h$ or $\hat{f}_{k-i}^k = \tilde{g} \circ h$ depending on the packet arrival sequence[2]. The inequality is from Lemma 2.2. ■

Define N_k as the number of consecutive packets not received by k, i.e.,

$$N_k \triangleq \min\{t \geq 0 : \gamma_{k-t} = 1\}. \tag{6.8}$$

Further define

$$\theta(k_i, D) \triangleq \prod_{j=0}^{k_i-1} \big(1 - \hat{\gamma}_j(D)\big). \tag{6.9}$$

It is easy to see that

$$\theta(k_1, D) \geq \theta(k_2, D).$$

Lemma 6.7 *Let k_1, k_2 and N_k be defined according to Eqn (6.6)–(6.8). Then*

$$\mathbf{Pr}[N_k \geq k_i | D] = \theta(k_i, D), i = 1, 2. \tag{6.10}$$

Proof:

$$\mathbf{Pr}[N_k \geq k_i | D] = \mathbf{Pr}[\gamma_{k-i} = 0, 0 \leq i \leq k_i - 1 | D] = \theta(k_i, D).$$

■

With all the supporting lemmas stated so far, we have the following main result of this chapter, which gives lower and upper bounds of $\mathbf{Pr}[P_k \leq M|D]$.

Theorem 6.8 *Assume $\overline{P} \leq P_0 \leq \overline{M}$. For any $M \geq \overline{M}$, we have*

$$1 - \theta(k_1, D) \leq \mathbf{Pr}[P_k \leq M|D] \leq 1 - \theta(k_2, D). \tag{6.11}$$

Proof: We divide the proof into two parts. For the remaining of the proof, all probabilities are conditioned on the given D. (1) Let us first prove $1 - \theta(k_1, D) \leq \mathbf{Pr}[P_k \leq M|D]$, or in

[2]Notice that we use the superscript k in $\hat{f}_{k-i}^k$ to emphasize that it depends on the current time k. For example, if $d_{k-i} = i + 1$, i.e., $\gamma_{k-i} = 0$ and $\gamma_{k-i}^{k+1} = 1$, then $\hat{f}_{k-i}^k = h$ and $\hat{f}_{k-i}^{k+1} = \tilde{g} \circ h$.

other words,

$$1 - \mathbf{Pr}[N_k \geq k_1 | D] \leq \mathbf{Pr}[P_k \leq M | D].$$

As $\gamma_k = 1$ or 0, there are in total 2^k possible realizations of γ_1 to γ_k as seen from Figure 6.3.

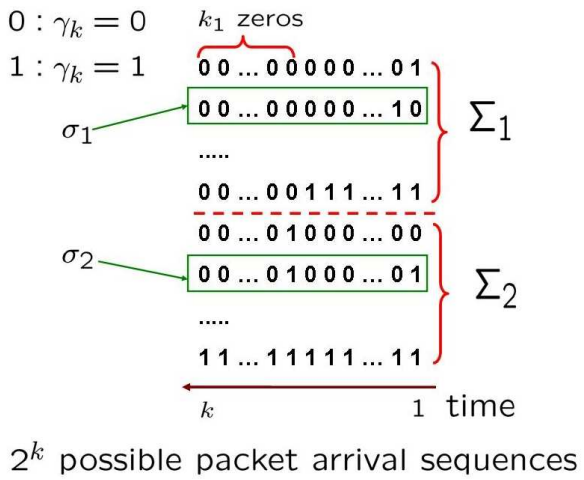

Figure 6.3: $N_k \geq k_1$

Let Σ_1 denote those packet arrival sequences of γ_1 to γ_k such that $N_k \geq k_1$. Similarly let Σ_2 denote those packet arrival sequences such that $N_k < k_1$. Let $P_k(\sigma_i)$ be the error covariance at time k when the underlying packet arrival sequence is σ_i, where $\sigma_i \in \Sigma_i, i = 1, 2$. Consider a particular $\sigma_2 \in \Sigma_2$. As $\gamma_{k-k_1+1} = 1$, from Lemma 6.3, $P_{k-k_1+1} \leq \overline{M}$. Therefore we have

$$P_k(\sigma_2) \leq h^{k_1-1}(P_{k-k_1+1}) \leq h^{k_1-1}(\overline{M}) \leq M,$$

where the first and second inequalities are from Lemma 2.2 and the last inequality is from the definition of k_1. In other words,

$$\mathbf{Pr}[P_k \leq M | \sigma_2] = 1.$$

Therefore we have

$$\begin{aligned}
\mathbf{Pr}[P_k \le M] &= \sum_{\sigma \in \Sigma_1 \cup \Sigma_2} \mathbf{Pr}[P_k \le M|\sigma]\mathbf{Pr}(\sigma) \\
&= \sum_{\sigma_1 \in \Sigma_1} \mathbf{Pr}[P_k \le M|\sigma_1]\mathbf{Pr}(\sigma_1) + \sum_{\sigma_2 \in \Sigma_2} \mathbf{Pr}[P_k \le M|\sigma_2]\mathbf{Pr}(\sigma_2) \\
&\ge \sum_{\sigma_2 \in \Sigma_2} \mathbf{Pr}[P_k \le M|\sigma_2]\mathbf{Pr}(\sigma_2) \\
&= \sum_{\sigma_2 \in \Sigma_2} \mathbf{Pr}(\sigma_2) = \mathbf{Pr}(\Sigma_2) = 1 - \mathbf{Pr}(\Sigma_1) = 1 - \mathbf{Pr}[N_k \ge k_1],
\end{aligned}$$

where the first equality is from the total probability theorem, the second equality holds as Σ_1 and Σ_2 are disjoint, and the third inequality holds as the first sum is non-negative. The remaining equalities are easy to see.

(2) We now prove $\mathbf{Pr}[P_k \le M|D] \le 1 - \theta(k_2, D)$, or in other words

$$\mathbf{Pr}[P_k \le M|D] \le 1 - \mathbf{Pr}[N_k \ge k_2|D].$$

Let Σ_1' denote those packet arrival sequences of γ_1 to γ_k such that $N_k \ge k_2$, and Σ_2' denote those packet arrival sequences such that $N_k < k_2$ (Figure 6.4).

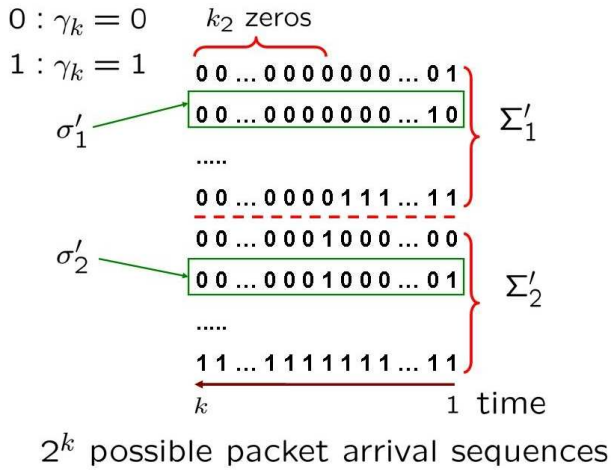

Figure 6.4: $N_k \ge k_2$

Consider $\sigma_1' \in \Sigma_1'$. Let

$$s(\sigma_1') = \min\{t \ge 1 : \gamma_{k-t} = 1|\sigma_1'\}.$$

As $\sigma'_1 \in \Sigma'_1$, we must have $s \geq k_2$. Consequently,

$$P_k(\sigma'_1) = h^{s(\sigma'_1)}(P_{k-s(\sigma'_1)}) \geq h^{s(\sigma'_1)}(\overline{P}),$$

where the inequality is from Lemma 6.6. Therefore we conclude that

$$P_k(\sigma'_1) \not\leq M.$$

Otherwise

$$h^{s(\sigma'_1)}(\overline{P}) \leq P_k(\sigma'_1) \leq M,$$

which violates the definition of k_2. In other words,

$$\mathbf{Pr}[P_k \leq M|\sigma'_1] = 0.$$

Therefore we have

$$\begin{aligned}
\mathbf{Pr}[P_k \leq M] &= \sum_{\sigma \in \Sigma_1 \cup \Sigma_2} \mathbf{Pr}[P_k \leq M|\sigma]\mathbf{Pr}(\sigma) \\
&= \sum_{\sigma'_1 \in \Sigma'_1} \mathbf{Pr}[P_k \leq M|\sigma'_1]\mathbf{Pr}(\sigma'_1) + \sum_{\sigma'_2 \in \Sigma'_2} \mathbf{Pr}[P_k \leq M|\sigma'_2]\mathbf{Pr}(\sigma'_2) \\
&= \sum_{\sigma'_2 \in \Sigma'_2} \mathbf{Pr}[P_k \leq M|\sigma'_2]\mathbf{Pr}(\sigma'_2) \\
&\leq \sum_{\sigma'_2 \in \Sigma'_2} \mathbf{Pr}(\sigma'_2) = \mathbf{Pr}(\Sigma'_2) = 1 - \mathbf{Pr}(\Sigma'_1) = 1 - \mathbf{Pr}[N_k \geq k_2],
\end{aligned}$$

where the inequality is from the fact that

$$\mathbf{Pr}[P_k \leq M|\sigma'_2] \leq 1 \text{ for any } \sigma'_2 \in \Sigma'_2.$$

■

Computing the Minimum D

Assume we require that

$$\mathbf{Pr}[P_k \leq M|D] \geq 1 - \epsilon, \tag{6.12}$$

then according to Eqn (6.11), a sufficient condition is that

$$\theta(k_1, D) \leq \epsilon. \tag{6.13}$$

And a necessary condition is that

$$\theta(k_2, D) \leq \epsilon. \tag{6.14}$$

For a given M, define

$$\epsilon_1(M) \triangleq \theta(k_1, k_1 - 1), \tag{6.15}$$

$$\epsilon_2(M) \triangleq \theta(k_2, k_2 - 1). \tag{6.16}$$

Sufficient Minimum D

Notice that $\theta(k_1, D)$ is decreasing when $1 \leq D \leq k_1 - 1$ and remains constant when $D \geq k_1$. Hence if $\epsilon < \epsilon_1(M)$, no matter how large D is, there is *no guarantee* that $\mathbf{Pr}[P_k \leq M|D] \geq 1 - \epsilon$. If $\epsilon \geq \epsilon_1(M)$, then the minimum D_s which *guarantees* $\mathbf{Pr}[P_k \leq M|D] \geq 1 - \epsilon$ is given by

$$D_s = \min\{D : \theta(k_1, D) \leq \epsilon, 1 \leq D \leq k_1 - 1\}. \tag{6.17}$$

Necessary Minimum D

Similarly, $\theta(k_2, D)$ is decreasing when $1 \leq D \leq k_2 - 1$ and remains constant when $D \geq k_2$. Hence if $\epsilon < \epsilon_2(M)$, no matter how large D is, it is guaranteed that $\mathbf{Pr}[P_k \leq M|D] < 1 - \epsilon$. If $\epsilon \geq \epsilon_1(M)$, then the minimum D_n such that it is *possible* that $\mathbf{Pr}[P_k \leq M|D] \geq 1 - \epsilon$ is given by

$$D_n = \min\{D : \theta(k_2, D) \leq \epsilon, 1 \leq D \leq k_2 - 1\}. \tag{6.18}$$

Example 6.9 Consider Eqn (6.1) and (6.2) with

$$A = 1.4, C = 1, Q = 0.2, R = 0.5.$$

We model the packet delay using a poisson distribution with mean d, i.e., the probability mass function $f(i)$ is given by

$$f(i) = \frac{d^i e^{-d}}{i!}, i = 0, 1, \cdots$$

where $d = \mathbb{E}[d_k]$ denotes the mean value of the packet delay.

When $M = 50$, it is calculated that $k_1(M) = k_2(M) = 7$, hence $\theta(k_1, D) = \theta(k_2, D)$ and $\theta(7, 6) = 0.0313$. Thus we can find the minimum D that guarantees $\mathbf{Pr}[P_k \leq 50] \geq 1 - \epsilon$ for any $\epsilon \geq 0.0313$. For any $\epsilon < 0.0313$, no matter how large D is, $\mathbf{Pr}[P_k \leq 50|D] < 1 - \epsilon$.

When $M = 150$, it is calculated that $k_1(M) = 8$ and $k_2(M) = 9$, hence $\theta(k_1, D) > \theta(k_2, D)$. We also find that $\theta(8, 7) = 0.0042$ and $\theta(9, 8) = 0.0003$. Therefore if $\epsilon > 0.0042$, we can find minimum D that guarantees $\mathbf{Pr}[P_k \leq 150] \geq 1 - \epsilon$; if $\epsilon < 0.0003$, no matter how large D is, $\mathbf{Pr}[P_k \leq 150|D] < 1 - \epsilon$.

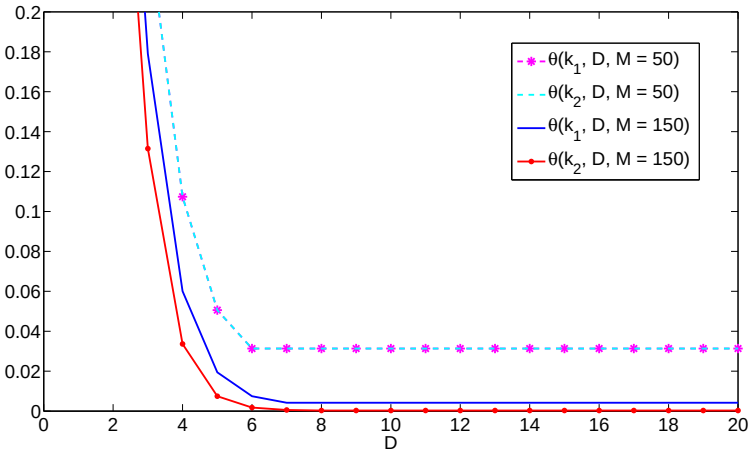

Figure 6.5: $\theta(k_i, D)$ for Different M

6.4 Sensor with Computation Capability

In this section, we consider the second scenario in Figure 6.1, i.e., the sensor has necessary computation capability and sends $\hat{x}_k^s$ to the remote estimator. All variables in this section, e.g., γ_t^k, γ_k, etc, are assumed to be the same as in Section 6.3 unless they are explicitly defined.

Consider the case when the Kalman filter enters steady state at the sensor side, i.e., $P_k^s = \overline{P}$. It is clear that the optimal estimation at the remote estimator is as follows: If $\gamma_k = 1$, then $\hat{x}_k = \hat{x}_k^s$ and $P_k = P_k^s = \overline{P}$. If $\gamma_k = 0$ and $\gamma_{k-1} = 1$, then $\hat{x}_k = A\hat{x}_k^s$ and $P_k = h(\overline{P})$. This is repeated until γ_{k-D+1} is examined. The full optimal estimation algorithm is presented in Figure 6.6.

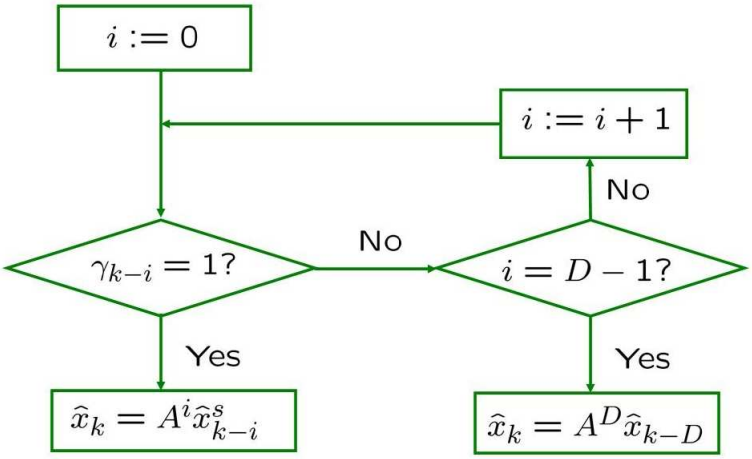

Figure 6.6: Optimal Estimation: Sensor with Computation Capability.

Notice that in the first scenario (Figure 6.2), i.e., sensor has no computation capability, we examine the sequence from γ_{k-D+1}^k to γ_k^k, while in the the second scenario, (Figure 6.6), we examine the sequence from γ_k to γ_{k-D+1}.

Different than in Theorem 6.8, where we give lower and upper bounds of $\mathbf{Pr}[P_k \leq M|D]$, with sensor having computation capability, we can give the exact expression of $\mathbf{Pr}[P_k \leq M|D]$. The closed form expression is given in the following theorem.

Theorem 6.10 *Assume the Kalman filter enters steady state at the sensor side so that* $P_k^s = \overline{P}$. *Then for any* $M \geq \overline{P}$, *we have*

$$\mathbf{Pr}[P_k \leq M|D] = 1 - \theta(k_2, D). \tag{6.19}$$

Proof: For the remaining proof, all probabilities are conditioned on D. Let σ_i' and $\Sigma_i', i = 1, 2$ be defined in the same way as in the proof of Theorem 6.8 (see Figure 6.4). Clearly for any $\sigma_2' \in \Sigma_2'$,

$$P_k(\sigma_2') \leq h^{k_2-1}(\overline{P}) \leq M.$$

The first inequality is from the fact that $\gamma_{k-k_2+1} = 1$ and hence $P_{k-k_2+1} = \overline{P}$. The second inequality is from the definition of k_2. In other words,

$$\mathbf{Pr}[P_k \leq M | \sigma_2'] = 1.$$

Similar to the proof of Theorem 6.8 , for $\sigma_1' \in \Sigma_1'$, let us define

$$s = s(\sigma_1') \triangleq \min\{t \geq 1 : \gamma_{k-t} = 1 | \sigma_1'\}.$$

As $\sigma_1' \in \Sigma_1'$, $s \geq k_2$. Therefore

$$P_k(\sigma_1') = h^s(\overline{P}) \not\leq M.$$

In other words,

$$\mathbf{Pr}[P_k \leq M | \sigma_1'] = 0.$$

Therefore

$$\begin{aligned}
\mathbf{Pr}[P_k \leq M] &= \sum_{\sigma' \in \Sigma_1' \cup \Sigma_2'} \mathbf{Pr}[P_k \leq M | \sigma'] \mathbf{Pr}(\sigma') \\
&= \sum_{\sigma_1' \in \Sigma_1'} \mathbf{Pr}[P_k \leq M | \sigma_1'] \mathbf{Pr}(\sigma_1') + \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}[P_k \leq M | \sigma_2'] \mathbf{Pr}(\sigma_2') \\
&= \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}[P_k \leq M | \sigma_2'] \mathbf{Pr}(\sigma_2') \\
&= \sum_{\sigma_2' \in \Sigma_2'} \mathbf{Pr}(\sigma_2') = \mathbf{Pr}(\Sigma_2') = 1 - \mathbf{Pr}(\Sigma_1') = 1 - \mathbf{Pr}[N_k \geq k_2].
\end{aligned}$$

■

Computing $\mathbf{Pr}[N_k \geq k_2 | D]$ follows exactly the same way as in Section 6.3. Since we have a strict equality in Eqn (6.19), in order that

$$\mathbf{Pr}[P_k \leq M | D] \geq 1 - \epsilon,$$

for a given ϵ, a necessary and sufficient condition is that

$$\mathbf{Pr}[N_k \geq k_2] \leq \epsilon. \tag{6.20}$$

Therefore the minimum D^* that guarantees Eqn (6.20) to hold is given by

$$D^* = \min\{D : \theta(k_2, D) \leq \epsilon, 1 \leq D \leq k_2 - 1\}. \tag{6.21}$$

Notice that since $\theta(k_2, D) \geq \theta(k_2, k_2 - 1) = \epsilon_2(M)$, D^* from the above equation exists if and only if $\epsilon \geq \epsilon_2(M)$ where $\epsilon_2(M)$ is defined in Eqn (6.16).

When C Is Not Full Rank

We use Theorem 6.10 to tackle the case when C is not full rank for the first scenario, i.e., sensor without computation capability. Since (A, C) is observable, there exists r $(2 \leq r \leq n)$ such that

$$\begin{bmatrix} C \\ CA \\ \cdots \\ CA^{r-1} \end{bmatrix}$$

is full rank. In this section, we consider the special case when $r = 2$, and in particular, we assume $\begin{bmatrix} C \\ CA \end{bmatrix}^{-1}$ exists. The idea readily extends to the general case.

Unlike the case when C^{-1} exists, and y_k is sent across the network, here we assume that the previous measurement y_{k-1} is sent along with y_k. This only requires that the sensor has a buffer that stores y_{k-1}. Then if $\gamma_k = 1$, both y_k and y_{k-1} are received. Thus we can use the following linear estimator to generate $\hat{x}_k$

$$\hat{x}_k \;=\; A \begin{bmatrix} CA \\ C \end{bmatrix}^{-1} \begin{bmatrix} y_k \\ y_{k-1} \end{bmatrix}.$$

The corresponding error covariance can be calculated as

$$P_k \;=\; AM_1A' + Q,$$

where

$$M_1 = \begin{bmatrix} CA \\ C \end{bmatrix}^{-1} \begin{bmatrix} CQC' + R & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} CA \\ C \end{bmatrix}^{-1'}.$$

Therefore once the packet for time k is received, i.e., $\gamma_k = 1$, we have

$$P_k = AM_1A' + Q \triangleq \tilde{P}.$$

Now if we treat $\tilde{P}$ as the steady-state error covariance at the sensor side, i.e., by letting $P_k^s = \tilde{P}$, and define

$$k_v \triangleq \min\{t \geq 1 : h^t(\tilde{P}) \not\leq M\},$$

we immediately obtain

$$\mathbf{Pr}[P_k \leq M|D] = 1 - \theta(k_v, D). \tag{6.22}$$

Remark 6.11 Though we give the exact expression of $\mathbf{Pr}[P_k \leq M|D]$ in Eqn (6.22), we have to point out that $\theta(k_v, D) \geq \theta(k_2, D)$, as $\tilde{P} \geq \overline{P}$ due to the optimality of Kalman filter. Thus the case that sensor has computation capability leads to better filter performance, which is illustrated from the vector system example in the next section.

6.5 Examples

Scalar System

Consider the same parameters as in Example 6.9, i.e.,

$$A = 1.4, C = 1, Q = 0.2, R = 0.5$$

and

$$f(i) = \frac{d^i e^{-d}}{i!}, i = 0, 1, \cdots$$

Figure 6.7 shows the values of $f(i)$ for $0 \leq i \leq 20$ for $d = 3$ and 5, respectively.

Sensor without Computation Capability

We run a Monte Carlo simulation for different parameters. Figure 6.8 to 6.10 show the results when D and d take different values. From Figure 6.11, we can see that both smaller d and larger D lead to larger $\mathbf{Pr}[P_k \leq M|D]$, which confirms the theory developed in this chapter. We also notice that when $d = 3$, the filter's performances using $D = 10$ and $D = 5$

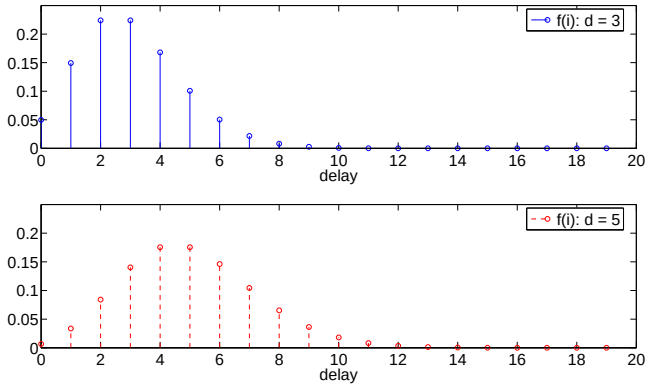

Figure 6.7: Poisson Distribution with $d = 3$ and $d = 5$

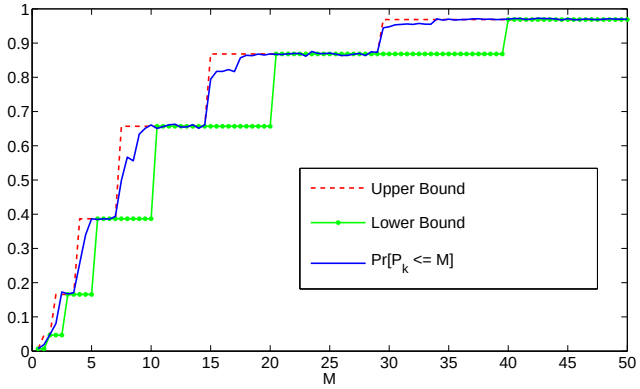

Figure 6.8: $\mathbf{Pr}[P_k \leq M|D = 10], d = 5$

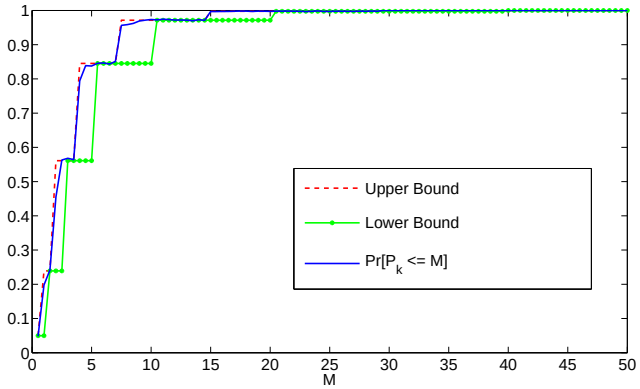

Figure 6.9: $\mathbf{Pr}[P_k \leq M|D = 10], d = 3$

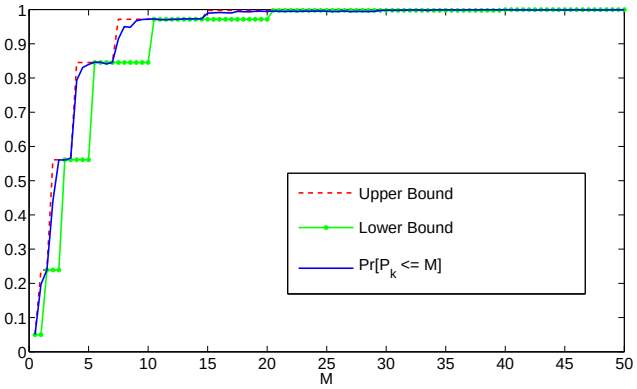

Figure 6.10: $\mathbf{Pr}[P_k \leq M|D = 5], d = 3$

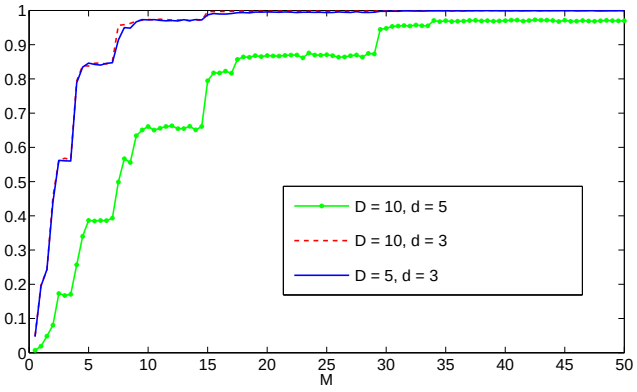

Figure 6.11: Comparison of the Three Simulations

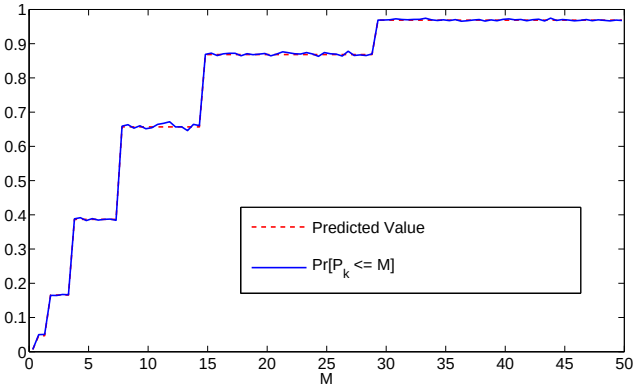

Figure 6.12: $\mathbf{Pr}[P_k \leq M|D = 10], d = 5$

only differ slightly (though the former one is better than the latter one), which confirms that using a large buffer may not improve the filter performance drastically.

Sensor with Computation Capability

We run a Monte Carlo simulation for the case when the sensor has computation capability. Figure 6.12 shows the result when $D = 10$ and $d = 5$. As we can see, the predicted value of $\mathbf{Pr}[P_k \leq M|D]$ from Eqn (6.19) matches well with the actual value.

Vector System

Consider a vehicle moving in a two dimensional space according to the standard constant acceleration model, which assumes that the vehicle has zero acceleration except for a small perturbation. The state of the vehicle consists of its x and y positions as well as velocities. Assume a sensor measures the positions of the vehicle and sends the measurements to a remote estimator over a packet delaying network. The system parameters are given according to Eqn (6.1)–(6.2) as follows:

$$A = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The process and measurement noise covariances are $Q = \mathbf{diag}(0.01, 0.01, 0.01, 0.01)$ and $R = \mathbf{diag}(0.001, 0.001)$. We assume the same delay profile as in the scalar system example with $D = 5$ and $d = 3$.

We run a Monte Carlo simulation for both cases when the sensor has or has not computation capability. As we can see from Figure 6.13, the predicted values of $\mathbf{Pr}[P_k \leq M|D]$ from Eqn (6.19) and Eqn (6.22) match well with the actual values. We also notice that when sensor has computation capability, the actual filter performance is better than when sensor has no computation capability, as stated in Remark 6.11. In Figure 6.13, the M in the x-axis means $M \times I_4$, where I_4 is the identity matrix of dimension 4.

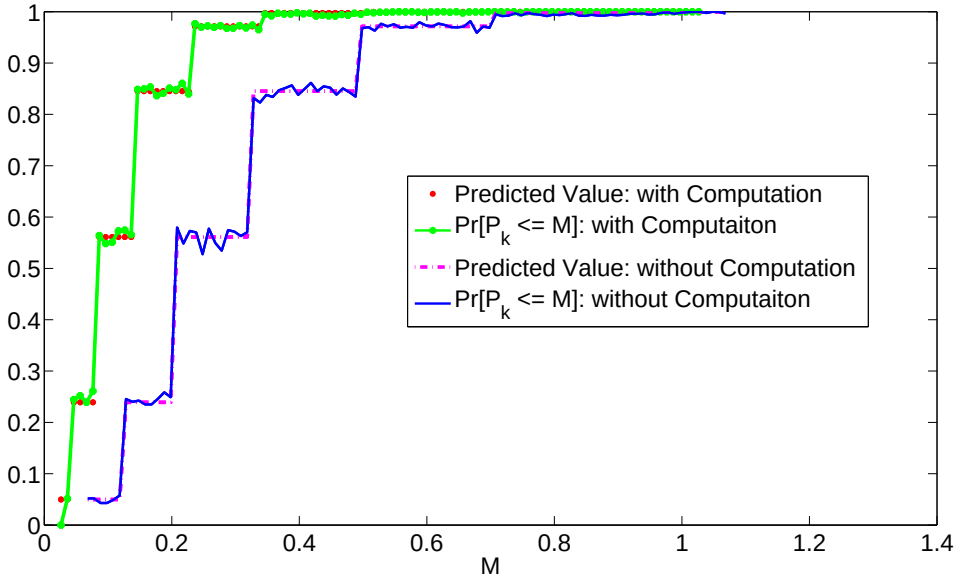

Figure 6.13: $\mathbf{Pr}[P_k \leq M | D = 5], d = 3$

Chapter 7

Conclusions and Future Work

7.1 Concluding Remarks

In this thesis I consider the problem of state estimation over a (sensor) network. With optimizing the network resources being the objective, I study the following three problems.

1. How we can minimize the sensor energy usage?
2. How we can maximize the network lifetime?
3. How we can minimize the buffer length at the fusion center?

To answer those questions, I first study the problem of optimal estimation over a static sensor tree in Chapter 3. I show that the optimal estimator is a chain of Kalman filters and the length of the chain corresponds to the depth of the tree. Closed-form expression on the steady-state error covariance is also obtained, which tells us that how each sensor contributes to the overall estimation.

In Chapter 4, I look at the sensor energy minimization problem, and I propose the Tree Reconfiguration Algorithm to find the minimum energy tree T subject to the constraint that $P_\infty(T) \leq P_{\text{desired}}$. The Tree Reconfiguration Algorithm consists of three subroutines with the first one generating the initial sensor tree, the second one switching the tree topology if the estimation constrain is violated, and the third one finding the minimum energy subtree.

In Chapter 5, I study the network lifetime maximization problem, and I propose the Tree Construction and Scheduling Algorithm. The algorithm constructs a set of sensor trees, $\{T_1, \cdots, T_M\}$, that fulfill the estimation quality constraints, i.e., $P_\infty(T_j) \leq P_{\text{desired}}\ \forall\ 1 \leq j \leq M$, yet with different energy cost of individual sensor node due to the randomness

introduced by the algorithm. Those trees are then scheduled in an optimal way such that the network lifetime is maximized.

Finally in Chapter 6, I consider minimizing the buffer length when state estimation is carried over a packet delaying network. As random delays of the measurement data are typically seen, I study the problem from a probabilistic angle, i.e., the minimum buffer length D is found subject to the constraint that $\mathbf{Pr}[P_k \leq M|D] \geq 1-\epsilon$ for a given M and ϵ. When the sensor does not have computation capability, lower and upper bounds are found for $\mathbf{Pr}[P_k \leq M|D]$; and when the sensor has computation capability, exact expression is found for $\mathbf{Pr}[P_k \leq M|D]$. Therefore, given M and ϵ, we can find the sufficient and necessary buffer length D that guarantees $\mathbf{Pr}[P_k \leq M|D] \geq 1-\epsilon$.

7.2 Future Directions

There are many interesting directions to go along the line of this thesis work, and they are briefly stated below.

Topology versus Performance: An important and interesting problem is to study the tradeoffs between different sensors, controllers, and actuators topologies and system performances, as the topologies determine how to deploy the different nodes, and re-deploying in general incurs very high cost. If sensor energy is not a big concern, but rather the quality of the estimation when the communication link is not reliable, it would be interesting to study the tradeoff between the estimation quality, the underlying graph that represents the sensor communication, the quality of the communication link and the energy cost of the sensors. The quality of the communication link can be represented by the average packet drop rate, so the higher the drop rate, the poorer the quality of the link. This will lead to a unified framework between the analysis of packet drops, delays, sensor topologies and system performance.

New Algorithms and Better Bounds: For the first two problems considered in the thesis, it would be interesting to find better algorithms that approximate the global optimal solution. Due to the complexity of the problem itself, we do not have a detailed performance analysis of the algorithms proposed for the network lifetime maximization problem, as we did for the algorithms in the energy minimization problem. It would be nice to have some performance description of the algorithms (possibly with some extra conditions). For the

third problem, it would be interesting to find tighter bounds on $\mathbf{Pr}[P_k \leq M]$. Current efforts are moving towards this direction.

Protocols versus Performance: Existing network infrastructure have many different communication protocols for different applications (TCP, UDP and etc). It is natural to imagine that such complex system of systems might utilize several different protocols simultaneously. It would be interesting to first characterize the effect that different communication protocols have on system performances. For example, TCP guarantees message delivery over a network while UDP does not. However, the time that TCP takes to send a message is usually longer than UDP. In some time critical applications, eg, in an rescue scenario, it might be better to use UDP than TCP. It is important to understand and characterize under which circumstance one protocol is better than the other to determine when and where each protocol is more applicable. New protocols to enhance system performance are also needed. Current protocols have not been designed specifically for use in a feedback control system. Future communication protocols should be optimized for this setting.

Sensor Scheduling and Fundamental Limitations: Sensor scheduling within sensor networks is an interesting problem. Some earlier work [43] (not included in the thesis) was toward this direction. In general, it would be interesting to find out the optimal sensor scheduling scheme that achieves desired system performance, minimizes total sensor energy/maximizes network lifetime, and subject to communication/computaiton constraints.

It would also be interesting to find the fundamental limitations in effective sensor scheduling, e.g., what is the minimum number of sensor communication channels needed in order to produce a bounded estimation error? given all the constraints, what is the best performance of the system? etc.

Closed loop control: Closing the loop using the estimation algorithms developed in this thesis is also interesting. If the separation principle holds, i.e., the estimator has full knowledge of the control inputs to the process, then the estimation algorithm is still optimal in the closed-loop setting similar to the work on TCP-like LQG control over packet dropping network [46]. The UDP-like communication protocol makes the closed-loop performance analysis much harder yet more interesting.

Bibliography

[1] P. Alriksson and A. Rantzer, "Distributed kalman filtering using weighted averaging." Proc. 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006.

[2] B. Anderson and J. Moore, *Optimal Filtering*. Prentice Hall, 1990.

[3] K. J. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," vol. J. In Preprints 14th World Congress of IFAC, Beijing, P.R. China, pp. 301–306.

[4] L. Bao, M. Skoglund, and K. H. Johansson, "Encoder–decoder design for event-triggered feedback control over bandlimited channels." American Control Conference, Minneapolis, Minnesota, USA, 2006.

[5] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1991.

[6] A. Bonivento, C. Fischione, and A. Sangiovanni-Vincentelli, "Randomized protocol stack for ubiquitous networks in indoor environment," *IEEE CCNC*, vol. 1, April 2006.

[7] R. W. Brockett and D. Liberzon, "Quantized Feedback Stabilization of Linear Systems," *IEEE Trans. on Automatic Control*, vol. 45, no. 7, pp. 1279–1289, July 2000.

[8] A. Chamam and S. Pierre, "Optimal scheduling of sensors' states to maximize network lifetime in wireless sensor networks." IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, Oct 2007.

[9] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin, "Transmission scheduling for optimizing sensor network lifetime: A stochastic shortest path approach," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2294–2309, May 2007.

[10] S. Coleri, A. Puri, and P. Varaiya, "Power efficient system for sensor networks." Proceedings of the Eighth IEEE International Symposium on Computers and Communication, 2003.

[11] L. B. Cremean, T. B. Foote, J. H. Gillula, G. H. Hines, D. Kogan, K. L. Kriechbaum, J. C. Lamb, J. Leibs, L. Lindzey, A. D. Stewart, J. W. Burdick, and R. M. Murray, "Alice: An information-rich autonomous vehicle for high-speed desert navigation," *Journal of Field Robotics*, vol. 23(9), 2006.

[12] D. Culler, D. Estrin, and M. Srivastava, "Overview of wireless sensor networks," *IEEE Computer, Special Issue in Sensor Networks*, Aug 2004.

[13] N. P. M. (Ed.), *Sensor Networks and Configuration*. Springer, 2007.

[14] M. Epstein, L. Shi, A. Tiwari, and R. M. Murray, "Probabilistic performance of state estimation across a lossy network," *Automatica*, 2008.

[15] M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice*. Prentice Hall, 1993.

[16] F. Gustafsson, *Adaptive Filtering and Change Detection*. John Wiley & Sons Inc, 2000.

[17] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," vol. 95, no. 1. Proceedings of the IEEE, January 2007.

[18] http://en.wikipedia.org/wiki/Wireless_sensor_network.

[19] http://www.hartcomm2.org.

[20] M. Huang and S. Dey, "Stability of kalman filtering with markovian packet losses," *Automatica*, vol. 43, pp. 598–607, 2007.

[21] Z. Jin, V. Gupta, and R. M. Murray, "State estimation over packet dropping networks using multiple description coding," *Automatica*, vol. 42, no. 9, pp. 1441–1452, 2006.

[22] F. Z. Juan Liu and D. Petrovic, "Information-directed routing in ad hoc sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, April 2005.

[23] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[24] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME Journal of Basic Engineering*, pp. 35–45, March 1960.

[25] H. Kopetz, "Should responsive systems be event triggered or time triggered?" *IEICE Trans. on Information and Systems*, vol. E76-D(10), pp. 1525–1532, 1993.

[26] C.-C. Lai, C.-K. Ting, and R.-S. Ko, "An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications." IEEE Congress on Evolutionary Computation, Sept 2007, pp. 3531 – 3538.

[27] W. Lai and I. C. Paschalidis, "Routing through noise and sleeping nodes in sensor networks: latency vs. energy trade-offs." Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA, Dec 2006, pp. 551 – 555.

[28] S. Liang, Y. Tang, and Q. Zhu, "Passive wake-up scheme for wireless sensor networks." Second International Conference on Innovative Computing, Information and Control, Sept 2007.

[29] X. Liu and A. Goldsmith, "Kalman Filtering with Partial Observation Losses," in *IEEE Conference on Decision and Control*, Dec 2004.

[30] A. S. Matveev and A. V. Savkin, "The problem of state estimation via asynchronous communication channels with irregular transmission times," *IEEE Trans. Automat. Contr.*, vol. 48, pp. 670–676, 2003.

[31] R. K. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Transactions on Automatic Control*, vol. 15, no. 2, April 1970.

[32] M. J. Miller and N. H. Vaidya, "A mac protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228 – 242, May-June 2005.

[33] G. N. Nair and R. J. Evans, "Communication-Limited Stabilization of Linear Systems," in *IEEE Conference on Decision and Control*, December 2000.

[34] S. Nakamori, R. Caballero-Aguila, A. Hermoso-Carazo, , and J. Linares-Perez, "Recursive estimators of signals from measurements with stochastic delays using covariance information," *Applied Mathematics and Computation*, vol. 162, pp. 65–79, 2005.

[35] E. Ngai, M. Lyu, and J. Liu, "A real-time communication framework for wireless sensor-actuator networks." Aerospace Conference, 2006 IEEE, 2006.

[36] T. N. Nguyen and D. T. Huynh, "Extending sensor networks lifetime through energy efficient organization." International Conference onWireless Algorithms, Systems and Applications, Aug 2007.

[37] A. Panousopoulou, G. Nikolakopoulos, A. Tzes, and J. Lygeros, "Experimental evaluation of a mobile ad-hoc networked (manet)controlled system." Proc. 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, 2006.

[38] I. R. Petersen and A. V. Savkin, "Multi-Rate Stabilization of Multivariable Discrete-time Linear Systems Via a Limited Capacity Communication Channel," in *IEEE Conference on Decision and Control*, Dec 2001.

[39] A. Ray, L. W. Liou, and J. Shen, "State estimation using randomly delayed measurements," *J. Dyn. Syst., Measurement Contr.*, vol. 115, pp. 19–26, 1993.

[40] M. Sahebsara, T. Chen, and S. L. Shah, "Optimal h2 filtering with random sensor delay, multiple packet dropout and uncertain observations," *Int. J of Control*, vol. 80, pp. 292–301, 2007.

[41] L. Shi, M. Epstein, A. Tiwari, and R. M. Murray, "Estimation With Information Loss: Asymptotic Analysis and Error Bounds," in *IEEE Conference on Decision and Control*, Dec 2005.

[42] L. Shi, M. Epstein, and R. M. Murray, "Kalman filtering over a packet dropping network: a probabilistic approach," in *Tenth International Conference on Control, Automation, Robotics and Vision, Dec 2008, Hanoi, Vietnam. To Appear.*

[43] L. Shi, M. Epstein, B. Sinopoli, and R. M. Murray, "Effective sensor scheduling schemes in a sensor network by employing feedback in the communication loop." Proceedings of the IEEE Conference on Control Applications, Singapore, 2007.

[44] D. Siljak, *Large-Scale Dynamic Systems: Stability and Structure*. North-Holland, New York, 1978.

[45] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, September 2004.

[46] ——, "Optimal Control with Unreliable Communication: the TCP Case," in *American Control Conference*, 2005.

[47] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S.S.Sastry, "Distributed control applications within sensor networks," vol. 91. Proceedings of the IEEE, Aug 2003, pp. 1235 – 1246.

[48] S. Sun, L. Xie, W. Xiao, and Y. Soh, "Optimal linear estimation for systems with multiple packet dropouts," *Automatica, to appear.*

[49] S. C. Tatikonda, "Control Under Communication Constraints," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[50] N. A. Vasanthi and S. Annadurai, "Sleep schedule for fast and efficient control of parameters in wireless sensor-actor networks." First International Conference on Communication System Software and Middleware, Jan 2006.

[51] Z. Wang, D. W. C. Ho, and X. Liu, "Robust filtering under randomly varying sensor delay with variance constraints," *IEEE Trans. Circuits and systems-II: Express briefs*, vol. 51, no. 6, pp. 320–326, 2004.

[52] E. Witrant, P. G. Park, M. Johansson, C. Fischione, and K. H. Johansson, "Predictive control over wireless multi-hop networks." Preceedings of the IEEE Conference on Control Applications, Singapore, 2007.

[53] W. S. Wong and R. W. Brockett, "Systems with Finite Communication Bandwidth-Part I: State Estimation Problems," *IEEE Trans. Automatic Control*, vol. 42, no. 9, pp. 1294–1299, September 1997.

[54] ——, "Systems with Finite Communication Bandwidth-Part II: Stabilization with Limited Information Feedback," *IEEE Trans. Automatic Control*, vol. 44, no. 5, pp. 1049–1053, May 1999.

[55] L. Xie and L. Xie, "Stability of a random riccati equation with markovian binary switching," *IEEE Trans. Automat. Contr., to appear*.

[56] Q. Xue and A. Ganz, "Maximizing sensor network lifetime: analysis and design guidelines." IEEE Confeence on Military Communications, Nov 2004.

[57] E. Yaz and A. Ray, "Linear unbiased state estimation for random models with sensor delay," in *Proceedings of IEEE Conf. on Decision and Control*, Dec 1996, pp. 47–52.

[58] R. Yu, Z. Sun, and S. Mei, "Scalable topology and energy management in wireless sensor networks." IEEE Conference on Wireless Communications and Networking Conference, March 2007.

[59] H. Zhang and L. Xie, *Control and Estimation of Systems with Input/Output Delays*. Springer, 2007.

[60] ——, "Optimal estimation for systems with time-varying delay," in *Proceedings of IEEE Conf. on Decision and Control*, Dec 2007.

[61] H. Zhang, L. Xie, D. Zhang, and Y. Soh, "A re-organized innovation approach to linear estimation," *IEEE Trans. Automat. Contr.*, vol. 49, no. 10, pp. 1810–1814, 2004.