

University of Alberta

**Distributed Sampling, Filtering and Synchronization in
Wireless Sensor Networks**

by

Salman Ahmed

A thesis submitted to the Faculty of Graduate Studies and Research in partial
fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical & Computer Engineering

©Salman Ahmed
Spring 2014
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

Abstract

A wireless sensor network (WSN) consists of spatially distributed sensor nodes which are deployed to monitor some process of interest. Although WSNs are very promising, the distributed nature, attributes of wireless networks, and availability of limited resources in WSNs introduce significant theoretical and practical challenges. First, the cooperative control of sensor nodes requires to consider subsystems instead of a single system. Second, the communication capabilities and connectivity of the sensor nodes are limited. Third, the information exchange in the wireless sensor network may be unreliable and suffer transmission delays. Fourth, the availability of limited resources imposes constraints on the sampling rates and time synchronization of sensor nodes. Motivated by these challenges, this thesis studies the design of distributed filtering and sampling techniques in resource constrained WSNs.

For distributed filtering, one of the most promising techniques are the linear consensus protocols. A motivating example for studying the application of consensus protocols is to investigate the distributed time synchronization problem in WSNs. In this thesis, we study and propose distributed time synchronization protocols which consider an asynchronous framework where the sensor nodes can have different time-periods, starting times and input update times. The clocks in a WSN are modeled by a time-varying system with time-delay terms. By employing tools from nonnegative matrix and graph theories, the convergence analysis is presented.

Most of the standard control and monitoring techniques rely on uniform and synchronized sampling. A sensor node has limited battery resources and their efficient utilization imposes constraints on the time synchronization of the sensor nodes which introduces sampling jitters. In this thesis, we model WSNs employing distributed sampling using filter banks and present the design of synthesis filters to minimize the effects of sampling jitters. We consider two cases for the design of synthesis filters. In the first case, we consider a hybrid filter bank and assume that the sampling jitter is known for each sensor. We employ tools from sampled-data control theory and present a procedure to design optimal \mathcal{H}_2 synthesis filter bank to handle sampling jitters and reconstruct uniformly sampled measurements. In the second case, we consider a discrete-time filter bank and allow the sampling jitters to be time-varying. Using polytopic matrices to encompass all possible representations of the system matrices, the problem is reduced to an \mathcal{H}_∞ optimization problem and the design of pre-processing filters is presented. All the theoretical development and the proposed techniques in this thesis are validated using simulation examples.

Acknowledgements

I am indebted to my supervisor, Prof. Tongwen Chen, for his guidance, inspiring discussions and support throughout this research. He is a knowledgeable mentor whose suggestions and feedback have been very helpful in the completion of this research. He taught me to be clear and organized in research and demonstrate professionalism from the early stages of my career. I really appreciate to be a student under his supervision.

I am grateful of my PhD candidacy and final exam committee members: Prof. Qing Zhao, Prof. Alan Lynch, Prof. Mahdi Tavakoli, Prof. Tony Qiu and Prof. Henry Leung. They spent their valuable time to review my thesis and provide valuable suggestions and comments. I am extremely thankful to Prof. Feng Xiao (currently at Harbin Institute of Technology) for his enlightening discussions, comments and help during his stay at University of Alberta. My numerous conversations with him have inspired many ideas of this work. Special thanks to all the administrative staff of ECE Department who have helped me on several occasions.

I also acknowledge the financial support from Natural Sciences and Engineering Research Council of Canada, which was made available by Prof. Tongwen Chen in the form of research assistantships, and University of Alberta (Provost Doctoral Entrance Award). I am thankful to my fellows in the Advanced Control Group for sharing their research work.

Finally, I owe special thanks to my parents, brother, sisters and my wife (the major nonscientific discovery of my life) and children for being constant sources of encouragement, support and optimism throughout my studies.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Literature Review	5
1.2.1	Distributed filtering and time synchronization	5
1.2.2	Distributed sampling	6
1.3	Summary of Contributions	7
1.4	Applications	8
1.4.1	Distributed filtering	8
1.4.2	Distributed sampling	8
1.5	Thesis Outline	8
2	Distributed Time Synchronization in WSNs	10
2.1	Introduction	10
2.2	Preliminaries	12
2.3	Problem Formulation	13
2.4	Main Results	15
2.5	Convergence Analysis	17
2.6	Numerical Examples	18
2.7	Concluding Remarks	21
3	Time Synchronization using Unreliable Communication Channels	23
3.1	Preliminaries	24
3.2	Problem Formulation	24
3.2.1	System model	25
3.2.2	Communication strategy	25
3.3	Main Results	27
3.4	Technical Analysis	31
3.5	Numerical Examples	34
3.6	Concluding Remarks	38

4	Optimal \mathcal{H}_2 Filtering in WSNs	39
4.1	Introduction	39
4.2	Preliminaries	40
4.2.1	Step-invariant discretization	41
4.2.2	Norm-invariant discretization	41
4.2.3	Norm of sampled systems	41
4.2.4	Discretizing a time-delay transfer matrix	43
4.2.5	Multi-rate components	44
4.3	Problem Formulation	44
4.4	\mathcal{H}_2 Filter Design	46
4.4.1	Reduction to a model-matching problem	46
4.4.2	Design of IIR filters	50
4.4.3	Design of FIR filters	50
4.5	Numerical Example	52
4.6	Concluding Remarks	53
5	\mathcal{H}_∞ Filtering in WSNs	55
5.1	Introduction	55
5.2	Problem Formulation	56
5.3	Main Results	58
5.3.1	Discrete-time representation of the error system	58
5.3.2	Polytopic representation of uncertain system matrices	59
5.3.3	Design of an \mathcal{H}_∞ filter for polytopic systems	60
5.4	Numerical Example	62
5.5	Concluding Remarks	62
6	Summary and Future Work	64
6.1	Major Thesis Contributions	65
6.2	Directions for Future Research Studies	66
	Bibliography	68

List of Tables

2.1	Parameter values used to simulate the WSN as shown in Fig. 2.1	19
3.1	Parameter values used to simulate the WSN as shown in Fig. 3.2	35
4.1	Performance comparison using different design techniques.	53

List of Figures

- 1.1 Schematic of a sensor node in a WSN. 2
- 1.2 Distributed sampling with 4 sensors labeled as s_1, s_2, s_3 and s_4 . The dashed lines represent the uniformly sampled measurements and the solid lines represent the sampled measurements in the presence of sampling jitters. 4
- 2.1 Communication topology of a WSN considered in Example 1 having 10 sensor nodes. 19
- 2.2 Plot of the local clock readings of a WSN considered in Example 1. The topology of the WSN is shown in Fig. 2.1. 20
- 2.3 Plot of the local control inputs applied to the WSN considered in Example 1. The topology of the WSN is shown in Fig. 2.1. 20
- 2.4 The graph of union of communication topologies of a WSN considered in Example 2 having 45 sensor nodes labeled as $1, \dots, 45$. The lines indicate the presence of communication channels. 21
- 2.5 Plot of the sum of the relative time synchronization errors in the WSN considered in Example 2. The graph of the union of communication topologies of the WSN is shown in Fig. 2.4. 22
- 3.1 Control input update and transmission timings of a WSN having 3 sensor nodes labeled as i, j , and h . The union of transmission topologies in the WSN and the transmission topologies at some particular time instants are also shown. 27
- 3.2 The union of possible communication topologies of a WSN considered in Example 1 having 8 sensor nodes. The lines indicate the presence of possible communication channels for transmitting information. 35

3.3	Plot of the local clock readings of a WSN considered in Example 1. The plot demonstrates the effectiveness of the proposed protocol to achieve time synchronization. The union of possible communication topologies of the WSN is shown in Fig. 3.2.	36
3.4	Plot of the local control inputs applied to the WSN considered in Example 1. The union of possible communication topologies of the WSN is shown in Fig. 3.2.	36
3.5	Plot of the sum of the relative time synchronization errors in the WSN considered in Example 1. The graph of the union of communication topologies of the WSN is shown in Fig. 3.2.	37
3.6	The graph of union of communication topologies of a WSN considered in Example 2 having 60 sensor nodes. The lines indicate the presence of communication channels only and do not indicate if at any particular time instant any information is exchanged on the communication link.	37
3.7	Plot of the sum of the relative time synchronization errors in the WSN considered in Example 2. The graph of the union of communication topologies of the WSN is shown in Fig. 3.6.	38
4.1	Schematic of the error system, K	45
4.2	The hybrid error system, K , modeling a WSN employing distributed sampling. The solid lines show continuous-time signals and the dashed lines depict discrete-time signals.	46
4.3	Equivalent discrete-time LTI error system, K_d	48
4.4	Standard \mathcal{H}_2 optimization problem.	51
4.5	Plot of the averaged error between actual output and desired output of the filter bank.	54
5.1	A filter bank showing a WSN employing distributed sampling.	57
5.2	The error system, K , considered for designing pre-processing filter for each channel.	58
5.3	Plot of estimation error using using the proposed \mathcal{H}_∞ filter design technique.	63

List of Symbols

\mathbb{R}	Set of real numbers
\mathbb{Z}	Set of integers
\mathbb{N}	Set of nonnegative integers
s_{pq}	A matrix with s_{pq} as the p -th row and j -th column element
\mathcal{H}_2	\mathcal{H}_2 norm of a system
\mathcal{H}_∞	\mathcal{H}_∞ norm of a system
U	Unit time-delay operator
$\delta(t)$	Continuous-time unit impulse function
$1(t)$	Continuous-time unit step function
$\mathbf{1}$	The vector $[1, \dots, 1]^T$ of compatible dimension
I	Identity matrix of compatible dimension
$\prod_{i=1}^k S_i = S_k S_{k-1} \dots S_1$	Left product of the matrices S_k, S_{k-1}, \dots, S_1
\mathcal{G}	Undirected or directed graph
$\mathcal{V}(\mathcal{G})$	Vertex set of undirected or directed graph, \mathcal{G}
$\mathcal{E}(\mathcal{G})$	Edge set of undirected or directed graph, \mathcal{G}
\mathcal{N}_i	Index set of neighbors of sensor node i
$\mathcal{N}(\mathcal{G}(t), i)$	Index set of available neighbors of sensor node i at t
$\mathcal{G}(S)$	Graph associated with nonnegative matrix S
$\left[\begin{array}{c c} A & B \\ \hline C & D \end{array} \right]$	Compact representation of state-space model

Chapter 1

Introduction

1.1 Motivation

With the rapid technological merging of communication systems, control engineering, and computing science, researchers and scientists in various fields are attracted to the exciting developments and technological challenges in the field of wireless sensor networks (WSNs). A wireless sensor network (WSN) consists of spatially distributed sensor nodes which are deployed to monitor some process of interest. A fundamental design for a sensor node in a WSN includes sensors, a microcontroller, wireless communication hardware, and battery resources as shown in Fig. 1.1. Each unit in a sensor node consumes battery power which must be efficiently utilized. Although each sensor node has limited battery resources and computation capabilities, when suitably deployed in large scale, potentially powerful networks can be constructed to accomplish tasks which can not be achieved with a single sensor. WSNs are widely used in many fields of engineering such as environment monitoring [7, 102, 66, 3], industrial and machine health monitoring [84, 48, 64], transportation systems [16, 79, 49] and biomedical engineering [25, 59].

Spatially distributed components for control systems have been used for many years in chemical process plants, oil and gas refineries, nuclear power plants and air traffic control [34, 67, 60]. Traditionally, these control systems had a centralized or a hierarchical architecture which offered advantages in the control design process. However, these systems tend to grow large and complex with the expansion in business units, requiring more computation capabilities, consuming more space and often requiring wiring over long distances. The recent trend to avoid these problems is to deploy WSNs that can reduce the wiring costs and compensate for the computation limitations [93, 103, 44, 36]. Although WSNs are very promising, the distributed nature, attributes of wireless networks, and availability of limited

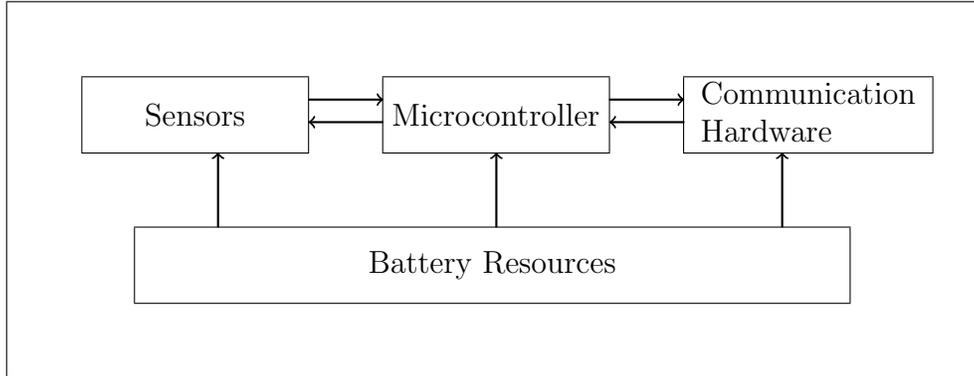


Figure 1.1: Schematic of a sensor node in a WSN.

resources in WSNs introduce significant theoretical and practical challenges to deploy WSNs. First, the cooperative control of sensor nodes requires to consider subsystems instead of a single system. Second, the communication capabilities and connectivity of the sensor nodes are limited. Third, the information exchange in the wireless sensor network may be unreliable and suffer transmission delays. Fourth, the availability of limited resources imposes constraints on the sampling rates and time synchronization of sensor nodes.

It is important to mention that research in the field of WSNs is mainly at the crossroads of three research areas: control sciences and signal processing, wireless communication and information theory, and computing science. Though many useful results have been obtained separately in each field, merging them would be very beneficial to the WSN community. Generally, traditional control theory assumes periodic and synchronized sampling, whereas communication theory focuses on the transmission of information over unreliable channels, and computing science focuses on the implementation of distributed computing algorithms and services in WSNs. Since a WSN consists of distributed sensor nodes, having limited resources, which communicate with each other using a shared network to monitor a process, combining these frameworks is essential to study WSNs.

A sensor node has limited computation and information processing capabilities. In many applications such as distributed estimation [62, 83], formation control [81, 73], exploration and surveillance [105], attitude control of satellite clusters and time synchronization [88, 24, 85], sensor nodes need to coordinate by exchanging information and processing the received information. Distributed filtering refers to the local processing of information, received from other sensor nodes as well as self

information, by a sensor node. The objective of distributed filtering is to ensure that all the sensor nodes reach agreement on some quantity of interest. Recent years have witnessed significant interest and research activity in the areas of distributed filtering and coordinated control of WSNs (see, e.g., [55, 22, 77, 50, 109] and the survey paper [18]).

Most of distributed filtering techniques presented in the literature require all the sensor nodes to periodically exchange and process information at the same time instants. This is based on the assumption that all the sensor nodes are time synchronized. The operating environment conditions as well as the unavoidable manufacturing flaws in the sensor nodes cause different sensor nodes to have different clock readings. Moreover, a WSN typically consists of a large number of sensor nodes and it is not easy to ensure that all sensor nodes start at the same time; this could lead to non-synchronization of sensor nodes. Therefore, it is important to design distributed time synchronization techniques for WSNs. In this thesis, we study and design distributed time synchronization techniques for WSNs.

The availability of limited computation, communication and battery resources imposes constraints on the sampling periods of the sensor nodes. For example, the MICA2 Berkeley nodes [97] use an 8-bit microcontroller running at 4 MHz, which can transmit up to 39.32 kilobits per second (kbps) and are battery powered by a pair of AA batteries. In [1], the MICA2 Berkeley nodes were used to conduct an experiment and the effective sampling rate (from control perspective) was found to be between 90 to 100 samples per second. For several real-time industrial monitoring applications, such as gas leakage and fire detection, the sampling period requirements are much higher. Therefore, we need a technique to obtain fast sampling period by employing sensor nodes having slow sampling periods. Distributed sampling was proposed in [53, 39], which utilizes a combination of sensor nodes, having slow sampling periods, to produce sampled measurements of a signal which are the same as the sampled measurements of the signal by a fictitious sensor node having fast sampling period.

Distributed sampling employs time-interleaved sampling to share the sampling load among the sensor nodes. Each sensor node samples a common input signal at a slower rate. The output measurements of these sensor nodes are multiplexed to produce a fast sampled signal. The advantage of distributed sampling is that slow sampling sensor nodes can be added in parallel to act like a fast sampling sensor node. The implementation of distributed sampling requires precise time synchronization among the sensor nodes.

A sensor node has limited battery resources which must be efficiently utilized. The efficient utilization of battery resources imposes constraints on the time synchronization of sensor nodes at every sampling instant [2, 76]. The non-synchronization of sensor nodes introduces sampling jitters, which is the difference between the ideal and physical sampling instants. Fig. 1.2 shows an example of the distributed sampling pattern in a WSN having 4 sensors which are labeled as s_1 , s_2 , s_3 and s_4 . The measurements of these 4 sensors are denoted by $s_{i(j)}$ where i denotes a sensor and j denotes a measurement by the sensor i . Thus $s_{1(1)}$ means measurement 1 from sensor 1, and so on. Fig. 1.2 also highlights the effect of sampling jitters. The dashed lines show the time instants and corresponding signal values if there are no sampling jitters. The solid lines show the time instants and corresponding signal values in presence of sampling jitters.

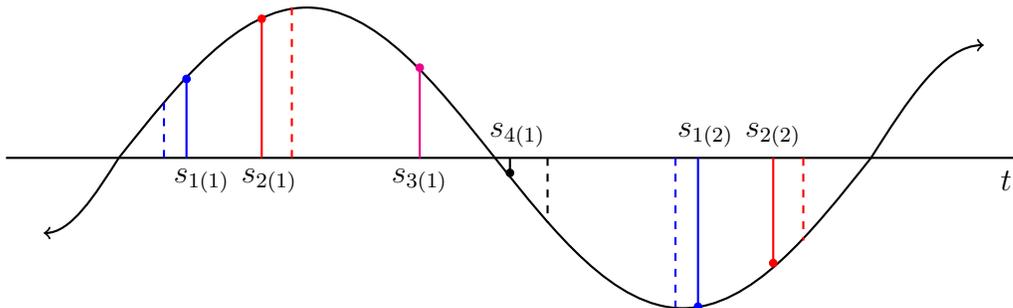


Figure 1.2: Distributed sampling with 4 sensors labeled as s_1 , s_2 , s_3 and s_4 . The dashed lines represent the uniformly sampled measurements and the solid lines represent the sampled measurements in the presence of sampling jitters.

The presence of sampling jitters gives rise to non-uniform sampling. Most of the standard control and monitoring techniques rely on uniform and synchronized sampling. Therefore, to apply the standard techniques we need to recover the uniformly sampled measurements from the non-uniformly sampled measurements. This motivates the development of new techniques to compensate the effects of sampling jitters. In this thesis, we study WSNs employing distributed sampling and present the design of techniques to minimize the effects of sampling jitters. In the next section, we present a brief overview of the recent progress in the fields of distributed filtering and sampling in WSNs.

1.2 Literature Review

In this section, we present an overview of the existing approaches related to the modeling and design of distributed filtering, time synchronization and distributed sampling techniques in WSNs.

1.2.1 Distributed filtering and time synchronization

The design of distributed filtering techniques has been a popular subject in signal processing and control theory over the last couple of years. One of the most promising distributed filtering design tools are the linear consensus protocols [77], which require limited sensor network knowledge, and minimal computation and communication capabilities to achieve agreement among the sensor nodes.

The earlier work related to the development of consensus techniques can be traced back to the work presented in [32] and [43], where the authors applied consensus techniques in the fields of management science and statistics. In the control community, the seminal work was presented by Borkar and Varaiya [14], and Tsitsiklis and Athans [98]; they considered the application of consensus techniques in distributed decision making systems. More recently, consensus protocols were applied by researchers in the control and multi-agent communities for cooperative control and coordination of multi-agent systems [77, 81]. In this thesis, we study and apply consensus algorithms to design distributed time synchronization protocols for WSNs. However, most of the consensus protocols presented in the literature require time synchronization and periodic communication among the sensor nodes. Therefore, we need to extend the current results and investigations to asynchronous systems. Some of the other existing approaches to design distributed time synchronization protocols are summarized below.

A common approach to design time synchronization protocols in WSNs is to create a hierarchical structure. The most prominent examples are the Reference Broadcast Synchronization (RBS) [33], Timing-sync Protocol for Sensor Networks (TPSN) [37], and Flooding Time Synchronization Protocol (FTSP) [68]. A limitation of the hierarchy-based protocols is that they are not robust to changes in the communication topology. Distributed time synchronization protocols which do not require any hierarchical structure have been proposed in [95, 91, 86, 20, 65]. In [95], the authors proposed a distributed time synchronization protocol using the coordinate descent optimization algorithm; the protocol takes into consideration the natural network constraint that the sum of relative clock offsets in a network

loop should be zero. In [91, 86] and [20], the authors presented consensus-based time synchronization protocols which ensure that the clocks in a WSN achieve time synchronization with reference to a virtual clock. However, the design of the protocols presented in [95, 91, 86, 20] assumes that the sensor nodes should periodically exchange and update their clock states at the same time instants; this assumption is unrealistic since it requires synchronization of the clocks, which is the goal of the protocol. This assumption is partially relaxed in [85] and [21], where the authors assume that all the sensor nodes should update their clock states at the same time instants but the information exchange among neighbor nodes may occur at non-regular time instants. In [65], a consensus-based time synchronization protocol is presented which compensates for initial time offsets and skew deviations. The protocol assumes synchronous clocks update and information exchange. To our best knowledge, the design of consensus-based time synchronization protocol using asynchronous time updates and unreliable communication links has not been studied in WSNs.

1.2.2 Distributed sampling

The interest to develop techniques to overcome the limitation of slow sampling rates of sensing devices can be traced back to the earlier works by Black and Hodges in 1980 [58], where it was proposed to employ an array of time-interleaved analog-to-digital converters to increase the sampling rate. More recently, this technique was rediscovered and applied by researchers in the field of WSNs to increase the sampling rate [53, 39, 40].

A popular approach to model a WSN employing distributed sampling sensors is to use filter banks [57, 30, 101]. Filter banks have been widely studied in the signal processing literature; Vaidyanathan's book [100] gives a comprehensive historical survey of the literature till 1992. The design of multi-rate filters using \mathcal{H}_∞ optimization was originally proposed in [89]. This design was extended to discrete-time filter banks in [26] where the authors assumed time synchronization in the sampling process. The design of synthesis filters using hybrid filter banks was presented in [90], where the authors used the continuous-time lifting technique to obtain an equivalent discrete-time filter bank problem. In [56], the authors considered time-delays, which are integer multiples of the sampling periods, for the design of filter bank. In [75], the authors considered fractional delays for the design of synthesis filter banks using \mathcal{H}_∞ optimization. It should be noted that the presence of sampling jitters make the sampling pattern non-uniform and results in

a time-varying and uncertain system; therefore the design of synthesis filter banks should take this observation into consideration.

The design of filters for uncertain systems has been extensively studied in the fields of control theory and signal processing (see, e.g., [41, 9, 70]). Some of the earlier design approaches employed a common Lyapunov function (see, e.g., [41, 99]). However, such filter design techniques result in conservative results due to a single common Lyapunov function. The conservatism can be reduced by using parameter-dependant Lyapunov functions [29, 9, 42]. The main difficulty in designing filters using parameter-dependant Lyapunov functions is to find a change of coordinates to separate the Lyapunov stability matrix from the system matrices to obtain linear matrix inequalities (LMIs). A robust filter design technique using a parameter-dependant Lyapunov function was proposed in [13]. However, the filter design conditions were expressed in terms of bilinear matrix inequalities (BMIs), which are non-convex, and the computation complexity is high compared to linear matrix inequalities (LMIs). In this thesis, we study and present the design of pre-processing filters in terms of LMIs by extending the techniques presented in [13, 31].

1.3 Summary of Contributions

The main contributions of this thesis are summarized as follows:

1. Presented the design of a consensus-based time synchronization protocol in WSNs which does not require periodic communication and control input updates.
2. Extended the design of above-mentioned protocol to consider information exchange over unreliable communication channels and time-varying communication topologies in a WSN.
3. Modeled a WSN employing distributed sampling sensors with sampling jitters using a hybrid filter bank and proposed a novel technique to reconstruct the uniformly sampled measurements by designing synthesis filters based on the minimization of the \mathcal{H}_2 norm of the estimation error system.
4. Using an approximation of the above-mentioned model, proposed another filter design technique in terms of LMIs to recover the uniformly sampled measurements by minimizing the \mathcal{H}_∞ norm of the estimation error system.

1.4 Applications

1.4.1 Distributed filtering

Distributed filtering in WSNs has received considerable attention in the control and signal processing communities because of its extensive applications such as coordinated motion and formation control of multi-agent systems [55, 81, 73], distributed estimation and path planning of mobile robots [77]. In [82, 81], the authors studied and implemented distributed filtering techniques to cooperatively control and coordinate multiple robot systems. In [12], the authors presented a distributed filtering algorithm to estimate a parameter using WSN and implemented it on an experimental testbed.

1.4.2 Distributed sampling

Distributed sampling has a rich history of applications. Initially, it was used to increase the sampling rate of analog-to-digital convertors. In the WSN community, this technique was rediscovered by the authors in [53] to increase the sampling rate. In [56], the authors experimentally implemented the distributed sampling technique using microphone arrays to obtain a high resolution signal. In [46], the authors considered the implementation of a WSN for healthcare monitoring and employed 16 wireless sensor nodes employing distributed sampling to achieve a sampling rate of 20 GHz.

1.5 Thesis Outline

The remaining of this thesis is organized as follows:

In Chapter 2, we study and present the design of a novel distributed time synchronization protocol using consensus algorithms. The proposed protocol does not require all the sensor nodes to have the same time-periods or starting times. Furthermore, the communication topology of the WSN is not assumed to be static. The WSN is modeled by a time-varying discrete-time system. The objective is to design local control inputs to achieve relative time synchronization by using the information of neighboring sensor nodes. By employing tools from nonnegative matrix and graph theories, the convergence analysis is presented. Numerical examples are presented to demonstrate the effectiveness of the proposed time synchronization protocol.

In Chapter 3, we extend the distributed time synchronization protocol presented in Chapter 2 to asynchronous frameworks. We consider unreliable communication links. We model the clocks in a WSN by a time-varying system with time-delay terms and present the convergence analysis of the proposed protocol. Numerical examples are given to demonstrate the effectiveness of the proposed time synchronization protocol.

In Chapter 4, we study WSNs employing distributed sampling sensors. We model such systems by hybrid filter banks. By utilizing the properties of continuous-time lifting operator and discretizing a fractional time-delay system, we obtain a norm-invariant discretized system. Next, by using the polyphase representation along with the lifting technique, we convert the system into a standard model-matching form for \mathcal{H}_2 optimal filter design. A numerical example is presented to show the effectiveness of the proposed filter design approach.

In Chapter 5, we present the design of \mathcal{H}_∞ pre-processing filters to compensate the effects of time-varying sampling jitters. The presence of sampling jitters causes uncertainty in the system matrices. Using polytopic matrices, we encompass all possible representations of the system matrices and then reduce the problem to an \mathcal{H}_∞ optimization problem. We present sufficient conditions for the design of filters in terms of linear matrix inequalities. A numerical example is also presented to show the effectiveness of the proposed filter design approach.

Chapter 6 gives a conclusion of the PhD work and presents a number of important research directions for future studies.

Chapter 2

Distributed Time Synchronization in WSNs^{*}

2.1 Introduction

Distributed filtering in WSNs refers to the local processing of information received at some sensor nodes. The objective of distributed filtering is to ensure that all the sensor nodes reach agreement on some quantity of interest. Examples of distributed filtering can be found in many areas such as distributed estimation [62, 83], coordinated motion and formation control [55, 81, 73], force allocation in paper moving devices [35], control of directional sensitivity of smart antennas [51], and attitude control of satellite clusters [77].

The design of distributed filtering techniques in WSNs has to consider some important constraints. First, each sensor node has access to limited information about the global goal of the sensor network. Second, the global goal can not be computed by a single sensor node due to several reasons such as the network is too large to transmit information to one point, or the topology of the network is changing and centralized decision making is not efficient. Third, each sensor node has limited memory and computation resources – even if a sensor node acquires all the information to compute the global goal, the computation load would be too high. Fourth, each sensor node communicates only with a limited number of nearby sensor nodes which are termed as its neighbors. Although these constraints introduce limits on the distributed filtering design techniques, the motivation comes from the fact that the simple control laws can be used by sensor nodes to achieve the global goal. One of the most promising distributed filtering design techniques are the linear consensus protocols (see, e.g., [77]), which require limited sensor network

^{*}A version of this chapter has been published in [6].

knowledge, and minimal computation and communication capabilities to achieve agreement among the sensor nodes. A motivating example for studying the application of consensus protocols is to consider the distributed time synchronization problem in WSNs. In this chapter, we study the design of consensus protocols to achieve distributed time synchronization in WSNs.

Distributed time synchronization in wireless sensor networks (WSNs) has been an important topic of research in the control community ([20], [85], [88]). A popular approach to design time synchronization protocols is to create a hierarchical structure within the WSN. The most prominent examples are the Reference Broadcast Synchronization (RBS) [33], Timing-sync Protocol for Sensor Networks (TPSN) [37], and Flooding Time Synchronization Protocol (FTSP) [68]. The implementation of RBS requires that the WSN should be divided into distinct clusters and each cluster should have a leader or reference node; all the leader nodes synchronize themselves with respect to each other, and sensor nodes within the same cluster synchronize themselves with respect to their leader node. The implementation of TPSN and FTSP requires that a spanning tree rooted at a reference node should be built in the WSN and the time difference of any sensor node can be obtained with respect to the reference node. Although the hierarchy-based protocols have been experimentally tested in [68] and their performance is remarkable, they require substantial overhead to rebuild the spanning trees or clusters if a sensor node dies or a new sensor node is added.

Distributed time synchronization protocols which do not require a hierarchical structure have been proposed in [88, 95, 86], where the authors have considered a continuous-time system to represent the clock of a sensor node. It is assumed that the clock of a sensor node has the same continuity property as the physical time and it can measure in any time interval. In actual implementation, the clock of a sensor node has some frequency or time-period. A clock cannot measure a time interval which is shorter than its time-period. Hence, it is more reasonable to use a discrete-time system to model a clock.

Discrete-time systems to model clocks in a WSN have been used in [20, 92, 11]. For the design of time synchronization protocol, the authors have assumed that all the sensor nodes should periodically exchange and update their clock states at the same time instants. This assumption is unrealistic since it requires synchronization of all clocks, which is the goal of the protocol. In this chapter, we propose a time synchronization protocol which relaxes the three main assumptions used in the literature. First, the proposed protocol does not require all the sensor nodes to

have the same time-periods or starting times. Second, all the sensor nodes are not restricted to exchange and update their clock states at the same time instants. Third, the communication topology of the WSN is not assumed to be static. We use the discrete-time clock model presented in [20]. We assume that each sensor node knows its time-period but does not know its exact starting time. Each sensor node exchanges its clock readings with its neighbors. The objective of the time synchronization protocol is to synchronize the clocks in a WSN with respect to a virtual clock, which is not physically present or accessible to any sensor node. The virtual clock is only used to demonstrate the concept of time synchronization in a WSN as the time-periods and starting times of sensor nodes can be different.

The remainder of this chapter is structured as follows: Section 2.2 presents some basic definitions and results from graph and matrix theories. The problem is formulated in Section 2.3. The main results and analysis are given in Sections 2.4 and 2.5, respectively. Numerical examples to demonstrate the effectiveness of the proposed protocol are presented in Section 2.6. Finally, concluding remarks are given in Section 2.7.

2.2 Preliminaries

Graphs provide a natural abstraction for information exchange within a WSN. In this section, we present some useful definitions and results from graph and matrix theories which will be used in this thesis.

Let $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^n$, $I \in \mathbb{R}^{n \times n}$ denotes an identity matrix, and the symbol \mathbb{N} denotes the set of nonnegative integers. A matrix S is nonnegative if all its elements are non-negative [47]. A non-negative, finite and square matrix S is called stochastic if the sum of each row is 1. A stochastic matrix S is called indecomposable and aperiodic (SIA) if there exists a column vector w such that $\lim_{k \rightarrow \infty} S^k = \mathbf{1}w^T$ [104]. Let $\prod_{i=1}^k S_i = S_k S_{k-1} \dots S_1$ denote the left product of the matrices S_k, S_{k-1}, \dots, S_1 . For any matrix S , we write s_{pq} to represent the element which lies in the p^{th} row and q^{th} column of S .

Consider an undirected (directed) graph \mathcal{G} which consists of a vertex set, $\mathcal{V}(\mathcal{G}) = \{v_1, \dots, v_n\}$, and an edge set, $\mathcal{E}(\mathcal{G}) \subset \{(v_i, v_j) : v_i, v_j \in \mathcal{V}(\mathcal{G})\}$ where an edge is an unordered (ordered) pair of vertices [69]. If (v_i, v_j) is an edge in a directed graph, \mathcal{G} , then v_i and v_j are defined as the parent and child vertices, respectively. The index set of neighbors of vertex v_i is denoted by $\mathcal{N}_i = \{j : (v_j, v_i) \in \mathcal{E}(\mathcal{G}), j \neq i\}$ and its cardinality by $|\mathcal{N}_i|$. A path from v_{i_1} to v_{i_k} is a sequence, v_{i_1}, \dots, v_{i_k} of

distinct vertices such that $(v_{i_j}, v_{i_{j+1}}) \in \mathcal{E}(\mathcal{G})$ for any $j = 1, \dots, k-1$. An undirected (directed) graph \mathcal{G} is connected (strongly connected) if there is a path between any two distinct vertices. A directed tree is a directed graph where every vertex except the root vertex has exactly one parent vertex and the root vertex can be connected to any other vertices via paths. A spanning tree of directed graph \mathcal{G} is a directed tree whose edge set belong to $\mathcal{E}(\mathcal{G})$ and vertex set is the same as $\mathcal{V}(\mathcal{G})$. The union of a group of undirected (directed) graphs $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$ having common vertex set \mathcal{V} is an undirected (directed) graph with vertex set \mathcal{V} and edge set given by the union of the edge sets $\mathcal{E}(\mathcal{G}_i)$, where $i = 1, \dots, k$. The Laplacian matrix of \mathcal{G} is denoted by $L = [l_{pq}] \in \mathbb{R}^{n \times n}$ and is defined as follows:

$$l_{pq} = \begin{cases} -1 & \text{if } p \neq q \text{ and } q \in \mathcal{N}_p, \\ |\mathcal{N}_p| & \text{if } p = q, \\ 0 & \text{otherwise.} \end{cases}$$

Let $S = [s_{pq}] \in \mathbb{R}^{n \times n}$ be a nonnegative matrix. Then $\mathcal{G}(S)$ is defined as a directed graph having vertex set $\mathcal{V}(\mathcal{G}(S)) = \{v_1, \dots, v_n\}$, and edges from v_q to v_p if and only if $s_{pq} > 0$. We present some lemmas which will be used later.

Lemma 2.1. [106] *Let S be a stochastic matrix. If $\mathcal{G}(S)$ contains at least one spanning tree such that the root vertex of that spanning tree has a self-loop in $\mathcal{G}(S)$, then S is SIA.*

Lemma 2.2. [80] *Let S_1, S_2, \dots, S_m be stochastic matrices. If the graph obtained by the union of graphs $\mathcal{G}(S_1), \mathcal{G}(S_2), \dots, \mathcal{G}(S_m)$ has a spanning tree, then the matrix product $S_m S_{m-1} \dots S_1$ is SIA.*

Lemma 2.3. [104] *Let $\Gamma = \{S_1, S_2, \dots, S_k\}$ be a finite set of SIA matrices with the property that for each sequence $S_{i_1}, S_{i_2}, \dots, S_{i_j}$ of positive length, the matrix product $S_{i_j} S_{i_{j-1}} \dots S_{i_1}$ is SIA. Then, for each infinite sequence S_{i_1}, S_{i_2}, \dots , there exists a column vector w such that*

$$\lim_{j \rightarrow \infty} S_{i_j} S_{i_{j-1}} \dots S_{i_1} = \mathbf{1} w^T.$$

2.3 Problem Formulation

We consider a WSN having n sensor nodes. Each sensor node has a local clock which consists of a software program and a hardware oscillator. The oscillator periodically generates time interrupts using its time-period and the software computes the local time of the sensor node at those time instants when the interrupts are

generated [20]. Let τ_i , x_i and t_0^i denote the time-period, clock value and starting time of sensor node i , respectively, where $i = 1, \dots, n$. Each sensor node knows its time-period but does not exactly know its starting time. Rather, its estimate, denoted by \hat{t}_0^i for sensor node i , is known. Let $T_i = \{\tilde{t}_k^i : k = 0, 1, \dots, \}$ represent the sequence of time instants when the clock of sensor node i updates its state, where $\tilde{t}_k^i := k\tau_i + t_0^i$. The time computed locally by sensor node i can be expressed as follows:

$$x_i(\tilde{t}_{k+1}^i) = x_i(\tilde{t}_k^i) + \tau_i, \quad x_i(\tilde{t}_0^i) = \hat{t}_0^i, \quad i = 1, \dots, n.$$

If each sensor node does not know its starting time exactly, then the clocks in the WSN would not be synchronized. To achieve time synchronization, we apply a control input to adjust the clock readings of sensor nodes. The control input is applied periodically at a certain known integer multiple of the local clock time-period. Let $t_s^i := \tilde{t}_{0+sz_i}^i$ be the time instants when the control input is applied to the clock of sensor node i , where z_i is a known integer, $s \in \mathbb{N}$ and $i = 1, \dots, n$. We can write the following:

$$x_i(t_{s+1}^i) = x_i(t_s^i) + z_i\tau_i + u_i(t_s^i), \quad x_i(t_0^i) = x_i(\tilde{t}_0^i), \quad (2.1)$$

where $u_i(t_s^i)$ denotes the control input or protocol which is to be designed. A virtual continuous-time clock can be expressed as follows:

$$v(t) = at + b, \quad (2.2)$$

where t represents the time of WSN, v represents the clock readings, a and b represent the skew and initial offset of the virtual clock, respectively. The virtual clock is not available to any sensor node. It is only used to demonstrate the concept of distributed time synchronization. The clocks in a WSN are said to be synchronized if there exists a common virtual clock in the form of (2.2) such that the following holds:

$$\lim_{\tilde{t}_k^i \rightarrow \infty} (x_i(\tilde{t}_k^i) - (\tilde{t}_k^i + b)) = 0, \quad \forall i = 1, \dots, n.$$

As each clock knows its own time-period, we use $a = 1$ for the common virtual clock. The sensor nodes communicate with their neighbors and exchange clock readings. The communication topology of the WSN can be modeled by a time-varying undirected graph, $\mathcal{G}(t_s^i)$, where $i = 1, \dots, n$ and $s \in \mathbb{N}$. The neighbors of sensor node j at t_s^i consist of those sensor nodes which are able to communicate with node j at t_s^i , where $i, j = 1, \dots, n$ and $s \in \mathbb{N}$. Let $\mathcal{N}_i(t_s^i)$ denotes the index set

of sensor node i neighbors and $L(t_s^i)$ represents the Laplacian matrix associated with $\mathcal{G}(t_s^i)$. The following assumptions are required to ensure that the clocks can be synchronized:

- (A1) There exists a known integer κ such that each sensor node i can obtain information at least once from any possible neighbor during $[t_s^i, t_{s+\kappa}^i]$, where $i = 1, \dots, n$ and $s \in \mathbb{N}$.
- (A2) There exists an integer $p \geq 0$ such that the union of graphs $\mathcal{G}(t_s^i), \mathcal{G}(t_{s+1}^i), \dots, \mathcal{G}(t_{s+p}^i)$, is connected, for all $i = 1, \dots, n$, and $s \in \mathbb{N}$.
- (A3) For any two sensor nodes i and j , the following holds:

$$t_s^i \in T_j, \quad \forall s.$$

It should be noted that (A1) is made to ensure that each sensor node can receive information from its possible neighbors in a finite interval of time. Assumption (A2) is made to ensure that union of communication topologies during a specific time interval remains connected, although at some particular time instants the communication topology may not be connected. Assumption (A3) is made so that by using the proposed protocol the relative time synchronization error in the WSN can be driven to zero. In actual implementation, if (A3) can not be ensured to hold for all sensor nodes, then it can be relaxed to consider neighboring nodes only. If (A3) is not ensured for neighboring nodes, then a small approximation error may exist due to the fact that the starting times and time-periods of the sensor nodes may be different. In the next section, we present the main results to achieve distributed time synchronization.

2.4 Main Results

We present the following protocol for achieving time synchronization:

$$u_i(t_s^i) = \alpha_i(t_s^i) \sum_{j \in \mathcal{N}_i(t_s^i)} (x_j(t_s^i) - x_i(t_s^i)), \quad i = 1, \dots, n, \quad (2.3)$$

where $\alpha_i(t_s^i)$ denotes a gain which needs to be designed. For synchronized distributed control systems, a protocol similar to (2.3) has been presented in [77]. In this chapter, we study protocol (2.3) for asynchronous distributed control systems. The following theorem presents the main results.

Theorem 2.1. Consider a WSN which satisfies assumptions (A2) and (A3). Then the clocks of the sensor nodes achieve distributed time synchronization using protocol (2.3) if $\alpha_i(t_s^i) \in \left(0, \frac{1}{|\mathcal{N}_i(t_s^i)|}\right)$, $\forall i = 1, \dots, n$ and $s \in \mathbb{N}$.

Remark. If the communication topology of the WSN is static, then we can allow $\alpha_i(t_s^i) \in \left(0, \frac{1}{d}\right)$, $\forall i = 1, \dots, n$ and $s \in \mathbb{N}$, where d denotes the maximum of the diagonal entries of L .

The proof of Theorem 2.1 consists of three main steps. In the first step, we combine the local clock models and define a new time-sequence to represent the local clock models by a single time-varying system. In the second step, we use the new time-sequence to compute an upper bound on the time interval in which each sensor node obtains information at least once from all its neighbors. Finally, we complete the proof by utilizing the properties of connected graphs and product of SIA matrices. Before proceeding, we define a new state as follows:

$$y_i(t_s^j) = x_i(t_s^j) - (t_s^j - t_0), \quad i, j = 1, \dots, n.$$

We re-write the model expressed by (2.1) and control protocol (2.3) as follows:

$$\begin{aligned} y_i(t_{s+1}^i) &= y_i(t_s^i) + u_i(t_s^i), & y_i(t_0^i) &= x_i(t_0^i). \\ u_i(t_s^i) &= \alpha_i(t_s^i) \sum_{j \in \mathcal{N}_i(t_s^i)} (y_j(t_s^i) - y_i(t_s^i)), & i &= 1, \dots, n. \end{aligned}$$

Let $t_h = \max\{\tilde{t}_0^i, i = 1, \dots, n\}$, and we define a new time-sequence $T := \{t_r\}$, where $r \in \mathbb{N}$, which is the union of the distinct values of the sequences $\{t_s^i \geq t_h, i = 1, \dots, n, s \in \mathbb{N}\}$. We order the values of T such that $t_r < t_{r+1}$. We mention here that the distances between consecutive elements of T are non-uniform. The time-synchronization objective can be re-written as follows:

$$\lim_{t_r \rightarrow \infty} (y_i(t_r) - b) = 0, \quad i = 1, \dots, n,$$

for some b . Let $y(t_r) = [y_1(t_r), \dots, y_n(t_r)]^T$ denote the group state vector. The dynamics of the overall system can be written as follows:

$$y(t_{r+1}) = (I - A(t_r)L(t_r))y(t_r), \quad (2.4)$$

where $A(t_r) = [a_{pq}]$, $L(t_r) = [l_{pq}]$,

$$a_{pq} = \begin{cases} \alpha_p(t_r) & \text{if } p = q \text{ and } \exists s \text{ s.t. } t_r = t_s^p, \\ 0 & \text{otherwise.} \end{cases}$$

and

$$l_{pq} = \begin{cases} -1 & \text{if } q \in \mathcal{N}_p(t_r) \text{ and } \exists s \text{ s.t. } t_r = t_s^p, \\ |\mathcal{N}_p(t_r)| & \text{if } p = q \text{ and } \exists s \text{ s.t. } t_r = t_s^p, \\ 0 & \text{otherwise.} \end{cases}$$

It should be noted that $L(t_r)$ is a time-varying Laplacian matrix which can be un-symmetric and the graph associated with $L(t_r)$ may not be strongly connected. Let $\Phi(t_r) = I - A(t_r)L(t_r)$, the solution of (2.4) can be written as follows:

$$y(t_r) = \Phi(t_r)\Phi(t_{r-1})\dots\Phi(t_0)y(t_0), \quad r = 0, 1, \dots$$

If $\alpha_i(t_s^i) \in \left(0, \frac{1}{|\mathcal{N}_i(t_s^i)|}\right) \forall i, s$, then $\Phi(t_r)$ are always stochastic matrices with positive diagonal entries. The product $\Phi(t_r)\Phi(t_{r-1})\Phi(t_{r-2})\dots\Phi(t_0)$, is also a stochastic matrix with positive diagonal entries. With this preparation, we present the following lemma, which is quite useful in the computation of an upper bound on the time interval in which each sensor node can obtain information at least once from its neighbors.

Lemma 2.4. *Let $g = \frac{\max\{z_i\tau_i\}}{\min\{z_i\tau_i\}}$ where $i = 1, \dots, n$. For any sensor node $i = 1, \dots, n$ and $s = 1, 2, \dots$, the number of elements in the set $\{t_p : t_p \in [t_s^i, t_{s+1}^i]\}$ is not greater than h , where $h = (\lceil g \rceil)(n - 1) + 1$, and $\lceil g \rceil$ denotes the smallest integer not less than g .*

Proof. Let j be the sensor node such that $z_j\tau_j \geq z_i\tau_i$ where $i = 1, \dots, n, i \neq j$. When sensor node j updates its control input at t_s^j , the maximum number of times any sensor node i can update its control input before the time instant t_{s+1}^j can not exceed $\lceil g \rceil$, where $i = 1, \dots, n, i \neq j$. Since there are $n - 1$ possible values of sensor node i (excluding $i = j$), we conclude that the number of elements in $\{t_p : t_p \in [t_s^j, t_{s+1}^j]\}$ is not greater than h . \square

Using Lemma 2.4 and assumption (A1), we can state that each sensor node obtains information from all its neighbors at least once during the time interval $[t_{m-\kappa h}, t_m]$, where $m = \kappa h, \kappa h + 1, \dots$

2.5 Convergence Analysis

Let $\bar{\mathcal{G}}(t_m)$ be the graph obtained by the union of $\mathcal{G}(\Phi(t_m)), \dots, \mathcal{G}(\Phi(t_{m-\kappa h}))$. Using (A2), we conclude that $\bar{\mathcal{G}}(t_m)$ is strongly connected. If $\bar{\mathcal{G}}(t_m)$ is strongly connected, then by using Lemma 2.1 together with Lemma 2.2, we establish that the product of $\Phi(t_m)\dots\Phi(t_{m-\kappa h})$ is SIA.

Let $F_0 = \prod_{l=0}^{\kappa h-1} \Phi(t_l)$, $F_1 = \prod_{l=\kappa h}^{2\kappa h-1} \Phi(t_l)$, $F_2 = \prod_{l=2\kappa h}^{3\kappa h-1} \Phi(t_l)$ and so on. Then it can be easily shown that all possible unique F_j 's are finite where $j = 0, 1, \dots$. Any product which involves a finite combination of the unique F_j 's is SIA. So, we can apply Lemma 2.3 to get the following conclusion:

$$\lim_{k \rightarrow \infty} \prod_{j=0}^k F_j = \mathbf{1}w^T.$$

The vector w depends on the order in which the matrices $\Phi(t_l)$ are multiplied, where $l = 0, 1, \dots$, and it can not be computed in advance. We can further write the following:

$$\lim_{t_r \rightarrow \infty} y(t_r) = \mathbf{1}w^T y(t_0) = \begin{pmatrix} w^T y(t_0) \\ \vdots \\ w^T y(t_0) \end{pmatrix}.$$

Hence we complete the proof of time synchronization protocol by taking $b = w^T y(t_0)$.

2.6 Numerical Examples

In this section, we present two examples to demonstrate the effectiveness of the proposed time synchronization protocol.

Example 1: Static topology, small-size WSN

We consider a WSN having 10 sensor nodes and label them as $1, \dots, 10$. We assume the communication topology of the WSN is not changing with respect to time and is shown in Fig. 2.1. Using Theorem 2.1, we choose $\alpha_i = 0.3, \forall i$. The values for different parameters satisfying (A2) and (A3) are listed in Table 2.1. By applying protocol 2.3, the WSN was simulated for 50 seconds. Fig. 2.2 shows a plot of the clock readings for 50 seconds. Fig. 2.3 illustrates a plot of the local control inputs applied to the sensor nodes. Based on the plots, it can be concluded that this WSN achieves distributed time synchronization using the proposed protocol.

Example 2: Switching topology, medium-size WSN

We consider a WSN having 45 sensor nodes and label them with $1, \dots, 45$. We consider the communication links in the WSN are changing randomly with respect to time. Fig. 2.4 shows the union of possible communication topologies of the WSN. We randomly select the parameter values for different sensor nodes from

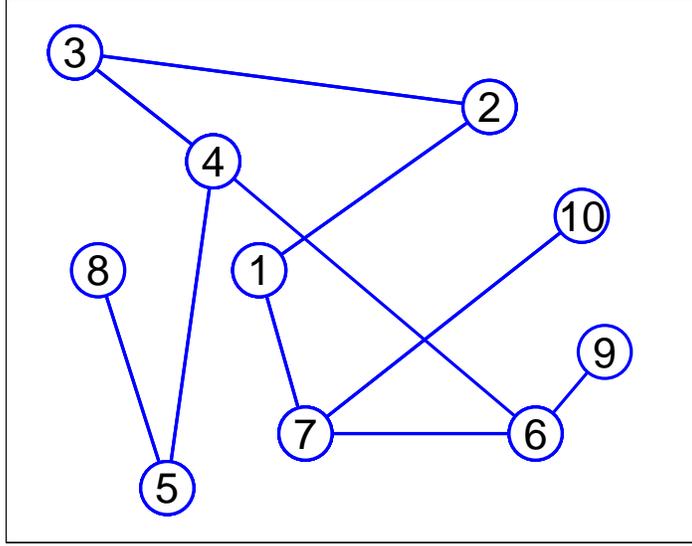


Figure 2.1: Communication topology of a WSN considered in Example 1 having 10 sensor nodes.

Table 2.1: Parameter values used to simulate the WSN as shown in Fig. 2.1

Sensor Node	Time-period (sec)	Actual starting time (sec)	Estimated starting time (sec)	Value of z
1	0.01	0.1	1.5	80
2	0.02	0	-12.5	100
3	0.01	1.3	3.19	100
4	0.05	1.01	-25.19	100
5	0.04	1	10.04	80
6	0.02	0.04	-7.34	100
7	0.1	0.12	2.19	80
8	0.01	0.4	3.25	10
9	0.2	0.16	0	10
10	0.4	0	14.58	10

Table 2.1 and choose $\kappa = 3$. For simulation purposes, we compute the relative time synchronization error of sensor node i with respect to its neighbor nodes, denoted by $e_i(t_r)$, as follows:

$$e_i(t_r) = \sum_{j \in \mathcal{N}(\mathcal{G}_i(t_r))} |(x_i(t_r) - x_j(t_r))|, \quad i = 1, \dots, 10.$$

Applying Theorem 2.1, we simulate the WSN for 80 seconds and select the values

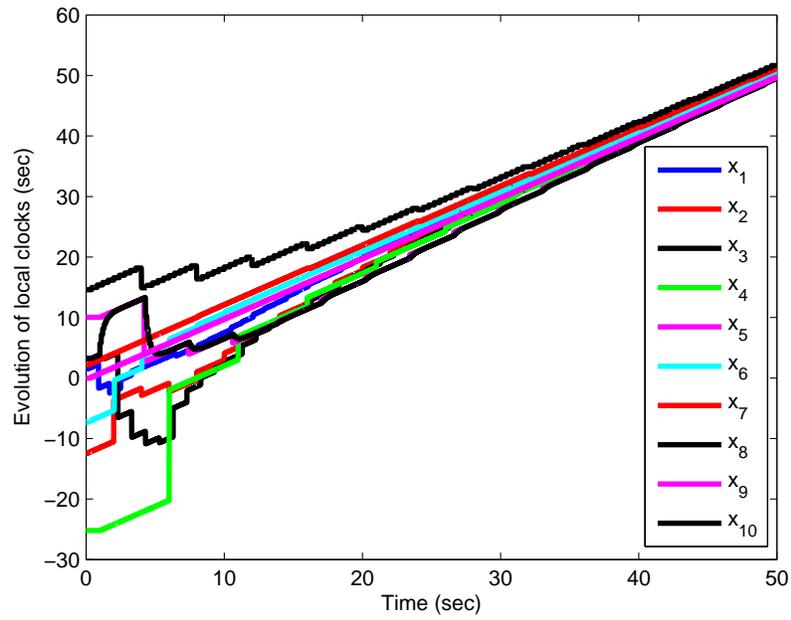


Figure 2.2: Plot of the local clock readings of a WSN considered in Example 1. The topology of the WSN is shown in Fig. 2.1.

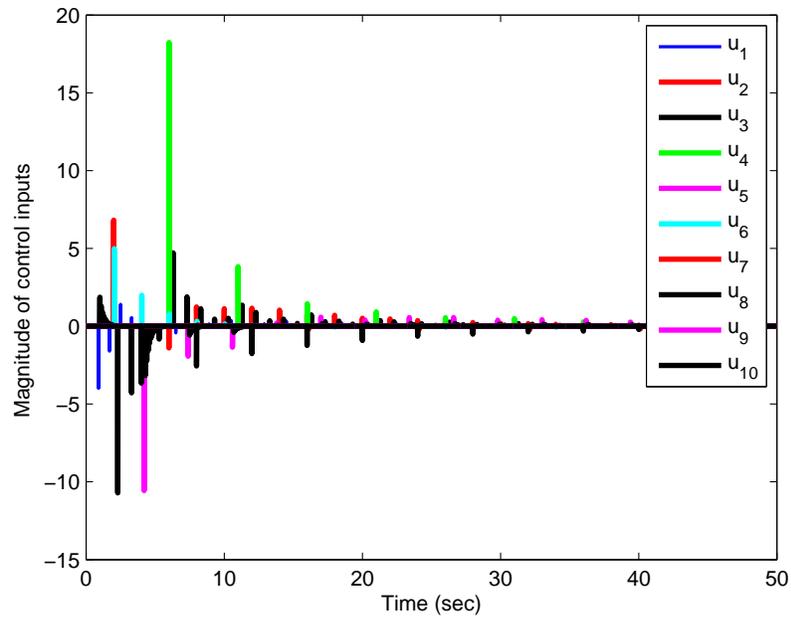


Figure 2.3: Plot of the local control inputs applied to the WSN considered in Example 1. The topology of the WSN is shown in Fig. 2.1.

for α_i at run-time based on the communication topology. Fig. 2.5 shows a plot of the sum of time synchronization errors. Based on the plots, it can be concluded that by using protocol (2.3) and applying Theorem 2.1, this WSN also achieves distributed time synchronization.

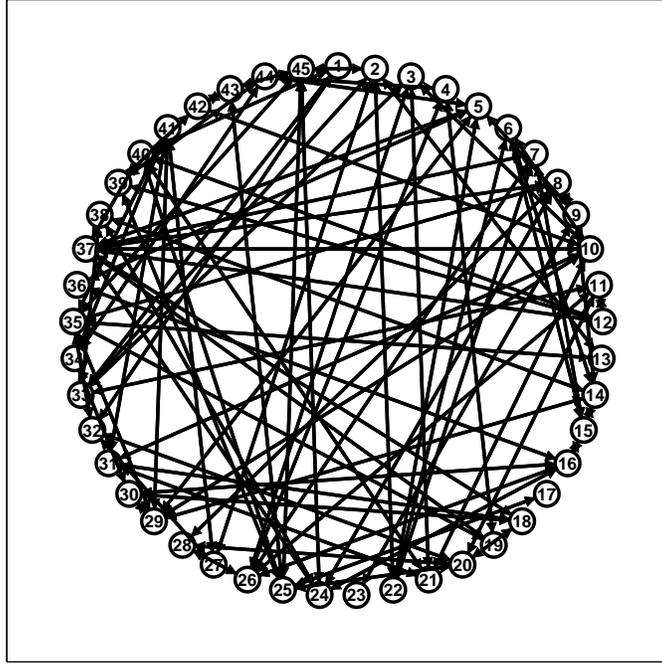


Figure 2.4: The graph of union of communication topologies of a WSN considered in Example 2 having 45 sensor nodes labeled as $1, \dots, 45$. The lines indicate the presence of communication channels.

2.7 Concluding Remarks

In this chapter, we presented a distributed time synchronization protocol for WSNs. The time-periods and starting times of all the sensor nodes were not restricted to be the same. By employing tools from stochastic matrix and algebraic graph theories, we presented the convergence analysis. In this chapter, we considered that a sensor node could access the clock readings of its available neighbors without any delays. In the next chapter, we relax this assumption by modifying the protocol to handle time-delayed measurements from neighbors.

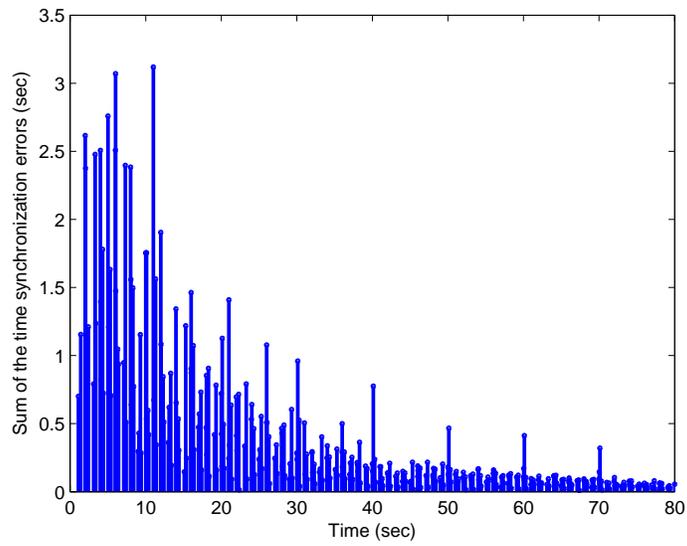


Figure 2.5: Plot of the sum of the relative time synchronization errors in the WSN considered in Example 2. The graph of the union of communication topologies of the WSN is shown in Fig. 2.4.

Chapter 3

Time Synchronization using Unreliable Communication Channels^{*}

In the previous chapter, we studied distributed time synchronization protocols for WSNs and assumed that a sensor node could access the clock readings of its available neighbors without any delays. In this chapter, we relax this assumption and study distributed time synchronization protocols which can handle time delayed clock readings from the neighbor nodes. We assume that each sensor node transmits its clock readings to its neighbor nodes at the time instants when it applies the control input. We consider the communication channels to be unreliable and propose a consensus-based protocol to achieve time synchronization.

Consensus-based time synchronization protocols for WSNs have been studied in [91], [86] and [20]. However, the design of the protocols presented in [91], [86] and [20] assumes that the sensor nodes should periodically exchange and update their clock states at the same time instants; this assumption is unrealistic since it requires synchronization of the clocks, which is the goal of the protocol. This assumption is partially relaxed in [85] and [21], where the authors assume that all the sensor nodes should update their clock states at the same time instants but the information exchange among neighbor nodes may occur at irregular time instants. In [65], a consensus-based time synchronization protocol is presented which compensates for initial time offsets and skew deviations. The protocol assumes synchronous clocks update and information exchange. To our best knowledge, the design of consensus-based time synchronization protocol using asynchronous time updates and unreliable communication links has not been studied in WSNs.

^{*}A version of this chapter has been submitted for publication in [5].

In this chapter, we investigate this problem and present a distributed time synchronization protocol which can work in the presence of unreliable communication channels. It may happen that a communication link is available between two nodes at a given time instant and it may not be available at some other time instants.^(**) We assume that each sensor node knows its time-period but does not know its exact starting time. The objective of the proposed distributed time synchronization protocol is to ensure that the clocks of the sensor nodes are synchronized to a virtual clock, which is not physically present or accessible to any sensor node.

The remainder of this chapter is structured as follows: In Section 3.1, some basic definitions and results from graph and matrix theories are presented. We formulate the problem in Section 3.2 and then present the main results in Section 3.3. The analysis and proof are presented in Section 3.4. Numerical examples to illustrate the effectiveness of the proposed protocol are given in Section 3.5. Finally, concluding remarks are stated in Section 3.6.

3.1 Preliminaries

We refer the interested reader to Section 2.2 for some basic and useful definitions from graph and non-negative matrix theories. For any stochastic matrix $M = [m_{pq}]$, the coefficient of ergodicity is defined as follows [87, 52]:

$$\lambda(M) := 1 - \min_{p_1, p_2} \sum_q \min(m_{p_1 q}, m_{p_2 q}).$$

M is said to be scrambling if $\lambda(M) < 1$. We present a useful lemma here which will be used later.

Lemma 3.1 ([108]). *Let Γ be a finite set of $r \times r$ SIA matrices with the property that for each sequence $S_1, S_2, \dots, S_k \in \Gamma$ of positive length, $\prod_{i=1}^k S_i$ is SIA. Then if $k > 2r^2$, $\prod_{i=1}^k S_i$ is scrambling; and for any infinite sequence of S_1, S_2, \dots , of matrices from Γ , there exists a column vector w such that $\lim_{l \rightarrow \infty} \prod_{i=1}^l S_i = \mathbf{1}w^T$.*

3.2 Problem Formulation

In this section, we formulate the time synchronization problem by presenting a model of each clock in the WSN and the communication strategy for exchanging information.

^(**)Note that this also covers the case of half-duplex communication in a WSN.

3.2.1 System model

We consider a WSN having n sensor nodes. Let x_i , τ_i , t_0^i and \hat{t}_0^i denote the clock value, time-period, actual starting time and estimate of the starting time of sensor node i , respectively, where $i = 1, \dots, n$. Let $T_i = \{\tilde{t}_k^i : k = 0, 1, \dots, \}$ represents the sequence of time instants when the clock of sensor node i updates its state, where $\tilde{t}_k^i := k\tau_i + t_0^i$. Then the time computed locally by sensor node i is expressed as follows:

$$x_i(\tilde{t}_{k+1}^i) = x_i(\tilde{t}_k^i) + \tau_i, \quad x_i(\tilde{t}_0^i) = \hat{t}_0^i, \quad i = 1, \dots, n.$$

To achieve time synchronization, we need to design local control inputs to adjust the clock readings of each sensor node. Let $t_s^i := \tilde{t}_{0+sz_i}^i$ be the time instants when the control input is applied to the clock of sensor node i , where z_i is a known integer, $i = 1, \dots, n$ and $s \in \mathbb{N}$. We can write the following:

$$x_i(t_{s+1}^i) = x_i(t_s^i) + z_i\tau_i + u_i(t_s^i), \quad x_i(t_0^i) = \hat{t}_0^i, \quad i = 1, \dots, n, \quad (3.1)$$

where $u_i(t_s^i)$ denotes the control input or protocol which is to be designed. The clocks in a WSN are said to be synchronized if there exists a common virtual clock in the form of (2.2) such that the following holds:

$$\lim_{k \rightarrow \infty} (x_i(\tilde{t}_k^i) - (\tilde{t}_k^i + b)) = 0, \quad i = 1, \dots, n. \quad (3.2)$$

To compute the local control input, each sensor node communicates with its neighbor nodes. In the next subsection, we present the communication strategy.

3.2.2 Communication strategy

Each sensor node interacts with its available neighbors by transmitting its clock values at those time instants when it updates its control input. As we mentioned earlier, the communication links are unreliable. We consider bidirectional communication links but it may happen that at some particular time instants the communication links are either unidirectional or not available. We make the following assumption:

- (B1) There exists a positive integer c such that sensor node i transmits its information at least once to any possible neighbors during $[t_s^i, t_{s+c-1}^i]$ for any $i = 1, \dots, n$ and $s \in \mathbb{N}$ with $t_s^i > \max\{\tilde{t}_0^j : j = 1, \dots, n\}$.

Assumption (B1) provides an upper bound on the time interval in which each sensor node can successfully transmit its information at least once to any possible

neighbors. In literature, this assumption is also known as bounded intercommunication interval (see e.g. [10, 74]). As the control input update and transmission time instants are asynchronous, we introduce three interaction topologies. The first one describes the transmission of information in the WSN and is modeled by a time-varying graph, denoted by $\mathcal{G}_f(t_s^i)$ where $i = 1, \dots, n$ and $s \in \mathbb{N}$, with vertex set $\mathcal{V}(\mathcal{G}_f(t_s^i)) = \{v_i : i = 1, \dots, n\}$ and edge set $\mathcal{E}(\mathcal{G}_f(t_s^i))$. Vertex v_i represents sensor node i . If sensor node i transmits its state information to node j at t_s^i , then we write $(v_j, v_i) \in \mathcal{E}(\mathcal{G}_f(t_s^i))$. Let $\mathcal{N}(\mathcal{G}_f(t_s^i), i)$ denote the index set of available neighbors of sensor node i at t_s^i .

The second interaction topology represents the union of the transmission topologies, $\mathcal{G}_f(t_s^i)$, where $i = 1, \dots, n$ and $s \in \mathbb{N}$, and is modeled by an undirected graph denoted by \mathcal{G}_n with vertex set $\mathcal{V}(\mathcal{G}_n) = \{v_i : i = 1, \dots, n\}$ and edge set $\mathcal{E}(\mathcal{G}_n)$. If a communication link between sensor node i and j exists at some time instant during the execution of the WSN, then $(v_j, v_i) \in \mathcal{E}(\mathcal{G}_n)$. Let $\mathcal{N}(\mathcal{G}_n, i)$ denote the index set of neighbors of sensor node i , where $i = 1, \dots, n$. It should be noted that the edges of topology \mathcal{G}_n correspond to the existence of communication channels only and they do not guarantee if at some particular time instant, any information has been transmitted on the communication channels. It can be seen that $\mathcal{N}(\mathcal{G}_f(t_s^i), i) \subset \mathcal{N}(\mathcal{G}_n, i) \forall i, s$.

The third interaction topology characterizes the utilization of transmitted information in the WSN and is modeled by another time-varying graph, denoted by $\mathcal{G}_u(t_s^i)$ where $i = 1, \dots, n$ and $s \in \mathbb{N}$, with vertex set $\mathcal{V}(\mathcal{G}_u(t_s^i)) = \mathcal{V}(\mathcal{G}_n)$ and edge set $\mathcal{E}(\mathcal{G}_u(t_s^i))$. If the state information from sensor node i is used at t_s^j to compute $u_i(t_s^j)$, then we write $(v_i, v_j) \in \mathcal{E}(\mathcal{G}_u(t_s^j))$. Let $\mathcal{N}(\mathcal{G}_u(t_s^j), j)$ denote the index set of the neighbors of sensor node j . We mention here that if sensor node i does not receive any information from its neighbor j during $[t_s^i, t_{s+1}^i]$, then it considers the last transmitted (most recent one) information from j to compute its input. We note here that if any sensor node i transmits to j at time instant t_s^i , then we write $(v_j, v_i) \in \mathcal{E}(\mathcal{G}_f(t_s^i))$, but t_s^i may not be the update time of j . Hence, the information transmitted by node i may be used by node j either at the present time instant or some future time instants to compute its input. To ensure that clocks in a WSN can be synchronized, we assume the following:

(B2) The graph, \mathcal{G}_n , is connected.

(B3) For any two sensor nodes i and j , the following holds:

$$t_s^i \in T_j, \quad \forall s.$$

Fig. 3.1 portrays an example of a WSN having three sensor nodes which are labeled with i , j and h . The starting times and time-periods of the sensor nodes are different and we let $c = 4$. The dashed lines represent the time instants when the control update is applied to some sensor node(s). From Fig. 3.1, it can be seen that at time instant t_1 , the communication link between node j and i is not available, and node j transmits to node h only. At time instant t_2 , sensor node h utilizes the delayed state information transmitted by sensor node j at t_1 , but transmits its own state information to sensor node i only. We write that $\mathcal{N}(\mathcal{G}_f(t_1), j) = \{h\}$, $\mathcal{N}(\mathcal{G}_u(t_2), h) = \{j\}$ and $\mathcal{N}(\mathcal{G}_f(t_2), h) = \{i\}$. At time instants t_3 and t_4 , no communication occurs in the WSN. We can also see that during the time interval $[t_1^j, t_4^j]$, sensor node j successfully transmit information at least once to its neighbors.

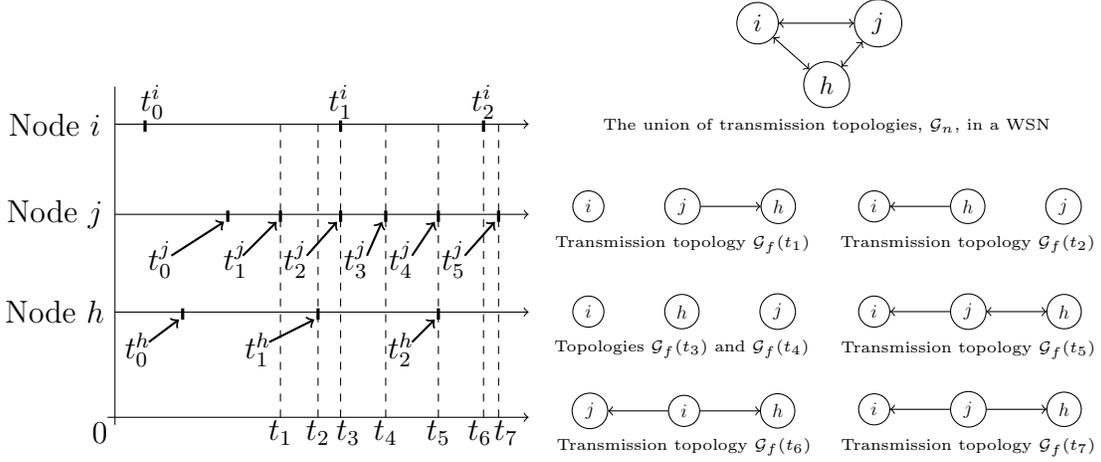


Figure 3.1: Control input update and transmission timings of a WSN having 3 sensor nodes labeled as i , j , and h . The union of transmission topologies in the WSN and the transmission topologies at some particular time instants are also shown.

3.3 Main Results

Before presenting the main results, we define a new time-sequence $T := \{t_r\}$, where $r = 0, 1, 2, \dots$, which is the union of the distinct values of the sequences $\{t_s^i \geq \max\{\tilde{t}_0^j, j = 1, \dots, n\}, i = 1, \dots, n, s \in \mathbb{N}\}$. We order the elements of T such that $t_r < t_{r+1}$. We redefine the interaction topology \mathcal{G}_f in terms of the new time-sequence, T . We write $(v_j, v_i) \in \mathcal{E}(\mathcal{G}_f(t_r))$ if sensor node i transmits information

to j at t_r . If there exists s such that $t_s^i = t_r$ and $j \in \mathcal{N}(\mathcal{G}_n, i)$, and only the most-recent transmitted data is considered, then there exists a time instant, denoted by $t_{r-d_{ij}(t_r)}$, where $d_{ij}(t_r) \in \mathbb{N}$, such that state information from sensor node j was transmitted to i and we can write $i \in \mathcal{N}(\mathcal{G}_f(t_{r-d_{ij}(t_r)}), j)$ and $j \in \mathcal{N}(\mathcal{G}_u(t_r), i)$. We let $d_{ii}(t_r) = 0, \forall i, r$ and define the following:

$$\bar{\alpha}_i(t_r) = \frac{1}{\sum_{j \in \{i\} \cup \mathcal{N}(\mathcal{G}_u(t_r), i)} \alpha_{ij}(t_r)},$$

where $\alpha_{ij}(t_r)$ are time-varying weighting factors. We present the following protocol for achieving time synchronization:

$$u_i(t_r) = \bar{\alpha}_i(t_r) \left(\sum_{j \in \{i\} \cup \mathcal{N}(\mathcal{G}_u(t_r), i)} \alpha_{ij}(t_r) (x_j(t_{r-d_{ij}(t_r)}) - x_i(t_{r-d_{ij}(t_r)})) \right), \quad (3.3)$$

where $\alpha_{ij}(t_r)$ are chosen from a known non-negative and finite set Λ , $\alpha_{ii}(t_r) > 0$, and $\alpha_{ij}(t_r)$ satisfy the following:

$$0 < \sum_{q=d_2}^{d_1} (\bar{\alpha}_i(t_q) \alpha_{ij}(t_q)) < 1, j \neq i, \quad (3.4)$$

where t_{d_1} is the most recent time instant when node j transmitted to node i and t_{d_2} is the second most recent time instant when node j transmitted to node i .

Remark. It should be noted that a sensor node i updates its control input at t_s^i , where $i = 1, \dots, n$ and $s \in \mathbb{N}$. From the definition of time-sequence T , it can be observed that $u_i(t_r) = 0$ when $t_r \in (t_s^i, t_{s+1}^i)$, no matter what values $\alpha_{ii}(t_r)$ takes.

In the literature, a protocol similar to (3.3) has been presented for synchronized distributed systems in various papers (see e.g. [77], [108], [19], [63]). For asynchronous continuous-time systems, a protocol similar to (3.3) has been presented in [107], [17], [38]. For asynchronous discrete-time systems, a protocol similar to (3.3) has been presented in [91] without time-delays and using reliable communication links. In this chapter, we study protocol (3.3) for asynchronous distributed systems where the transmission links are unreliable, the communication topology is changing (and may not be connected at some time instants), and the protocol has to consider time-delayed state information from neighbors. We now present the following theorem.

Theorem 3.1. *If the WSN satisfies assumptions (B1), (B2) and (B3), and the weighting factors belong to Λ and satisfy (3.4), then the clocks of the sensor nodes achieves time synchronization using protocol (3.3).*

To analyze the correctness of Theorem 3.1, we need to substitute protocol (3.3) in the local clock models of the sensor nodes and represent them by a single system. The resulting augmented system is a time-varying system with time-delay terms and has a special structure which will be explored in the next section and the proof will be presented there. We define a new state as follows:

$$y_i(t_r) = x_i(t_r) - (t_r - t_0), \quad i, j = 1, \dots, n. \quad (3.5)$$

We re-write the model expressed in (3.1) as follows:

$$y_i(t_{r+1}) = y_i(t_r) + u_i(t_r), \quad y_i(t_0) = x_i(t_0). \quad (3.6)$$

It should be noted that the value of $y_i(t_p)$ during the time interval $t_p \in [t_s^i, t_{s+1}^i)$ does not change, where $i = 1, \dots, n$ and $s \in \mathbb{N}$. The time-synchronization objective (3.2) can also be expressed as follows:

$$\lim_{r \rightarrow \infty} (y_i(t_r) - b) = 0, \quad i = 1, \dots, n, \quad (3.7)$$

for some b . We re-write the control protocol expressed in (3.3) as follows:

$$u_i(t_r) = \bar{\alpha}_i(t_r) \left(\sum_{j \in \{i\} \cup \mathcal{N}(\mathcal{G}_u(t_r), i)} \alpha_{ij}(t_r) (y_j(t_{r-d_{ij}(t_r)}) - y_i(t_r)) \right). \quad (3.8)$$

The control protocols expressed in (3.8) and (3.3) are the same using the state transformation defined by (3.5). The following lemma states the equivalence between the system models expressed in (3.6) and (3.1).

Lemma 3.2. *If system (3.6) using protocol (3.8) achieves the time-synchronization objective as expressed by (3.7), then system (3.1) using protocol (3.3) achieves time synchronization described by (3.2).*

Proof. Let us consider any initial values of system (3.6). For any sensor node i at time t_r , if there exists s such that $t_r = t_s^i$, then we compute and apply $u_i(t_r)$ to sensor node i . Furthermore, for any node i if $t_r \neq t_s^i$ for any value of s , then we write $y_i(t_{r+1}) = y_i(t_r)$. Therefore, the value of $y_i(t_p)$ during the interval $t_p \in [t_s^i, t_{s+1}^i)$ remains the same, where $i = 1, \dots, n$ and $s \in \mathbb{N}$. By applying protocol (3.8), if system (3.6) achieves the time synchronization objective as expressed by (3.7), then $\lim_{r \rightarrow \infty} u_i(t_r) = 0$ and we can write the following:

$$\lim_{r \rightarrow \infty} (x_i(t_r) - (t_r - t_0)) = b, \quad i = 1, \dots, n,$$

for some b . Let $m_x = \max_i \{z_i \tau_i\}$. For any $\tilde{t}_k^i > t_0$ and $i = 1, \dots, n$, we state that there exists $r \in \mathbb{N}$ such that $t_{r(k)} \leq \tilde{t}_k^i < t_{r(k)+1}$ and $|\tilde{t}_k^i - t_{r(k)}| < m_x$. We can further state that if $k \rightarrow \infty$, then $r(k) \rightarrow \infty$. As $\lim_{r(k) \rightarrow \infty} u_i(t_{r(k)}) = 0$, under assumption (B2) we conclude the following:

$$\lim_{k \rightarrow \infty} (x_i(\tilde{t}_k^i) - (\tilde{t}_k^i - t_0 + b)) = 0, \quad i = 1, \dots, n.$$

□

The following lemma computes an upper bound on the time interval in which each sensor node can utilize the information transmitted by its neighbor.

Lemma 3.3. *Let $g = \frac{\max_i \{z_i \tau_i\}}{\min_i \{z_i \tau_i\}}$ where $i = 1, \dots, n$. For any sensor node $i = 1, \dots, n$ and $s \in \mathbb{N}$, the number of elements in the set $\{t_p : t_p \in [t_s^i, t_{s+1}^i]\}$ is not greater than m , where $m = (\lceil g \rceil)(n-1) + 1$, and $\lceil g \rceil$ denotes the smallest integer not less than g . For any two neighboring nodes i and j , if node i transmits to node j at t_r , then node j utilizes the information, transmitted by node i at t_r , no longer than $t_{r+m(c-1)+1}$.*

Proof. Let j be the sensor node such that $z_j \tau_j \geq z_i \tau_i$ where $i = 1, \dots, n, i \neq j$. When sensor node j updates its control input at t_s^j , the maximum number of times any sensor node i can update its control input before the time instant t_{s+1}^j can not exceed $\lceil g \rceil$, where $i = 1, \dots, n, i \neq j$. Since there are $n-1$ possible values of sensor node i (excluding $i = j$), we conclude that the number of elements in $\{t_p : t_p \in [t_s^j, t_{s+1}^j]\}$ is not greater than m . Furthermore, using assumption (B1), we can state that the number of elements in $\{t_p : t_p \in [t_s^j, t_{s+c-1}^j]\}$ is not greater than $m(c-1)$. □

Let $e = 1, \dots, m(c-1)$ where m is computed using Lemma 3.3 and c is a known integer as defined in assumption (B1). Let us define the matrices $A_e(t_r) = [a_{pq}^e(t_r)]$ as follows:

$$a_{pq}^e(t_r) = \begin{cases} \bar{\alpha}_p(t_r) \times \alpha_{pq}(t_r) & \text{if } p \in \mathcal{N}(\mathcal{G}_f(t_{r-e}), q) \text{ and } e = \min\{e' : t_{r-e'} = t_s^q, \exists s\}, \\ 0 & \text{otherwise.} \end{cases}$$

We define another matrix $A_0(t_r) = [a_{pq}^0(t_r)]$ as follows:

$$a_{pq}^0(t_r) = \begin{cases} \bar{\alpha}_p(t_r) \times \alpha_{pq}(t_r) & \text{if } p \in \mathcal{N}(\mathcal{G}_f(t_r), q) \text{ and } \exists s \text{ s.t. } t_r = t_s^q, \\ 1 - \sum_{d=0}^{m(c-1)} \sum_{l=1, l \neq p}^n a_{pl}^d(t_r) & \text{if } p = q, \\ 0 & \text{otherwise.} \end{cases}$$

Let $y(t_r) = [y_1(t_r), \dots, y_n(t_r)]^T$ denote the augmented state vector. The dynamics of the augmented system can be written as follows:

$$y(t_{r+1}) = \sum_{d=0}^{m(c-1)} A_d(t_r) y(t_{r-d}). \quad (3.9)$$

It should be noted that augmented system expressed in (3.9) is a time-varying system having time-delay terms. With this preparation, we present the proof in the next section.

3.4 Technical Analysis

To proceed with the proof, let $\Phi(t_r) = [y(t_r)^T, y(t_{r-1})^T, y(t_{r-2})^T, \dots, y(t_{r-m(c-1)})^T]^T$, where $r > m(c-1) + 1$; then we can write the following:

$$\Phi(t_{r+1}) = \Psi(t_r) \Phi(t_r),$$

where

$$\Psi(t_r) = \begin{bmatrix} A_0(t_r) & A_1(t_r) & \dots & A_{m(c-1)}(t_r) & A_{m(c-1)}(t_r) \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{bmatrix}. \quad (3.10)$$

The following lemma states some useful properties of $\Psi(t_r)$.

Lemma 3.4. *Let α_{\min} and α_{\max} denote the minimum and maximum elements of Λ . Given the matrix $\Psi(t_r)$ as expressed in (3.10), the following statements hold:*

- (a) *All non-zero elements of $\Psi(t_r)$ are not less than η , where $\eta = \frac{\alpha_{\min}}{n\alpha_{\max}}$, and not greater than 1.*
- (b) $\Psi(t_r)\mathbf{1} = \mathbf{1}$.
- (c) *The possible numerical values of $\Psi(t_r)$ are finite.*

Proof. By the definition of the matrices $A_h(t_r)$, $h = 1, \dots, m(c-1)$, we can see that each non-zero element of $A_h(t_r)$ is less than 1. A sensor node can have maximum $n-1$ neighbors. Let us consider a WSN where sensor node k has $n-1$ neighbors, where $k = 1, \dots, n$. At a given time instant t_r , we consider three extreme possible values of the weighting factors as follows:

- (i) We choose $\alpha_{ki}(t_r) = \alpha_{\min}$, where $i, k = 1, \dots, n$, and obtain $\bar{\alpha}_k(t_r) = \frac{1}{n\alpha_{\min}}$.

- (ii) We choose $\alpha_{kj}(t_r) = \alpha_{\min}$ for some neighbor node j , where $k = 1, \dots, n$ and $\alpha_{ki}(t_r) = \alpha_{\max}$, where $i = 1, \dots, n$, $i \neq j$, and obtain $\bar{\alpha}_k(t_r) = \frac{1}{\alpha_{\min} + (n-1)\alpha_{\max}}$.
- (iii) We choose $\alpha_{ki}(t_r) = \alpha_{\max}$, where $i, k = 1, \dots, n$, and obtain $\bar{\alpha}_k(t_r) = \frac{1}{n\alpha_{\max}}$.

Using the definition of matrices $A_d(t_r)$, where $d = 1, \dots, m(c-1)$, and considering (i), (ii) and (iii), we can observe that all non-zero elements of $A_d(t_r)$ are greater than η for any values of the weighting factors. Using similar arguments, we can state that all off-diagonal elements of $A_0(t_r)$ are also greater than η . Let $k = 1, \dots, n$; we use the definition of diagonal elements of $A_0(t_r)$ to state the following:

$$\begin{aligned}
a_{kk}^0(t_r) &= 1 - \frac{\sum_{l=1, l \neq k}^n \alpha_{kl}(t_r)}{\alpha_{kk}(t_r) + \sum_{l=1, l \neq k}^n \alpha_{kl}(t_r)}, \\
&= \frac{\alpha_{kk}(t_r)}{\alpha_{kk}(t_r) + \sum_{l=1, l \neq k}^n \alpha_{kl}(t_r)}, \\
&> \frac{\alpha_{\min}}{n\alpha_{\max}}.
\end{aligned}$$

Thus, the diagonal elements of $A_0(t_r)$ are greater than η for any values of the weighting factors. The sum of each row in $\Psi(t_r)$ is 1. Hence, statements (a) and (b) are correct. We know the number of sensor nodes is finite, the elements of Λ are finite, and all possible communication topologies of the WSN are finite. Therefore, all possible values of $\Psi(t_r)$ where $r = 0, 1, 2, \dots$, are finite. Hence, statement (c) holds. \square

The following lemma presents useful results which are required to complete the proof.

Lemma 3.5 ([107]). *Let S_1, S_2, \dots, S_m be $n \times n$ nonnegative matrices and let*

$$P = \begin{bmatrix} S_1 & S_2 & S_3 & \dots & S_{m-1} & S_m \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix},$$

$$Q = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 \\ I & 0 & 0 & \dots & 0 & 0 \\ 0 & I & 0 & \dots & 0 & 0 \\ 0 & 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I & 0 \end{bmatrix},$$

and $R_k = P + Q^k$ for any $k \in \{1, 2, \dots, m-1\}$. If $\mathcal{G}(\sum_{i=1}^m S_i)$ has a spanning tree, then $\mathcal{G}(R_k)$ contains a spanning tree such that the root vertex of that spanning tree has a self-loop in $\mathcal{G}(R_k)$.

Proof. Let $S = S_1 + S_2 + \dots + S_m$, the vertex set of $\mathcal{G}(S)$ be $\mathcal{V}(\mathcal{G}(S)) = \{s_1, s_2, \dots, s_n\}$ and edge set $\mathcal{E}(\mathcal{G}(S))$. Let the vertex sets of $\mathcal{G}(P)$ and $\mathcal{G}(Q)$ be $\mathcal{V}(\mathcal{G}(P)) = \{p_1, p_2, \dots, p_{mn}\}$ and $\mathcal{V}(\mathcal{G}(Q)) = \{q_1, q_2, \dots, q_{mn}\}$, respectively, and edge sets of $\mathcal{G}(P)$ and $\mathcal{G}(Q)$ be $\mathcal{E}(\mathcal{G}(P))$ and $\mathcal{E}(\mathcal{G}(Q))$, respectively. Let the vertex set for the graphs $\mathcal{G}(R_k)$, where $k = 1, 2, \dots, m-1$, be the same and expressed as $\mathcal{V}(\mathcal{G}(R_k)) = \{r_1, r_2, \dots, r_{mn}\}$. Let the edge set of $\mathcal{G}(R_k)$ be denoted by $\mathcal{E}(\mathcal{G}(R_k))$, where $\mathcal{E}(\mathcal{G}(R_k)) = \mathcal{E}(\mathcal{G}(P)) \cup \mathcal{E}(\mathcal{G}(Q^k))$. It can be observed that for any $i = 1, \dots, n$, we have $(q_i, q_i) \in \mathcal{E}(\mathcal{G}(Q))$. Furthermore, for any $i = 1, \dots, n$, we have $(q_i, q_{i+n}), (q_{i+n}, q_{i+2n}), (q_{i+2n}, q_{i+3n}), \dots, (q_{i+(m-2)n}, q_{i+(m-1)n}) \in \mathcal{E}(\mathcal{G}(Q))$. Investigating the elements in $\mathcal{E}(\mathcal{G}(Q^k))$, we observe the following:

$$\left\{ (q_i, q_i), (q_i, q_{i+n}), (q_i, q_{i+2n}), \dots, (q_i, q_{i+kn}), (q_{i+n}, q_{i+n(k+1)}), (q_{i+2n}, q_{i+n(k+2)}), \dots, (q_{i+n(m-k-1)}, q_{i+n(m-1)}) \right\} \in \mathcal{E}(\mathcal{G}(Q^k)), \quad \forall i = 1, \dots, n.$$

If $(s_i, s_j) \in \mathcal{E}(\mathcal{G}(S))$, then we can always find $p' \in [0, mn-1]$ such that $(r_{i+p'}, r_j) \in \mathcal{E}(\mathcal{G}(R))$. Therefore if $\mathcal{G}(S)$ has a spanning tree and s_i is its root vertex, then we can claim that $\mathcal{G}(R_k)$ also has spanning tree and r_i is its root vertex. Furthermore, if $S_1 \geq \mu I$ and $\mathcal{G}(S)$ has a spanning tree, then the root vertex of the spanning tree in $\mathcal{G}(R_k)$ has a self-loop. \square

Using Lemma 3.4, we observe that $\Psi(t_r)$ is a stochastic matrix, $\forall r$ such that $r > m(c-1)$. We further obtain that during the time interval $[t_r, t_{r-m(c-1)}]$, every sensor node receives information at least once from its neighbors. Under assumption (B2), we then conclude that the graph obtained by the union of topologies $\mathcal{G}_u(t_r), \mathcal{G}_u(t_{r-1}), \dots, \mathcal{G}_u(t_{r-m(c-1)})$ contains a spanning tree for any value of r such that $r > m(c-1)$. Applying Lemmas 3.5 and 2.1, we can establish that $\Psi(t_r)$ is SIA.

Using Lemma 3.4(c), we know that all possible $\Psi(t_r)$ are finite. Any product which involves a finite combination of $\Psi(t_r)$, where $r > m(c-1)$, is SIA. So,

we can apply Lemma 3.1 to get the following conclusion:

$$\lim_{l \rightarrow \infty} \prod_{r=c(m-1)}^l \Psi(t_r) = \mathbf{1}w^T.$$

The vector w depends on the order in which the matrices $\Psi(t_r)$ are multiplied, where $r = m(c-1), m(c-1)+1, m(c-1)+2, \dots$, and it can not be computed in advance. We can further write the following:

$$\lim_{r \rightarrow \infty} y(t_r) = \mathbf{1}w^T y(t_0) = \begin{pmatrix} w^T y(t_0) \\ \vdots \\ w^T y(t_0) \end{pmatrix}.$$

Hence we complete the proof by taking $b = w^T y(t_0)$.

3.5 Numerical Examples

In this section, we present two examples to illustrate the effectiveness of the proposed protocol.

Example 1: Small-size WSN

We consider a WSN having 8 sensor nodes. We assume the communication links are unreliable and choose $c = 5$, $\Lambda = \{1, 2, 3\}$. The union of possible communication topologies in the WSN is shown in Fig. 3.2. The values of different parameters are listed in Table 3.1. For simulation purposes, we compute the relative time synchronization error of sensor node i with respect to its neighbor nodes, denoted by $e_i(t_r)$, as follows:

$$e_i(t_r) = \sum_{j \in \mathcal{N}(\mathcal{G}_u(t_r), i)} |(x_i(t_r) - x_j(t_r))|, \quad i = 1, \dots, 8.$$

By applying the proposed protocol in 3.3, the WSN was simulated for 70 seconds. Fig. 3.3 illustrates a plot of the local clock readings in the WSN. The magnitudes of the local control inputs are plotted in Fig. 3.4. The sum of relative time synchronization errors is shown in Fig. 3.5. Based on the plots, we can conclude that this WSN achieves time synchronization using the proposed protocol.

Example 2: Medium-size WSN

We consider a medium-size WSN having 60 sensor nodes. We assume the communication links are unreliable and choose $c = 7$, $\Lambda = \{0.1, 0.2, 0.3, 0.4, \dots, 3\}$. Fig. 3.6

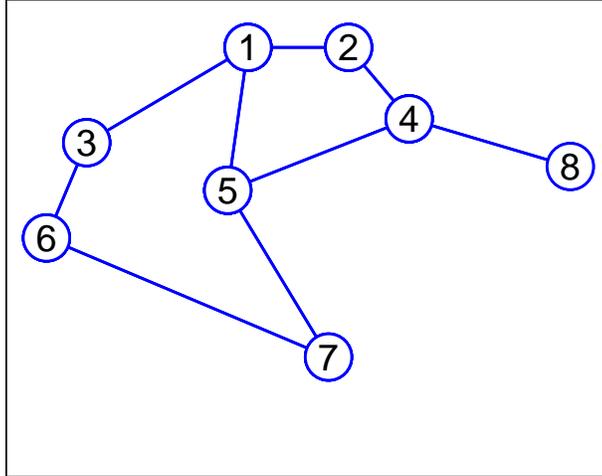


Figure 3.2: The union of possible communication topologies of a WSN considered in Example 1 having 8 sensor nodes. The lines indicate the presence of possible communication channels for transmitting information.

Table 3.1: Parameter values used to simulate the WSN as shown in Fig. 3.2

Sensor Node	Time-period (sec)	Actual starting time (sec)	Estimated starting time (sec)	Value of z
1	0.05	1.4	-3.5	100
2	0.01	0.3	4.69	100
3	0.1	2	0	10
4	0.04	2.01	-12.5	10
5	0.02	1	13.25	100
6	0.01	0.04	-15.19	10
7	0.1	0.12	20.04	10
8	0.2	0	-7.34	10

shows the union of possible communication topologies of the WSN. We randomly select the parameter values for different sensor nodes from Table 3.1. For simulation purposes, we compute the relative time synchronization error of sensor node i with respect to its neighbor nodes, denoted by $e_i(t_r)$, as follows:

$$e_i(t_r) = \sum_{j \in \mathcal{N}(\mathcal{G}_u(t_r), i)} |(x_i(t_r) - x_j(t_r))|, \quad i = 1, \dots, 60.$$

We simulate the WSN for 35 seconds by applying the proposed protocol. Fig. 3.7 shows a plot of the sum of relative time synchronization errors. Based on the simulation examples, it can be concluded that by using protocol (3.3), this WSN

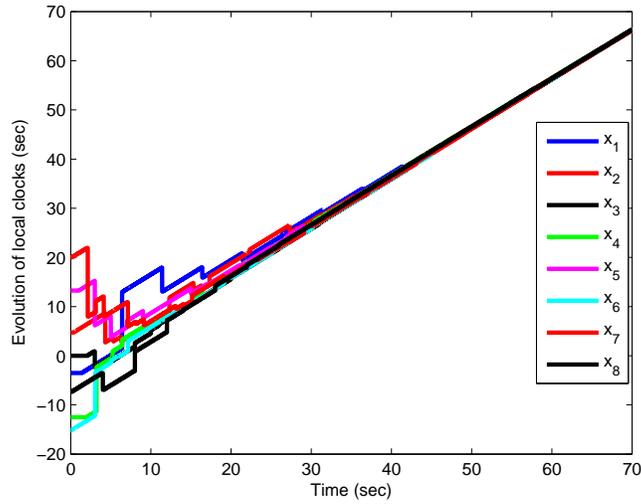


Figure 3.3: Plot of the local clock readings of a WSN considered in Example 1. The plot demonstrates the effectiveness of the proposed protocol to achieve time synchronization. The union of possible communication topologies of the WSN is shown in Fig. 3.2.

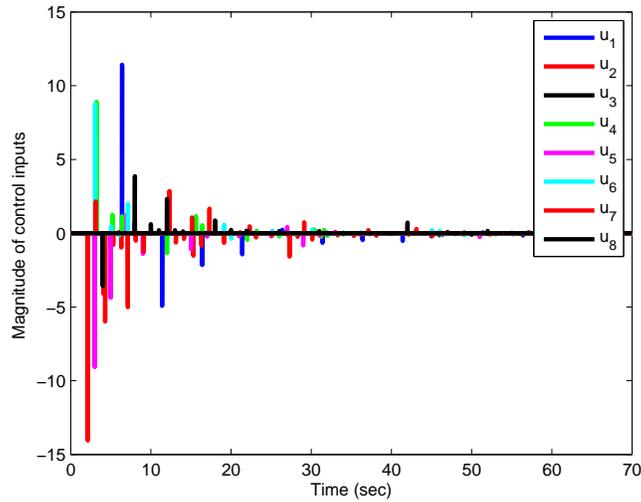


Figure 3.4: Plot of the local control inputs applied to the WSN considered in Example 1. The union of possible communication topologies of the WSN is shown in Fig. 3.2.

also achieves relative time synchronization.

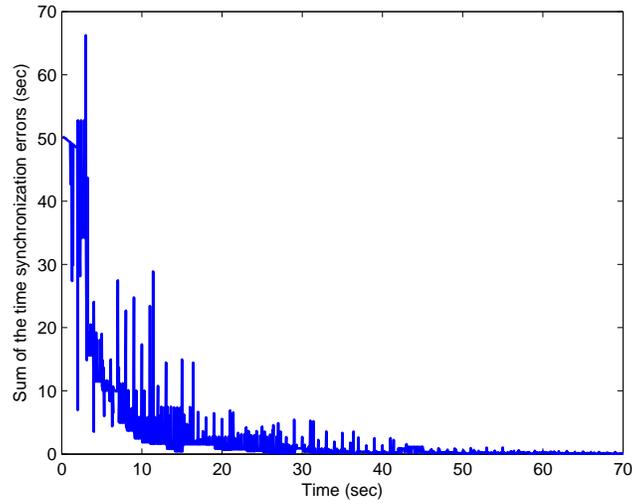


Figure 3.5: Plot of the sum of the relative time synchronization errors in the WSN considered in Example 1. The graph of the union of communication topologies of the WSN is shown in Fig. 3.2.

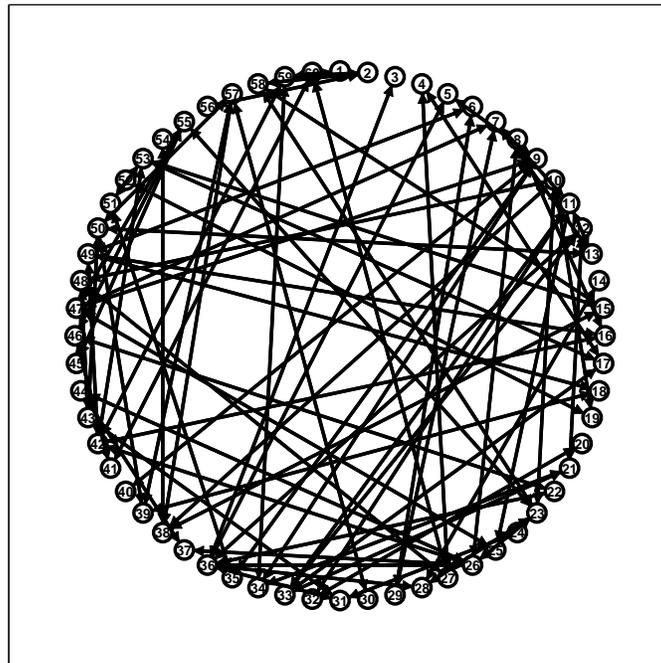


Figure 3.6: The graph of union of communication topologies of a WSN considered in Example 2 having 60 sensor nodes. The lines indicate the presence of communication channels only and do not indicate if at any particular time instant any information is exchanged on the communication link.

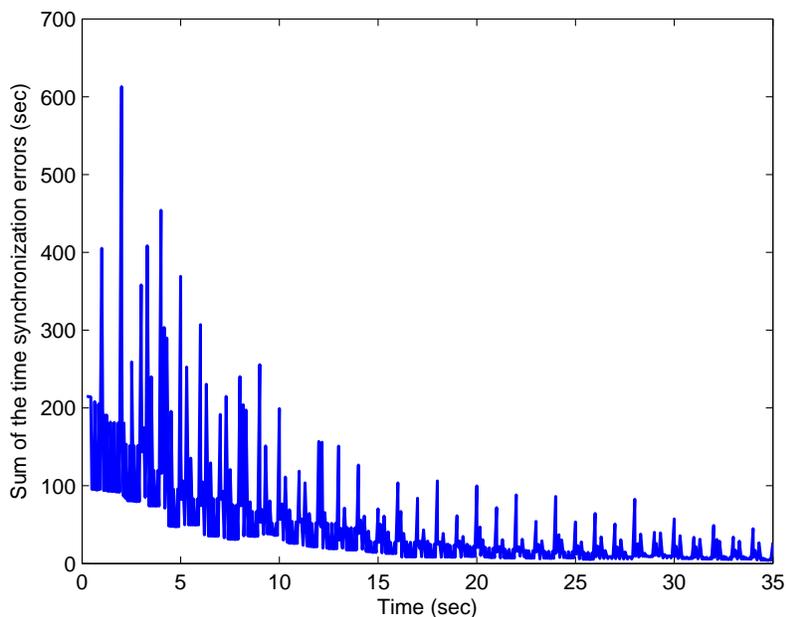


Figure 3.7: Plot of the sum of the relative time synchronization errors in the WSN considered in Example 2. The graph of the union of communication topologies of the WSN is shown in Fig. 3.6.

3.6 Concluding Remarks

Time synchronization is a basic assumption in most distributed control and monitoring applications. In this chapter, we investigated the design of a distributed time synchronization protocol for WSNs by borrowing tools from linear control theory and consensus algorithms. The protocol design considers asynchronous framework where the interaction graphs are not connected at some time instants, the communication channels are unreliable and time-delayed information is utilized to compute local control inputs. By employing tools from stochastic matrix and algebraic graph theories, we presented the convergence analysis of the proposed algorithm. Simulation results were provided to illustrate the effectiveness of the proposed protocol.

Chapter 4

Optimal \mathcal{H}_2 Filtering in WSNs^{*}

4.1 Introduction

A wireless sensor network (WSN) consists of sensor nodes which are spatially distributed to monitor some process of interest. Each sensor node has limited computation, communication and battery resources, which must be efficiently utilized. The efficient utilization of battery resources introduces constraints on the time synchronization and the sampling rates of the sensor nodes. The non-synchronization of sensor nodes introduces sampling jitters. The presence of sampling jitters gives rise to non-uniform sampling. Most of the standard control and monitoring techniques rely on uniform and synchronized sampling. Therefore, to apply the standard techniques we need to design an algorithm to minimize the effect of unsynchronized sampling. In this chapter, we propose a method to handle sampling jitters and reconstruct uniformly sampled measurements obtained using a WSN. We assume that the sampling jitter is known and fixed for each sensor, therefore, it can be modeled by a continuous-time drift (delay/advance) transfer function.

To increase the sampling rate of a WSN, distributed sampling can be employed [53, 39]. As illustrated in Chapter 1, distributed sampling uses a combination of M sensors to sample a common input signal. Each sensor samples at a rate of $1/M$ samples/sec. The outputs of these sensors are multiplexed to produce a fast sampled signal. Therefore, multiple sensors can be used to distribute the sampling load across the sensors. The advantage of distributed sampling is that slow sampling sensors can be added in parallel to act like a fast sampling sensor.

A popular approach to model distributed sampling sensors is to use filter banks [57, 30, 101]. Digital multi-rate filter banks have been widely studied in signal processing; Vaidyanathan's book [100] gives a comprehensive historical survey of

^{*}A version of this chapter has been published in [4].

the literature till 1992. The design of multi-rate filters using \mathcal{H}_∞ optimization was originally proposed in [89]. This design was later extended to discrete-time filter banks in [26] and hybrid filter banks in [90]. However, in both cases the authors considered rational transfer functions whereas in this chapter fractional delays are considered which are not rational transfer functions in continuous-time. In [56], the authors considered delays which are integer multiples of the sampling period, for the design of filter bank. In [75], the authors considered fractional delays for the design of filter bank using \mathcal{H}_∞ optimization. To our best knowledge, the filter bank design for fractional delays using \mathcal{H}_2 optimization has not been studied in WSNs. The main contribution of this chapter is to propose a technique for discretization of fractional delays and reconstruction of uniformly sampled measurements by designing synthesis filters based on the minimization of the \mathcal{H}_2 norm of the error system.

The \mathcal{H}_2 norm of a stable system has a meaningful interpretation for deterministic as well as stochastic inputs (see, e.g., [27]). If we consider the input to be a unit impulse, then the average of the output energy equals the square of the \mathcal{H}_2 norm of the system. Furthermore, if we consider the input to be white noise having zero mean and unit variance, then the root-mean-square of the output equals the \mathcal{H}_2 norm of the system. Therefore, the design objective was chosen based on the \mathcal{H}_2 norm optimization, which results in the optimal and stable synthesis filters. The design procedure is implemented in a control room where computing power is abundant.

The remainder of this chapter is structured as follows: In Section 4.2, the methods of discretization of a sampled-data and a fractional delay system are presented. The problem formulation and system modeling using multi-rate hybrid filter bank are discussed in Section 4.3. The optimal \mathcal{H}_2 filter design procedure for obtaining infinite impulse response (IIR) and finite impulse response (FIR) filters is presented in Section 4.4. An example to demonstrate the effectiveness of the proposed design procedure is given in Section 4.5.

4.2 Preliminaries

Let us consider a linear time-invariant system, G , in continuous (discrete) time. The impulse response of G is denoted by the lower case g and its state-space realization by a 2×2 packed notation. Thus in continuous-time

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] := D + C(sI - A)^{-1}B,$$

and in discrete-time

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] := D + C(zI - A)^{-1}B.$$

There are different methods to discretize a continuous-time system such as step-invariant, impulse-invariant, bilinear and norm-invariant discretization. In this thesis, the step-invariant and norm-invariant discretizations are considered, which are briefly explained below.

4.2.1 Step-invariant discretization

Let us consider a continuous-time system denoted by G and having the following representation:

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right], \quad (4.1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$ and $C \in \mathbb{R}^{q \times n}$. The step-invariant discretization of G is defined as $G_d = SGH$, where H is the zero-order hold and S is the sampler, both operating with period h . The discrete-time system, G_d , is given as follows (see, e.g., [27]):

$$G_d(z) = \left[\begin{array}{c|c} e^{hA} & \int_0^h e^{\tau A} d\tau B \\ \hline C & 0 \end{array} \right]. \quad (4.2)$$

4.2.2 Norm-invariant discretization

The norm-invariant discretization of a continuous-time system as in (4.1) is given as

$$G_j(z) = \left[\begin{array}{c|c} A_j & B_j \\ \hline C & 0 \end{array} \right], \quad (4.3)$$

where $A_j = e^{hA}$ and B_j is a matrix satisfying $B_j B_j^T = \int_0^h e^{\tau A} B B^T e^{A^T \tau} d\tau$. The mathematical details for computation of B_j are given in [27].

4.2.3 Norm of sampled systems

A sampled system, $SG : \mathcal{L}_2(\mathbb{R}) \rightarrow \ell_2(\mathbb{Z})$, maps continuous-time signals to discrete-time signals. The \mathcal{H}_2 norm of SG is defined as the average of total energy of outputs when impulses are applied in one period to the input channels. The \mathcal{H}_2 norm of SG can be expressed as follows [27]:

$$\|SG\|_2 = \left(\frac{1}{h} \int_0^h \left(\sum_{i=1}^p \|SG_c \delta(t - \tau) e_i\|_2^2 \right) d\tau \right)^{1/2},$$

where $\delta(t)$ represents a continuous-time unit impulse function and e_i represent the standard basis vectors in \mathbb{R}^p . The following lemma is quite useful to relate the \mathcal{H}_2 norm of SG to the norm a discrete-time system.

Lemma 4.1. [54] *The \mathcal{H}_2 norm of SG is related to the norm of a certain discrete-time system as expressed below:*

$$\|SG\|_2 = \frac{1}{\sqrt{h}} \|G_j\|_2,$$

where G_j is defined as in 4.3.

Proof. The impulse response, $g(t)$, of G can be written as follows:

$$G\delta(t) = g(t) = Ce^{At}B1(t).$$

where $1(t)$ is the continuous-time unit step function. Then, we can write the following:

$$\begin{aligned} G\delta(t - \tau) &= Ce^{A(t-\tau)}B1(t - \tau), \\ SG\delta(t - \tau) &= \{0, Ce^{A(h-\tau)}B, Ce^{A(2h-\tau)}B, \dots, Ce^{A(kh-\tau)}B, \dots\}, \\ SG\delta(t - \tau) &= \sum_{k=1}^{\infty} (Ce^{A(kh-\tau)}B). \end{aligned}$$

For multi-input system, we can write the following:

$$\begin{aligned} \sum_{i=1}^p \|SG\delta(t - \tau)e_i\|_2^2 &= \text{trace} \left(\sum_{k=1}^{\infty} \left(Ce^{A(kh-\tau)}BB^T e^{A^T(kh-\tau)}C^T \right) \right), \\ \frac{1}{h} \int_0^h \left(\sum_{i=1}^p \|SG\delta(t - \tau)e_i\|_2^2 \right) d\tau &= \\ \frac{1}{h} \text{trace} \left(\sum_{k=1}^{\infty} \left(Ce^{Akh} \int_0^h \left(e^{-A\tau}BB^T e^{-A^T\tau} d\tau \right) e^{A^T kh}C^T \right) \right). \end{aligned}$$

A change of variables from $(k - 1) \rightarrow k$ and $(h - \tau) \rightarrow \tau$ yields the following:

$$\begin{aligned} \|SG\|_2^2 &= \frac{1}{h} \text{trace} \left(\sum_{k=0}^{\infty} Ce^{Akh}B_jB_j^T e^{A^T kh}C^T \right), \\ \|SG\|_2^2 &= \frac{1}{h} \text{trace} \left(\sum_{k=0}^{\infty} CA_j^k B_j B_j^T A_j^{T k} C^T \right), \\ \|SG\|_2^2 &= \frac{1}{h} \|G_j\|_2^2. \end{aligned}$$

□

4.2.4 Discretizing a time-delay transfer matrix

Let us consider a linear, time-invariant (LTI) time-delay system with transfer matrix given as

$$G_{td}(s) = e^{-\tau s}G(s),$$

where $G(s)$ is a real rational and strictly-proper transfer matrix. We note here that the delay transfer matrix is not rational. If τ is an integer multiple of the sampling period h , then it can be discretized by using a discrete-time delay. However, if τ is not an integer multiple of h , then we need a method to discretize the time-delay system. In this section, we derive a finite-dimensional discrete-time transfer matrix for a time-delay system where the delay is not an integer multiple of h . For such a system, the delay can be decomposed as

$$\tau = lh - \tau_0,$$

where $l \in \mathbb{Z}$ and τ_0 is a fractional delay. The time-delay transfer matrix can also be described by equations as follows:

$$G_{td}(s) = \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t - \tau). \end{cases} \quad (4.4)$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^r$ and $y(t) \in \mathbb{R}^m$ are the state, input and output vectors, respectively; A , B and C are constant matrices of appropriate dimensions. Using the step-invariant discretization, let the discrete-time system be denoted by $G_d = SG_{td}H$. Let the discrete-time state and input vectors be denoted by x_d and u_d , respectively. The discrete-time state equation can be written as follows:

$$x_d(k+1) = e^{hA}x_d(k) + \int_0^h e^{tA}dtBu_d(k).$$

The output equation can be written as

$$y_d(k) = Cx[(k-l)h + \tau_0]. \quad (4.5)$$

Using the state equation, we can also write

$$x[(k-l)h + \tau_0] = e^{\tau_0 A}x_d(k-l) + \int_0^{\tau_0} e^{tA}dtBu_d(k-l).$$

Substituting this into (4.5), the following is obtained:

$$y_d(k) = Ce^{\tau_0 A}x_d(k-l) + C \int_0^{\tau_0} e^{tA}dtBu_d(k-l).$$

The discrete-time transfer matrix for the LTI time-delay system, expressed by (4.4), can be written as

$$G_d(z) = z^{-l} \left[\begin{array}{c|c} A_d & B_d \\ \hline C_d & D_d \end{array} \right], \quad (4.6)$$

where $A_d = e^{hA}$, $B_d = \int_0^h e^{tA} dt B$, $C_d = C e^{\tau_0 A}$ and $D_d = C \int_0^{\tau_0} e^{tA} dt B$. Using a similar derivation for the norm-invariant discretization, the discrete-time transfer matrix can be written in a similar way as (4.6), where A_d and C_d remain the same and B_d and D_d are redefined. B_d is a matrix satisfying

$$B_d B_d^T = \int_0^h e^{\tau A} B B^T e^{A^T \tau} d\tau.$$

D_d is defined as $D_d = C B_{dd}$, where B_{dd} is a matrix satisfying

$$B_{dd} B_{dd}^T = \int_0^{\tau_0} e^{\tau A} B B^T e^{A^T \tau} d\tau.$$

4.2.5 Multi-rate components

Let us consider an up-sampler, denoted by E , whose function is to increase the sampling rate by inserting zeros between the input measurements [100]. Let the discrete-time input signal to E be denoted by $u[n]$. If we consider the up-sampling rate to be 3, then we can write the following:

$$E : \{u[0], u[1], u[2], \dots\} \mapsto \{u[0], 0, 0, u[1], 0, 0, u[2], 0, 0, \dots\}.$$

A down-sampler decreases the sampling rate. Let a down-sampler be denoted by W and the input to W by $f[n]$. If we consider the down-sampling rate to be 3, then we can write the following:

$$W : \{f[0], f[1], f[2], f[3], f[4], \dots\} \mapsto \{f[0], f[3], \dots\}.$$

With these definitions, we proceed to the next section and formulate the problem.

4.3 Problem Formulation

A WSN employing distributed sampling can be modeled by a filter bank having analysis and synthesis filter banks [56]. The analysis filter bank represents the slow sampling sensor nodes. Each slow sampling sensor node samples the input signal and transmits it over a communication channel. Due to sampling jitters, the sampling pattern is non-uniform and the objective is to design a synthesis filter

bank to minimize the effects of sampling jitters. In order to optimally design the synthesis filters, we consider an error system, denoted by K . The schematic of K is shown in Fig. 4.1, where $f(t)$, $\hat{y}_o[n]$ and $e[n]$ denote the continuous-time input signal, discrete-time reconstructed output and reconstruction error, respectively. The fast sampling sensor node represents a fictitious sensor which does not have sampling jitters; it is considered for design purposes only and is not present in the actual implementation. The slow sampling sensor nodes have sampling jitters.

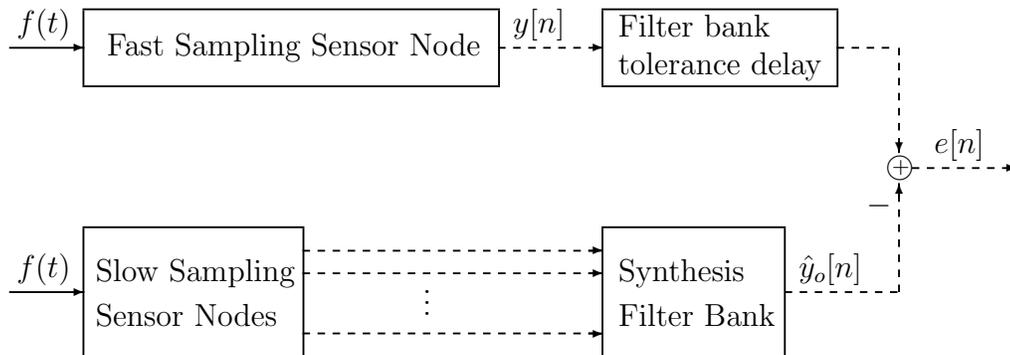


Figure 4.1: Schematic of the error system, K .

The error system, K , is modeled by a hybrid filter bank as illustrated in Fig. 4.2. The anti-aliasing filter, $\Phi_0(s)$, along with the sampler, S_h , in the top channel of the error system model the fast sampling sensor node having period h and without any sampling jitters. The remaining channels model slow sampling sensors having anti-aliasing filters, $\Phi_i(s)$, and samplers, S_{Mh} , where $i = 1, \dots, M$ and M is the total number of sensors. The synthesis filters, which are needed to be designed, are denoted by $F_i(z), i = 1, \dots, M$. The delay transfer matrices are denoted by $e^{-T_i s}, i = 1, \dots, M$; whereas the filter bank tolerance delay is denoted by U^m with transfer function z^{-m} . The up-samplers are denoted by E and they increase the sampling rate by a factor of M . It should be noted that for ideal distributed sampling (without sampling jitters), the delay transfer function is $e^{-T_i s}$ with $T_i = ih, 1 \leq i \leq M$. In the case of sampling jitters, $T_i = ih \pm d_i$, where d_i are the sampling jitters. We assume that the sampling jitters are bounded and can be expressed as $-h/2 < d_i < h/2$. Therefore, if sampling jitters are positive or negative, the time delay transfer function, $e^{-T_i s}$, can be treated as a fractional delay transfer function.

The hybrid system, K , is a time-varying but Mh -periodic system. Hence, there

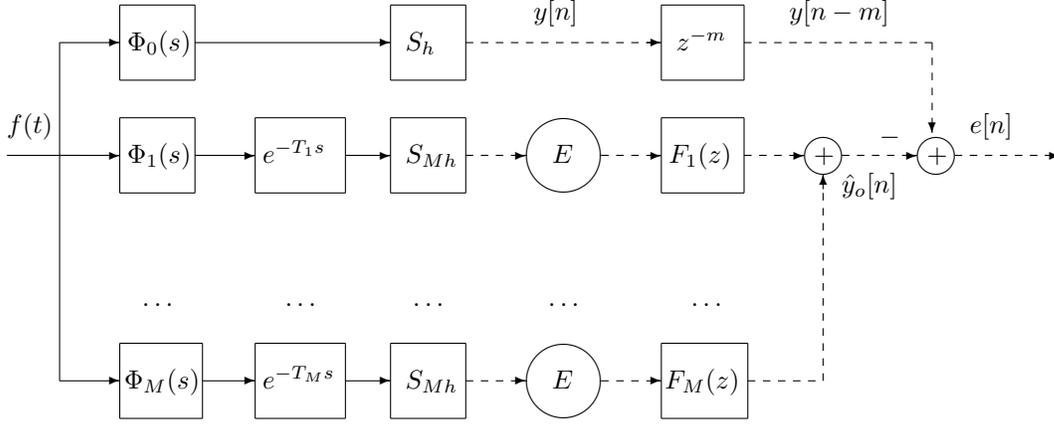


Figure 4.2: The hybrid error system, K , modeling a WSN employing distributed sampling. The solid lines show continuous-time signals and the dashed lines depict discrete-time signals.

is no transfer function whose \mathcal{H}_2 norm can be defined. The \mathcal{H}_2 norm of K is defined as the average of the output energy when impulses are applied in one period at the inputs. The \mathcal{H}_2 norm of K can be expressed as follows:

$$\|K\|_2 = \left(\frac{1}{Mh} \int_0^{Mh} \|K\delta(t - \tau)\|_2^2 d\tau \right)^{1/2},$$

where $\delta(t)$ denotes the unit impulse function. The objective of the hybrid filter bank is to design the synthesis filters to minimize $\|K\|_2$.

4.4 \mathcal{H}_2 Filter Design

In this section, we convert the hybrid filter bank problem into a discrete-time model-matching problem. This is achieved in two steps. First, the continuous-time part of the hybrid filter bank is discretized to obtain a time-varying discrete-time system, K_d . Next, by using the polyphase representation along with the lifting technique, the system is converted into a standard model-matching form for \mathcal{H}_2 filter design.

4.4.1 Reduction to a model-matching problem

The hybrid error system involves samplers S_h and S_{Mh} , with different periods. Let W be a down-sampler by a factor of M . It can be easily verified that

$$S_{Mh} = WS_h.$$

Thus, with the introduction of down-samplers, all the samplers in the hybrid multi-rate system are having the same period, h . The continuous-time part of the hybrid error system can be discretized using the norm-invariant discretization as expressed by (4.6). Let the discrete-time transfer functions for the continuous-time part of each channel be denoted by $H_i(z)$, $i = 0, \dots, M$. The hybrid system after discretization is denoted by K_d , which can be expressed as follows:

$$K_d = U^m H_0 - (F_1 E W H_1 + \dots + F_M E W H_M).$$

It should be noted that K_d is time-varying due to the presence of up- and down-samplers. For ease of derivation, we consider the case for 2 channels which can be extended to any finite number of channels. Hence, $M = 2$ and we can write the following equation:

$$K_d = U^m H_0 - [F_1 \quad F_2] \begin{bmatrix} E & 0 \\ 0 & E \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}.$$

Next, the polyphase matrices are introduced for the analysis and synthesis filters. For the analysis filters, the type I polyphase decomposition is used as follows [100]:

$$\begin{bmatrix} H_1(z) \\ H_2(z) \end{bmatrix} = Q(z^2) \begin{bmatrix} I \\ z^{-1}I \end{bmatrix}.$$

For the synthesis filters, the type II decomposition is used:

$$[F_1(z) \quad F_2(z)] = [z^{-1} \quad I] R(z^2). \quad (4.7)$$

Using the polyphase decompositions and the noble identities (see [100], Fig. 5.5-3), we obtain the following:

$$\begin{aligned} K_d &= U^m H_0 - [U \quad I] \begin{bmatrix} E & 0 \\ 0 & E \end{bmatrix} RQ \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} \begin{bmatrix} I \\ UI \end{bmatrix}, \\ &= U^m H_0 - [UE \quad E] RQ \begin{bmatrix} W \\ WU \end{bmatrix}. \end{aligned}$$

As the blocking/lifting operator preserves the ℓ_2 norm [26], so the norm of K_d remains unchanged if we pre-multiply and post-multiply K_d correspondingly by

$$\begin{bmatrix} W \\ WU \end{bmatrix} U^{-1}, \quad [E \quad U^{-1}E].$$

Hence, it can be written as

$$\begin{aligned} \|K_d\| &= \left\| \begin{bmatrix} W \\ WU \end{bmatrix} U^{-1} \left\{ U^m H_0 - [UE \quad E] RQ \begin{bmatrix} W \\ WU \end{bmatrix} \right\} [E \quad U^{-1}E] \right\|, \\ &= \|P - RQ\|. \end{aligned}$$

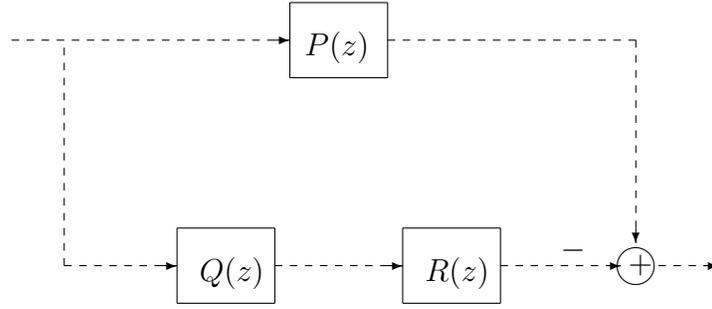


Figure 4.3: Equivalent discrete-time LTI error system, K_d .

Finally, we obtain the equivalent discrete-time system as shown in Fig. 4.3. Hence, minimizing $\|P - RQ\|_2$ is equivalent to minimizing the norm of the corresponding transfer matrix, namely, $\|P(z) - R(z)Q(z)\|_2$.

Let the discrete-time representation for the continuous-time part in each channel be given as follows:

$$H_i(z) = \left[\begin{array}{c|c} A_{H_i} & B_{H_i} \\ \hline C_{H_i} & D_{H_i} \end{array} \right], \quad i = 0, 1, 2.$$

It should be noted that the top channel of the filter bank does not have any fractional delay, hence $D_{H_0} = 0$. The following lemma is quite useful to obtain the state-space representation of Q .

Lemma 4.2. *Let N be a causal, LTI discrete-time system having the following representation:*

$$N(z) = \left[\begin{array}{c|c} A_N & B_N \\ \hline C_N & D_N \end{array} \right].$$

Let the polyphase components of $N(z)$ be $N_0(z^2)$ and $N_1(z^2)$, which can also be written as follows:

$$N(z) = N_0(z^2) + z^{-1}N_1(z^2). \quad (4.8)$$

Then, $N_0(z)$ and $N_1(z)$ are representations of WNE and $WU^{-1}NE$, respectively, and given as follows:

$$\begin{aligned} N_0(z) &= \left[\begin{array}{c|c} A_N^2 & B_N \\ \hline C_N A_N & D_N \end{array} \right] = \left[\begin{array}{c|c} A_N^2 & A_N B_N \\ \hline C_N & D_N \end{array} \right], \\ N_1(z) &= \left[\begin{array}{c|c} A_N^2 & A_N B_N \\ \hline C_N A_N & C_N B_N \end{array} \right] = \left[\begin{array}{c|c} A_N^2 & A_N^2 B_N \\ \hline C_N & C_N B_N \end{array} \right]. \end{aligned} \quad (4.9)$$

Proof. Using the polyphase decomposition, we can easily obtain Eq. (4.8). To prove that WNE has the transfer matrix $N_0(z)$, we can write the following:

$$WNE = W(N_0(z^2) + z^{-1}N_1(z^2))E.$$

Using the identities $WE = I$, $WUE = 0$, and the noble identities, we can write the following:

$$WNE = N_0,$$

which completes the proof for N_0 . Similarly, we can write the following:

$$WU^{-1}NE = WU^{-1}(N_0(z^2) + z^{-1}N_1(z^2))E.$$

Using the noble identities, we can write the following:

$$\begin{aligned} WU^{-1}NE &= WU^{-1}EN_0 + WU^{-1}UEN_1, \\ &= WU^{-1}EN_0 + WEN_1, \\ &= N_1. \end{aligned}$$

To obtain state-space representations of $N_0(z)$ and $N_1(z)$, we write the following:

$$\begin{aligned} \left[\begin{array}{c|c} A_N & B_N \\ \hline C_N & D_N \end{array} \right] &= [D_N + z^{-1}C_N B_N + z^{-2}C_N A_N B_N + z^{-3}C_N A_N^2 B_N + \dots], \\ &= [D_N + z^{-2}C_N A_N B_N + \dots] + [z^{-1}C_N B_N + z^{-3}C_N A_N^2 B_N + \dots] \\ &= [D_N + z^{-2}C_N A_N B_N + \dots] + z^{-1} [C_N B_N + z^{-2}C_N A_N^2 B_N + \dots] \\ &= N_0(z^2) + z^{-1}N_1(z^2). \end{aligned}$$

We can further write the following:

$$\begin{aligned} N_0(z^2) &= [D_N + z^{-2}C_N A_N B_N + \dots], \\ N_0(z) &= [D_N + z^{-1}C_N A_N B_N + \dots], \\ &= \left[\begin{array}{c|c} A_N^2 & B_N \\ \hline C_N A_N & D_N \end{array} \right] = \left[\begin{array}{c|c} A_N^2 & A_N B_N \\ \hline C_N & D_N \end{array} \right]. \end{aligned}$$

Similarly, we obtain the following:

$$\begin{aligned} N_1(z^2) &= z^{-1} [C_N B_N + z^{-2}C_N A_N^2 B_N + \dots], \\ N_1(z) &= [C_N B_N + z^{-1}C_N A_N^2 B_N + \dots], \\ &= \left[\begin{array}{c|c} A_N^2 & A_N B_N \\ \hline C_N A_N & C_N B_N \end{array} \right] = \left[\begin{array}{c|c} A_N^2 & A_N^2 B_N \\ \hline C_N & C_N B_N \end{array} \right]. \end{aligned}$$

□

Using Lemma 4.2, the representation for Q can be written as follows:

$$Q(z) = \left[\begin{array}{cc|cc} A_{H_1}^2 & 0 & A_{H_1}B_{H_1} & A_{H_1}^2B_{H_1} \\ 0 & A_{H_2}^2 & A_{H_2}B_{H_2} & A_{H_2}^2B_{H_2} \\ \hline C_{H_1} & 0 & D_{H_1} & C_{H_1}B_{H_1} \\ 0 & C_{H_2} & D_{H_2} & C_{H_2}B_{H_2} \end{array} \right]. \quad (4.10)$$

The representation for P takes different forms depending on the filter bank tolerance delay, m . Let us assume m is even, so that $m = 2d$ for some $d \geq 0$. Then, the representation for P can be written as follows:

$$P(z) = z^{-d} \left[\begin{array}{cc|cc} A_{H_0}^2 & & A_{H_0}B_{H_0} & A_{H_0}^2B_{H_0} \\ \hline C_{H_0}A_{H_0} & & C_{H_0}B_{H_0} & C_{H_0}A_{H_0}B_{H_0} \\ & C_{H_0} & 0 & C_{H_0}B_{H_0} \end{array} \right]. \quad (4.11)$$

If m is odd, $m = 2d + 1$, the representation of P can be written as follows:

$$P(z) = z^{-d} \left[\begin{array}{cc|cc} A_{H_0}^2 & & B_{H_0} & A_{H_0}B_{H_0} \\ \hline C_{H_0}A_{H_0} & & 0 & C_{H_0}B_{H_0} \\ & C_{H_0} & 0 & 0 \end{array} \right]. \quad (4.12)$$

4.4.2 Design of IIR filters

Having the representations of P and Q , we need to convert the model-matching problem into a standard form as shown in Fig. 4.4. In order to make Fig. 4.3 and Fig. 4.4 equivalent, we can write the following:

$$G = \begin{bmatrix} P & -I \\ Q & 0 \end{bmatrix}.$$

This has the following realization:

$$G = \left[\begin{array}{c|c} A_G & B_G \\ \hline C_G & D_G \end{array} \right] = \left[\begin{array}{cc|cc} A_P & 0 & B_P & 0 \\ 0 & A_Q & B_Q & 0 \\ \hline C_P & 0 & D_P & -I \\ 0 & C_Q & D_Q & 0 \end{array} \right]. \quad (4.13)$$

This is now a standard \mathcal{H}_2 optimization problem, which can be solved using techniques available in the literature (see, e.g., [27, 78]).

4.4.3 Design of FIR filters

In this section, we present the design of FIR synthesis filters. We first derive a state-space representation for the polyphase matrix $R(z)$. Assuming that the FIR filter order is n , the transfer function for $R(z)$ can be written as follows:

$$R(z) = a_0 + a_1z^{-1} + a_1z^{-1} + \cdots + a_nz^{-n}$$

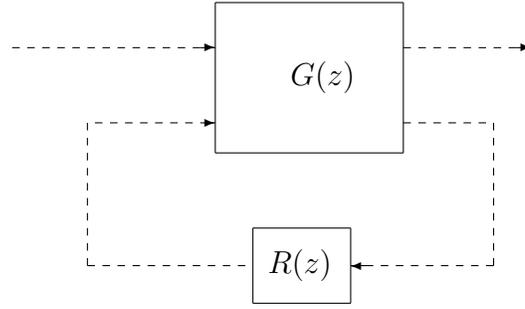


Figure 4.4: Standard \mathcal{H}_2 optimization problem.

The state-space realization for $R(z)$ can be expressed as follows:

$$A_R = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \quad B_R = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$C_R = [a_n \ a_{n-1} \ \cdots \ a_1], \quad D_R = a_0,$$

where $A_R \in \mathbb{R}^{n \times n}$ and $B_R \in \mathbb{R}^n$. Hence, designing FIR filters is equivalent to finding the matrices C_R and D_R to minimize $\|K_d\|_2$. Let K_{clp} denote the close-loop transfer matrix for the system shown in Fig. 4.4. The realization for K_{clp} can be expressed as follows:

$$\begin{aligned} K_{clp} &= \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right] \\ &= \left[\begin{array}{cc|c} A_G + B_{2G}D_R C_{2G} & B_{2G}C_R & B_{1G} + B_{2G}D_R D_{21G} \\ \hline B_R C_{2G} & A_R & B_R D_{21G} \\ C_{1G} + D_{12G}C_{2G} & D_{12G}C_R & D_{11G} + D_{12G}D_R D_{21G} \end{array} \right] \quad (4.14) \\ &= \left[\begin{array}{ccc|c} A_P & 0 & 0 & B_P \\ 0 & A_Q & 0 & B_Q \\ 0 & B_R C_Q & A_R & B_R D_Q \\ \hline C_P & -D_R C_Q & -C_R & D_P - D_R D_Q \end{array} \right]. \end{aligned}$$

We observe that the state-space matrices for K_{clp} depend on C_R and D_R in a linear fashion. Hence, linear matrix inequality (LMI) techniques can be used to solve for the matrices C_R and D_R [15]. The following lemma is presented for the design of FIR \mathcal{H}_2 filters.

Lemma 4.3. [45, 94] *For a given β , the close-loop system K_{clp} satisfies $\|K_{clp}\|_2^2 < \beta^2$ if and only if there exists a positive definite symmetric matrix X such that the*

following LMIs in X and S are feasible:

$$\begin{bmatrix} A'_K X A_K - X & & * \\ B'_K X A_K & & B'_K X B_K - I \end{bmatrix} < 0, \quad \begin{bmatrix} X & * & * \\ 0 & I & * \\ C_K & D_K & S \end{bmatrix} > 0, \quad (4.15)$$

$$\text{Trace}(S) - \beta^2 < 0.$$

Finally, the steps of this filter design procedure are summarized below:

- **Inputs:** Strictly proper transfer functions, $\{\Phi_i(s)\}_{i=0}^2$, sampling jitters, d_1, d_2 , the system tolerance delay, m , the sampling period h .
- **Outputs:** Synthesis filters, $F_1(z)$ and $F_2(z)$.

Step 1. Compute $H_0(z)$, the discrete-time state-space realization of $\Phi_0(s)$, using (4.3).

Step 2. Compute $H_1(z)$ and $H_2(z)$, the discrete-time state-space realizations of $\Phi_1(s)$ and $\Phi_2(s)$, respectively, by using the norm-invariant discretization and (4.6).

Step 3. Compute $Q(z)$ and $P(z)$ as expressed by (4.10) and (4.11) or (4.12), respectively.

Step 4. Compute the representation for G as expressed by (4.13).

Step 5. For designing IIR filters, solve the corresponding \mathcal{H}_2 optimal model-matching problem, given as

$$\min_{R(z)} \|P(z) - R(z)Q(z)\|_2.$$

Step 6. For designing FIR filters, compute K_{clp} using (4.14) and obtain R using (4.15) for \mathcal{H}_2 FIR synthesis filters.

Step 7. Obtain the synthesis filters, $F_1(z)$ and $F_2(z)$, as expressed by (4.7).

4.5 Numerical Example

In this section, we present an example to demonstrate the effectiveness of the proposed technique. We consider white noise with zero mean and unit variance as an input. The number of sensors considered is 2, hence $M = 2$. The filter bank tolerance delay considered is $m = 11$. The anti-aliasing filters are chosen as

$\Phi_i(s) = \omega_c^2 / (s + \omega_c)^2$ for $\omega_c = 0.80$ and $i = 0, 1, 2$. The sampling jitters are taken as $d_1 = 0.18$ and $d_2 = -0.23$. The sampling period chosen is $h = 1$. The synthesis filters were designed using the proposed IIR and FIR filter design technique and filter design technique based on rounding the fractional delays to the nearest integer multiple of h [56]. For FIR filter design, we used the Yalmip toolbox and Sedumi solver in MATLAB [61, 96]. The system of Fig. 4.2 was simulated for 5000 sec. Table 4.1 summarizes the system norm using the proposed and rounded delay design techniques. As white noise cannot be generated perfectly with computer simulations, therefore the system was simulated for 1000 times and the results were averaged. Fig. 4.5 shows a plot of the average error between the actual output and desired output for 250 sec. Based on the simulation results, it can be concluded that the effect of sampling jitters in WSNs can be minimized by the proposed technique.

Table 4.1: Performance comparison using different design techniques.

Filter design technique	System norm
Proposed IIR filter design method	0.0443
Proposed FIR filter design method	0.0485
Design based on rounding delay	0.3245

4.6 Concluding Remarks

This chapter presented the design of discrete-time synthesis filters to minimize the effect of sampling jitters in WSNs. The WSN was modeled using a hybrid filter bank. We showed that the hybrid system is \mathcal{H}_2 norm-equivalent to a discrete-time system. It was assumed that sampling jitters are fixed and known. In the next chapter, we relax this assumption and consider time-varying jitters.

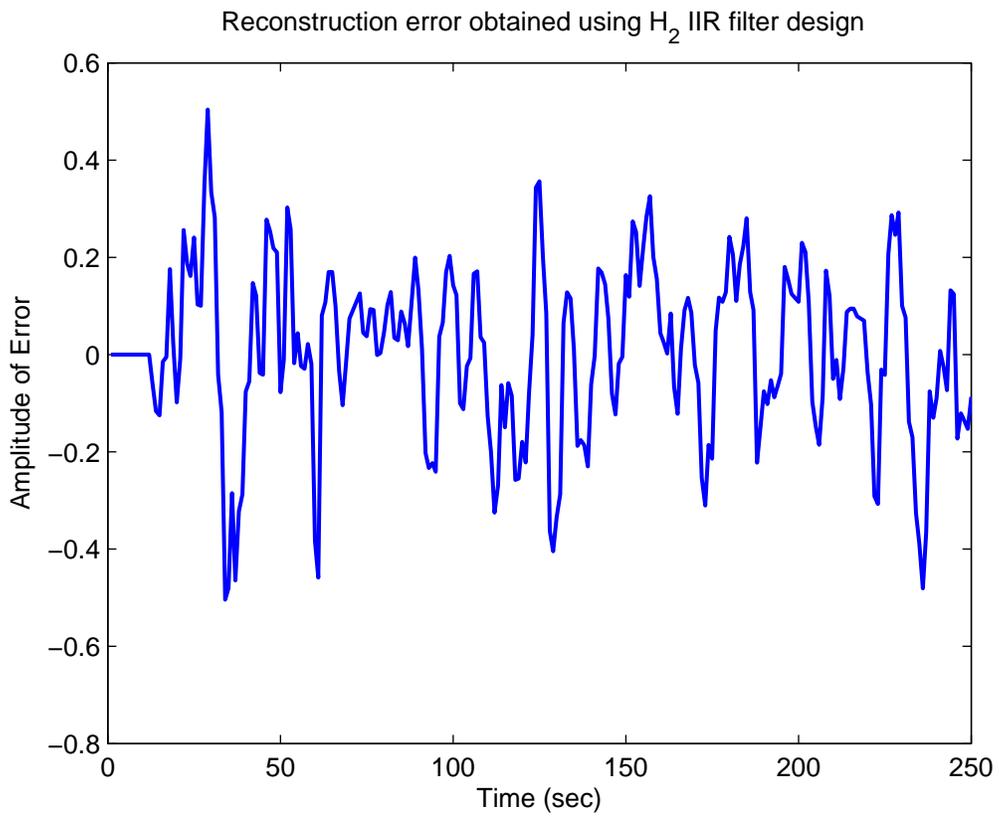


Figure 4.5: Plot of the averaged error between actual output and desired output of the filter bank.

Chapter 5

\mathcal{H}_∞ Filtering in WSNs

5.1 Introduction

In the pervious chapter, we modeled a wireless sensor network (WSN) using hybrid filter bank and designed optimal \mathcal{H}_2 synthesis filters to minimize the effects of sampling jitters. We assumed that the sampling jitters were know. In this chapter, we study the design of \mathcal{H}_∞ filters to compensate the effects of time-varying sampling jitters.

As stated in Chapter 4, the design of multi-rate filters using \mathcal{H}_∞ norm optimization was originally proposed in [89]. This design procedure was extended to discrete-time filter banks in [26] and hybrid filter banks in [90]. However, in both cases the authors considered uniformly sampled measurements whereas in this chapter non-uniformly sampled measurements are considered. To our best knowledge, the filter bank design problem using time-varying sampling rates has not been studied in WSNs. The main contribution of this chapter is to propose a technique for the design of pre-processing filters to reconstruct uniformly sampled measurements based on the minimization of the \mathcal{H}_∞ norm of the error system.

The design of optimal filters has received considerable attention in the fields of signal processing and control theory (see, e.g., [8, 27]). The filter design problem becomes complex when the sampling rate is varying and unknown due to the presence of sampling jitters. The variable sampling rates introduces uncertainty in the system matrices. In this chapter, we use polytopic matrices to encompass all possible representations of the uncertain system matrices and design robust filters to compensate the effects of sampling jitters. Some of the earlier approaches to design robust filters employed a common Lyapunov function (see, e.g., [41, 99]). However, such design techniques result in conservative results due to a single common Lyapunov function. The conservatism can be reduced by using parameter-

dependant Lyapunov functions [29, 9, 42]. The main difficulty in designing filters using parameter-dependant Lyapunov functions is to find a change of coordinates to separate the Lyapunov stability matrix from the system matrices to obtain linear matrix inequalities (LMIs). A robust filter design technique using parameter-dependant Lyapunov function was proposed in [13]. However, the filter design conditions were expressed in terms of bilinear matrix inequalities (BMIs), which are nonconvex, and the computation complexity is high compared to LMIs. This chapter extends the results presented in [31] to design filters to handle time-varying sampling rates.

The remainder of this chapter is structured as follows: In Section 5.2, the problem formulation is presented. The main results are discussed in Section 5.3. Section 5.4 presents a numerical example which demonstrates the effectiveness of the proposed technique. Finally, conclusions and future work are summarized in section 5.5.

5.2 Problem Formulation

A WSN employing distributed sampling can be modeled by a hybrid filter bank as shown in Fig. 4.2. In this chapter, we focus on the design of filters to handle time-varying sampling jitters. To make the problem tractable, we assume the input to slow sampling sensor nodes is in discrete-time as shown in Fig. 5.1. The input to the WSN is denoted by f_k . Let the period of a fast sampling sensor be denoted by h . When the sampling load is distributed between N sensors, the ideal period of each slow sampling sensor is Nh . To avoid clutter in notation, let $T = Nh$. When sampling jitters are present, the sensors operate at period T_i , where T_i satisfies the following:

$$T_i = T + \Delta_i \text{ where } \Delta_{\min} \leq \Delta_i \leq \Delta_{\max} \text{ and } i = 1, \dots, N. \quad (5.1)$$

The sampling jitters in each sensor are represented by Δ_i . The values of sampling jitters are unknown and time-varying. It is assumed that the lower bound, Δ_{\min} , and upper bound, Δ_{\max} , of the sampling jitters are known. The analysis filters in Fig. 5.1 are denoted by $\Phi_i(s)$ and the pre-processing filters by $F_i(z)$ where $i = 1, \dots, N$. S_{T_i} and H_{T_i} represent Sample and Hold devices, respectively, operating at period T_i , where T_i satisfies 5.1 and $i = 1, \dots, N$. The objective of the pre-processing filters is to minimize the effects of sampling jitters and reconstruct uniformly sampled measurements which are given to the synthesis filter bank. The

synthesis filter bank consists of up-samplers and synthesis filters, which reconstruct the input signal, denoted by \hat{f}_k .

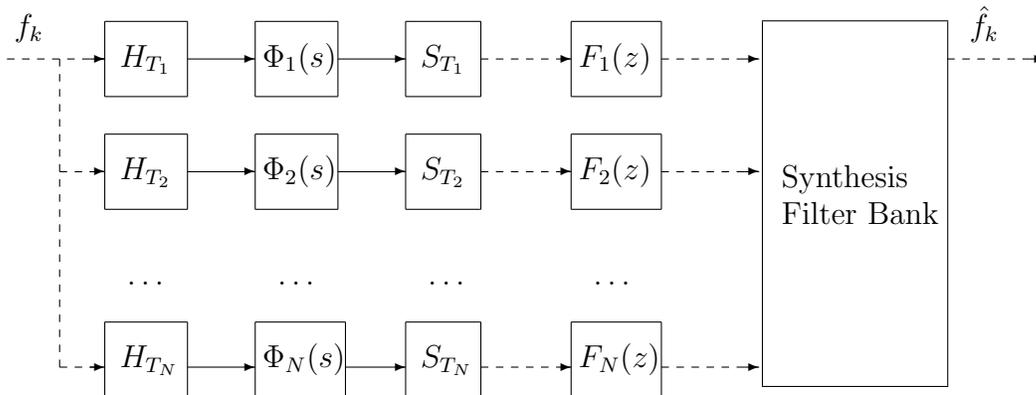


Figure 5.1: A filter bank showing a WSN employing distributed sampling.

The design of optimal pre-processing and synthesis filter bank simultaneously is a nonlinear optimization problem. In this chapter, we focus on the design of pre-processing filters only. The pre-processing filters are designed separately for each channel. Therefore, we present the design procedure for a given channel, which should be repeated for each channel. To design a pre-processing filter for a channel, we construct an error system, K , as shown in Fig. 5.2. The top channel of the error system is considered for design and simulation purposes only and it represents an ideal slow sampling sensor operating at time-period T . The bottom channel represents an actual sensor operating at time-period T_i , where T_i satisfies (5.1). The input to the anti-aliasing filter is denoted by $f(t)$ and the output by $y(t)$. For ease of notations, we have denoted the anti-aliasing filter by $\Phi(s)$ and pre-processing filter by $F(z)$. The output of the error system is denoted by e_k , which is the difference between the output of an ideal sensor, y_k , and output of the pre-processing filter, \hat{y}_k . We consider F having the following representation:

$$\begin{aligned}\hat{x}_{k+1} &= A_F \hat{x}_k + B_F \tilde{y}_k, \\ \hat{y}_k &= C_F \hat{x}_k + D_F \tilde{y}_k.\end{aligned}$$

The \mathcal{H}_∞ norm of K can be expressed as follows (see, e.g., [27]):

$$\|K\|_\infty = \sup \frac{\|e\|_2}{\|f\|_2}. \quad (5.2)$$

The objective is to design the filter parameters A_F, B_F, C_F, D_F to minimize (5.2).

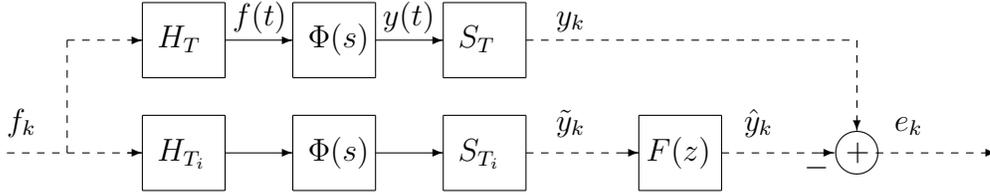


Figure 5.2: The error system, K , considered for designing pre-processing filter for each channel.

5.3 Main Results

In this section, first we represent the error system by a discrete-time, time-varying system. For the lower channel of the error system, we can not have a discrete-time representation which can be computed numerically because the actual values of the sampling periods are unknown. Next, we present a technique of constructing a polytopic system which encompasses all possible representations of the system matrices that can be generated due to the presence of sampling jitters. Finally, we present a filter design technique for uncertain systems.

5.3.1 Discrete-time representation of the error system

Let the representation of $\Phi(s)$ be given as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bf(t), \\ y(t) &= Cx(t). \end{aligned}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 1}$ and $C \in \mathbb{R}^{1 \times n}$. Applying the step-invariant discretization technique (see e.g., [27]), the discrete-time representation of $S_T \Phi(s) H_T$, denoted by $\Phi(z)$, can be expressed as follows:

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d f_k, \\ y_k &= C x_k. \end{aligned}$$

where $A_d = e^{TA}$ and $B_d = \int_0^T e^{\tau A} d\tau B$. For the lower channel of the error system, K , we can not numerically compute the discrete-time representation of $S_{T_i} \Phi(s) H_{T_i}$, because the values of T_i are unknown and changing. The variation of T_i causes uncertainty in the system matrices. Let $\tilde{A}_d(T_i) = e^{T_i A}$, $\tilde{B}_d(T_i) = \int_0^{T_i} e^{\tau A} d\tau B$ be the uncertain system matrices, and $\tilde{P}(T_i)$ represent the corresponding uncertain system. In the next section, we present a technique to construct polytopic system to encompass all possible representations of $\tilde{A}_d(T_i)$ and $\tilde{B}_d(T_i)$.

5.3.2 Polytopic representation of uncertain system matrices

The actual values of T_i are unknown and the extreme values are only known, which can be obtained using (5.1). Therefore, $\tilde{A}_d(T_i)$ and $\tilde{B}_d(T_i)$ can take values from an infinite set of system representations. This infinite set can be overapproximated using a polytopic representation. The advantage of polytopic representation is that the filter design conditions can be easily derived in terms of LMIs. The main difficulty lies in defining the vertices of the polytopic system due to uncertainty in the exponential terms of the system matrices. One approach is to use Cayley-Hamilton theorem (see, e.g., [23, 13]) to compute the matrix exponential and then find the vertices of the polytope. In the following, we apply Cayley-Hamilton theorem and present a technique to embed $\tilde{A}_d(T_i)$ and $\tilde{B}_d(T_i)$ into polytopic matrices.

Let us denote the eigenvalues of A by λ_j where $j = 1, \dots, n$. Using Cayley-Hamilton theorem, we can write the following:

$$\begin{aligned} e^{T_i A} &= \sum_{j=1}^n \beta_j(T_i) A^{j-1}, \\ &= \sum_{j=1}^n \theta_j(T_i) \Omega_j, \end{aligned} \tag{5.3}$$

where β_j can be determined by solving a set of linear equations defined in terms of λ_j , θ_j are T_i -varying coefficients and Ω_j are constant known matrices which are obtained by collecting the known terms. If we evaluate $\theta_j(T_i)$ at the extreme values of T_i , then we can write the following:

$$\underline{\theta}_j \leq \theta_j(T_i) \leq \bar{\theta}_j.$$

Let $A_p, p = 1, \dots, 2^n$, represent all combinations obtained by substituting $\underline{\theta}_j$ and $\bar{\theta}_j$ in (5.3), then all possible outcomes of $\tilde{A}_d(T_i)$ belong to the following polytope:

$$\mathcal{A} = \left\{ A(\alpha) = \sum_{p=1}^{2^n} \alpha_p A_p \right\}, \tag{5.4}$$

where $\sum_{p=1}^{2^n} \alpha_p = 1$, $\alpha_p \geq 0$ and α is a time-varying vector containing α_p . Similarly, we can write the following:

$$\begin{aligned} \tilde{B}_d(T_i) &= \int_0^{T_i} e^{\tau A} d\tau B = \int_0^{T_i} \sum_{j=1}^n \beta_j(\tau) A^{j-1} d\tau B \\ &= \sum_{j=1}^n \int_0^{T_i} \beta_j(\tau) d\tau A^{j-1} B = \sum_{j=1}^n \vartheta_j(T_i) \Psi_j, \end{aligned} \tag{5.5}$$

where $\vartheta_j(T_i) = \int_0^{T_i} \beta_j(\tau)$ and $\Psi_j = A^{j-1}B$. If we evaluate $\vartheta_j(T_i)$ at the extreme values of T_i , then we can write the following:

$$\underline{\vartheta}_j \leq \vartheta_j(T_i) \leq \bar{\vartheta}_j.$$

Let $B_p, p = 1, \dots, 2^n$, represent all combinations obtained by substituting $\underline{\vartheta}_j$ and $\bar{\vartheta}_j$ in (5.5), then all possible outcomes of $\tilde{B}_d(T_i)$ belong to the following polytope:

$$\mathcal{B} = \left\{ B(\alpha) = \sum_{p=1}^{2^n} \alpha_p B_p \right\}. \quad (5.6)$$

There are two main advantages of using Cayley-Hamilton theorem to generate a polytopic representation of the matrix exponential. The first one is that there is no approximation involved. Although, the polytope may contain matrices which cannot be generated by the matrix exponential, but this is the price to be paid in order to avoid approximating the matrix exponential. The second advantage is that the uncertain matrix exponential is given as a linear convex combination of known vertices and α_i . For an uncertain time-varying system, if the system matrices are modeled using polytopic representation, then it is easier to derive LMI conditions for filter design. In the next section, we present the design of an \mathcal{H}_∞ filter for polytopic systems.

5.3.3 Design of an \mathcal{H}_∞ filter for polytopic systems

Let us define $\bar{x}_k = [x_k \hat{x}_k]^T$ and $u_k = [f_k \ 0]^T$, then the estimation error dynamics can be written as follows:

$$\begin{aligned} \bar{x}_{k+1} &= \bar{A}(\alpha)x_k + \bar{B}(\alpha)u_k, \\ e_k &= \bar{C}\bar{x}_k, \end{aligned} \quad (5.7)$$

where

$$\begin{aligned} \bar{A}(\alpha) &= \sum_{p=1}^{2^n} \alpha_p \bar{A}_p = \sum_{p=1}^{2^n} \alpha_p \begin{bmatrix} A_p & 0 \\ B_F C & A_F \end{bmatrix}, \quad \bar{B}(\alpha) = \sum_{p=1}^{2^n} \alpha_p \bar{B}_p = \sum_{p=1}^{2^n} \alpha_p \begin{bmatrix} B_p \\ 0 \end{bmatrix} \quad \text{and} \\ \bar{C} &= [C - D_F C \quad -C_F]. \end{aligned}$$

To analyze the \mathcal{H}_∞ performance of the error system, we present the following lemma.

Lemma 5.1. *The error system (5.7) is asymptotically stable with an \mathcal{H}_∞ performance given by $\gamma > 0$ if there exists matrices G, Q and a symmetric positive*

definite matrix P such that the following matrix inequality holds:

$$\begin{bmatrix} G + G' - P & 0 & G\bar{A} - Q' & G\bar{B} \\ * & I & \bar{C} & 0 \\ * & * & P - Q\bar{A} - \bar{A}'Q' & -Q\bar{B} \\ * & * & * & \gamma^2 I \end{bmatrix} > 0.$$

Proof. It is an extension of discrete-time bounded real lemma to uncertain systems [28, 31]. \square

Lemma 5.1 can be used for analyzing \mathcal{H}_∞ performance, however it cannot be used for designing the filter parameters. The following lemma presents the design of filter parameters.

Lemma 5.2. *Assume there exists matrices G_{11} , G_{21} , G_2 , Q_{11} , Q_{21} , S_1 , S_2 , S_3 , S_4 , J_{pp} , K_{pp} , L_{pp} , and scalars λ_1 , λ_2 , where $p = 1, \dots, 2^n$ such that the following LMIs hold:*

$$\begin{bmatrix} T_{11} & T_{12} & 0 & T_{14} & S_1 - Q'_{21} & G_{11}B_p \\ * & T_{22} & 0 & T_{24} & S_1 - \lambda_2 G'_2 & G_{21}B_p \\ * & * & I & C - S_4 C & -S_3 & 0 \\ * & * & * & T_{44} & T_{45} & -Q_{11}B_p \\ * & * & * & * & T_{55} & -Q_{21}B'_p \\ * & * & * & * & * & \gamma^2 I \end{bmatrix} > 0,$$

where

$$\begin{aligned} T_{11} &= G_{11} + G'_{11} - J_{pp}, \\ T_{12} &= G_2 + G'_{21} - K_{pp}, \\ T_{14} &= G_{11}A_p + S_2C - Q_{11}, \\ T_{22} &= G_2 + G'_2 - L_{pp}, \\ T_{24} &= G_{21}A_p + S_2C - \lambda_1 G'_2, \\ T_{44} &= J_{pp} - Q_{11}A_p - \lambda_1 S_2 C - A'_p F'_{11} - \lambda_1 C' S'_2, \\ T_{45} &= K_{pp} - \lambda_1 S_1 - A'_p Q'_{21} - \lambda_2 C' S'_2, \\ T_{55} &= L_{pp} - \lambda_2 S_1 - \lambda_2 S'_1, \\ p &= 1, \dots, 2^n, \end{aligned}$$

then $A_F = G_2^{-1}S_1$, $B_F = G_2^{-1}S_2$, $C_F = S_3$ and $D_F = S_4$ are the matrices of the \mathcal{H}_∞ filter with cost γ .

Proof. The proof follows a similar derivation as in [31]. \square

5.4 Numerical Example

In this section, we present an example to demonstrate the effectiveness of the proposed filter design procedure. We consider two sensors, hence $N = 2$. The analysis filters are taken from [90] and expressed as follows:

$$\Phi_1(s) = \frac{0.145s^2 + 0.0677}{s^3 + 0.3776s^2 + 0.2893s + 0.0677},$$

$$\Phi_2(s) = \frac{0.0066s^5 + 0.0507s^3 + 0.0016}{s^6 + 0.3754s^5 + 1.7368s^4 + 0.4121s^3 + 0.851s^2 + 0.0901s + 0.1176}.$$

The fast sampling period is chosen as $h = 0.5$. The bounds of the sampling jitters are chosen as $\Delta_{\min} = -0.15$ and $\Delta_{\max} = 0.20$. The pre-processing filters were designed using the proposed design procedure. The LMIs of Lemma 5.2 were implemented in MATLAB using YALMIP [61] and Sedumi [96]. For the first sensor, the value of γ using the proposed design procedure is 0.1329, while for the second it is 0.3998. For simulation purposes, the following test input signal was considered:

$$f_k = 3e^{(-0.01k)} \sin(-0.9k).$$

The WSN model was simulated for 2000 sec. Fig. 5.3 shows a plot of the error between the actual output and desired output for 80 sec for both sensors. From the simulation example, it can be found that using the proposed technique, the \mathcal{H}_∞ norm ratio between the error and input for sensor 1 is 0.01345 and for sensor 2 is 0.2633. Based on the simulation results, it can be seen that the proposed filter design procedure minimizes the effects of sampling jitters. Hence, it can be concluded that the effects of sampling jitters in WSNs can be minimized by the proposed design procedure.

5.5 Concluding Remarks

This chapter presented the design of discrete-time pre-processing filters to compensate the effects of sampling jitters in WSNs. A WSN was modeled using filter bank. It was assumed that the sampling jitters were unknown but bounded. LMI conditions were presented for the design of \mathcal{H}_∞ filters. The LMI conditions involved some scalar parameters. For future work, the optimum values of those scalar parameters can be investigated. Similarly, for future research work the design of synthesis filters using non-uniform sampling can be considered.

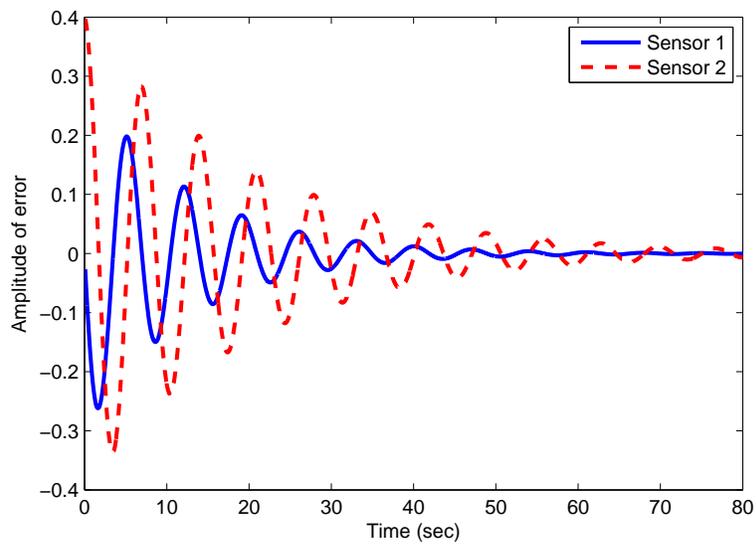


Figure 5.3: Plot of estimation error using using the proposed \mathcal{H}_∞ filter design technique.

Chapter 6

Summary and Future Work

The rapid technological merging of communication systems, control engineering, and computing sciences has led to exciting developments in the fields of wireless sensor networks. Traditionally, control systems had a centralized or a hierarchical architecture which offered many design advantages. However, these control systems tend to become complex with the expansion in business units, requiring more computation capabilities, consuming more space and often requiring wiring over long distances. The recent trend to avoid these problems is to deploy WSNs. Although WSNs are very promising, the distributed nature, attributes of wireless networks, and availability of limited resources in WSNs introduce significant theoretical and practical challenges. Motivated by these challenges, this thesis presented the design of distributed filtering and sampling techniques in resource constrained WSNs.

In this thesis, we studied the design of distributed time synchronization protocols which are motivating examples for studying the applications of distributed filtering techniques. We investigated the design of consensus-based protocols to achieve distributed time synchronization in Chapter 2. Compared to the popular distributed time synchronization protocols presented in the literature which require a hierarchical structure, our proposed protocols are fully distributed and robust to topology changes in WSNs. We presented the concept of a virtual clock to illustrate the phenomenon of time synchronization in sensor nodes.

In Chapter 3, we extended the distributed time synchronization protocol presented in Chapter 2 to an asynchronous framework. We considered unreliable communication links. We modeled the clocks in a WSN by a time-varying system with time-delay terms and presented the convergence analysis by employing tools from graph and nonnegative matrix theories. The effectiveness of the time synchronization protocols presented in Chapters 2 and 3 were demonstrated using

numerical examples.

In Chapter 4, we studied WSNs employing distributed sampling sensors. We modeled such systems by filter banks. By utilizing the properties of continuous-time lifting operator and discretizing a fractional time-delay system, we obtained a norm-invariant discretized system. Next, by employing the polyphase representation and using the discrete-time lifting technique, we converted the system into a standard model-matching \mathcal{H}_2 optimization problem. We also presented the design of finite impulse response \mathcal{H}_2 optimal filters. A numerical example was presented to illustrate the effectiveness of the proposed \mathcal{H}_2 filter design technique.

In Chapter 5, we presented the design of \mathcal{H}_∞ pre-processing filters to mitigate the effects of time-varying sampling jitters. The presence of sampling jitters introduces uncertainty in the exponential terms of the system matrices. We used polytopic matrices to encompass all possible representations of the system matrices and then reduced the problem to an \mathcal{H}_∞ optimization problem. We presented sufficient conditions for the design of pre-processing filters in terms of linear matrix inequalities. All the theoretical developments and proposed techniques in this thesis were validated using simulation examples.

6.1 Major Thesis Contributions

This thesis is concerned with design of distributed filtering, synchronization and sampling techniques in WSNs and the main contributions of this thesis are summarized below:

- Chapter 2 presented the design of a consensus-based time synchronization protocol in WSNs which does not require periodic communication and control input updates.
- Chapter 3 extended the design of the above-mentioned protocol to include information exchange over unreliable communication channels and time-varying communication topologies in WSNs.
- Chapter 4 presented the model of a WSN employing distributed sampling sensors with sampling jitters using a hybrid filter bank and proposed a novel technique to reconstruct the uniformly sampled measurements by designing synthesis filters based on the minimization of the \mathcal{H}_2 norm of the estimation error system.

- Chapter 5 extended the design of distributed sampling models and proposed another filter design technique in terms of LMIs to recover the uniformly sampled measurements by minimizing the \mathcal{H}_∞ norm of the estimation error system.

6.2 Directions for Future Research Studies

We mention here that WSNs are mainly at the crossroads of three research fields: control sciences and signal processing, wireless communication and information theory, and computing science. There are a lot of opportunities to advance this study and some of the suggested areas for future research studies are listed below:

- **Extension of distributed filtering techniques to high order systems**

In this thesis, we studied the time synchronization problem which is an application of the distributed filtering techniques. We investigated discrete-time, first order, integrator systems. An interesting extension of the proposed study is to consider high order systems and investigate distributed filtering techniques in asynchronous frameworks. In a recent study [38], the authors investigated the design of consensus-based techniques for second order systems using an asynchronous framework and presented sufficient conditions in terms of LMIs to ensure consensus. The authors in [81] applied contraction-based stability theory to present the convergence analysis for achieving consensus for first order systems. The application of contraction theory to study distributed filtering techniques for high order systems is still an open problem.

- **Time synchronization using noisy links**

The consensus-based time synchronization protocols presented in this thesis assume the absence of noise in the communication channels. For first order systems, the presence of noise generates extra terms when expressing the group dynamics for all the sensor nodes. In [72], the authors considered such a situation and termed the Laplacian matrix as a deformed Laplacian matrix and presented a deformed consensus protocol. The deformed consensus protocol worked for the synchronous framework, and future research studies may include the extension of such protocols to asynchronous frameworks.

- **Sampled-data design to handle time-varying jitters**

In this thesis, we considered a discrete-time filter bank and introduced a

zero-order hold at the input of the filter bank to make the problem tractable. A disadvantage of this approach is that it ignores the inter-sample behavior of the input signal. The presence of time-varying jitters makes the sampling pattern non-uniform and introduces uncertainty in the system matrices. For non-uniform sampling, the idea of non-uniform continuous-time lifting can be used [71]; however the resulting problem is much harder due to time-varying operators and discretizing the hybrid system. It requires more research efforts to have a better understanding and make such problems tractable.

Bibliography

- [1] Berkeley mote wireless throughput analysis. http://www.doc.ic.ac.uk/vip/ubimon/notice_board/Berkeley_Mote_Wireless_Throughput_Analysis.htm.
- [2] N. Aakvaag, M. Mathiesen, and G. Thonet. Timing and power issues in wireless sensor networks - an industrial test case. In *IEEE International Conference on Parallel Processing*, pages 419–426, 2005.
- [3] A. M. Abdelgawad and M. A. Bayoumi. Remote measuring for sand in pipelines using wireless sensor network. *IEEE Transactions on Instrumentation and Measurement*, 60(4):1443–1452, 2011.
- [4] S. Ahmed and T. Chen. Minimizing the effects of sampling jitters in wireless sensors networks. *IEEE Signal Processing Letters*, 18(4):219–222, 2011.
- [5] S. Ahmed, F. Xiao, and T. Chen. Asynchronous consensus-based time synchronization in wireless sensor networks using unreliable communication links. *submitted to IET Control Theory and Applications*.
- [6] S. Ahmed, F. Xiao, and T. Chen. Achieving relative time synchronization in wireless sensor networks. *Journal of Control Science and Engineering*, 2013:Article ID 538181, 7 pages, 2013.
- [7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [8] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, NJ, 1979.
- [9] K. A. Barbosa, C. E. de Souza, and A. Trofino. Robust \mathcal{H}_2 filter design via parameter-dependant Lyapunov functions. In *IFAC World Congress*, volume 15, pages 1881–1886, 2002.

- [10] P.-A. Bliman, A. Nedic, and A. Ozdaglar. Rate of convergence for consensus with delays. In *IEEE Conference on Decision and Control*, pages 4849–4854, 2008.
- [11] S. Bolognani, R. Carli, E. Lovisari, and S. Zampieri. A randomized linear algorithm for clock synchronization in multi-agent systems. In *IEEE Conference on Decision and Control*, pages 20–25, 2012.
- [12] S. Bolognani, S. D. Favero, L. Schenato, and D. Varagnolo. Consensus-based distributed sensor calibration and least-square parameter identification in WSNs. *International Journal of Robust and Nonlinear Control*, 20(2):176–193, 2010.
- [13] R. A. Borges, R. C. L. F. Oliveira, C. T. Abdallah, and P. L. D. Peres. \mathcal{H}_∞ filtering of networked systems with time-varying sampling rates. In *American Control Conference*, pages 3372–3377, 2009.
- [14] V. Borkar and P. P. Varaiya. Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control*, 27(3):650–655, 1982.
- [15] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, 1994.
- [16] K. Bur, P. Omiyi, and Y. Yang. Wireless sensor and actuator networks: Enabling the nervous system of the active aircraft. *IEEE Communications Magazine*, 48(7):118–125, 2010.
- [17] M. Cao, A. S. Morse, and B. D. O. Anderson. Agreeing asynchronously. *IEEE Transactions on Automatic Control*, 53(8):1826–1838, 2008.
- [18] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, 2013.
- [19] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, 2013.
- [20] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. A PI consensus controller for networked clocks synchronization. In *IFAC World Congress*, pages 10289–10294, 2008.

- [21] R. Carli, E. D’Elia, and S. Zampieri. A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks. In *IEEE Conference on Decision and Control and European Control Conference*, pages 7512–7517, 2011.
- [22] C. G. Cassandras and W. Li. Sensor networks and cooperative control. *European Journal of Control*, 11(4-5):436–463, 2005.
- [23] C.-T. Chen. *Linear System Theory and Design*. Oxford University Press, 1998.
- [24] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun. Distributed collaborative control for industrial automation with wireless sensor and actuator networks. *IEEE Transactions on Industrial Electronics*, 57(12):4219–4230, 2010.
- [25] S.-K. Chen, T. Kao, C.-T. Chan, C.-N. Huang, C.-Y. Chiang, C.-Y. Lai, T.-H. Tung, and P.-C. Wang. A reliable transmission protocol for ZigBee-based wireless patient monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 16(1):6–16, 2012.
- [26] T. Chen and B. A. Francis. Design of multirate filter banks by \mathcal{H}_∞ optimization. *IEEE Transactions on Signal Processing*, 43(12):2822–2830, 1995.
- [27] T. Chen and B. A. Francis. *Optimal Sampled-Data Control Systems*. Springer, London, 1995.
- [28] M.C. de Oliveira, J. C. Geromel, and J. Bernussou. Extended \mathcal{H}_2 and \mathcal{H}_∞ norm characterizations and controller parameterizations for discrete-time system. *International Journal of Control*, 75(9):666–679, 2002.
- [29] C. E. de Souza, A. Trofino, and J. de Oliveria. Robust \mathcal{H}_∞ control of uncertain linear systems via parameter-dependant lyapunov functions. *IEEE Conference on Decision and Control*, 4:3194–3199, 2000.
- [30] V. Divi and G. W. Wornell. Blind calibration of timing skew in time-interleaved analog-to-digital converters. *IEEE Journal of Selected Topics in Signal Processing*, 3(3):509–522, 2009.
- [31] Z. Duan, J. Zhang, C. Zhang, and E. Mosca. Robust \mathcal{H}_2 and \mathcal{H}_∞ filtering for uncertain linear systems. *Automatica*, 42(11):1919–1926, 2006.

- [32] E. Eisenberg and D. Gale. Consensus of subjective probabilities: The Pari-Mutuel method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959.
- [33] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *ACM SIGOPS Operating Systems Review*, pages 147–163, 2002.
- [34] K. J. Fisher. Distributed process control system for an iron ore processing plant. *IEEE Transactions on Industry Applications*, IA-18(2):199–203, 1982.
- [35] M. P. J. Fromherz and W. B. Jackson. Force allocation in a large-scale distributed active surface. *IEEE Transactions on Control Systems Technology*, 11(5):641–655, 2003.
- [36] B. Galloway and G. P. Hancke. Introduction to industrial control networks. *IEEE Communications Surveys Tutorials*, 15(2):860–880, 2013.
- [37] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *ACM SenSys*, pages 138–149, 2003.
- [38] Y. Gao and L. Wang. Asynchronous consensus of continuous-time multi-agent systems with intermittent measurements. *International Journal of Control*, 83(3):552–562, 2010.
- [39] M. Gastpar and M. Vetterli. Power, spatio-temporal bandwidth, and distortion in large sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):745–754, 2005.
- [40] M. Gastpar, M. Vetterli, and P. L. Dragotti. Sensing reality and communicating bits: a dangerous liaison. *IEEE Signal Processing Magazine*, 23(4):70–83, 2006.
- [41] J. C. Geromel. Optimal linear filtering under parameter uncertainty. *IEEE Transactions on Signal Processing*, 47(1):168–175, 1999.
- [42] J. C. Geromel, M. C. de Oliveira, and J. Bernussou. Robust filtering of discrete-time linear systems with parameter dependent lyapunov functions. *SIAM Journal on Control and Optimization*, 41(3):700–711, 2002.
- [43] M. H. De Groot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

- [44] V. C. Gungor and G. P. Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56(10):4258–4265, 2009.
- [45] H. Hindi, B. Hassibi, and S. Boyd. Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$ -optimal control via finite dimensional q -parametrization and linear matrix inequalities. In *American Control Conference*, pages 3244–3249, 1998.
- [46] C.-H. Hong, H.-H. Shen, H.-C. Wu, H. Shen, C.-W. Cheng, T.-S. Chu, and J.-M. Wu. Fast selection of time-interleaved samples for wireless health-care monitoring with pulse radar. In *IEEE Biomedical Circuits and Systems Conference*, pages 45–48, 2012.
- [47] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press Textbooks, 2012.
- [48] L. Hou and N. W. Bergmann. Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 61(10):2787–2798, 2012.
- [49] S.-C. Hsia, S.-W. Hsu, and Y.-J. Chang. Remote monitoring and smart sensing for water meter system and leakage detection. *IET Wireless Sensor Systems*, 2(4):402–408, 2012.
- [50] Q. Hui and W. M. Haddad. Distributed nonlinear control algorithms for network consensus. *Automatica*, 44(9):2375 – 2381, 2008.
- [51] F. Hutu, S. Cauet, and P. Coirault. Robust synchronization of different coupled oscillators: Application to antenna arrays. *Journal of the Franklin Institute*, 346(5):413–430, 2009.
- [52] I. C. F. Ipsen and T. M. Selee. Ergodicity coefficients defined by vector norms. *SIAM Journal on Matrix Analysis and Applications*.
- [53] P. Ishwar, A. Kumar, and K. Ramchandran. Distributed sampling for dense sensor networks: a “bit-conservation principle”. *Information Processing in Sensor Networks, Lecture Notes in Computer Science*, 2634:17–31, 2003.
- [54] I. Izadi, T. Chen, and Q. Zhao. Norm invariant discretization for sampled-data fault detection. *Automatica*, 41(9):1633–1637, 2005.

- [55] A. Jadbabaie, J. J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [56] O. S. Jahromi and P. Aarabi. Theory and design of multirate sensor arrays. *IEEE Transactions on Signal Processing*, 53(5):1739–1753, 2005.
- [57] O. S. Jahromi, B. A. Francis, and R. H. Kwong. Relative information of multi-rate sensors. *Information Fusion*, 5(2):119–129, 2004.
- [58] W. C. Black Jr. and D. Hodges. Time interleaved converter arrays. *IEEE Journal of Solid-State Circuits*, 15(6):1022–1029, 1980.
- [59] K. Kang, J. Ryu, J. Hur, and L. Sha. Design and QoS of a wireless system for real-time remote electrocardiography. *IEEE Journal of Biomedical and Health Informatics*, 17(3):745–755, 2013.
- [60] V. Kecman. *State-Space Models of Lumped and Distributed Systems*. Springer-Verlag, Berlin, New York, 1988.
- [61] J. Lfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Computer-Aided Control System Design Conference*, 2004.
- [62] J. Li and G. AlRegib. Distributed estimation in energy-constrained wireless sensor networks. *IEEE Transactions on Signal Processing*, 57(10):3746–3758, 2009.
- [63] J. Li, S. Xu, Y. Chu, and H. Wang. Distributed average consensus control in networks of agents using outdated states. *IET Control Theory Applications*, 4(5):746–758, 2010.
- [64] Q. Ling, Z. Tian, Y. Yin, and Y. Li. Localized structural health monitoring using energy-efficient wireless sensor networks. *IEEE Sensors Journal*, 9(11):1596–1604, 2009.
- [65] M.K. M. K. Maggs, S. G. O’Keefe, and D. V. Thiel. Consensus clock synchronization for wireless sensor networks. *IEEE Sensors Journal*, 12(6):2269–2277, 2012.
- [66] Y. Ma, M. Richards, M. Ghanem, Y. Guo, and J. Hassard. Air pollution monitoring and mining based on sensor grid in London. *Sensors*, 8(6):3601–3623, 2008.

- [67] K. B. Mallory. Control through a system of small computers. *IEEE Transactions on Nuclear Science*, 22(3):1086–1087, 1975.
- [68] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *ACM SenSys*, pages 39–49, 2004.
- [69] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [70] L. Mirkin. Some remarks on the use of time-varying delay to model sample-and-hold circuits. *IEEE Transactions on Automatic Control*, 52(6):1109 – 1112, June 2007.
- [71] L. Mirkin. Some remarks on the use of time-varying delay to model sample-and-hold circuits. *IEEE Transactions on Automatic Control*, 52(6):1109–1112, 2007.
- [72] F. Morbidi. The deformed consensus protocol. *Automatica*, 49(10):3049–3055, 2013.
- [73] R. M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems Measurement and Control*, 129(5):571–583, 2007.
- [74] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
- [75] H. T. Nguyen and M. N. Do. Hybrid filter banks with fractional delays: Minimax design and application to multichannel sampling. *IEEE Transactions on Signal Processing*, 56(7-2):3180–3190, 2008.
- [76] B. Nickerson. Issues in designing practical wireless sensors. *IEEE Instrumentation Measurement Magazine*, 15(1):22–26, 2012.
- [77] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [78] M. C. De Oliveira, J. C. Geromel, and J. Bernussou. Extended \mathcal{H}_2 and \mathcal{H}_∞ norm characterizations and controller parametrizations for discrete-time systems. *International Journal of Control*, 75(9):666–679, 2002.

- [79] A. Pascale, M. Nicoli, F. Deflorio, B. D. Chiara, and U. Spagnolini. Wireless sensor networks for traffic management and road safety. *IET Intelligent Transport Systems*, 6(1):67–77, 2012.
- [80] W. Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.
- [81] W. Ren and W. R. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Springer London, 2007.
- [82] W. Ren, H. Chao, W. Bourgeois, N. Sorensen, and Y-Q. Chen. Experimental validation of consensus algorithms for multivehicle cooperative control. *IEEE Transactions on Control Systems Technology*, 16(4):745–752, 2008.
- [83] R. O. Saber and P. Jalalkamali. Coupled distributed estimation and control for mobile sensor networks. *IEEE Transactions on Automatic Control*, 57(10):2609–2614, 2012.
- [84] F. Salvadori, M. De Campos, P. S. Sausen, R. F. De Camargo, C. Gehrke, M. A. Spohn, and A. C. Oliveira. Monitoring in industrial systems using wireless sensor network with dynamic power management. *IEEE Transactions on Instrumentation and Measurement*, 58(9):3104–3111, 2009.
- [85] L. Schenato and F. Fiorentin. Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica*, 47(9):1878–1886, 2011.
- [86] L. Schenato and G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *IEEE Conference on Decision and Control*, pages 2289–2294, 2007.
- [87] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer, 2006.
- [88] I. Shames and A. N. Bishop. Distributed relative clock synchronization in wireless sensor networks. In *IFAC World Congress*, pages 11256–11270, 2011.
- [89] R. G. Shenoy. Analysis of multirate components and application to multirate filter design. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 121–124, 1994.

- [90] H. Shu, T. Chen, and B. A. Francis. Minimax design of hybrid multirate filter banks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 44(2):120–128, 1997.
- [91] O. Simeone and U. Spagnolini. Distributed time synchronization in wireless sensor network with coupled discrete-time oscillators. *EURASIP Journal on Wireless Communications and Networking*, 2007:Article ID 57054, 13 pages, 2007.
- [92] O. Simeone, U. Spagnolini, Y. Bar-ness, and S. H. Strogatz. Distributed synchronization in wireless networks. *IEEE Signal Processing Magazine*, 25(5):81–97, 2008.
- [93] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry. Distributed control applications within sensor networks. *Proceedings of the IEEE*, 91(8):1235–1246, 2003.
- [94] R. E. Skelton, T. Iwasaki, and K. M. Grigoriadis. *A Unified Algebraic Approach To Control Design*. Taylor and Francis, London, 1997.
- [95] R. Solis, V. S. Borkar, and P. R. Kumar. A new distributed time synchronization protocol for multihop wireless networks. In *IEEE Conference on Decision and Control*, pages 2734–2739, 2006.
- [96] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
- [97] Crossbow Technology. MICA2 868, 916MHz. <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=174>.
- [98] J.N. Tsitsiklis and M. Athans. Convergence and asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, 29(1):42–50, 1984.
- [99] H. D. Tuan, P. Apkarian, and T. Q. Nguyen. Robust and reduced-order filtering: new LMI-based characterizations and methods. *IEEE Transactions on Signal Processing*, 49(12):2975–2984, 2001.
- [100] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

- [101] C. Vogel. A signal processing view on time-interleaved adcs. In A. H. M. Roermund, H. Casier, and M. Steyaert, editors, *Analog Circuit Design*, pages 61–78. Springer, Netherlands, 2010.
- [102] H. Wang, D Estrin, and L Girod. Preprocessing in a tiered sensor network for habitat monitoring. *EURASIP Journal on Advances in Signal Processing*, 2003:392–401, 2003.
- [103] A. Willig, K. Matheus, and A. Wolisz. Wireless technology in industrial networks. *Proceedings of the IEEE*, 93(6):1130–1151, 2005.
- [104] J. Wolfowitz. Product of indecomposable, aperiodic and stochastic matrices. *Proceedings of American Mathematical Society*, 14(5):733–737, 1963.
- [105] D. Wu, S. Ci, H. Luo, Y. Ye, and H. Wang. Video surveillance over wireless sensor and actuator networks using active cameras. *IEEE Transactions on Automatic Control*, 56(10):2467–2472, 2011.
- [106] F. Xiao and L. Wang. State consensus for multi-agent systems with switching topologies and time-varying delays. *International Journal of Control*, 79(10):1277–1284, 2006.
- [107] F. Xiao and L. Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control*, 53(8):1804–1816, 2008.
- [108] F. Xiao and L. Wang. Consensus protocols for discrete-time multi-agent systems with time-varying delays. *Automatica*, 44(10):2577–2582, 2008.
- [109] W. Yu, G. Chen, Z. Wang, and W. Yang. Distributed consensus filtering in sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1568–1577, 2009.