

FAULT DETECTION AND DISTRIBUTED ESTIMATION WITH SENSOR NETWORKS

Yilun Zhou

Department of Electrical and Electronic Engineering
Imperial College London

This dissertation is submitted for the degree of
Doctor of Philosophy

April 2017

I would like to dedicate this thesis to my loving parents and my fiancée

Declaration of Originality

I hereby declare that this thesis is the result of my own work, all material in this dissertation which is not my own work has been properly acknowledged.

Yilun Zhou
April 2017

Declaration of Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Yilun Zhou
April 2017

Acknowledgements

This thesis is the result of research work carried out at the Department of Electrical and Electronic Engineering, Imperial College London. First and foremost, I wish here to sincerely thank my supervisor, Prof. Thomas Parisini, that supported and guided me with patience and efforts during my whole PhD. He continuously motivated me to reach my objectives and provided me an excellent atmosphere for doing research. His profound insights and attention to details have been true inspirations to my research.

I would also like to thank Dr. Francesca Boem who has provided me with invaluable support and continuous encouragement. I am very fortunate to have had her as a collaborator during my PhD. Without her insightful comments and suggestions, my research would not carry out so smoothly.

My thanks also goes to ABB Corporate Research Center, that made an interesting industrial research project possible during my PhD. This three-year project offered me an excellent opportunity to collaborate with the great company in automation industry and gain a valuable industrial research experience.

Thanks to all the people of the control and power research group with whom I worked during these years, and for the perfect environment you created. Special gratitude goes to my friends, Boli Chen, Mingyang Sun, Yang Wang, Peng Li, Chen Wang, Dixi Liu, for helping me in different ways, also for all the great lunches and coffee breaks we had together in the past years.

The work in this thesis has been made possible by financial support from the Department of Electrical and Electronic Engineering, Imperial College London.

Last, but not least, I would like to express my deepest gratitude to my parents, for their everlasting love and unconditional support. And my final thanks is for Cheng Cheng, I could hardly have completed this work without her constant support and encouragement.

Abstract

A sensor network is a distributed system, consisting of plenty of embedded sensors that can be deployed over a large-scale physical environment. One of the major applications of sensor networks is to monitor the state of systems that are evolving in the sensing field. Thanks to the emergence of advancements in high-performance processors, nodes in a sensor network can not only collect measurements but coordinate to estimate the state of systems as well. This thesis proposes a monitoring architecture, where distributed state estimation and fault detection algorithms are implemented by every node in the sensor networks to track the system's state while simultaneously detecting the faults occurred in either the monitored systems or the sensor networks. Several approaches for different monitoring tasks are presented in this thesis and classified into two main parts: distributed state estimation and fault detection algorithms in the monitoring architecture.

In the first part, we present two distributed state estimation algorithms in the sensor networks for the monitoring of a system, which can be described by a centralized, decentralized, or distributed dynamic model. The first one is implemented over a sensor network, where the local estimator in each node consists of a filtering step – which uses a weighted combination of neighboring sensors information – and a model-based state predictor. The filtering weights and prediction parameters jointly minimize both the mean and variance of the prediction error in a Pareto optimization framework at each time step. Since each predictor uses the model of the whole system monitored by the sensor network, the algorithm over a sensor network can only monitor a centralized system or each subsystem of a decentralized system. For a distributed system, where several subsystems interact with each other, the second algorithm implemented over several sensor networks is introduced so that each sensor network can coordinate with neighboring networks to monitor the corresponding subsystem of the distributed system.

The second part of the thesis is devoted to fault detection algorithms for process faults in monitored systems and sensor faults in sensor networks. The aforementioned estimation algorithm over a sensor network is applied to design process fault detection

algorithm for a centralized or decentralized system. A residual is defined, and suitable stochastic thresholds are designed, allowing to set the parameters so to guarantee an upper bound of false alarms probability. For detecting sensor faults in the sensor networks, the centralized, decentralized, and distributed sensor fault detection schemes are proposed in a discrete-time framework. And the detection performance is compared by an industrial benchmark simulation in a continuous stirred tank heater (CSTH) pilot plant. Then a rigorous fault detectability and detection time interval analysis of the centralized sensor fault detection scheme is presented.

The performance of proposed distributed estimation methods and effectiveness of presented fault detection methods are evaluated by extensive numerical and industrial benchmark simulations.

Table of contents

Notation	xv
List of figures	xix
List of tables	xxiii
1 INTRODUCTION	1
1.1 Motivations	3
1.2 State of the Art	8
1.2.1 Distributed Estimation	8
1.2.2 Fault Detection Methods	12
1.3 Contribution of the Present Work	14
1.4 Publications	16
I DISTRIBUTED ESTIMATION ALGORITHMS	17
2 THE ALGORITHM FOR CENTRALIZED OR DECENTRALIZED SYSTEMS	19
2.1 Problem Formulation	20
2.2 Distributed State Dynamic Estimation	22
2.2.1 Local Filtering and Prediction Errors	23
2.2.2 Global Filtering and Prediction Errors	24
2.2.3 Stability of the Prediction Error	26
2.3 The Optimization Problem	28
2.3.1 Approximation Approach to the Optimization	30
2.3.2 The Optimal Weights for the Approximated Problem	34
2.4 Implementation of the Distributed Prediction Algorithm	39
2.4.1 Computational complexity	40
2.5 Simulation Results	40

2.6	Concluding Remarks	43
3	THE ALGORITHM FOR DISTRIBUTED SYSTEMS	47
3.1	Problem Formulation	48
3.2	Distributed State Dynamic Estimation	49
3.2.1	Local Filtering and Prediction Errors	50
3.2.2	Global Filtering and Prediction Errors	51
3.2.3	Stability of the Prediction Error	53
3.3	The Optimization Problem	56
3.3.1	Approximation Approach to the Optimization	57
3.3.2	The Optimal Weights for the Approximated Problem	62
3.4	Simulation Results	66
3.5	Concluding Remarks	69
II	FAULT DETECTION ALGORITHMS	71
4	PROCESS FAULT DETECTION SCHEME USING SENSOR NETWORKS	73
4.1	Problem Formulation	74
4.2	Distributed State Estimation	75
4.3	Distributed Fault Detection	77
4.3.1	Local Residual Analysis	78
4.3.2	Local Autocorrelation Matrix	79
4.3.3	The Fault Detection Thresholds	83
4.3.4	The Algorithm	84
4.4	Simulation Results	86
4.5	Concluding Remarks	90
5	SENSOR FAULT DETECTION SCHEMES IN THREE ARCHITECTURES	93
5.1	Tank Heater Plant Dynamics	94
5.1.1	The Stirred Tank Heater Process Model	95
5.1.2	Pilot Plant Simulation	96
5.2	Centralized Sensor Fault Detection Scheme	97
5.2.1	Learning Approach	99
5.2.2	Centralized Sensor Fault Detector	102
5.3	Decentralized Sensor Fault Detection Scheme	104
5.3.1	Learning Block ($\mathcal{L}_\star$)	105
5.3.2	Stop Learning Block ($\mathcal{S}_\star$)	107

5.3.3	Local Sensor Fault Detection Block ($\mathcal{D}_\star$)	108
5.4	Distributed Sensor Fault Detection Scheme	111
5.4.1	Learning Block ($\mathcal{L}_\star$)	113
5.4.2	Stop Learning Block ($\mathcal{S}_\star$)	113
5.4.3	Local Sensor Fault Detection Block ($\mathcal{D}_\star$)	113
5.5	Simulation Outcomes	116
5.5.1	Learning Performance	117
5.5.2	Sensor Fault Detection Scheme	120
5.6	Concluding Remarks	121
6	DRIFT SENSOR FAULT DETECTABILITY ANALYSIS	127
6.1	Problem Formulation	128
6.1.1	Nominal Plant, Uncertainty and Fault Representation	128
6.1.2	Fault Detection Scheme	130
6.2	Enhanced Fault Detectability Analysis	131
6.3	Detection Time Interval Analysis	136
6.4	Simulation Results	140
6.5	Concluding Remarks	145
7	CONCLUSIONS AND FUTURE PROSPECTS	147
7.1	Concluding Remarks	147
7.2	Future Work	149
	References	151

Notation

$\exists$	there exists
$\forall$	for all
$\in$	is an element of
$\subset$	is a subset of
$\cap$	set-theoretic intersection
$\cup$	set-theoretic union
$\setminus$	set-theoretic complement
$\vee$	logical disjunction
$\triangleq$	define
$\neq$	not equal
$\mathbb{R}$	real numbers
$\mathbb{R}^n$	real valued n -vectors
$\mathbb{R}^{m \times n}$	real valued $m \times n$ -matrices
$\mathbb{Z}$	the set of integers
$\mathbf{1}_m$	the m elements vector $(1, \dots, 1)^\top$
$\mathbf{0}_m$	the m elements vector $(0, \dots, 0)^\top$
I_m	the identity matrix of dimension $m \times m$
$\mathbb{E}x$	expected value of the stochastic variable x
$ \cdot $	absolute value, taken component-wise if the argument is not a scalar, or cardinality if the argument is a set
$\ \cdot\ _\infty$	the uniform norm of a matrix
$\circ$	the component-by-component product operator
$\otimes$	the Kronecker product
$\oplus$	the direct sum of scalars
$\bigoplus$	the direct sum of matrices
$\min$	smallest value of the function
$\inf$	infimum or greatest lower bound
$\sup$	supremum or greatest upper bound

$\arg$	argument or solution of an optimization problem
$\lim$	the limit of a function
$\mathcal{D}$	a compact region
$\mathcal{G}$	an undirected graph
$\mathcal{S}$	the set of the sensor nodes
$\mathcal{E}$	the set of the edges connecting communicating nodes
Σ_I	the subsystem I in a distributed system
$\mathcal{P}_{\mathcal{A}}$	a projection operator on the domain $\mathcal{A}$
Θ	domain to which $\hat{\theta}$ belongs
M_{Θ}	radius of the hyper-sphere that contains Θ

List of figures

1.1	Pictorial representation of a centralized, a decentralized and a distributed physical system [12].	1
1.2	Pictorial representation of a centralized (red), a decentralized (yellow), and a distributed (green) architecture applied to monitor a distributed system (white). Physical interaction between subsystems is represented by black arrows, while white thick arrows illustrate information exchanging channels [12].	2
1.3	A classic three-layer monitoring architecture	4
1.4	Diagram of the sensor network in the three-layer monitoring architecture. The black dashed lines represent measurements acquired from the physical layer; the orange block works as a sink sensor node; the orange thick arrow means aggregate data sent to application layer.	4
1.5	Diagram of the sensor network in the two-layer monitoring architecture. The black dashed lines represent measurements acquired from the physical layer; the red blocks work as sensors, while the green blocks illustrate the embedded system attached to each sensor; the orange thick arrows mean local computational results produced in each sensor node.	5
1.6	Pictorial representation of the two-layer monitoring architecture using sensor networks for a centralized (red), a decentralized (yellow), and a distributed (green) modern system.	6
1.7	A diagram for model-based fault detection approach. The derivation of estimates $\hat{x}$ and real values x is compared to the constant or adaptive threshold $\bar{\epsilon}$ [12].	13

2.1	Performance comparison in terms of MSE of the five considered estimation methods, with different standard deviation values for the measurement noises and the process disturbance. The first three rows consider the Gaussian case. In the fourth row, the white Gaussian noise case is represented by solid lines, while the dashed lines show the non-Gaussian noise case performance.	45
2.2	The distance between the performance of the proposed approach E_{p1} and the Centralized Kalman Filter in terms of ratio of respective MSE values for the 30 different noise scenarios, both in the Gaussian and in the non-Gaussian case.	46
3.1	The 4-tank system monitored by four sensor networks. Each tank is monitored by a set of sensors (having the same color as the corresponding tank level). The communication network is represented by dotted arrows (black arrows for local communication, yellow arrows for communication with sensors monitoring neighboring subsystems)	67
3.2	MSE of 3 prediction methods with different measurement standard deviation. Disturbance standard deviation is 0.05.	69
3.3	MSE of 3 prediction methods with different measurement standard deviation. Disturbance standard deviation is 0.09.	70
4.1	The schematic diagram of the four-tank process.	87
4.2	Estimates, Residuals and Thresholds for state component 3.	88
4.3	Estimates, Residuals and Thresholds for state component 1.	88
4.4	Detection time with various leakage size	90
4.5	Detection time with various leakage size and chosen $\alpha = 0.8$	91
4.6	Detection time with various leakage size and chosen $\alpha = 1.0$	91
4.7	Detection time with various leakage size and chosen $\alpha = 1.2$	92
5.1	The continuous stirred tank heater pilot plant	94
5.2	Closed-loop control diagram in the benchmark	97
5.3	Centralized sensor fault detection scheme for the sensors in the benchmark	98
5.4	Decentralized fault detection scheme without information exchange among Subsystems (1), (2), and (3). The component in each subsystem is given: $Sub_{(1)} = \{x_1\}$, $Sub_{(2)} = \{x_2\}$, $Sub_{(3)} = \{x_3\}$	104

5.5	General architecture of decentralized sensor fault detection scheme for Subsystem (1). The timeline is shown enhancing the fact that learning is only activated during the time-horizon $(0, T_L)$ in which it is assumed that the system is not faulty.	106
5.6	Distributed fault detection scheme with information exchange among Subsystems (1), (2), and (3). The component in each subsystem is given: $Sub_{(1)} = \{x_1\}, Sub_{(2)} = \{x_2\}, Sub_{(3)} = \{x_3\}$	111
5.7	General architecture of distributed sensor fault detection scheme for Subsystem (3). The timeline is shown enhancing the fact that learning is only activated during the time-horizon $(0, T_L)$ in which it is assumed that the system is not faulty.	112
5.8	General Simulink model of centralized, decentralized, and distributed sensor fault detection schemes	116
5.9	Simulink Block Benchmark_f : Stirred tank heater pilot plant	117
5.10	Centralized, decentralized, and distributed learning cost functions . . .	117
5.11	Learning performance on output disturbance	118
5.12	Learning performance on real and learnt modeling uncertainties	119
5.13	Details of the learning performance on real and learnt modeling uncertainties	120
5.14	Learning performance on updated modeling uncertainties	121
5.15	Centralized sensor detection scheme in y_1	122
5.16	Decentralized sensor detection scheme in y_1	123
5.17	Distributed sensor detection in y_1	123
5.18	Centralized sensor detection scheme in y_2	124
5.19	Decentralized sensor detection scheme in y_2	124
5.20	Distributed sensor detection scheme in y_2	125
5.21	Centralized sensor detection scheme in y_3	125
5.22	Decentralized sensor detection scheme in y_3	126
5.23	Distributed sensor detection scheme in y_3	126
6.1	Time-evolution of a drift sensor fault (red line) and the corresponding lower bound (black line).	129
6.2	Time-evolution of a sensor fault (red curve) and the class of detectable faults in s^{th} sensor (purple shade).	136
6.3	Time-evolution of an arbitrary sensor fault (red curve) and the class of detectable faults in s^{th} sensor (purple shade).	137
6.4	Ideal drift sensor fault detected in Sensor 3.	142

6.5	Arbitrary drift sensor fault in Sensor 2	143
6.6	Relationship between upper bound $\bar{t}_d^s$ and norm of sensor fault saturation level $ \theta^s $	144
7.1	The Tennessee Eastman Challenge Process with Four Sensor Networks	150

List of tables

3.1	: The parameters in the the four-tank process.	66
4.1	: FAR with various α and SNR.	89
4.2	: MIS with various α and SNR.	89

Chapter 1

INTRODUCTION

Nowadays, a noticeable trend in modern industrial and commercial systems monitoring is to integrate communication, control, and computing into different levels of physical operations and information processes. This trend is justified by the emergence of complex modern systems, which are *large-scale* and/or *distributed*. The term *large-scale* refers to physical systems with a large number of state components, and the term *distributed* describes that the structure of physical systems can be constituted by multiple subsystems that interact with neighboring subsystems. According to this classification of physical systems structure, there exist the other two categories, namely *centralized* and *decentralized*, which both have less complexity in contrast with *distributed*. The term *centralized* means the structure where every component in the physical systems interacts with every other one. And the term *decentralized* refers to the physical systems made of multiple subsystems that do not interact with neighboring subsystems. A pictorial representation of these three categories is illustrated in Fig.1.1. To give an instance, the large-scale chemical plant, where many units, such as reactors, condensers, and separators, interact with each other in a distributed way to maintain efficient production. It is not reasonable to understand these units without considering the influence from other coordinated parts. On the other hand, the traditional communication approach for control systems, which has been widely applied for decades, is based on a centralized communication network. Specifically, the centralized control room receives each sensor measurement and sends orders to each actuator. However, increasing scale of the physical operations and functionalities of the control approach are pushing the limits of a centralized communication network. Meanwhile, another drawback of this communication approach is the limits of computation power at the centralized control room. Hence, the modern systems with complicated topologies require a more sophisticated monitoring *architecture* for the integration

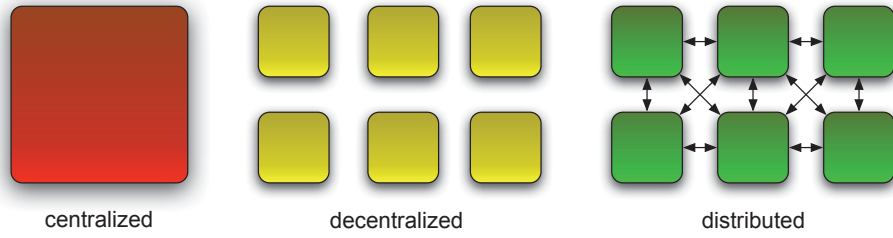


Fig. 1.1 Pictorial representation of a centralized, a decentralized and a distributed physical system [12].

of communication, control, and computing. In this work, the term *architecture* is denoted as a combination of hardware and software used to achieve modern systems monitoring tasks. The scientific research has been a gradual improvement in the design of architectures over the recent decades. Just as the classification in physical systems structure, it is shown a pictorial representation of the centralized, decentralized, and distributed architectures applied to monitor a distributed physical system in Fig. 1.2. The centralized architecture is unfeasible to receive all the measurements from multiple subsystems at a centralized location. The decentralized and distributed architectures are both composed of several subsystems monitoring the corresponding components in the physical systems. While, different from the subsystems in decentralized architecture, subsystems in the distributed architecture can interact with neighboring subsystems, which deals with distributed physical systems in a similar topology.

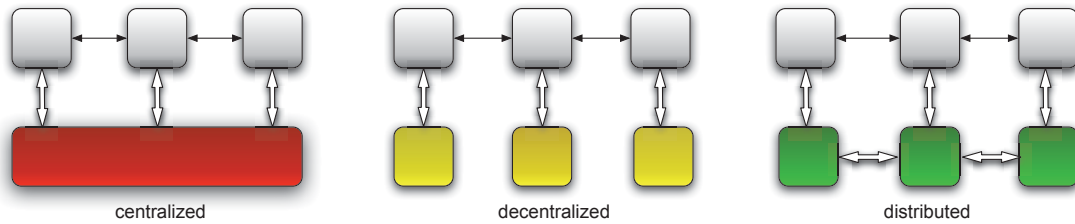


Fig. 1.2 Pictorial representation of a centralized (red), a decentralized (yellow), and a distributed (green) architecture applied to monitor a distributed system (white). Physical interaction between subsystems is represented by black arrows, while white thick arrows illustrate information exchanging channels [12].

And a new concept, Cyber-physical Systems (CPS), has been recently introduced and has attracted considerable attention from academia, industry, and government due to the potential revolution in the monitoring architectures of complex modern systems (see [6, 75, 126]). The concept refers to a new generation of engineered systems with integrated computational and physical capabilities that can achieve stability,

functionality, reliability, robustness, and efficiency in many application domains such as environment monitoring, critical infrastructure, automotive, and healthcare systems. Cyber-physical Systems bridge the cyber capabilities (e.g., communication, control, and computation) to the physical operations through lots of sensors, controllers, and actuators. This concept is not limited to the innovation of software side (e.g., architectures, algorithms, and implementation methods), another remarkable breakthrough is due to the hardware applications, such as sensor networks, embedded systems, and high-performance processors. Let us consider a sensor network [156], where a large number of sensors communicate with each other to achieve a designed task. The inherent characteristics of sensor networks allow more interaction among hardware devices and their surrounding physical environment; faster and easier deployment in a large-scale and needed topology; and larger observation redundancy and higher monitoring precision. Hence, sensor networks play a vital role to gather and refine information from a large amount of raw data in the physical operations. And they are critical to the monitoring of complex modern systems, such as water management, large-scale industrial processes, traffic management, and smart cities.

1.1 Motivations

The monitoring architecture is a long-standing research challenge along with the development of industrial and commercial systems. When these systems start to present the characteristics, including large-scale and/or distributed, the research interest in decentralized and distributed monitoring architecture emerges in the control science community. In practice, these monitoring architectures are developed without considering a rigorous software and hardware implementation approach. Although computer science researchers have achieved significant breakthroughs in real-time computing techniques, embedded systems architectures, and new programming languages, there does not exist a standardized methodology to fill the gap between existing research outcomes of control and computer science. Hence, it is difficult and costly to integrate these two research disciplines when dealing with complex modern systems monitoring. That is the main reason why both academia and industry actively promote CPS research to bring successful control theories and powerful computational capabilities together under a standardized and unified architecture. Let us consider a *classic monitoring architecture* from control community in [75], composed of three layers (see Fig. 1.3), namely physical layer, communication layer, and application layer. The physical layer represents the dynamical system that has to be estimated and monitored. The dynamic

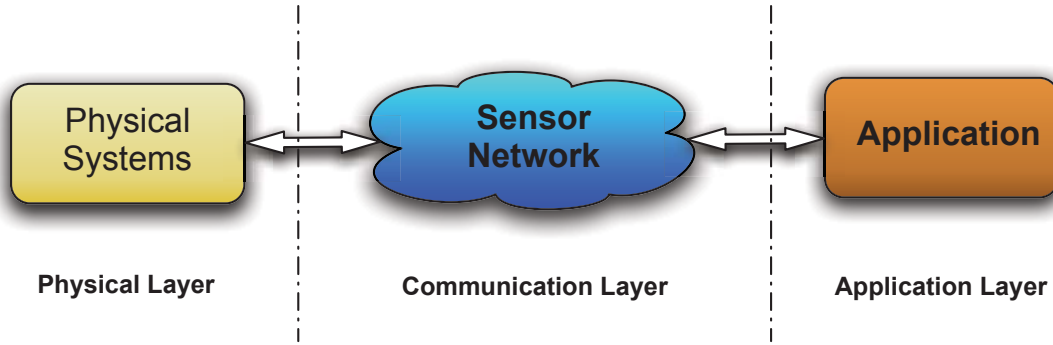


Fig. 1.3 A classic three-layer monitoring architecture

system is represented by a discrete-time or continuous-time mathematical model. Based on the complexity of physical systems, the mathematical model can be formatted as a centralized, decentralized, or distributed framework, whose pictorial representation is presented in Fig.1.1. The primary task of communication layer is to provide a reliable communication channel for exchanging information between physical and application layers. In this layer, a sensor network, illustrated in Fig.1.4, is responsible for collecting

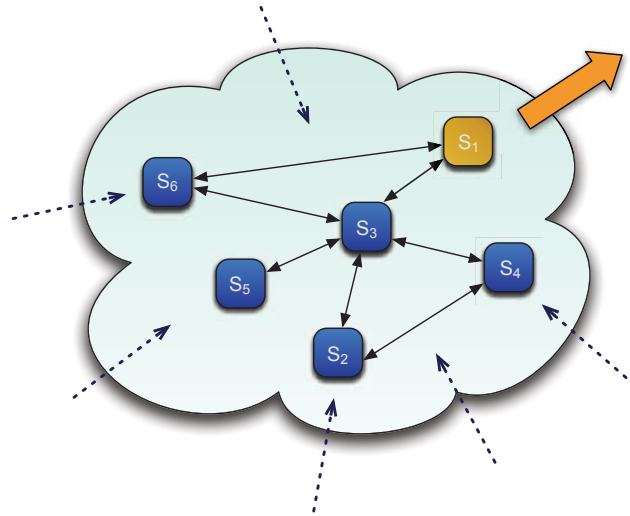


Fig. 1.4 Diagram of the sensor network in the three-layer monitoring architecture. The black dashed lines represent measurements acquired from the physical layer; the orange block works as a sink sensor node; the orange thick arrow means aggregate data sent to application layer.

and processing information, then sending the aggregate data to sink nodes S_1 in Fig. 1.4. The application layer takes charge of computational activities to process the aggregate data and to achieve the monitoring tasks in a centralized way. In this three-layer monitoring architecture, the sensing data flows from the sensor network to

a fusion center in the application layer, which means this approach requires a powerful central processor in the fusion center and stronger communication capability on the communication layer. Once the physical systems are complex or the monitoring tasks become challenging, there exist significant restrictions on communication bandwidth among sink nodes, energy resources in sink nodes, and computational capability in the application layer. Hence, to improve monitoring precision in an energy-efficient way, the design of data processing algorithms and networking structures in the sensor network should be developed in a compromise among monitoring precision, computational capability, and energy consumption.

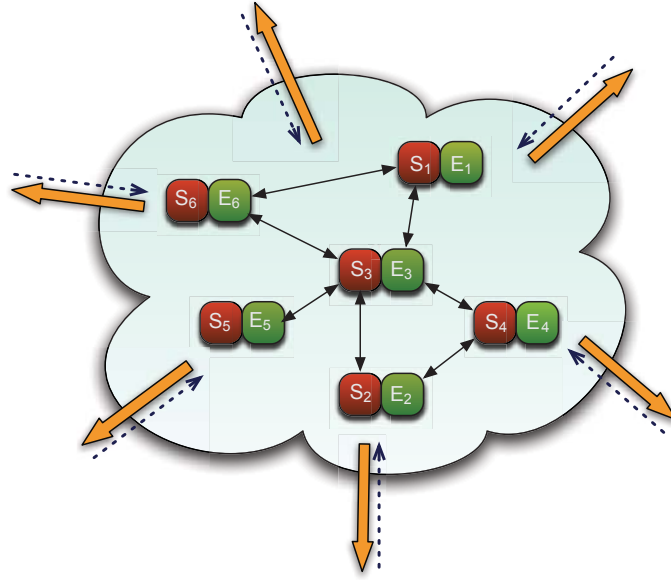


Fig. 1.5 Diagram of the sensor network in the two-layer monitoring architecture. The black dashed lines represent measurements acquired from the physical layer; the red blocks work as sensors, while the green blocks illustrate the embedded system attached to each sensor; the orange thick arrows mean local computational results produced in each sensor node.

However, the emergence of advancements in high-performance processors and real-time embedded systems offers an opportunity to develop a *two-layer monitoring architecture*, where sensor nodes can not only collect and process data but attach embedded systems to obtain computational capability as well. In other words, the functionality of application layer in the classic architecture can be achieved by the embedded system attached at each sensor node in the sensor network, presented in Fig.1.5. A sensor network is a distributed system, consisting of plenty of embedded sensors, distributed over a large-scale physical environment; therefore, distributed data

processing algorithms are more suitable to process information locally at various sensor nodes. More importantly, as the computational capability and memory resources of sensor nodes increase, more complicated networking structures and data processing algorithms can be designed without considering the strict restrictions on communication bandwidth and energy resources. Hence, it would be a useful approach to applying distributed estimation algorithms in sensor networks to achieve higher monitoring precision and lower energy consumption. Moreover, the distributed algorithms are not only applied in a single sensor network, where sensor nodes inside exchange information; when several sensor networks collaborate to monitor complex and networked systems, the algorithms can also support data exchange among sensor nodes located in various sensor networks. For the various structures of physical systems to be monitored, it

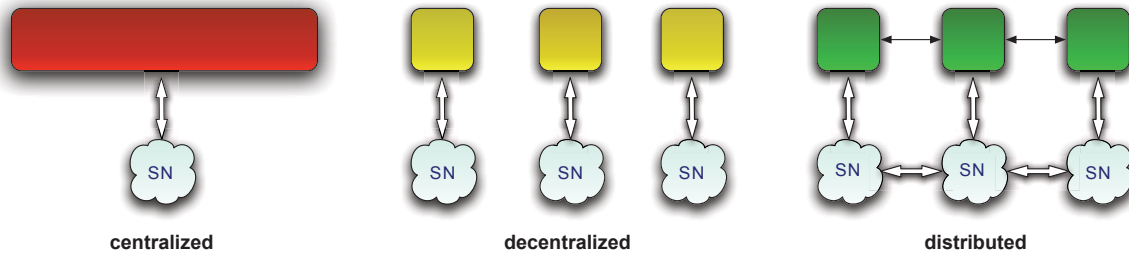


Fig. 1.6 Pictorial representation of the two-layer monitoring architecture using sensor networks for a centralized (red), a decentralized (yellow), and a distributed (green) modern system.

is shown in Fig. 1.6, a pictorial representation of the centralized, decentralized, and distributed topologies of sensor networks in the two-layer monitoring architecture. For centralized and decentralized modern systems, sensor networks are applied to monitor the systems independently. In the monitoring architecture of distributed physical systems, each sensor network can communicate with neighboring sensor networks to track the systems in a distributed manner.

Generally, in the monitoring architecture for centralized and decentralized systems, it is assumed that distributed estimation in a sensor network works according to the following procedure: Firstly, each node communicates measurements and estimates to neighboring nodes. Secondly, each node locally filters the measurements by taking a linear combination of its own and neighboring's measurements and estimates. Finally, the node uses the current filtered measurements and estimates to implement a model-based predictor, smoothing the previous estimation error. In the monitoring architecture for distributed systems, the model-based predictor in each sensor node not only interacts with neighboring nodes in the same network but communicates with neighboring nodes

located in other sensor networks. This distributed estimation approach makes full use of the characteristics of sensor networks so that the sensor nodes can process and exchange information in a distributed manner to improve monitoring precision significantly. Also, this two-layer architecture can achieve data processing in real time by avoiding information delay and dropouts on the communication channels between sensor networks and the fusion center.

Another fundamental research topic is *reliability* of the monitoring architecture. Frameworks, algorithms, control approaches, and implementation methods are developed to satisfy the *reliability* requirement for heterogeneous cooperating elements that interact with each other in a complex physical environment operating over various spatial and temporal scales [6]. In the two-layer monitoring architecture we proposed above, *reliability* can be considered as the ability to perform its intended functions: the architecture not only can monitor the physical systems accurately but also should be able to detect process faults in the physical operations and sensor faults in the sensor networks. Let us consider the complex modern systems, which are networked at multiple scales, both in time and space, especially with a considerable amount of system states. It is noteworthy that increased complexity of the modern systems implies a high fault ratio in the monitoring architectures, such as failures in part of the physical operations and sensor measurements drift faults. Hence, the design of fault detection schemes is the primary research task in monitoring architectures for complex modern systems.

Historically, the fault detection methods did not depend on a mathematical model of the system to be monitored but relied on the knowledge of a range in which the monitored variables are under normal conditions. Once the monitored variables are out of the range, the detector will raise a fault alarm immediately. Later on, owing to stronger computational capability, the model-based fault detection methods have been widely introduced in control community. The idea of model-based methods is to estimate monitored variables by using the mathematical model of the system in a healthy condition; then, the fault is detected when the difference between estimation and the real measurement exceeds a designed threshold. As the knowledge-based methods need a comprehensive understanding of the previous behavior of the system in a healthy condition, it is significantly costly to develop a robust and reliable fault detection scheme for the complex systems; therefore, the model-based fault detection methods are more suitable for developing a reliable monitoring architecture. Considering process faults in the physical operations, the model-based distributed estimation approach, implemented in sensor networks of the two-layer architecture, can be applied to design

a distributed fault detection method to monitor the physical operations locally in real time. More importantly, the reliability of raw measurements collected in each sensor is a prerequisite to the well-designed monitoring architecture. Any malfunction may occur in sensors, which could be due to an abrupt bias or an incipient drift in the instruments. The computational capability in sensor networks makes it possible to attach a local sensor fault detector to the sensor node, where the local detector estimates the states monitored by the corresponding sensor node. According to the structure of monitored physical systems, the local fault detection scheme can be designed in a centralized, decentralized, or distributed manner.

1.2 State of the Art

In the previous section, we have seen that sensor networks present enormous potential for complex systems monitoring. Apart from the monitoring precision, the reliability is also a fundamental prerequisite to a well-designed monitoring architecture. A wide range of methods for distributed estimation with sensor networks and fault detection are already available in the control community. These approaches exploit research topics such as distributed Kalman filtering, distributed state estimation based on stability analysis or mean squared error (MSE) analysis, and sensor fault detection schemes in centralized, decentralized, or distributed architecture. A comprehensive review of some approaches in these categories is carried out in this section.

1.2.1 Distributed Estimation

One of the fundamental applications of sensor networks is to estimate and track the state of targets or processes that are evolving in the sensing field. Useful in many monitoring scenarios, such as for example, target tracking and environment and agriculture monitoring ([3, 34, 79, 132, 133, 137]). Traditionally, in a centralized application, sensor nodes would aggregate all the data and communicate their measurements to a central fusion center for processing, and the calculated estimates would be sent back to all the nodes or actuators. As the fusion center has data from all the sensor nodes, the centralized methods can deal with the global algorithms, and will, therefore, be able to produce optimal estimates. Most of the research, introduced in several survey papers ([2], [118] and [90] and the references cited therein), focuses on data aggregation under the centralized scenario. However, when monitoring large-scale systems, the centralized methods require a large amount of communication and energy [44]. Moreover, this

monitoring architecture demonstrates the lack of robustness to any failures occurred in sensor networks or the fusion center. An alternative approach is the distributed architecture, where sensor nodes interact with their neighborings and no fusion center is needed [44, 80, 128, 129]. The state of the system has to be estimated in a distributed manner at each sensor in sensor networks. Hence, the problem of distributed estimation over sensor networks has been attracting growing research interests. Given their importance, distributed estimators have been the subject of many investigations in the area of networked control and monitoring (see, for example, [7, 32, 35, 47, 53, 58, 102, 140] and the references cited therein). Some notable application examples of distributed estimation using sensor networks are: target tracking ([60, 62, 81, 86, 162, 163]), source localization ([30, 70, 116]), and rendezvous of unmanned vehicles ([49, 66, 117, 120]).

Depending on the manner by which the sensor nodes communicate with each other, there exist two distributed estimation schemes: incremental schemes ([26, 88, 99]) and diffusion schemes ([22–25, 27, 89, 141]). In incremental schemes, data flows from one sensor node to a neighboring node on a cyclic path, where each sensor node can process data locally. Though the incremental schemes show global performance, it is expensive to design an optimized cyclic path for large-scale systems. Moreover, any faults on this path would cause a failure of the whole scheme. An alternative approach is diffusion schemes, where a sensor node can communicate with all its neighboring nodes without a cyclic path. Though the diffusion schemes have more communication and worse performance compared to the incremental schemes, the robustness to any faults is improved considerably, and distributed estimation algorithms under the diffusion mechanism become an active research area. In [22, 24], the diffusion least mean squares (LMS) method is introduced to achieve distributed estimation in sensor networks, while the diffusion Kalman filtering methods are proposed in [23, 25] to process data locally at each sensor node.

There are three arguably representative distributed estimation strategies have been proposed in the recent literature [41]: the distributed estimation algorithm to guarantee the stability or $\mathcal{H}_\infty$ performance of the estimation error dynamics; the distributed Kalman filter algorithm with consensus protocols; and the distributed state estimation based on mean square error (MSE) analysis. The distributed state estimation with stability analysis is a system reconstruction problem. The primary task is to exploit the Lyapunov stability theory to augmented systems, composed of the system dynamics and estimation error dynamics. Literature in [42, 84, 85] all demonstrate a distributed state estimation method in the sensor networks to guarantee the stability when dealing with randomly varying nonlinearities and measurements failure in the systems. Moreover,

the $\mathcal{H}_\infty$ filters have been widely applied to offer a constraint of the worst-case estimation error, which is achieved by the linear matrix inequality or Riccati equation approach. For a distributed topology in the sensor networks, the centralized $\mathcal{H}_\infty$ filter approach is replaced by a $\mathcal{H}_\infty$ consensus method (see, for example, [40, 130, 144, 155, 157]), where each local $\mathcal{H}_\infty$ filter is calculated in a recursive manner to reach the average consensus of the estimation errors in all sensor nodes.

The traditional Kalman filter keeps track of the estimated state of the system and the variance of the estimate in a two-step process. This method has been widely used in numerous applications: navigation and control of vehicles, robotic motion control, and target tracking. When the estimation problem is solved by a centralized approach, the Kalman filter is optimal under Gaussian assumption on the noises and represents the best linear filter also when disturbances are non-Gaussian [36]. For the monitoring tasks using sensor networks, the most fundamental objective is to design distributed estimation algorithms. Thus, the distributed Kalman filtering becomes an active area of research in control community, (see, [1, 33, 72, 74, 92, 106, 131] and two survey papers [41] and [91], covering distributed filtering methods over sensor networks and distributed Kalman filtering methods are presented, respectively.) The concept *consensus* in computer science becomes popular in the research of system and control theory (see, for example, [45, 58, 63, 97, 107, 119, 138]). It is introduced in [105] a *consensus algorithm*: an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network. The distributed Kalman filter methods based on the consensus strategies are proposed, e.g., in [71, 101, 103, 134], where the Kalman filter approximates the information form of the centralized Kalman filtering with two steps: Firstly, each sensor node fuses the measurements and covariance matrices of its neighbors by using the consensus algorithm; then, the local Kalman filter is performed on each sensor node with the fused measurement and covariance matrix. Moreover, in [104] an optimal decentralized Kalman-Consensus Filtering approach is designed with the objectives of estimating the state of the system and reaching a consensus with neighboring estimator agents on the estimate. Then in [21], the distributed Kalman-Consensus based approaches are investigated for the scalar case to obtain optimal Kalman gains and consensus weights such that the variance of the estimation error is minimized.

The distributed state estimation based on MSE analysis has recently become another active research approach to monitor systems using sensor networks, where each sensor node computes its estimate as a weighted sum of its own and its neighbors' measurements and estimates. The weights are adaptively updated to minimize the

variance of the estimation error. In [153], it is proposed a sensor fusion method based on consensus filtering, each node computes a locally weighted least-squares estimate that can converge to the maximum likelihood solution for the overall network. Moreover, in [136], a minimum variance distributed estimator for tracking of a noisy time-varying scalar signal is proposed. And both state estimation and the weight optimization are achieved in a distributed manner to minimize the variance of estimation error at each sensor node. Some papers exist addressing the distributed state estimation based on minimizing the variance of estimation error [8, 19, 24, 82, 93, 100, 127, 152, 154], but few works are dealing with both the variance and the bias of the estimation error as a multi-objective optimization problem. To develop adaptive weights of variance and bias in the optimization problem, Pareto optimality approach, proposed in [57], can be applied in the distributed estimation methods. Pareto optimization has been used in sensor networks for several tasks (see, [5, 29, 37, 68, 69]). Specifically in [68], the optimization of wireless sensor network layout is analyzed to determine the best sensor placement, where the competing objectives considered are the total sensor coverage and the lifetime of the network. To the best of author's knowledge, as an extended work of [136], it is presented in [15] the first approach to developed a distributed estimation method based on Pareto optimization to track a time-varying scalar signal by a sensor network despite the presence of noise. And the state estimation and Pareto optimization are achieved at each sensor node to minimize both the variance and the bias of the estimation error. Unlike the approach in [15] to achieve Pareto optimality at each sensor node, [29] introduces a solution to minimize the sum of individual cost function at each sensor node in a Pareto strategy, which is "optimal" in some sense for the entire sensor network.

However, the two-step approach has also been a popular research topic in the sensor network literature (see [21] for example). To serve different purposes, the distributed estimation method using sensor networks prefers keeping steps separate. In fact, in the literature, it is a common choice to present prediction methods employing a two-step strategy composed of a merging-update step and a prediction step (see, for instance, contributions about Distributed Kalman Filtering like [4, 21]). Within the context of distributed estimation over sensor networks, the paper [135] was the first one to propose this two-stage strategy based on i) computing the mean of the sensor measurements via consensus algorithms, and then ii) updating and predicting the local estimates using the centralized Kalman optimal gains. This approach has been then extended in the literature [101, 134]. Then, considering Pareto optimization framework in [15], it is introduced in [17] a two-step distributed estimator to using a

sensor network, where the estimator in each node consists of a filtering step – which uses a weighted combination of measurements and estimates – and a model-based predictor of the system’s state. This method is only assumed to know the mean and variance of disturbance, unlike [4, 20, 103, 104], which need assumption on the distribution of the disturbance, or [9], where it is necessary to know the PDFs of measurement noises and modeling uncertainties. Moreover, this work is not focused on static estimation parameters in [38], or where only the measurements are diffused over the network [22, 89]. Instead, the simultaneous design of the filtering weights and prediction parameters in a Pareto framework at each time-step is proposed in [17] to minimize both the bias and the variance of the prediction error. And the filtering and prediction steps are jointly optimized at each sensor node to achieve both instantaneous performance and asymptotic convergence.

Either in the scalar signal scenario [15] or in the multi-state system scenario [17], the distributed state estimation method based on a Pareto framework is implemented in a single sensor network, where sensor nodes only diffuse information inside the network. And these methods are not assumed to know the explicit topology of the sensor network. Hence, these two proposed methods are both suitable to be implemented in the monitoring architecture for centralized or decentralized systems, shown in Fig. 1.6. Nevertheless, there is an increasing demand for innovative methods for the monitoring of large-scale interconnected systems, characterized by the interconnection of some subsystems. This problem is challenging that a single sensor network architecture proposed in [17] is not feasible to monitor the whole systems without the assumption on the observability of every sensor node. However, the monitoring architecture for distributed systems in Fig. 1.6, provides a strategy that each sensor network can monitor one subsystem by using local model information and communicating with neighboring networks. Thus, the partition-based state estimation method becomes an active research area (see [46, 48, 50, 74, 124, 125, 139]), where each local agent estimates part of the global systems with information exchange with neighboring agents. Specifically, [46] introduces a Kalman-filter based distributed estimator without consensus strategy for non-overlapping interconnected subsystems. But none of these works use sensor networks for the estimation task allowing more than one sensors to measure the same state variables. However, the partition-based state estimation using sensor networks in a Pareto framework is a promising monitoring solution for large-scale distributed systems.

1.2.2 Fault Detection Methods

The process of fault diagnosis can be divided into two steps: a) fault detection determines whether a fault has occurred or not; b) fault isolation identifies and locates the fault. Various fault diagnosis architectures have been published during the last decades due to the growing demand for fault tolerance achieved by an efficient fault diagnosis scheme. Historically, the traditional engineering approach of physical redundancy is widely used in health monitoring system. The methods, such as limit checking and signal models in [65], often need additional space and cost, but did not depend on a dynamical model of the system. Thanks to the understanding and modeling of the dynamic processes, a dynamical model of the physical system can be used to estimate the variable states and parameters to detect any changes in the system dynamics. Residual generation can be a helpful method to show the difference of estimates and real values. And fixed or adaptive threshold is a straightforward and sensitive criterion of fault detection. So the model-based approach, illustrated in Fig.1.7, could be proven to achieve better performance. The books by Gertler [59], Blanke et

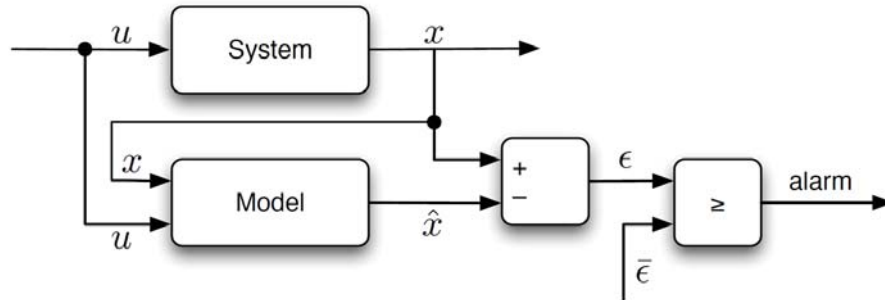


Fig. 1.7 A diagram for model-based fault detection approach. The derivation of estimates $\hat{x}$ and real values x is compared to the constant or adaptive threshold $\bar{\epsilon}$ [12].

al. [11] and Isermann [65] present a lot of achievement in fault diagnosis architectures for various class of dynamical systems. Usually, the method can be divided into two major sections, namely analytical redundancy approach and knowledge-based approach [56]. A survey paper by Isermann [64] summarizes estimation methods used in fault diagnosis approaches. Later in [55] introduces a comprehensive review of analytical and knowledge-based redundancy approaches used in model-based fault diagnosis schemes. The problem of automated fault diagnosis is motivated by the need to develop more autonomous and intelligent systems that have a reliable operation in the presence of system faults. With the development of computer technology, the restriction of computational capability is relaxed so that analytical redundancy approaches can be

widely applied in fault diagnosis architectures. Specifically, learning approaches show advantages in fault diagnosis schemes for nonlinear systems. [112, 115] summarize a general learning approach using on-line approximation and adaptive algorithm. The structure such as neural networks, mentioned in [113], can be designed as a generic estimator for approximating the fault function, and a stable learning scheme is achieved by the adaptive algorithm in [110] and [111]. Later, there has been considerable research efforts [39, 108, 147, 148] on model-based fault diagnosis schemes under diversified assumptions and fault scenarios. All the fault diagnosis schemes mentioned above can be only applied in centralized systems. Let us consider the large-scale system composed of several subsystems, then distributed fault diagnosis architectures recently become a new research direction (see, for example, [13, 14, 52, 83, 158]). Specifically, Ferrari et al. [51] introduce a distributed fault detection methodology for distributed large-scale nonlinear dynamical systems, where local fault detectors can interconnect with their neighbors. Then a novel filtering approach is designed in [73], where the filtering approach can remove the high-frequency measurement uncertainty to avoid false-positive alarms by replacing the conservative thresholds.

There is an assumption by now in all the fault diagnosis schemes that the measurements by sensors are ideal without any faults. Any malfunction may also occur in sensors, which could be due to an abrupt bias or an incipient drift in the instruments. Hence, sensor fault detection schemes attract much attention over the last few decades. A unified framework for fault diagnosis approach is proposed in [143], considering both state and sensor faults. Kalman filters are used to detect sensor fault in [96], [78]. Adaptive methods for diagnosing sensor faults are investigated by Wang et al. [150], Reppa et al. [122] and Vemuri [145], [146]. Learning approaches are applied in sensor fault diagnosis as well. In [149], it is introduced a learning-based diagnosis scheme for input dependent bias in sensors. For the sensor networks in a large-scale system, either sensor faults or the stochastic delays and packet losses in the communication network between the local fault diagnosers are inevitable. There exist some works addressing the problem of fault diagnosis of sensor networks, such as sensor fault detecting, packet losses, and energy consumption monitoring (see, [28, 76, 87, 98, 121, 151]).

There exist many works in the literature dealing with distributed estimation methods using sensor networks (see, as examples, introduced at Section 1.2.1). But, the methods of using sensor networks to detect the process faults is a relatively new field of research. An exception is [54], where a distributed fault detection and isolation technique is designed, relying on decentralized Kalman filtering and based on a distributed hypothesis testing method. [31] uses sensor networks for distributed

estimation, but the fault detection decision is centralized based on a fusion center. Then in [16], a distributed fault detection scheme using sensor network is proposed based on the distributed filtering method presented in [15], where each sensor node monitors a scalar signal in a Pareto optimization framework. Thanks to the way it is designed, it allows to compute mean and variance of the estimation error, thus allowing to obtain suitable thresholds for some proper residual signals with known confidence levels and definable rate of false alarms. Later on, based on the distributed estimation method in [17], it is proposed a distributed fault detection scheme in [164], where each sensor node can detect the process faults in multi-state centralized or decentralized system in a Pareto multi-objective optimization.

Most of literature [39, 108, 143, 147–149, 159], discuss robustness and stability after introducing the fault diagnosis schemes. Unlike the robustness and stability achieved by modeling uncertainty bounds and adaptive algorithm respectively, another performance property, named detectability, is relatively difficult to be derived due to coupling between modeling uncertainty and sensor faults. The existing fault detectability analysis can be found in [143, 148, 161]. Moreover, Polycarpou et al. [114] investigate a rigorous detectability analysis for system fault diagnosis scheme with full-state measurement. Consider a class of nonlinear uncertain systems with an incipient sensor fault, [165] proposes the off-line assessment criterion for sensor fault detectability, where the set of detectable sensor faults can be characterized before activating the corresponding sensor fault detection scheme. Moreover, the upper bound on detection time interval can be analytically outlined as well.

1.3 Contribution of the Present Work

The primary objective of this thesis is to provide a monitoring architecture using sensor networks, where distributed state estimation algorithms are implemented by every sensor node to track the systems while simultaneously to detect the faults occurred in the systems. Meanwhile, the centralized, decentralized, and distributed sensor fault detection schemes are introduced and compared by an industrial benchmark simulation in a continuous stirred tank heater (CSTH) pilot plant. Also, the rigorous fault detectability and detection time interval analysis for a centralized sensor fault detection scheme is presented. The thesis consists of two main parts. Part I contains distributed state estimation algorithms for the monitoring of modern systems, whereas Part II includes various fault detection schemes for process faults and sensor faults. The contribution of each chapter is briefly outlined in the following.

In Chapter 2, a novel model-based distributed state estimation method using a sensor network is proposed for the monitoring architecture of centralized and decentralized systems. Each sensor can measure only a subset of the state variables, and no probability distribution knowledge is assumed for the noise variables. The local estimator in each sensor node consists of a filtering step – which uses a weighted combination of neighboring sensors information – and a model-based state predictor of the monitored system. The filtering weights and prediction parameters jointly minimize both the bias and the variance of the prediction error in a Pareto optimization framework at each time step.

In Chapter 3, a new model-based distributed state estimation method using sensor networks is proposed for the monitoring architecture of distributed systems. No probability distribution knowledge is required for the noise variables. Each node filters the information of neighboring nodes in the local sensor network; then, computes a prediction of the state of the monitored subsystem based on the knowledge of the local model and communicating only with the connected nodes in the neighboring sensor networks without the knowledge of the global model. The filtering weights and prediction parameters are locally optimized to minimize at each time-step bias and variance of the prediction error using a multi-objective Pareto optimization framework.

Chapter 4 deals with a distributed process fault detection method using a sensor network in the monitoring architecture of centralized and decentralized systems. In this chapter, the individual correlations between the state, the measurements, and the noise components are considered. No assumptions on the probability distribution knowledge are needed for the noise variables. Each sensor communicates with the neighboring nodes without centralized coordination to compute an estimate of the state of the system to monitor. A residual is defined, and suitable stochastic thresholds are designed, allowing to set the parameters so to guarantee a maximum false alarms probability. Moreover, an extensive sensitivity analysis is introduced, concerning fault magnitude and measurement noise.

In Chapter 5, the centralized, decentralized, and distributed sensor fault detection schemes are presented in a discrete-time framework. These sensor detection schemes are tested and compared by an industrial benchmark simulation in a continuous stirred tank heater (CSTH) pilot plant. Because of the great modeling uncertainty between the highly nonlinear system of CSTH and imprecise linearized model, an adaptive approximation approach introduced in [122], was utilized to learn the system of CSTH during the initial stage of benchmark simulation. Then the upgraded model can be

used to develop various sensor fault detection schemes without time-consuming efforts on system modeling.

Finally, Chapter 6 presents a centralized drift fault detection scheme for sensors in the monitoring architecture. A rigorous fault detectability and detection time interval analysis for this fault detection scheme is proposed. The set of detectable sensor faults is characterized using off-line information. Then, the upper bound on fault detection interval is derived. Finally, it is proven there exists monotonicity of the detection time interval upper bound concerning the magnitude of sensor faults.

1.4 Publications

The research results shown in this thesis have been presented in several international conferences and published in the international journal. The list is reported in the following.

- Y. Zhou, T. Parisini, M. M. Polycarpou, “*Detection of Drift Sensor Faults in a Class of Nonlinear Uncertain Systems*,” in Proc. of the Conference on Decision and Control, Osaka, 2015, pp. 3169-3174.
- Y. Zhou, F. Boem, C. Fischione, T. Parisini, “*Distributed Fault Detection with Sensor Networks using Pareto-Optimal Dynamic Estimation Method*,” in Proc. of the European Control Conference, Aalborg, 2016
- Y. Zhou, F. Boem, T. Parisini, “*Partition-based Pareto-Optimal State Prediction Method for Interconnected Systems using Sensor Networks*,” American Control Conference, Seattle, 2017, (to appear).
- F. Boem, Y. Zhou, C. Fischione, T. Parisini, “*Distributed Pareto-Optimal State Estimation Using Sensor Networks*,” Automatica, 2017(submitted).

Part I

DISTRIBUTED ESTIMATION ALGORITHMS

Chapter 2

THE ALGORITHM FOR CENTRALIZED OR DECENTRALIZED SYSTEMS

In this chapter, we address the problem of distributed state estimation over a sensor networks in a multi-objective Pareto optimization framework for the monitoring of centralized or decentralized systems. And the proposed approach only assumes the knowledge of the mean and variance of the process and measurement noises without need of any further assumption on their probability distributions. Generally, it is assumed that distributed estimation works according to the following procedure [41, 58]: Firstly, each node communicates measurements and predictions to neighboring nodes. Secondly, each node locally filters the measurements by taking a linear combination of its own and neighboring's measurements and predictions. Finally, the node uses the current filtered measurements to implement a model-based predictor, smoothing the previous prediction error.

However, there are several aspects in this procedure that have not yet been considered. The first important aspect is the number of accessible states. Due to geographic nodes distributions, technological constraints, etc., it is possible that, although the overall network observes the entire state, each node measures only a subset of the monitored variables. However, if the state components are interconnected, then a node could still in principle perform an accurate estimation of the state components it has not directly access to. The second aspect is to consider the mean of estimation error as another performance criterion of the estimators proposed in this chapter. Usually, the performance criterion of the estimators in the literature is essentially based only on the variance of the estimation error (see [19, 94, 136, 154] as examples), which leads to poor performance of the distributed estimation process when biases (mean of the estimation error) are present. Hence, we consider two indicators of the quality of the proposed

estimators: the mean of the estimation error (bias) and the variance. A third vital aspect is related to the fact that in the literature the filtering step and the prediction step are designed independently, e.g., [4, 21] for the sake of tractability and ease of implementation. This separation may lead to suboptimal solutions. Instead, in this chapter the filtering weights and the prediction parameters are jointly optimized by each sensor node at each time step in a multi-objective Pareto optimization framework. The last aspect is the instantaneous performance in the distributed estimation algorithm. Although any well-designed distributed estimator can asymptotically perform well, in the transient mean and variance of the estimation error may take on unacceptably large values. The proposed multi-objective optimization problem is solved by satisfying asymptotic convergence conditions at each time step to minimize the mean and the variance of the estimation error; therefore, the good instantaneous performance is guaranteed at each sensor node. This is in contrast to the well-known Kalman-based approaches (see [104]), where only the asymptotic convergence is achieved.

This chapter is organized as follows. In Section 2.1, we introduce the problem formulation. We describe the distributed dynamic estimation method in Section 2.2. We derive some convergence conditions in Subsection 2.2.3 and we then formulate in Section 2.3 the multi-objective optimization problem to select the filtering weights and the prediction parameters. The optimal solution is derived in Section 2.3.2. Finally, some simulation results are provided in Section 2.5 and concluding remarks in Section 2.6.

2.1 Problem Formulation

We aim at computing the one-step ahead prediction $\hat{x}(t+1)$ of the state $x(t)$ of a linear stochastic system described as

$$x(t+1) = Ax(t) + w(t), \quad (2.1)$$

where t is the discrete time, $x \in \mathbb{R}^m$ denotes the state vector and $w \in \mathbb{R}^m$ is modelled as

$$w(t) = \bar{w}(t) + \tilde{w}(t),$$

with $\bar{w}(t)$ denoting a known time-varying bias possibly including some known nonlinearities and process disturbances, while $\tilde{w}(t)$ models uncertainties and process noises.

Assumption 1 *We assume that $\tilde{w}(t)$ is a zero-mean process with covariance matrix $\Sigma_w(t)$.*

The state prediction $\hat{x}(t+1)$ is computed in a distributed way by a *sensor network*, made of n nodes, that monitors system (2.1) by taking measurements at each time-instant t . More specifically, each sensor knows the model (2.1) of the monitored system and may measure directly some, possibly not all, state components, so that, for each sensor node i , with $i = 1, \dots, n$, we have

$$y_i(t) = C_i x(t) + v_i(t), \quad (2.2)$$

where $y_i \in \mathbb{R}^{p_i}$, with $p_i \leq m$, denotes the measurements vector taken by sensor i , $v_i \in \mathbb{R}^{p_i}$ is a zero-mean measurement noise, with Σ_{v_i} as covariance matrix, and $C_i \in \mathbb{R}^{p_i \times m}$ is the output matrix, each row having a single element equal to 1 in correspondence with the measured state component, and 0 otherwise. Since each sensor may measure directly one or more components of the state vector, the matrix C_i satisfies the following property:

$$C_i \mathbf{1}_m = \mathbf{1}_{p_i}$$

and $C_i^\top \mathbf{1}_{p_i}$ is a vector whose components take on values in the set $\{0, 1\}$. The following further assumption is needed:

Assumption 2 *The measurement noise v_i and process disturbances w are not correlated, for each $i = 1, \dots, n$.*

Assumption 3 *We assume that graph $\mathcal{G}$ is connected.*

Each node of the network exchanges measurements and predictions only with neighboring nodes. The associated communication network is modeled as an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where $\mathcal{S}$ is the set of the nodes and $\mathcal{E}$ is the set of the edges connecting communicating nodes. We denote $\mathcal{N}_i = \{j \in \mathcal{S} : (j, i) \in \mathcal{E}\} \cup \{i\}$ the set of neighbors of node $i \in \mathcal{S}$ plus the node itself. For well-posedness, the following assumption is clearly needed.

Considering the entire sensor network, the global measurement output equation becomes

$$y(t) = Cx(t) + v(t),$$

with $y, v \in \mathbb{R}^{p_E}$, $C \in \mathbb{R}^{p_E \times m}$, $p_E = \sum_{j \in 1, \dots, n} p_j$, where vectors and matrices collect all the local vectors y_j and v_j and local matrices C_j of the sensor network, with $j = 1, \dots, n$. Hence, we have $C\mathbf{1}_m = \mathbf{1}_{p_E}$, which means each measurement refers to a state component.

Assumption 4 *We assume that C satisfies the following condition*

$$C^\top \mathbf{1}_{p_E} \geq \mathbf{1}_m.$$

This means that each state component is directly measured by one or more nodes, so that the states are completely measurable by the sensors in the sensor network. In the subsequent sections, the distributed state prediction algorithm is described and analyzed.

2.2 Distributed State Dynamic Estimation

In the proposed state estimator, each node i of the sensor network implements a two steps dynamic estimator: a filtering-merging step and a prediction step. First, by communicating with neighboring nodes, the estimator at the i -th node filters the measurement noise in a consensus-like fashion by computing a linear combination of its own and neighboring available measurements and predictions:

$$\bar{x}_i(t) = \sum_{j \in \mathcal{N}_i} k_{i,j}(t) \hat{x}_j(t) + \sum_{j \in \mathcal{N}_i} h_{i,j}(t) C_j^\top y_j(t), \quad (2.3)$$

where $k_{i,j}(t)$ and $h_{i,j}(t) \in \mathbb{R}^{m \times m}$ are time-varying filter weights diagonal matrices that we intend to design. The term $C_j^\top$ guarantees that the available measurements are used for each node. By communicating with neighbors, each node shares its available information and spreads it through the network. The objective of this first step is for each node to reduce its own measurement uncertainty, *without the use of centralized coordination*. Sensors are allowed to communicate only once between subsequent time-instants.

In qualitative terms, in case at the i -th node no measurement is available for a certain state component (neither directly, nor indirectly through the neighbors), more weight should be assigned to the predictions. In fact, these could be more accurate because they use the process model and could take advantage of the measurements of other nodes which are not immediate neighbors of node i .

After the filtering-merging step, each node implements a classical model-based prediction, using filtered measurements (obtained from the first step), instead of the observed measurements. The one-step-ahead prediction is computed as

$$\hat{x}_i(t+1) = A\bar{x}_i(t) + \bar{w}(t) + \lambda'_i(t)(\hat{x}_i(t) - \bar{x}_i(t)), \quad (2.4)$$

where $\lambda'_i(t) \in \mathbb{R}^{m \times m}$ is a matrix collecting suitable time-varying filter parameters that will be designed later on in the chapter. The term $\lambda'_i(t)(\hat{x}_i(t) - \bar{x}_i(t))$ represents a local correction of the previous prediction error, taking into account the system dynamics. The estimates are initialized with $\bar{x}_i(0) = C_i^\top y_i(0)$ and $\hat{x}_i(0) = \bar{x}_i(0)$.

We can rewrite (2.3) and (2.4) as:

$$\begin{aligned} \bar{x}_i(t) &= \kappa_i(t)\hat{x}_i^{\text{reg}}(t) + \eta_i(t)C_{iE}^\top y_i^{\text{reg}}(t) \\ \hat{x}_i(t+1) &= A\bar{x}_i(t) + \bar{w}(t) + A\lambda_i(t)(\hat{x}_i(t) - \bar{x}_i(t)), \end{aligned} \quad (2.5)$$

where $\hat{x}_i^{\text{reg}}$ and y_i^{reg} are two column vectors collecting the prediction vectors and the measurements vectors respectively available at node i , ordered according to their indexes: $\hat{x}_i^{\text{reg}} = (\hat{x}_{i_1}^\top, \dots, \hat{x}_{i_{N_i}}^\top)^\top$, $i_1 < \dots < i_{N_i}$, and $y_i^{\text{reg}} = (y_{i_1}^\top, \dots, y_{i_{N_i}}^\top)^\top$, $i_1 < \dots < i_{N_i}$, with N_i being the cardinality of the set $\mathcal{N}_i$. Moreover, $\kappa_i(t)$ and $\eta_i(t)$ are the time varying row block matrices $\in \mathbb{R}^{m \times N_i m}$ collecting matrices $k_{i,j}$ and $h_{i,j}$ respectively; $\lambda_i(t)$ is a diagonal matrix collecting the local components of matrix $\lambda(t)$. C_{iE} is a block matrix whose diagonal elements are matrices C_j , with $j \in \mathcal{N}_i$.

The main goal is to devise a design principle of the time-varying $k_{i,j}(t)$, $h_{i,j}(t)$, and $\lambda_i(t)$, for each $i = 1, \dots, n$, $j \in \mathcal{N}_i$, so to minimize mean and variance of the estimation error in each sensor node at each time step.

2.2.1 Local Filtering and Prediction Errors

Let us now define the following estimation errors: the local filtering error

$$\bar{e}_i(t) = \bar{x}_i(t) - x(t)$$

and the local prediction error

$$\hat{E}_i(t) = \hat{x}_i(t) - x(t)$$

We introduce the following assumption:

Assumption 5 Suppose that at each time t

$$(\kappa_i(t) + \eta_i(t)C_{iE}^\top C_{iE})\mathbf{1}_{N_i m} = \mathbf{1}_m.$$

Then the following expression hold:

$$\kappa_i(t)x_i^E(t) + \eta_i(t)C_{iE}^\top C_{iE}x_i^E(t) = x_i^E(t)$$

being x_i^E a column vector repeating N_i times the state vector x . We use this expression to derive the local filtering error and the local prediction error:

$$\bar{e}_i(t) = \kappa_i(t)\hat{e}_i(t) + \eta_i(t)C_{iE}^\top v_{\epsilon_i}(t), \quad (2.6)$$

$$\hat{E}_i(t+1) = A(I - \lambda_i(t))\kappa_i(t)\hat{e}_i(t) + A\lambda_i(t)\hat{E}_i(t) + A(I - \lambda_i(t))\eta_i(t)C_{iE}^\top v_{\epsilon_i}(t) - \tilde{w}(t), \quad (2.7)$$

where $\hat{e}_i$ collects the prediction error vectors available at node i and v_{ϵ_i} collects the measurement noise vectors related to the measurements available at node i , both ordered following their indexes.

2.2.2 Global Filtering and Prediction Errors

To facilitate the subsequent analysis, the distributed estimation formulation can be expressed in a compact vector form that includes all the nodes in the sensor network:

$$\begin{aligned} \bar{x}(t) &= K(t)\hat{x}(t) + H(t)C_E^\top y(t) \\ \hat{x}(t+1) &= A_E\bar{x}(t) + \bar{w}_E(t) + \lambda'(t)(\hat{x}(t) - \bar{x}(t)), \end{aligned} \quad (2.8)$$

where $\bar{x}, \hat{x} \in \mathbb{R}^{mn \times 1}$ and $y \in \mathbb{R}^{p_E \times 1}$ are column vectors collecting the local vectors $\bar{x}_i$, $\hat{x}_i$ and y_i , respectively, with $i = 1, \dots, n$; A_E is a diagonal block matrix, with each block on the diagonal equal to A , and $\bar{w}_E$ is a column vector of appropriate dimension, where the process disturbance vector $\bar{w}$ is repeated n times. $K(t)$ and $H(t)$ are block matrices, where each (i, j) -th block, with $i = 1, \dots, n$ and $j = 1, \dots, n$, is a diagonal matrix. The (i, j) -th block is a null matrix if i and j are not neighboring nodes, whereas it collects the coefficients with which the i -th nodes weights measurements or estimates components developed by the j -th node, if they are neighbors. As an example, $k_{i,j}^l$ is the diagonal element of the l -th row of the (i, j) -th block of matrix K and it represents

how the i -th node weights the l -th estimate component received from the j -th node. C_E is a block matrix having the matrices C_j , with $j = 1, \dots, n$, on the diagonal. Finally, λ' is a diagonal block matrix, collecting the matrices λ'_i , $i = 1, \dots, n$, on the diagonal.

Let us introduce the following global estimation errors: the filtering error

$$\bar{e}(t) = \bar{x}(t) - x_E(t)$$

and the prediction error

$$\hat{E}(t) = \hat{x}(t) - x_E(t),$$

with x_E being a column vector of appropriate length, where the process state vector x is repeated n times. Given Assumption 5, it holds

$$(K(t) + H(t)C_E^\top C_E)\mathbf{1}_{mn} = \mathbf{1}_{mn}.$$

and

$$H(t)C_E^\top C_E x_E(t) + K(t)x_E(t) = x_E(t).$$

Therefore, the global filtering error can be computed as

$$\begin{aligned} \bar{e}(t) &= K(t)\hat{x}(t) + H(t)C_E^\top y(t) - x_E(t) \\ &= K(t)\hat{E}(t) + H(t)C_E^\top v(t). \end{aligned} \tag{2.9}$$

For the sake of simplicity, we express the prediction parameter as $\lambda'(t) = A_E \lambda(t)$, with $\lambda(t)$ being a diagonal matrix. Consequently, the global prediction error is:

$$\begin{aligned} \hat{E}(t+1) &= A_E \bar{x}(t) + \bar{w}_E(t) + \lambda'(t)(\hat{x}(t) - \bar{x}(t)) - A_E x_E(t) - \bar{w}_E(t) - \tilde{w}_E(t) \\ &= A_E [(I - \lambda(t))K(t) + \lambda(t)] \hat{E}(t) + A_E (I - \lambda(t))H(t)C_E^\top v(t) - \tilde{w}_E(t). \end{aligned} \tag{2.10}$$

Let us compute mean and variance of the global filtering and prediction errors. The expected value of the errors are

$$\mathbb{E}\bar{e}(t) = K(t)\mathbb{E}\hat{E}(t), \tag{2.11}$$

$$\mathbb{E}\hat{E}(t+1) = A_E [(I - \lambda(t))K(t) + \lambda(t)] \mathbb{E}\hat{E}(t). \tag{2.12}$$

The covariance matrices can be computed as

$$\mathbb{E}[(\bar{e}(t) - \mathbb{E}\bar{e}(t))(\bar{e}(t) - \mathbb{E}\bar{e}(t))^\top] = K(t)\Gamma_{\hat{E}}(t)K(t)^\top + H(t)C_E^\top \Sigma_v(t)C_E H(t)^\top, \tag{2.13}$$

where

$$\Gamma_{\hat{E}}(t) = \mathbb{E} \left[(\hat{E}(t) - \mathbb{E}\hat{E}(t))(\hat{E}(t) - \mathbb{E}\hat{E}(t))^\top \right]$$

and

$$\Sigma_v(t) = \mathbb{E} \left[(v(t) - \mathbb{E}v(t))(v(t) - \mathbb{E}v(t))^\top \right],$$

noting that $\text{Cov}(\hat{E}(t), v(t)) = 0$. We can thus compute the covariance matrix $\Gamma_{\hat{E}}(t+1)$ as

$$\begin{aligned} & \mathbb{E} \left[(\hat{E}(t+1) - \mathbb{E}\hat{E}(t+1))(\hat{E}(t+1) - \mathbb{E}\hat{E}(t+1))^\top \right] \\ &= W_1(t)\Gamma_{\hat{E}}(t)W_1(t)^\top + W_2(t)\Sigma_v(t)W_2(t)^\top + \Sigma_{w_E}(t), \end{aligned} \quad (2.14)$$

with

$$W_1(t) = A_E [(I - \lambda(t))K(t) + \lambda(t)],$$

$$W_2(t) = A_E(I - \lambda(t))H(t)C_E^\top,$$

and

$$\Sigma_{w_E}(t) = \mathbb{E} \left[(w_E(t) - \mathbb{E}w_E(t))(w_E(t) - \mathbb{E}w_E(t))^\top \right],$$

being $\text{Cov}(\hat{E}(t), v(t)) = 0$, $\text{Cov}(\hat{E}(t), \tilde{w}_E(t)) = 0$ and $\text{Cov}(v(t), \tilde{w}_E(t)) = 0$, owing to Assumption 2.

2.2.3 Stability of the Prediction Error

In this subsection, we show that the mean of the global prediction error given by (2.12) converges to zero under some local conditions on the time-varying filtering weights κ_i , η_i and prediction parameter λ_i (which are the local parts of matrices $K(t)$, $H(t)$ and $\lambda(t)$). Moreover, the convergence property is guaranteed in all network nodes, only based on local conditions. Given two vectors v and v_1 , we denote $v < v_1$ or $v > v_1$ the component-by-component inequality. We use a similar notation for matrices. For the sake of notation simplicity, we omit the dependence on t of the matrices.

Proposition 2.2.1 *Consider the global prediction error mean given by (2.12). The following local conditions are sufficient to guarantee the asymptotic stability of the prediction error. For each l -th row of each i -th block row of the global matrices K , H and λ , with $i = 1, \dots, n$ and $l = 1, \dots, m$:*

◦ if $1/\|A\|_\infty > 1$, then

$$\sum_{j=1}^n |k_{i,j}^l| < \frac{1}{\|A\|_\infty},$$

$$\frac{-\frac{1}{\|A\|_\infty} + \sum_{j=1}^n |k_{i,j}^l|}{1 + \sum_{j=1}^n |k_{i,j}^l|} < \lambda_i^l < \frac{\frac{1}{\|A\|_\infty} + \sum_{j=1}^n |k_{i,j}^l|}{1 + \sum_{j=1}^n |k_{i,j}^l|};$$
(2.15)

◦ if $1/\|A\|_\infty = 1$, then

$$\sum_{j=1}^n |k_{i,j}^l| < \frac{1}{\|A\|_\infty},$$

$$\frac{-\frac{1}{\|A\|_\infty} + \sum_{j=1}^n |k_{i,j}^l|}{1 + \sum_{j=1}^n |k_{i,j}^l|} \leq \lambda_i^l < 1;$$
(2.16)

◦ if $1/\|A\|_\infty < 1$, then

$$\sum_{j=1}^n |k_{i,j}^l| < \frac{1}{\|A\|_\infty},$$

$$\frac{-\frac{1}{\|A\|_\infty} + \sum_{j=1}^n |k_{i,j}^l|}{1 + \sum_{j=1}^n |k_{i,j}^l|} < \lambda_i^l < 1 - \frac{1 - \frac{1}{\|A\|_\infty}}{1 - \sum_{j=1}^n |k_{i,j}^l|}.$$
(2.17)

Proof: Eq. (2.12) represents the dynamics of a linear time-varying system. A necessary and sufficient condition for global asymptotic stability is given in [10]: asymptotic stability is ensured if there exists a finite $k > 0$ such that

$$\left\| \prod_{t=1}^k A_E [(I - \lambda(t))K(t) + \lambda(t)] \right\|_\infty < 1.$$
(2.18)

This condition is implied by the satisfaction, for each t , of the condition

$$\|A_E [(I - \lambda(t))K(t) + \lambda(t)]\|_\infty < 1.$$

Recall that $\lambda(t)$ and $I - \lambda(t)$ are two diagonals matrices; by definition $K(t)$ is a block matrix, where each block is a diagonal matrix; therefore also $(I - \lambda(t))K(t) + \lambda(t)$ is a block matrix with all diagonal blocks. Thanks to the submultiplicative property of the

norm, the convergence condition is implied by

$$\|(I - \lambda)K + \lambda\|_\infty < \frac{1}{\|A_E\|_\infty}.$$

This is satisfied if, for each l -th row of each i -th block row of matrix $(I - \lambda(t))K(t) + \lambda(t)$, with $i = 1, \dots, n$ and $l = 1, \dots, m$, we have

$$|1 - \lambda_i^l| \sum_{j=1}^n |k_{i,j}^l| + |\lambda_i^l| < \frac{1}{\|A\|_\infty}, \quad (2.19)$$

since $\|A_E\|_\infty = \|A\|_\infty$. The analysis of (2.19) in all possible different scenarios concerning the value of λ_i^l immediately leads to prove the statement of the proposition. $\square$

Note that Proposition 2.2.1 gives local convergence conditions on the filtering weights and the prediction parameters, meaning that these weights and parameters can be computed at each sensor node and, if the conditions of the proposition are satisfied, we have global convergence to zero of the mean prediction error.

Remark 2.2.1 The global convergence to zero of the mean prediction error implies that a consensus among the local state estimates is asymptotically achieved. This interesting aspect relates the proposed two-steps estimation method to the one proposed in [135] (see also [4, 21]).

2.3 The Optimization Problem

The goal of the proposed distributed dynamic estimation method is to predict the state minimizing the mean and variance of the prediction error at each time instant. To do that, we propose that each sensor at each time step computes the optimal time-varying weights and parameters by solving a multi-objective optimization problem, where the first objective is the squared prediction error mean, and the second objective is the variance of the prediction error.

Now, the expressions of the mean and the variance for the local filtering and prediction errors can be computed. The mean values are given by

$$\mathbb{E}\bar{e}_i(t) = \kappa_i(t)\mathbb{E}\hat{e}_i(t),$$

$$\begin{aligned}\mathbb{E}\hat{E}_i(t+1) &= A(I - \lambda_i(t))\kappa_i(t)\mathbb{E}\hat{e}_i(t) + A\lambda_i(t)\mathbb{E}\hat{E}_i(t) \\ &= A[(I - \lambda_i(t))\kappa_i(t) + \lambda_i^0(t)]\mathbb{E}\hat{e}_i(t),\end{aligned}\tag{2.20}$$

being $\lambda_i^0(t)$ a $m \times mN_i$ a block matrix having the sub-matrix $\lambda_i(t)$ in the block position corresponding to the i -th index in the neighboring set $\mathcal{N}_i$, and all the other blocks components equal to 0. An algebra gives the expression of the variance as

$$\begin{aligned}&\mathbb{E}[(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))^\top] \\ &= W_{1i}(t)\Gamma_{\hat{e}_i}(t)W_{1i}(t)^\top + W_{2i}(t)\Sigma_{v_{\hat{e}_i}}W_{2i}(t)^\top + \Sigma_w(t),\end{aligned}\tag{2.21}$$

where

$$W_{1i}(t) = A[(I - \lambda_i(t))\kappa_i(t) + \lambda_i^0(t)],\tag{2.22}$$

$$W_{2i}(t) = A(I - \lambda_i(t))\eta_i(t)C_{iE}^\top,\tag{2.23}$$

$$\Gamma_{\hat{e}_i}(t) = \mathbb{E}[(\hat{e}_i(t) - \mathbb{E}\hat{e}_i(t))(\hat{e}_i(t) - \mathbb{E}\hat{e}_i(t))^\top]\tag{2.24}$$

and $\Sigma_{v_{\hat{e}_i}}$ is the measurement noise covariance matrix, including correlations between neighboring sensors. given in (2.20) and (2.21)

Since these two objectives are convex in the decision variables (namely the filtering weights and the prediction parameter), then we can reduce the multi-objective optimization problem [18] to a scalar case and pose the following Pareto optimization problem:

$$\min_{\kappa_i, \eta_i, \lambda_i} \text{tr} [\rho_i B_i^2 + (1 - \rho_i)V_i]\tag{2.25a}$$

$$\text{s.t. } (\kappa_i(t) + \eta_i(t)C_{iE}^\top C_{iE})\mathbf{1}_{m_i} = \mathbf{1}_m,\tag{2.25b}$$

$$\text{convergence conditions Eq.(2.15)} \vee (2.16) \vee (2.17),\tag{2.25c}$$

where $m_i = mN_i$, $0 \leq \rho_i \leq 1$ is the Pareto parameter, $B_i = \mathbb{E}\hat{E}_i(t+1)$ is the prediction error mean given in (2.20), $V_i = \mathbb{E}[(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))^\top]$ is the variance of the prediction error given in (2.21). The first constraint in problem (2.25) is the local condition given by Assumption 5; the other constraints are the convergence conditions derived in Section 2.2.3, depending on the norm of matrix A . According to the derived expressions for the mean and variance of the prediction error,

the cost function of problem (2.25) is

$$\begin{aligned}
& \text{tr} [\rho_i B_i^2 + (1 - \rho_i) V_i] \\
&= \text{tr} [A(I - \lambda_i(t)) \kappa_i(t) [\rho \mathbb{E} \hat{\epsilon}_i(t) \mathbb{E} \hat{\epsilon}_i^\top(t) + (1 - \rho_i) \Gamma_{\hat{\epsilon}_i}(t)] \kappa_i(t)^\top (I - \lambda_i(t))^\top A^\top \\
&\quad + (1 - \rho_i) \Sigma_w(t) + A \lambda_i(t) [\rho \mathbb{E} \hat{E}_i(t) \mathbb{E} \hat{E}_i^\top(t) + (1 - \rho_i) \Gamma_{\hat{E}_i}(t)] \lambda_i(t)^\top A^\top \\
&\quad + A(I - \lambda_i(t)) \eta_i(t) C_{iE}^\top [(1 - \rho_i) \Sigma_{v_{\hat{\epsilon}_i}}] C_{iE} \eta_i(t)^\top (I - \lambda_i(t))^\top A^\top],
\end{aligned} \tag{2.26}$$

which is obtained by using the definitions of W_{1i} and W_{2i} given in (2.22) and (2.23). To highlight the dependencies on the decision variables $\kappa_i(t)$, $\eta_i(t)$ and $\lambda_i(t)$, we rewrite the Pareto optimization problem (2.25) by substituting the following expressions that are data of the problem or can be computed on-line using samples:

$$\begin{aligned}
M_i(t, \rho_i) &= \rho_i \mathbb{E} \hat{\epsilon}_i(t) \mathbb{E} \hat{\epsilon}_i^\top(t) + (1 - \rho_i) \Gamma_{\hat{\epsilon}_i}(t), \\
M_i(t, \rho_i)^{\text{loc}} &= \rho_i \mathbb{E} \hat{E}_i(t) \mathbb{E} \hat{E}_i^\top(t) + (1 - \rho_i) \Gamma_{\hat{E}_i}(t), \\
S_i(t, \rho_i) &= (1 - \rho_i) \Sigma_{v_{\hat{\epsilon}_i}}(t).
\end{aligned}$$

We rewrite the objective function of (2.25) as

$$\begin{aligned}
\min_{\kappa_i(t), \eta_i(t), \lambda_i(t)} \quad & \text{tr} [A(I - \lambda_i(t)) \kappa_i(t) M_i(t, \rho_i) \kappa_i(t)^\top (I - \lambda_i(t))^\top A^\top + (1 - \rho_i) \Sigma_w(t) \\
& + A(I - \lambda_i(t)) \eta_i(t) C_{iE}^\top S_i(t, \rho_i) C_{iE} \eta_i(t)^\top (I - \lambda_i(t))^\top A^\top \\
& + A \lambda_i(t) M_i(t, \rho_i)^{\text{loc}} \lambda_i(t)^\top A^\top] \tag{2.27}
\end{aligned}$$

Note that the second convergence conditions in (2.15), (2.16), (2.17) are non convex expressions depending on two sets of decision variables, which further complicates the derivation of the solution especially for what concerns the use of necessary optimality conditions. We now analyze optimization problem (2.25) in order to obtain a convex approximation problem that can be solved in optimal way.

2.3.1 Approximation Approach to the Optimization

In this section, we propose a convex approximation of problem (2.25). We start by first briefly analyzing the convergence conditions (2.15), (2.16), (2.17). The first condition is common to all the three cases. Since the absolute value would make the

problem more difficult to solve, we use the more conservative conditions

$$\begin{cases} \kappa_i^\top(t) \mathbf{1}_m \geq \mathbf{0}_{m_i} \\ \kappa_i(t) \mathbf{1}_{m_i} < \frac{1}{\|A\|_\infty} \mathbf{1}_m \end{cases} \quad (2.28)$$

instead of

$$\sum_{j=1}^n |k_{i,j}^l| < 1/\|A\|_\infty, \forall l.$$

In fact, the block matrix κ_i is a row block matrix collecting the diagonal matrices $K_{i,j}$, $j = 1, \dots, n$, having on the diagonal the values $k_{i,j}^l$ for each l -th row, if j is a neighbor of i . By forcing κ_i to have only positive values (first condition), it is possible to remove the absolute value from the convergence condition and we then obtain the second condition.

Moreover, as regards the expressions in (2.15), (2.16), (2.17), we note that if the first condition $\sum_{j=1}^n |k_{i,j}^l| < 1/\|A\|_\infty$ holds, then

$$\frac{-\frac{1}{\|A\|_\infty} + \sum_{j=1}^n |k_{i,j}^l|}{1 + \sum_{j=1}^n |k_{i,j}^l|} < 0;$$

if, in addition, $1/\|A\|_\infty > 1$, then

$$\frac{\frac{1}{\|A\|_\infty} + \sum_{j=1}^n |k_{i,j}^l|}{1 + \sum_{j=1}^n |k_{i,j}^l|} > 1,$$

while if $1/\|A\|_\infty < 1$, then

$$0 < 1 - \frac{1 - \frac{1}{\|A\|_\infty}}{1 - \sum_{j=1}^n |k_{i,j}^l|} < 1.$$

We can then use the above relations to approximate the convergence conditions for the definition of convex constraints. We obtain the following sufficient convergence conditions, implying conditions (2.15), (2.16), (2.17): condition (2.28) and

◦ if $1/\|A\|_\infty > 1$,

$$0 \leq \lambda_i^l \leq 1; \quad (2.29)$$

◦ if $1/\|A\|_\infty = 1$,

$$0 \leq \lambda_i^l < 1; \quad (2.30)$$

◦ if $1/\|A\|_\infty < 1$,

$$0 \leq \lambda_i^l < 1, \quad (2.31)$$

for every $l = 1, \dots, m$. Using the element-wise inequality, (2.29) is equivalent to

$$0 \leq \lambda_i \leq I, \quad (2.32)$$

and (2.30), (2.31) to

$$0 \leq \lambda_i < I. \quad (2.33)$$

We then formulate a convex approximation of problem (2.25):

$$\begin{aligned} \min_{\kappa_i(t), \eta_i(t), \lambda_i(t)} \quad & \text{tr} [A(I - \lambda_i(t))\kappa_i(t)M_i(t, \rho_i)\kappa_i(t)^\top (I - \lambda_i(t))^\top A^\top + (1 - \rho_i)\Sigma_w(t) \\ & + A(I - \lambda_i(t))\eta_i(t)C_{iE}^\top S_i(t, \rho_i)C_{iE}\eta_i(t)^\top (I - \lambda_i(t))^\top A^\top \\ & + A\lambda_i(t)M_i(t, \rho_i)^{\text{loc}}\lambda_i(t)^\top A^\top] \end{aligned} \quad (2.34a)$$

$$\text{s.t. } (\kappa_i(t) + \eta_i(t)C_{iE}^\top C_{iE})\mathbf{1}_{m_i} = \mathbf{1}_m, \quad (2.34b)$$

$$\kappa_i^\top(t)\mathbf{1}_m \geq \mathbf{0}_{m_i} \quad (2.34c)$$

$$\kappa_i(t)\mathbf{1}_{m_i} < \frac{1}{\|A\|_\infty}\mathbf{1}_m \quad (2.34d)$$

$$\lambda_i(t)\mathbf{1}_m + \varphi\mathbf{1}_m \leq \mathbf{1}_m, \quad (2.34e)$$

$$\lambda_i(t)\mathbf{1}_m \geq \mathbf{0}_m, \quad (2.34f)$$

being φ a small positive constant, where the last two constraints are equivalent to (2.32), if $\varphi = 0$, or (2.33) if $\varphi \neq 0$, depending on matrix A norm, since $\lambda_i(t)$ is a diagonal matrix.

Remark 2.3.1 Problem (2.34) and (2.25) share the same cost function and the first constraint. The error that we commit on the variable λ_i with this approximation depends on the norm $\|A\|_\infty$. For example, in the case $1/\|A\|_\infty > 1$, since $0 \leq \sum_{j=1}^n |k_{i,j}^l| < 1/\|A\|_\infty$, the second condition in (2.15) can vary from

$$\frac{-1}{\|A\|_\infty} < \lambda_i^l < \frac{1}{\|A\|_\infty}$$

if $\sum_{j=1}^n |k_{i,j}^l| = 0$, to

$$\frac{-\epsilon}{1 + \frac{1}{\|A\|_\infty} - \epsilon} < \lambda_i^l < \frac{\frac{2}{\|A\|_\infty} - \epsilon}{1 + \frac{1}{\|A\|_\infty} - \epsilon}$$

if $\sum_{j=1}^n |k_{i,j}^l| = 1/\|A\|_\infty - \epsilon$, with ϵ arbitrarily small, where

$$\frac{\frac{2}{\|A\|_\infty} - \epsilon}{1 + \frac{1}{\|A\|_\infty} - \epsilon} > 1.$$

The choice of using the constraint $0 \leq \lambda_i \leq I$, instead of the convergence conditions (2.15), may affect each component of λ_i with a maximum error of $1/\|A\|_\infty$, in the worst case. In fact, as it is possible to see by substituting the inferior and superior values that $\sum_{j=1}^n |k_{i,j}^l|$ can assume in conditions (2.15), the largest interval where λ_i can be constrained based on Eq.(2.15) is $-1/\|A\|_\infty < \lambda_i^l < 1/\|A\|_\infty$.

Note that problem (2.25) and problem (2.34) have the same objective function, but problem (2.34) is constrained by more conservative conditions. This gives an optimization problem which is tractable. In fact, problem (2.34) is convex since the objective function has a quadratic form and M_i , M_i^{loc} and S_i are positive definite matrices, and the constraints are convex. Coherently, we can use Lagrangian duality to solve problem (2.34). Let introduce the dual variables ξ_{i1} , ξ_{i2} , ξ_{i4} and ν_i , which are $m \times 1$ vectors, and the $m_i \times 1$ vector ξ_{i3} . To simplify the notation, we omit the dependence on time in the following lemma and proof. Then we have the following result:

Lemma 2.3.1 *Consider optimization problem (2.34). Let ϵ_1 be a positive constant. Then, the optimal values of the primal (κ_i , η_i and λ_i) and dual (ξ_{i1} , ξ_{i2} , ξ_{i3} , ξ_{i4} and ν_i) variables satisfy the following equations and inequalities:*

$$(\kappa_i + \eta_i C_{iE}^\top C_{iE}) \mathbf{1}_{m_i} - \mathbf{1}_m = 0_m, \quad (2.35)$$

$$\lambda_i \mathbf{1}_m + \varphi \mathbf{1}_m - \mathbf{1}_m \leq 0_m, \quad (2.36)$$

$$-\lambda_i \mathbf{1}_m \leq 0_m, \quad (2.37)$$

$$\xi_{i1}^\top (\lambda_i \mathbf{1}_m + \varphi \mathbf{1}_m - \mathbf{1}_m) = 0 \quad \xi_{i1} \geq 0, \quad (2.38)$$

$$\xi_{i2}^\top(-\lambda_i \mathbf{1}_m) = 0 \quad \xi_{i2} \geq 0, \quad (2.39)$$

$$-\kappa_i \mathbf{1}_m \leq 0_m, \quad (2.40)$$

$$\kappa_i \mathbf{1}_{m_i} - \frac{1}{\|A\|_\infty} \mathbf{1}_m + \epsilon_1 \leq 0_m, \quad (2.41)$$

$$\xi_{i3}^\top(-\kappa_i \mathbf{1}_m) = 0 \quad \xi_{i3} \geq 0, \quad (2.42)$$

$$\xi_{i4}^\top(\kappa_i \mathbf{1}_{m_i} - \frac{1}{\|A\|_\infty} \mathbf{1}_m + \epsilon_1) = 0 \quad \xi_{i4} \geq 0, \quad (2.43)$$

$$\begin{aligned} & 2[M_i(\rho_i)\kappa_i^\top(I - \lambda_i)^\top A^\top A(I - \lambda_i)] \circ (\mathbf{1} \otimes I) \\ & + [\mathbf{1}_{m_i}\nu_i^\top] \circ (\mathbf{1} \otimes I) + [\mathbf{1}_{m_i}\xi_{i4}^\top] \circ (\mathbf{1} \otimes I) - [\xi_{i3}\mathbf{1}_m^\top] \circ (\mathbf{1} \otimes I) = 0, \end{aligned} \quad (2.44)$$

$$\begin{aligned} & 2[C_{iE}^\top S_i(\rho_i)C_{iE}\eta_i^\top(I - \lambda_i)^\top A^\top A(I - \lambda_i)] \circ (\mathbf{1} \otimes I) \\ & + [C_{iE}^\top C_{iE}\mathbf{1}_{m_i}\nu_i^\top] \circ (\mathbf{1} \otimes I) = 0, \end{aligned} \quad (2.45)$$

$$\begin{aligned} & 2[[\kappa_i M_i(\rho_i)\kappa_i^\top + \eta_i C_{iE}^\top S_i(\rho_i)C_{iE}\eta_i^\top + M_i^{\text{loc}}(\rho_i)]\lambda_i A^\top A \\ & - [\kappa_i M_i(\rho_i)\kappa_i^\top + \eta_i C_{iE}^\top S_i(\rho_i)C_{iE}\eta_i^\top]A^\top A] \circ I \\ & + [\mathbf{1}_m \xi_{i1}^\top] \circ I - [\mathbf{1}_m \xi_{i2}^\top] \circ I = 0 \end{aligned} \quad (2.46)$$

Proof: Problem (2.34) is convex. Therefore the KKT conditions are both necessary and sufficient for optimality. In (2.35)–(2.37) we derive the canonical form of the constraints. Since κ_i , η_i and λ_i are composed by diagonal blocks, (2.44)–(2.46) are obtained by using matrix derivatives for diagonal and symmetric matrices [95, 109]: given matrices A , $M \succeq \mathbf{0}$ and diagonal matrix X , then

$$\frac{d \operatorname{tr}(AXMX^T A^T)}{dX} = 2(MXA^T A) \circ I. \quad (2.47)$$

As an extension to this result, given block-diagonal matrix κ , positive semidefinite matrix M and matrix A , then

$$\frac{d \operatorname{tr}(A\kappa M \kappa^T A^T)}{d\kappa} = 2(M\kappa^T A^T A) \circ (\mathbf{1} \otimes I). \quad (2.48)$$

This concludes the proof. $\square$

In the following we analyze the derived KKT conditions to obtain the optimal weight values.

2.3.2 The Optimal Weights for the Approximated Problem

In the previous section, we formulated the optimization problem (2.25) to minimize mean and variance of the local prediction error and we obtained some optimality conditions for the equivalent problem (2.34) with more restrictive constraints than problem (2.25). Here we analyze the KKT conditions to derive the optimal values for the decision variables $\kappa_i(t)$, $\eta_i(t)$ and $\lambda_i(t)$. Given a vector V , we denote $\text{diag}(V)$ the diagonal matrix whose diagonal entries are the elements of V , and $V = \text{diag}^{-1}(\text{diag}(V))$. We need the following preliminary result:

Lemma 2.3.2 *Consider square matrices M , A and X with dimension $n \times n$, where A is symmetric and X is diagonal. Then*

$$\text{diag}^{-1}(MXA) = (M \circ A)\text{diag}^{-1}(X),$$

Proof: We have

$$MX = \begin{bmatrix} M_{11}X_1 & M_{12}X_2 & \cdots & M_{1n}X_n \\ M_{21}X_1 & M_{22}X_2 & \cdots & M_{2n}X_n \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1}X_1 & M_{n2}X_2 & \cdots & M_{nn}X_n \end{bmatrix},$$

then we obtain

$$\text{diag}^{-1}(MXA) = \begin{bmatrix} \sum_{i=1}^n M_{1i}A_{i1}X_i \\ \sum_{i=1}^n M_{2i}A_{i2}X_i \\ \vdots \\ \sum_{i=1}^n M_{ni}A_{in}X_i \end{bmatrix}.$$

On the other side, we have $(M \circ A)\text{diag}^{-1}(X) =$

$$= \begin{bmatrix} M_{11}A_{11} & M_{12}A_{12} & \cdots & M_{1n}A_{1n} \\ M_{21}A_{21} & M_{22}A_{22} & \cdots & M_{2n}A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1}A_{n1} & M_{n2}A_{n2} & \cdots & M_{nn}A_{nn} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix},$$

Thus for $j = 1, \dots, n$ we have

$$\begin{aligned} [\text{diag}^{-1}(MXA)]_j &= \sum_{i=1}^n M_{ji}A_{ij}X_i = \sum_{i=1}^n M_{ji}A_{ji}X_i \\ &= [(M \circ A)\text{diag}^{-1}(X)]_j. \end{aligned} \quad (2.49)$$

□

We use this lemma in the following proposition to solve (2.44) - (2.46) in order to obtain some linear relations between the variables.

We now define the $m_i \times 1$ vector $\kappa_i^{\text{vec}}(t) = \kappa_i^\top(t)\mathbf{1}_m$, collecting all the decisional variables of the diagonals of $\kappa_i(t)$ on a column vector. Similarly, we define also $\eta_i^{\text{vec}}(t) = \eta_i^\top(t)\mathbf{1}_m$, $\lambda_i^{\text{vec}}(t) = \lambda_i^\top(t)\mathbf{1}_m$ and $\nu_i^E(t) = [\mathbf{1}_{m_i}\nu_i^\top(t)] \circ (\mathbf{1} \otimes I)\mathbf{1}_m$, collecting N_i times the vector ν_i . We are ready to provide the optimal solution for problem (2.34). Note that the results of the following propositions will be enough in each node to compute the optimal weights at each time step.

Proposition 2.3.1 *The optimal solution for problem (2.34), for each node i , is given by $\xi_{i_1} = 0$, $\xi_{i_2} = 0$, $\xi_{i_3} = 0$, $\xi_{i_4} = 0$,*

$$\kappa_i^{\text{vec}}(t) = - (2M_i(t, \rho_i) \circ \mathbf{D}_i(t))^{-1} \nu_i^E(t), \quad (2.50a)$$

$$\eta_i^{\text{vec}}(t) = - (2C_{iE}^\top S_i(t, \rho_i) C_{iE} \circ \mathbf{D}_i(t))^{-1} C_{iE}^\top C_{iE} \nu_i^E(t), \quad (2.50b)$$

$$\nu_i(t) = -2(I_{m \times m_i}((M_i(t, \rho_i) \circ \mathbf{D}_i(t))^{-1} + C_{iE}^\top C_{iE} \quad (2.50c)$$

$$(C_{iE}^\top S_i(t, \rho_i) C_{iE} \circ \mathbf{D}_i(t))^{-1} C_{iE}^\top C_{iE}) I_{m \times m_i}^\top)^{-1} \mathbf{1}_m,$$

$$\lambda_i^{\text{vec}}(t) = (2(M_i^{\text{loc}}(t, \rho_i) + \kappa_i(t)M_i(t, \rho_i)\kappa_i^\top(t)) \quad (2.50d)$$

$$+ \eta_i(t)C_{iE}^\top S_i(t, \rho_i) C_{iE} \eta_i^\top(t)) \circ (A^T A))^{-1}$$

$$(\text{diag}^{-1}(2(\kappa_i(t)M_i\kappa_i^\top(t) + \eta_i(t)C_{iE}^\top S_i(t, \rho_i) C_{iE} \eta_i^\top(t))A^T A)),$$

with $\mathbf{D}_i(t) = \mathbf{1}_{N_i}\mathbf{1}_{N_i}^\top \otimes D_i(t)$ and $D_i(t) := (I - \lambda_i(t))^T A^T A (I - \lambda_i(t))$, if $\kappa_i(t)$ and $\lambda_i(t)$ so computed in addition satisfy (2.36), (2.37), (2.40), (2.41).

Proof: In the following we will omit the dependence on time and on the ρ_i Pareto parameter for the sake of notation simplicity. We observe that (2.46) is equivalent to

$$\begin{aligned} &\text{diag}^{-1} (2(M_i^{\text{loc}} + \kappa_i M_i \kappa_i^T + \eta_i C_{iE}^\top S_i C_{iE} \eta_i^T) \lambda_i A^T A) \\ &= \text{diag}^{-1} (2(\kappa_i M_i \kappa_i^T + \eta_i C_{iE}^\top S_i C_{iE} \eta_i^T) A^T A) + \xi_{i_2} - \xi_{i_1}. \end{aligned} \quad (2.51)$$

By using the result from Lemma 2.3.2, we have

$$\lambda_i^{\text{vec}} = \left(2(M_i^{\text{loc}} + \kappa_i M_i \kappa_i^T + \eta_i C_{iE}^\top S_i C_{iE} \eta_i^T) \circ (A^T A) \right)^{-1} \left(\text{diag}^{-1} \left(2(\kappa_i M_i \kappa_i^T + \eta_i C_{iE}^\top S_i C_{iE} \eta_i^T) A^T A \right) + \xi_{i_2} - \xi_{i_1} \right). \quad (2.52)$$

Let us now analyze KKT conditions in (2.44) and (2.45). We find a solution with $\xi_{i_3} = 0$ and $\xi_{i_4} = 0$. Let note that (2.44) holds if and only if it holds

$$2(M_i \kappa_i^T (I - \lambda_i)^T A^T A (I - \lambda_i)) \circ (\mathbf{1} \otimes I) \mathbf{1}_m = -\mathbf{1}_{m_i} \nu_i^T \circ (\mathbf{1} \otimes I) \mathbf{1}_m = -\nu_i^E. \quad (2.53)$$

Let denote $\kappa_i = [\kappa_{i_1}, \dots, \kappa_{i_{N_i}}]$, where κ_{i_j} is the (j) -th $m \times m$ diagonal block of κ_i , with $j = 1, \dots, N_i$. Similarly we consider $M_{i_{pq}}$, which is the $m \times m$ block of matrix M_i :

$$M_i = \begin{bmatrix} M_{i_{11}} & \cdots & M_{i_{1N_i}} \\ \vdots & \ddots & \vdots \\ M_{i_{N_i 1}} & \cdots & M_{i_{N_i N_i}} \end{bmatrix}.$$

Then, we use the block operations and we achieve

$$\begin{cases} 2(\sum_{k=1}^{N_i} M_{i_{1k}} \kappa_{i_k} D_i) \circ I \mathbf{1}_m = -\nu_i \\ \vdots \\ 2(\sum_{k=1}^{N_i} M_{i_{N_i k}} \kappa_{i_k} D_i) \circ I \mathbf{1}_m = -\nu_i \end{cases},$$

with $D_i := (I - \lambda_i)^T A^T A (I - \lambda_i)$. Thus by using block by block the result in Lemma 2.3.2:

$$2M_i \circ (\mathbf{1}_{N_i} \mathbf{1}_{N_i}^T \otimes D_i) \kappa_i^{\text{vec}} = -\mathbf{1}_{N_i} \otimes v_i = -\nu_i^E. \quad (2.54)$$

Remembering that by definition κ_i^{vec} is the vector that collects all the elements from κ_i , from (2.54), we obtain

$$\kappa_i^{\text{vec}} = - \left(2M_i \circ (\mathbf{1}_{N_i} \mathbf{1}_{N_i}^T \otimes D_i) \right)^{-1} \nu_i^E,$$

and similarly

$$\eta_i^{\text{vec}} = - \left(2C_{iE}^\top S_i C_{iE} \circ (\mathbf{1}_{N_i} \mathbf{1}_{N_i}^T \otimes D_i) \right)^{-1} C_{iE}^\top C_{iE} \nu_i^E,$$

as in the statement of the proposition. According to the first KKT condition, we have

$$(\kappa_i + \eta_i C_{iE}^\top C_{iE}) \mathbf{1}_{m_i} = \mathbf{1}_{N_i}^T \otimes I (\kappa_i^{\text{vec}} + C_{iE}^\top C_{iE} \eta_i^{\text{vec}}) = \mathbf{1}_m, \quad (2.55)$$

which implies that

$$\nu_i = -2(I_{m \times m_i}((M_i \circ \mathbf{D}_i)^{-1} + C_{iE}^\top C_{iE} (C_{iE}^\top S_i C_{iE} \circ \mathbf{D}_i)^{-1} C_{iE}^\top C_{iE}) I_{m \times m_i}^T)^{-1} \mathbf{1}_m, \quad (2.56)$$

where $I_{m \times m_i}$ denotes $\mathbf{1}_{N_i}^T \otimes I_m$, and $\mathbf{D}_i = \mathbf{1}_{N_i} \mathbf{1}_{N_i}^T \otimes D_i$. Thus we have obtained the optimal value of the dual variable ν_i and, by substituting the result in (2.50a) and (2.50b), we obtain the optimal values of κ_i and η_i depending on λ_i . Remember that the weights are adaptive and time-varying, depending on the measurements, and can be computed locally by node i at each step. This is an optimal solution for the Pareto optimization problem (2.34), if, once computed the solutions on-line, in addition they satisfy the constraints (2.36), (2.37), (2.40) and (2.41), which concludes the proof. $\square$ The solution given by the previous proposition, is the optimal solution of problem (2.34). Since in problem (2.34) we have imposed more restrictive conditions than the ones obtained in Section 2.2.3 for (2.25), so the proposed solution could be sub-optimal w.r.t. problem (2.25).

In order to use the result in Prop. 2.3.1, we propose a computational method (Algorithm 1) to find at each step the time-varying optimal values κ_i^* , η_i^* and λ_i^* . At each step, we initialize $\lambda_i = I_m/2$, and define a small positive constant ϵ . We first compute κ_i and η_i with the initial given value of λ_i . Then we update λ_i with the computed values of κ_i and η_i . We repeat this procedure until the difference between current value of λ_i and the updated one is smaller than ϵ as shown in Algorithm 1. At each step, we verify that the obtained values satisfy the constraints (2.36), (2.37), (2.40) and (2.41). In order to guarantee that these are satisfied, we project at each iteration the computed values of κ_i and λ_i in the corresponding sets defined by the constraints. This method can be used with any matrix A norm, since the convergence conditions constraints are not active.

Proposition 2.3.2 *Algorithm 1 converges to the optimal solutions κ_i^* , η_i^* and λ_i^* .*

Proof: In this proof, we first show that Algorithm1 converges, then we show that it converges to the optimal solution. Let note that the optimization problem (2.34) fulfills Slack's conditions, which implies that the optimal solution have finite optimal value. Furthermore, the two iterative sub-optimization steps in Algorithm 1 ensure that the intermediate solutions improve (decrease) the current optimal value. Thus it is sure that Algorithm 1 converges.

Moreover, let recall that the cost function is convex on all the variables. Suppose that Alg. 1 converges to a point $(\kappa'_i, \eta'_i, \lambda'_i)$ different from a nearby point $(\kappa_i^*, \eta_i^*, \lambda_i^*)$.

Algorithm 1 Optimal weights computation

Set $\lambda_i^+ = I_m/2, \epsilon$
 $I_{m \times m_i} = \mathbf{1}_{N_i}^T \otimes I_m$
repeat
 $\lambda_i = \lambda_i^+$
 $\mathbf{D}_i = \mathbf{1}_{N_i} \mathbf{1}_{N_i}^T \otimes (I_m - \lambda_i)^T A^T A (I_m - \lambda_i)$
 $\nu_i = -2[I_{m \times m_i} ((M_i \circ \mathbf{D}_i)^{-1} + C_{iE}^\top C_{iE} (C_{iE}^\top S_i C_{iE} \circ \mathbf{D}_i)^{-1} C_{iE}^\top C_{iE}) I_{m \times m_i}^\top]^{-1} \mathbf{1}_m$ as in (2.50c)
 $\kappa_i^{\text{vec}} = \mathbf{P}_K \left(- (2M_i \circ \mathbf{D}_i)^{-1} \mathbf{1}_{N_i} \otimes I_m \nu_i \right)$
 $\eta_i^{\text{vec}} = - (2C_{iE}^\top S_i C_{iE} \circ \mathbf{D}_i)^{-1} \mathbf{1}_{N_i} \otimes I_m C_{iE}^\top C_{iE} \nu_i$ as in (2.50b)
 $R_i = \kappa_i M_i \kappa_i^T + \eta_i C_{iE}^\top S_i C_{iE} \eta_i^T$
 $\lambda_i^{\text{vec}} = \mathbf{P}_\Lambda ((M_i^{\text{loc}} + R_i) \circ (A^T A))^{-1} \text{diag}^{-1} (R_i A^T A)$
 $\lambda_i^+ = \text{diag}(\lambda_i^{\text{vec}})$
until $|\lambda_i - \lambda_i^+| \leq \epsilon$
return $\kappa_i(t) = \text{diag}(\kappa_i^{\text{vec}}), \eta_i(t) = \text{diag}(\eta_i^{\text{vec}})$, and $\lambda_i(t) = \lambda_i^+$

Then we have

$$f(\kappa_i^*, \eta_i^*, \lambda_i^*) \geq f(\kappa_i', \eta_i', \lambda_i^*) \geq f(\kappa_i', \eta_i', \lambda_i'),$$

where $f(\kappa_i, \eta_i, \lambda_i)$ is the cost function. Since κ_i^*, η_i^* and λ_i^* are the optimal solutions, κ_i', η_i' and λ_i' must be equal to κ_i^*, η_i^* and λ_i^* respectively, which completes the proof. $\square$

Remark 2.3.2 In Section 2.3, we have proposed the solution to the optimization problem (2.34), where weights and parameters in the proposed distributed estimator are optimized to minimize the same cost function in (2.34). A question that may rise concerns the possible separability of the consensus and prediction.

Consider the optimization problem (2.34). The separate design of the consensus-filtering weights κ_i and η_i and of the model-based prediction parameters λ_i leads to suboptimal solutions to the problem (2.25). In fact, from Prop. 2.3.1, it follows that the optimal value of λ_i depends on the optimal values of κ_i and η_i , and vice versa. In (2.50d) we find an optimal solution of λ_i depending on κ_i and η_i , and in (2.50a) and (2.50b) we find an optimal solution of κ_i and η_i depending on λ_i . Therefore, the computation of the solution of two separate optimization problems would certainly lead to a suboptimal solution unless additional assumption on the optimization problem are introduced. The proposed joint optimization approach in Algorithm 1 can lead to improved solutions. And Prop. 2.3.2 proved that the weights and parameters can be jointly optimized.

Even considering only error variance minimization, thus choosing $\rho_i = 0$ in the proposed method, the choice of the filtering weights and prediction parameters has

to be jointly designed. This is an important result, since in many works of the state of the art (see [4], [25] and [21], as examples, or whenever using a filtering method for dynamic prediction without optimizing the prediction parameters) the two steps (filtering and prediction) are designed separately, thus possibly causing suboptimal performances.

2.4 Implementation of the Distributed Prediction Algorithm

In this section, we address how to implement the proposed distributed prediction method. Each node has to implement the estimator given by Algorithm 2. The nodes are not required to on-line solve the optimization problem (2.25) formulated in the previous sections, since the optimal weights values can be found using the computation procedure described by Algorithm 1. In the literature, it is common to determine the best value of ρ_i by building the Pareto trade-off curve and selecting the "knee-point" of this curve, that is, choosing ρ_i^* such that B_i and V_i , computed with the optimal values $\kappa_i^*(\rho_i^*)$, $\eta_i^*(\rho_i^*)$ and $\lambda_i^*(\rho_i^*)$, are $V_i = B_i^2$. This can be obtained by solving the following further problem:

$$\min_{\rho_i} (V_i(\kappa_i^*(\rho_i), \eta_i^*(\rho_i), \lambda_i^*(\rho_i)) - B_i^2(\kappa_i^*(\rho_i), \eta_i^*(\rho_i), \lambda_i^*(\rho_i)))^2.$$

This problem is highly non-linear. Numerical methods can be used to compute the optimal value and, in the literature, genetic algorithms are often used (see, for instance, [69], [68]). In this and previous papers, we tested different approaches for the definition of the Pareto parameter. It is possible to choose it locally, using the Nelder-Mead simplex algorithm as described in [77], to minimize the cost function $(1 - \rho_i)V_i + \rho_i B_i^2$ with the values of parameters and weights obtained at the previous step. Note that the values of V_i and B_i are function of ρ_i . The Nelder-Mead algorithm is one of the most widely used methods for nonlinear unconstrained optimization problem adopting a direct search method that allows to avoid the computation of numerical or analytic gradients, which are difficult to obtain in our case due to the presence of matrix inverses. Once the parameter ρ_i has been set, the optimal weights $\kappa_i(t)$, $\eta_i(t)$ and $\lambda_i(t)$ are computed using the result in Prop. 2.3.1 (following Algorithm 1) and so the local prediction of the state $x_i(t+1)$ can be obtained. After that, the values of the estimates of $\hat{\Gamma}_i(t)$ and $\hat{m}_{\epsilon_i}(t)$ can be updated using new signal estimates and measurements samples (see [136] for details about the estimation from samples procedure).

Algorithm 2 Prediction algorithm for node i

```

 $t := 0$ 
 $\hat{m}_{\epsilon_i}(0) := 0$ 
 $\hat{\Gamma}_i(0) := \Sigma_{v_i}$ 
 $\rho_i := 0.5$ 
 $\bar{x}_i(0) := C_i^\top y_i(0)$ 
 $\hat{x}_i(1) := A\bar{x}_i(0)$ 
repeat
   $N_i := |\mathcal{N}_i|$ 
   $t := t + 1$ 
  Collect predictions  $\hat{\mathbf{x}}_i(t) := (\hat{x}_{i_1}(t), \dots, \hat{x}_{i_{N_i}}(t))^T$  where  $\{i_1, \dots, i_{N_i}\} \in \mathcal{N}_i$ 
  Collect measurements  $\mathbf{y}_i(t) := (C_{i_1}^\top y_{i_1}(t), \dots, C_{i_{N_i}}^\top y_{i_{N_i}}(t))^T$  where  $\{i_1, \dots, i_{N_i}\} \in \mathcal{N}_i$ 
   $\hat{\epsilon}_i := \frac{\hat{\mathbf{x}}_i}{1+\nu} - \frac{\nu \mathbf{1}^\top \hat{\mathbf{x}}_i + (1+\nu) \mathbf{1}^\top \mathbf{y}_i}{N_i(1+2\nu)(1+\nu)} \mathbf{1}$ 
   $\hat{m}_{\epsilon_i}(t) := \frac{t-1}{t} \hat{m}_{\epsilon_i}(t-1) + \frac{1}{t} \hat{\epsilon}_i(t)$ 
   $\hat{\Gamma}_i(t) := \frac{t-1}{t} \hat{\Gamma}_i(t-1) + \frac{1}{t} (\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))(\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))^\top$ 
  Compute  $\rho_i$ 
   $S_i := (1 - \rho_i) \Sigma_{v_{\epsilon_i}}(t)$ 
   $M_i := \rho_i (\hat{m}_{\epsilon_i}(t) \hat{m}_{\epsilon_i}^\top(t)) + (1 - \rho_i) \hat{\Gamma}_i(t)$ 
  Compute  $\kappa_i(t)$ 
  Compute  $\eta_i(t)$ 
  Compute  $\lambda_i(t)$ 
   $\bar{x}_i(t) = \kappa_i(t) \hat{\mathbf{x}}_i(t) + \eta_i(t) \mathbf{y}_i(t)$ 
   $\hat{x}_i(t+1) = A\bar{x}_i(t) + \bar{w}(t) + \lambda_i(t)(A\hat{x}_i(t) - A\bar{x}_i(t))$ 
until forever

```

2.4.1 Computational complexity

The computational complexity of the proposed distributed estimator is given mainly by two components: the computational complexity of matrices inverse and that needed for the estimation of the covariance matrix. The computation of matrices inverse is required to compute the optimal weights $\kappa_i(t)$, $\eta_i(t)$ and $\lambda_i(t)$. It has complexity $O((m \times |\mathcal{N}_i|)^3)$. The computational complexity of Algorithm 1 is mainly given by the computation of matrices inverse. In the simulation, the typical number of iterations N_{Iter1} at each step is less than 10. The computation of the covariance matrix is required to compute the approximate estimates $\hat{\Gamma}_i(t)$ and $\hat{\Lambda}_i(t)$: the complexity is $O(\text{Table}_{\text{size}} \log(\text{Table}_{\text{size}}))$, where the $\text{Table}_{\text{size}}$ is the size of a look-up table used to speed up the computation of a quadratically constrained least-square problem [136]. We set $\text{Table}_{\text{size}} = 100$.

2.5 Simulation Results

In this section, simulation results are given with the purpose of illustrating the analysis. In [15], it is explained how to derive the estimates of the needed quantities to implement the proposed method. To test the performance of the distributed estimator proposed in this paper, we consider a linear system, whose dynamics are described by the matrix:

$$A = \begin{pmatrix} 0.8 & 0.1 & 0 & 0.05 \\ 0.1 & 0.55 & 0 & 0.2 \\ 0.04 & 0 & 0.7 & 0.2 \\ 0 & 0.15 & 0.3 & 0.5 \end{pmatrix}.$$

We assume only 2 or 3 states can be measured by each node. We then randomly choose the matrix C on the basis of this assumption and the conditions in Section 2.1, so that the state is completely observable at least by the set of all the nodes in the sensor network.

We analyzed the state of the art in order to identify some methods for comparison. The distributed estimation methods in [19], [154] and [94] consider a partially-measurable case; however the first two approaches require the assumption of a strongly connected sensor network and so we can not use them for comparison. Hence, we selected from the literature the following prediction methods:

E_{CKF} : Centralized Kalman filter. The estimator collects all the measurements y_i available at the entire sensor network and the related output matrices C_i , for all $i = 1, \dots, n$ and implements the classical steady-state Kalman predictor.

E_{DKF} : Decentralized Kalman filter as in [103].

E_{DE1} : Distributed state estimation approach as in [9]. The estimator is adopted by formulating consensus on probability density functions of the states.

E_{DE2} : Distributed state estimation with event triggered communication protocols as in [94]

E_{p1} : The proposed estimator using the computational solution in Algorithm 1.

The use of the centralized Kalman predictor is included to provide a baseline performance index and, for the sake of simplicity, the *steady-state* Kalman predictor has been considered. This latter predictor, in specific scenarios may show suboptimal performance during the transient when compared to time-varying algorithms like the

proposed one. For simulation consistency, since we are considering the case where each node has no access to the entire state vector, the centralized Kalman predictor processes all the measurements available from all nodes of the sensor network in a centralized way. For all the methods, we compare the one-step ahead prediction error.

A 15-nodes network is obtained by distributing the nodes randomly over a squared area of size $N/2$ and the graph by letting two nodes communicate if their relative distance is less than $1.7\sqrt{N}$. We investigate the performance of the proposed methods with different noise probability distributions and different standard deviations for the measurement and modeling disturbance noises. We repeat the experiment 80 times with different random network topologies for each scenario. After many simulation experiments, we decided to fix $\rho_i = 0.5$ for each node i in the simulations, in order to reduce the computational complexity, since the obtained performances were similar to the knee-point method described in Section 2.4. The evaluated performance metric in one experiment is the mean square error of the predictions at each node, that is then averaged over all the nodes of the network. We then average this value over 80 experiments for each noise scenario (in the figures we use the term MSE to refer to this final average).

We firstly consider Gaussian white noise for both the measurement noise and system disturbance. The results are presented in the first three rows of Fig. 2.1. Each figure shows the performance of the considered prediction methods with different noise scenarios, letting varying the standard deviation of the measurement noise from 1 to 3 on the x axis and the standard deviation of the disturbance noise from 1 to 2.6 (different figures on the same row). In the first row, all the five considered methods are illustrated. The second row shows the performances of only three of the considered methods in order to appreciate, with the different scale, how the prediction error increases as the standard deviation values increase. The third row is needed to properly compare the performances of the three best methods, using different scales for each figure. We see that the proposed estimators always have better results than all the other distributed methods, in all the considered scenarios. Compared with the centralized steady-state Kalman filter, which is optimal in the Gaussian case compared to steady-state linear estimators, the error of the proposed algorithms is lower than that of the centralized Kalman filter in a single scenario, when the standard deviation of measurement noise is far larger than that of the disturbance (as it is possible to see in Fig. 2.1, third row, first column). This is due to the fact that the time-varying proposed algorithm optimizes the used information at each step, also in the transient.

We then tested our method on a more challenging scenario with non-Gaussian distribution of the noises (see fourth row in Fig. 2.1). We consider the Tri-Gaussian noise, which is a non-Gaussian noise introduced in [61]. We use this distribution both for the measurement noise and the system disturbance. The Tri-Gaussian distribution is composed of three Gaussian distributions, and can be modeled as

$$p(x) = \sum_{i=1}^3 \alpha_i \mathcal{N}(x; \mu_i, R_i),$$

where $\sum_i \alpha_i = 1, \alpha_i > 0$ and $R_i > 0$. By taking $\alpha_1 = \alpha_3, \mu_1 = -\mu_3, \mu_2 = 0, R_i = 1 - \mu_1^2(1 - \alpha_2)$, a unit standard deviation Tri-Gaussian distribution can be obtained, having two degrees of freedom. By randomly choosing the pair $(\alpha_2, \mu_1) = (0.55, 0.1)$, the noise can be generated by using Monte Carlo simulation method. From Fig. 2.1, last row, we can see the performance of the best three estimation methods in each of the 30 noise scenarios, for both the probability distributions (Gaussian and non-Gaussian). Obviously in the Gaussian case all the methods perform better than in the non-Gaussian case. It is anyway interesting to see that the distance between our method and the centralized Kalman Filter performance decreases in the non-Gaussian case, as shown in Figs. 2.2, where the ratio between our method MSE and CKF MSE is illustrated for all the considered noise scenarios.

2.6 Concluding Remarks

In this chapter, we propose a novel distributed estimation method using a sensor network for the monitoring of a centralized or decentralized system. The state may be not entirely measured by each node. And we do not require Gaussian distribution for the noises. The optimal filtering weights and prediction parameters are computed by each sensor node at each time step to minimize both the mean and the variance of the prediction error. We show that the filtering weights and the prediction parameters have to be jointly optimized. For the monitoring architecture of a distributed system, a new distributed estimation algorithm over several sensor networks will be presented in the next chapter.

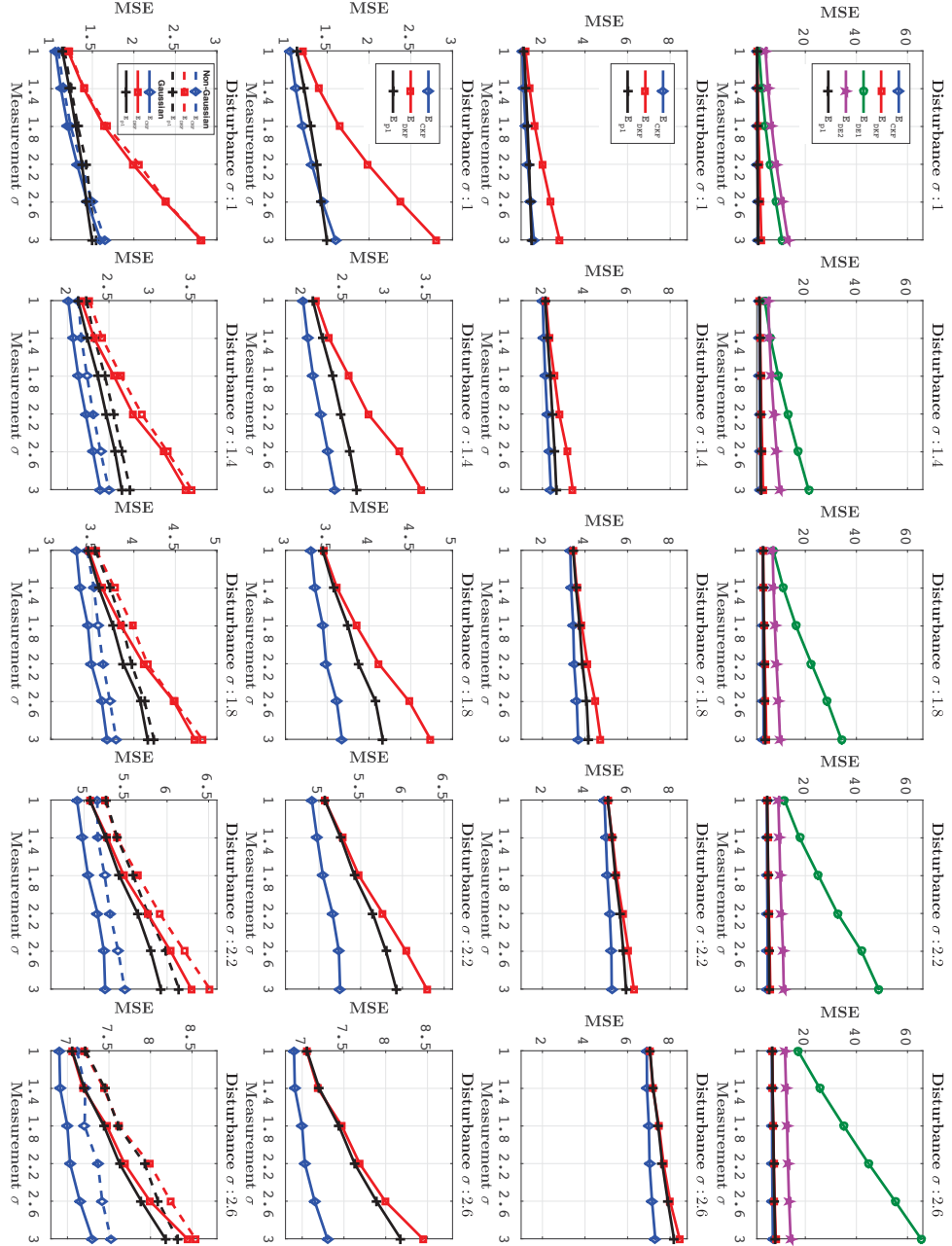


Fig. 2.1 Performance comparison in terms of MSE of the five considered estimation methods, with different standard deviation values for the measurement noises and the process disturbance. The first three rows consider the Gaussian case. In the fourth row, the white Gaussian noise case is represented by solid lines, while the dashed lines show the non-Gaussian noise case performance.

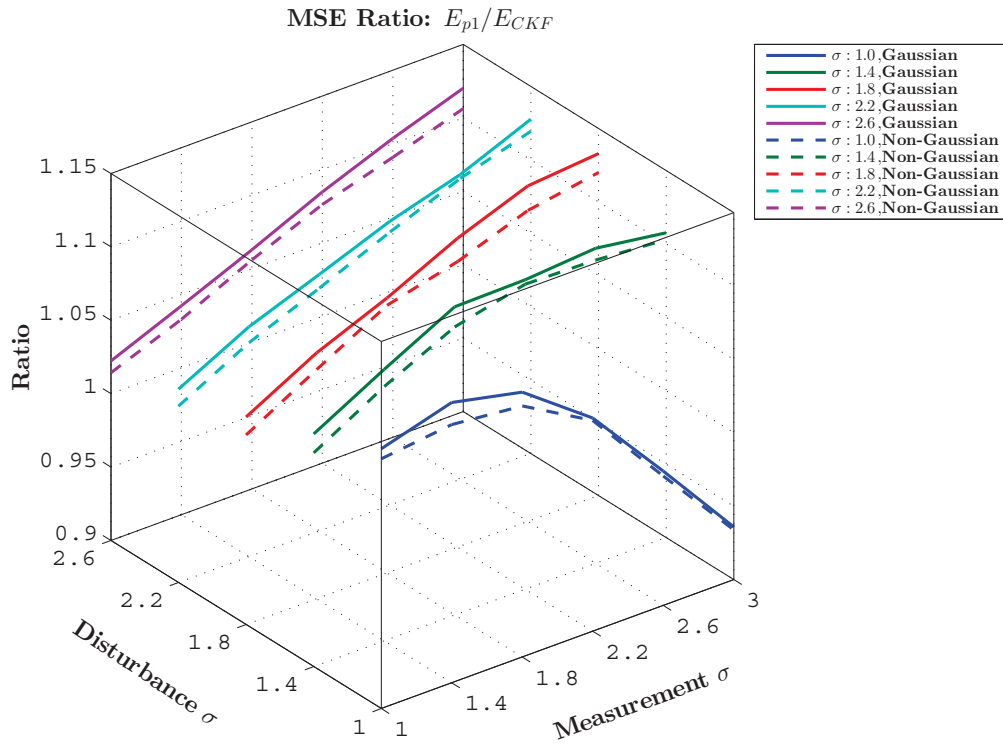


Fig. 2.2 The distance between the performance of the proposed approach E_{p1} and the Centralized Kalman Filter in terms of ratio of respective MSE values for the 30 different noise scenarios, both in the Gaussian and in the non-Gaussian case.

Chapter 3

THE ALGORITHM FOR DISTRIBUTED SYSTEMS

In this chapter, a novel distributed state estimation algorithm over sensor networks is proposed to estimate the partitioned state of the distributed system in a multi-objective optimization framework. Each subsystem is monitored by a sensor network, where each sensor may communicate both with other sensors monitoring the same subsystem and with sensors measuring the state of neighboring subsystems. There are no assumptions on the topology of the communication network, apart from connectedness. Generally, the distributed estimation works according to the following procedure: Firstly, each node communicates measurements and predictions to neighboring nodes in the local sensor network and the neighboring sensor networks. Secondly, each node filters the measurements by taking a linear combination of measurements and predictions from itself and its neighboring nodes in the local sensor network. Finally, the node uses the current filtered measurements and predictions from neighboring nodes in the neighboring sensor networks to implement a model-based predictor, smoothing the previous prediction error.

When designing the multi-objective optimization framework, two indicators of the quality of the distributed estimation approach: the mean of the estimation error (bias) and the variance are considered. Moreover, the time-varying filtering weights and prediction parameters are jointly optimized by each sensor node at each time step in a Pareto optimization framework. We consider correlations between the local state and neighboring systems' states, thus dealing with a more challenging scenario for the optimization, with respect to the previous works ([15–17]). Asymptotic convergence conditions are provided. The on-line computation of the time-varying optimal filtering weights and prediction parameters allows to consider the transient performance, which

gives the proposed method an advantage over other methods [46, 125, 139], where only asymptotical performance is investigated.

The chapter is organized as follows. In Section 3.1, we introduce the problem formulation and give the assumptions. In Section 3.2, the distributed dynamic estimation method is presented, followed by some convergence conditions. The multi-objective optimization problem and its optimal weighs are demonstrated in Section 3.3. Then, some simulation results are illustrated in Section 3.4 and concluding remarks are given in Section 3.5.

3.1 Problem Formulation

The system we want to monitor is composed of (or can be decomposed in) N interconnected subsystems. Each subsystem is modeled as

$$\Sigma_I : x_I(t+1) = A_{II}x_I(t) + \sum_{J \in \mathcal{O}_I} A_{IJ}x_J(t) + w_I(t), \quad (3.1)$$

with $I = 1, \dots, N$, where t is the discrete time, $x_I \in \mathbb{R}^{m_I}$ denotes the local state vector, $w_I \in \mathbb{R}^{m_I}$ represents modeling uncertainties and process noises and disturbances, $\mathcal{O}_I$ is a set collecting parents of subsystem Σ_I , that is, the subsystems Σ_J whose state x_J influence the dynamics of Σ_I . Matrix A_{II} describes local dynamics, while A_{IJ} models the coupling between Σ_I and Σ_J , $J \in \mathcal{O}_I$.

Assumption 6 *We assume that w_I is a noise with known mean $\bar{w}_I$ and covariance matrix Σ_{w_I} , for each $I = 1, \dots, N$.*

The system is monitored by sensor networks $\mathcal{S}$, composed of n sensors. Each subsystem Σ_I , $I = 1, \dots, N$, is monitored by a sensor network $\mathcal{S}_I$, composed of n_I sensors. Each sensor $i \in \mathcal{S}_I$ measures the state x_I , according to the following:

$$y_I^i(t) = x_I(t) + v_I^i(t), \quad (3.2)$$

where $y_I^i \in \mathbb{R}^{m_I}$, denotes the measurements vector taken by sensor $i \in \mathcal{S}_I$ and $v_I^i \in \mathbb{R}^{m_I}$ is a zero-mean measurement noise, with $\Sigma_{v_I^i}$ its covariance matrix.

Assumption 7 *The measurement noise v_I^i and process disturbances w_I are not correlated, for each $i = 1, \dots, n$ and $I = 1, \dots, N$.*

Assumption 8 *We assume that graph $\mathcal{G}$ is connected.*

A communication network is defined, allowing the sensors to communicate. It is modeled as an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where $\mathcal{S}$ is the set of the nodes (the sensors) and $\mathcal{E}$ is the set of the edges connecting the nodes.

Each sensor $i \in \mathcal{S}_I$ may communicate with two different subsets of neighbouring nodes:

- some neighbouring sensors in $\mathcal{S}_I$, which we denote $\mathcal{N}_I^i$;
- some sensors in $\mathcal{S}_J$, $J \in \mathcal{O}_I$, collected in the sets $\mathcal{N}_{IJ}^i$.

Specifically, we denote $\mathcal{N}_I^i = \{l \in \mathcal{S} \cap \mathcal{S}_I : (l, i) \in \mathcal{E}\} \cup \{i\}$ the set of neighbors of node $i \in \mathcal{S}$ monitoring subsystem Σ_I plus the node itself. On the other hand, $\mathcal{N}_{IJ}^i = \{l \in \mathcal{S} \cap \mathcal{S}_J : (l, i) \in \mathcal{E}\}$. Each node can communicate measurements and estimates to the neighboring sensors.

3.2 Distributed State Dynamic Estimation

In this chapter each node i of the sensor network implements a two steps dynamic estimator: a filtering step and a prediction step. First, by communicating with neighboring nodes in $\mathcal{N}_I^i$, it filters the measurements noise in a consensus-like fashion by computing a linear combination of its own and neighbors' available measurements and predictions:

$$\bar{x}_I^i(t) = \sum_{l \in \mathcal{N}_I^i} k_I^{i,l}(t) \hat{x}_I^l(t) + \sum_{l \in \mathcal{N}_I^i} h_I^{i,l}(t) y_I^l(t), \quad (3.3)$$

where $k_I^{i,l}(t)$ and $h_I^{i,l}(t) \in \mathbb{R}^{m_I \times m_I}$ are diagonal matrices collecting the time-varying filtering weights. The objective of this first step is for each node to reduce its own measurement uncertainty, without the use of centralized coordination. Sensors are allowed to communicate only once between subsequent time-instants.

After the filtering-merging step, each node implements a model-based prediction, using filtered measurements (obtained from the first step), instead of the observed measurements. In this second step each node $i \in \mathcal{S}_I$ communicates with neighboring nodes $\mathcal{N}_{IJ}^i$, $J \in \mathcal{O}_I$, in order to estimate the coupling influence. The one-step-ahead prediction is computed as

$$\hat{x}_I^i(t+1) = A_{II} \bar{x}_I^i(t) + \bar{w}_I(t) + \lambda_I^{i,i}(t) (\hat{x}_I^i(t) - \bar{x}_I^i(t)) + \sum_{J \in \mathcal{O}_I} A_{IJ} \sum_{j \in \mathcal{N}_{IJ}^i} \omega_{IJ}^{i,j}(t) \hat{x}_J^j(t), \quad (3.4)$$

where $\lambda_I^i(t) = A_{II}\lambda_I^i(t)$, being $\lambda_I^i(t) \in \mathbb{R}^{m_I \times m_I}$ and $\omega_{IJ}^{i,j}(t) \in \mathbb{R}^{m_J \times m_J}$ diagonal matrices collecting the time-varying filter parameters. The term $\lambda_I^i(t)(\hat{x}_I^i(t) - \bar{x}_I^i(t))$ represents a local correction of the previous prediction error, taking into account the system dynamics.

The goal is to design $k_I^{i,l}(t)$, $h_I^{i,l}(t)$, $\lambda_I^i(t)$ and $\omega_{IJ}^{i,j}(t)$, for each $I = 1, \dots, N$, $i = 1, \dots, n_I$, $l = 1, \dots, m_I$, $J \in \mathcal{O}_I$, so as to minimize at each time step the mean and variance of the local estimation error.

3.2.1 Local Filtering and Prediction Errors

Let us define the following estimation errors: the local filtering error

$$\bar{e}_I^i(t) = \bar{x}_I^i(t) - x_I^i(t)$$

and the local prediction error

$$\hat{E}_I^i(t) = \hat{x}_I^i(t) - x_I^i(t).$$

We can rewrite Eqs. (3.3) and (3.4) as:

$$\begin{aligned} \bar{x}_I^i(t) &= \kappa_I^i(t)\hat{x}_I^{i\text{reg}}(t) + \eta_I^i(t)y_I^{i\text{reg}}(t) \\ \hat{x}_I^i(t+1) &= A_{II}\bar{x}_I^i(t) + A_{II}\lambda_I^i(t)(\hat{x}_I^i(t) - \bar{x}_I^i(t)) + \bar{w}_I(t) + A_I^{\text{inei}}\omega_I^i(t)\hat{x}_I^{i\text{nei}}(t), \end{aligned} \quad (3.5)$$

where $\hat{x}_I^{i\text{reg}}$ and $y_I^{i\text{reg}}$ are two column vectors collecting the prediction vectors and the measurements vectors respectively available at node i related to subsystem Σ_I , ordered according to their indexes: $\hat{x}_I^{i\text{reg}} = (\hat{x}_I^{i_1\top}, \dots, \hat{x}_I^{i_{N_I^i}\top})^\top$, $i_1 < \dots < i_{N_I^i}$, and $y_I^{i\text{reg}} = (y_I^{i_1\top}, \dots, y_I^{i_{N_I^i}\top})^\top$, $i_1 < \dots < i_{N_I^i}$, with N_I^i being the cardinality of the set $\mathcal{N}_I^i$. Moreover, $\hat{x}_I^{i\text{nei}}$ is a column vector collecting the prediction vectors available at node i related to subsystems Σ_J , $J \in \mathcal{O}_I$, ordered according to their indexes. Furthermore, $\kappa_I^i(t)$ and $\eta_I^i(t)$ are the time varying row block matrices $\in \mathbb{R}^{m_I \times N_I^i m_I}$ collecting matrices $k_I^{i,l}$ and $h_I^{i,l}$ respectively; A_I^{inei} is a row block matrix collecting matrices A_{IJ} , $J \in \mathcal{O}_I$. Finally, $\omega_I^i(t)$ is a block matrix collecting on the diagonal blocks matrices $\omega_{IJ}^i(t)$, which is row block matrices and collects $\omega_{IJ}^{i,j}(t)$ according to the order followed in $\hat{x}_I^{i\text{nei}}$.

We introduce the following assumptions:

Assumption 9 Suppose that at each time t

$$(\kappa_I^i(t) + \eta_I^i(t))\mathbf{1}_{N_I^i m_I} = \mathbf{1}_{m_I}$$

and

$$\omega_I^i(t) \mathbf{1}_{p_I^i} = \mathbf{1}_{m_I^i},$$

being N_{IJ}^i the cardinality of set $\mathcal{N}_{IJ}^i$ and $p_I^i = \sum_{J \in \mathcal{O}_I} N_{IJ}^i m_J$, $m_I^i = \sum_{J \in \mathcal{O}_I} m_J$.

These are needed so that the following expressions hold:

$$\kappa_I^i(t) x_I^E(t) + \eta_I^i(t) x_I^E(t) = x_I^E(t)$$

being x_I^E a column vector repeating N_I^i times the state vector x_I , and

$$\omega_{IJ}^i(t) x_J^{iE}(t) = x_J^{iE}(t)$$

being x_J^{iE} a column vector repeating N_{IJ}^i times the state vector x_J , for each $J \in \mathcal{O}_I$. We use these expressions to derive the local filtering error and the local prediction error:

$$\bar{e}_I^i(t) = \kappa_I^i(t) \hat{e}_I^i(t) + \eta_I^i(t) v_{\epsilon_I^i}(t), \quad (3.6)$$

$$\begin{aligned} \hat{E}_I^i(t+1) = & A_{II}(I - \lambda_I^i(t)) \kappa_I^i(t) \hat{e}_I^i(t) + A_{II} \lambda_I^i(t) \hat{E}_I^i(t) - w_I(t) + \bar{w}_I \\ & + A_{II}(I - \lambda_I^i(t)) \eta_I^i(t) v_{\epsilon_I^i}(t) + A_I^{\text{inei}} \omega_I^i(t) \hat{e}_I^{\text{inei}}(t), \end{aligned} \quad (3.7)$$

where $\hat{e}_I^i$ collects the prediction error vectors available at node i and $v_{\epsilon_I^i}$ collects the measurement noise vectors related to the measurements available at node i , both related to subsystem Σ_I and ordered following their indexes; $\hat{e}_I^{\text{inei}}$ collects the prediction error vectors available at node i related to subsystems Σ_J , $J \in \mathcal{O}_I$.

3.2.2 Global Filtering and Prediction Errors

For analytical purposes, the distributed estimator-predictor formulation can be expressed in a compact vector form that includes all the nodes in the sensor network:

$$\begin{aligned} \bar{x}^E(t) &= K(t) \hat{x}^E(t) + H(t) y^E(t) \\ \hat{x}^E(t+1) &= A^E \bar{x}^E(t) + A^E \lambda^E(t) (\hat{x}^E(t) - \bar{x}^E(t)) + A_{\text{co}}^E \omega^E(t) \hat{x}^E(t) + \bar{w}^E(t), \end{aligned} \quad (3.8)$$

where $\bar{x}^E$, $\hat{x}^E$ and $y^E \in \mathbb{R}^{p \times 1}$, $p = \sum_{J=1}^N m_J n_J$ are column vectors collecting the local vectors $\bar{x}_I^i$, $\hat{x}_I^i$ and y_I^i , respectively, with $I = 1, \dots, N$, $i = 1, \dots, n_I$. We have

$$\begin{aligned}\bar{x}^E &= \text{col}(\bar{x}_J, J = 1, \dots, N), & \bar{x}_J &= \text{col}(\bar{x}_J^j, j \in \mathcal{S}_J), \\ \hat{x}^E &= \text{col}(\hat{x}_J, J = 1, \dots, N), & \hat{x}_J &= \text{col}(\hat{x}_J^j, j \in \mathcal{S}_J), \\ y^E &= \text{col}(y_J, J = 1, \dots, N), & y_J &= \text{col}(y_J^j, j \in \mathcal{S}_J).\end{aligned}$$

Consequently, A^E is a diagonal block matrix, having on the diagonal the matrices A_{II} , eventually repeated according to the index in $\bar{x}^E$. Similarly, λ^E is a diagonal matrix, collecting the matrices λ_I^i , $I = 1, \dots, N$, $i = 1, \dots, n_I$. Furthermore, $\bar{w}^E$ is a column vector of appropriate length, collecting the process disturbance mean vector $\bar{w}_I$, $I = 1, \dots, N$ repeated n_I times.

Define a $\delta \times 1$ vector e_δ^γ , where only γ^{th} component is 1, others are null. $K(H$ respectively) is diagonal block matrix, where I^{th} diagonal block matrix K_I equals to $\sum_{i=1}^{n_I} e_{n_I}^i \otimes K_I^i$, $I = 1, \dots, N$. The $m_I \times n_I m_I$ row block matrix K_I^i (H_I^i respectively) collects the diagonal matrices $k_I^{i,l}$ ($h_I^{i,l}$ respectively), $l = 1, \dots, n_I$, where $k_I^{i,l}$ is null if i and l are not neighboring nodes. Given $m = \sum_{J=1}^N m_J$ and $n = \sum_{J=1}^N n_J$, A_{co}^E is a $p \times mn$ diagonal block matrix, where I^{th} diagonal block matrix $\mathbf{A}_I^E = \sum_{i=1}^{n_I} (e_{n_I}^i e_{n_I}^{i\top}) \otimes A_I^E$, $I = 1, \dots, N$. The $m_I \times m$ row block matrix A_I^E includes matrices A_{IJ} , $J = 1, \dots, N$, ordered according the indices, and it is a null matrix if $J \notin \mathcal{O}_I$. Finally, $\omega^E(t)$ is a $mn \times p$ column block matrix, equals to $\sum_{I=1}^N e_N^I \otimes (\sum_{i=1}^{n_I} e_{n_I}^i \otimes \omega_I^{iE})$. Each $m \times p$ matrix ω_I^{iE} , $i = 1, \dots, n_I$, $I = 1, \dots, N$, is a diagonal block matrix, where the J^{th} $m_J \times m_J n_J$ diagonal block ($J = 1, \dots, N$) is null if Σ_I , to which sensor i belongs, is not neighbour of Σ_J (or it I itself); while it collects the matrices $\omega_{IJ}^{i,j}(t)$, $j \in \mathcal{S}_J$ if $J \in \mathcal{O}_I$. Now, let us define the following extended estimation errors: the global filtering error

$$\bar{e}(t) = \bar{x}^E(t) - x^E(t)$$

and the global prediction error

$$\hat{E}(t) = \hat{x}^E(t) - x^E(t),$$

with x^E being a $p \times 1$ column vector, where each local state vector x_I is repeated n_I times, for each $I = 1, \dots, N$. Given Assumption 9, it holds

$$(K(t) + H(t))\mathbf{1}_p = \mathbf{1}_p$$

and

$$\omega^E(t)\mathbf{1}_p = \mathbf{1}_{mn}.$$

Therefore, the filtering error can be computed as

$$\begin{aligned}\bar{e}(t) &= K(t)\hat{x}^E(t) + H(t)y^E(t) - x^E(t) \\ &= K(t)\hat{E}(t) + H(t)v^E(t).\end{aligned}\tag{3.9}$$

Then, the prediction error is:

$$\begin{aligned}\hat{E}(t+1) &= A^E(I - \lambda^E(t))H(t)v^E(t) + A^E((I - \lambda^E(t))K(t) \\ &\quad + \lambda^E(t))\hat{E}(t) + A_{\text{co}}^E\omega^E(t)\hat{E}(t) - w^E(t) + \bar{w}^E.\end{aligned}$$

In order to express prediction error in a compact format, a $p \times mn$ diagonal block matrix A_{global}^E is introduced, where I^{th} diagonal block matrix $\mathbf{A}_I^{Eg} = \sum_{i=1}^{n_I} (e_{n_I}^i e_{n_I}^{i\top}) \otimes A_I^{Eg}$, $I = 1, \dots, N$. The $m_I \times m$ row block matrix A_I^{Eg} includes matrices A_{IJ} , $J = 1, \dots, N$, ordered according the indices, and it is a null matrix if $J \notin \mathcal{O}_I$ and $J \neq I$. Then we get

$$A_{\text{co}}^E\omega^E(t) = A_{\text{global}}^E\omega^E(t).$$

Term $[(I - \lambda^E)K + \lambda^E]$ is a diagonal block matrix, where the I^{th} block is denoted by $\phi_I = \sum_{i=1}^{n_I} e_{n_I}^i \otimes \phi_I^i$, $I = 1, \dots, N$. The $m_I \times n_I m_I$ row block matrix ϕ_I^i equals to $(I - \lambda_I^i(t))K_I^i(t) + \lambda_I^{iE}(t)$, where $\lambda_I^{iE}(t)$ is a $m_I \times m_I n_I$ block matrix having the sub-matrix $\lambda_I^i(t)$ in the block position corresponding to the i^{th} index in set $\mathcal{S}_I$, and all the other blocks components equal to $\mathbf{0}$. Define a $m \times p$ diagonal block matrix φ_I^{iE} , where only I^{th} diagonal block equals to ϕ_I^i and all others are null. So we have

$$A^E[(I - \lambda^E(t))K(t) + \lambda^E(t)] = A_{\text{global}}^E\varphi^E(t),$$

where $\varphi^E = \sum_{I=1}^N e_N^I \otimes (\sum_{i=1}^{n_I} e_{n_I}^i \otimes \varphi_I^{iE})$. Given the compact weight matrix $\Omega^E \triangleq \omega^E + \varphi^E$,

$$\hat{E}(t+1) = A_{\text{global}}^E\Omega^E(t)\hat{E}(t) + \bar{w}^E - w^E(t) + A^E(I - \lambda^E(t))H(t)v^E(t).\tag{3.10}$$

Now, we can derive the expressions of the expected value for the global filtering and prediction errors. The mean values can be computed as

$$\mathbb{E}\bar{e}(t) = K(t)\mathbb{E}\hat{E}(t), \quad (3.11)$$

$$\mathbb{E}\hat{E}(t+1) = A_{\text{global}}^E \Omega^E(t) \mathbb{E}\hat{E}(t). \quad (3.12)$$

3.2.3 Stability of the Prediction Error

In this subsection, we analyze the stability of the proposed estimation method. Some local conditions on the time-varying weights κ_I^i , η_I^i , λ_i , and ω_I^i are introduced so that the expected value of the global prediction error $\mathbb{E}\hat{E}(t)$ can converge to zero. Moreover, the convergence property is guaranteed in all network nodes, by deriving local conditions. Given two vectors v and v_1 , we denote $v < v_1$ or $v > v_1$ the component-by-component inequality. We use a similar notation for matrices. For the sake of notation simplicity, we omit the dependence on t of the matrices in the following proposition.

Proposition 3.2.1 *Consider the global prediction error mean $\mathbb{E}\hat{E}$. The following local conditions are sufficient to guarantee the asymptotic stability of the prediction error. For each r -th row of K_I^i , r -th diagonal of λ_I^i and each r' -th row of J -th diagonal block matrix of ω_I^{iE} , with $I = 1, \dots, N$, $i = 1, \dots, n_I$, $r = 1, \dots, m_I$ and $r' = 1, \dots, m_J$:*

◦ if $1/\|A_I^{Eg}\|_\infty > 1$, then

$$\begin{aligned} \sum_{l=1}^{n_I} |k_I^{i,l^r}| &< \frac{1}{\|A_I^{Eg}\|_\infty} \\ -\frac{\frac{1}{\|A_I^{Eg}\|_\infty} + \sum_{l=1}^{n_I} |k_I^{i,l^r}|}{1 + \sum_{l=1}^{n_I} |k_I^{i,l^r}|} &< \lambda_I^{i^r} < \frac{\frac{1}{\|A_I^{Eg}\|_\infty} + \sum_{l=1}^{n_I} |k_I^{i,l^r}|}{1 + \sum_{l=1}^{n_I} |k_I^{i,l^r}|} \\ \sum_{l=1}^{n_J} |\omega_{I,J}^{i,l^{r'}}| &< \frac{1}{\|A_I^{Eg}\|_\infty}, J \in \mathcal{O}_I. \end{aligned} \quad (3.13)$$

◦ if $1/\|A_I^{Eg}\|_\infty = 1$, then

$$\begin{aligned} \sum_{l=1}^{n_I} |k_I^{i,l^r}| &< \frac{1}{\|A_I^{Eg}\|_\infty} \\ \frac{-\frac{1}{\|A_I^{Eg}\|_\infty} + \sum_{l=1}^{n_I} |k_I^{i,l^r}|}{1 + \sum_{l=1}^{n_I} |k_I^{i,l^r}|} &< \lambda_I^{i^r} < 1 \\ \sum_{l=1}^{n_J} |\omega_{I,J}^{i,l^{r'}}| &< \frac{1}{\|A_I^{Eg}\|_\infty}, J \in \mathcal{O}_I. \end{aligned} \quad (3.14)$$

◦ if $1/\|A_I^{Eg}\|_\infty < 1$, then

$$\begin{aligned} \sum_{l=1}^{n_I} |k_I^{i,l^r}| &< \frac{1}{\|A_I^{Eg}\|_\infty} \\ \frac{-\frac{1}{\|A_I^{Eg}\|_\infty} + \sum_{l=1}^{n_I} |k_I^{i,l^r}|}{1 + \sum_{l=1}^{n_I} |k_I^{i,l^r}|} &< \lambda_I^{i^r} < 1 - \frac{1 - \frac{1}{\|A_I^{Eg}\|_\infty}}{1 - \sum_{l=1}^{n_I} |k_I^{i,l^r}|} \\ \sum_{l=1}^{n_J} |\omega_{I,J}^{i,l^{r'}}| &< \frac{1}{\|A_I^{Eg}\|_\infty}, J \in \mathcal{O}_I. \end{aligned} \quad (3.15)$$

Proof: Eq. (3.12) represents the dynamics of a linear time-varying system. We use the results in [10] to derive convergence conditions on the time-varying matrix $\Omega^E(t)$. A necessary and sufficient condition for global asymptotic stability is given in [10], by which there exists a finite $k > 0$ such that

$$\left\| \prod_{t=1}^k A_{\text{global}}^E \Omega^E(t) \right\|_\infty < 1. \quad (3.16)$$

This sufficient condition of (3.16) is given, for each t ,

$$\|A_{\text{global}}^E \Omega^E(t)\|_\infty < 1.$$

Denote $A_- \Omega_I^i(t)$ as i -th row block of I -th $m_I n_I \times p$ row block matrix of $A_{\text{global}}^E \Omega^E(t)$,

$$A_- \Omega_I^i(t) = A_I^{Eg} [\omega_I^{iE}(t) + \varphi_I^{iE}(t)]. \quad (3.17)$$

This condition is sufficiently satisfied if, for each I and i ,

$$\|A_- \Omega_I^i(t)\|_\infty < 1, I = 1, \dots, N, i = 1, \dots, n_I.$$

Thanks to the submultiplicative property of the norm, the convergence condition is implied by

$$\|\omega_I^{iE}(t) + \varphi_I^{iE}(t)\|_\infty < \frac{1}{\|A_I^{Eg}\|_\infty}.$$

For each r -th row of i -th row block of matrix $\varphi_I^E(t)$, and each r' -th row of J -th diagonal block matrix of ω_I^{iE} , with $r = 1, \dots, m_I$ and $r' = 1, \dots, m_J$, we have

$$\begin{aligned} |1 - \lambda_I^{i^r}| \sum_{l=1}^{n_I} |k_I^{i,l^r}| + |\lambda_I^{i^r}| &< \frac{1}{\|A_I^{Eg}\|_\infty} \\ \sum_{l=1}^{n_J} |\omega_{I,J}^{i,l^{r'}}| &< \frac{1}{\|A_I^{Eg}\|_\infty}, J \in \mathcal{O}_I. \end{aligned} \quad (3.18)$$

We analyzed this condition for all the alternative cases to extract it from the absolute value and we obtained the convergence conditions in the proposition statement. $\square$

3.3 The Optimization Problem

The goal of the proposed distributed dynamic estimation method is to predict the local state minimizing the mean and variance of the prediction error at each node. To do that, we propose that each sensor at each step computes the optimal time-varying weights by solving a multi-objective optimization problem, where the first objective is the squared mean of the prediction error and the second objective is the variance of the prediction error. We derive the expressions of the bias and the variance for the local filtering and prediction errors, given in (3.19) and (3.20), respectively. Given $\mathbb{E}\bar{e}_I^i(t) = \kappa_I^i(t)\mathbb{E}\hat{e}_I^i(t)$, the mean values can be computed as

$$\mathbb{E}\hat{E}_I^i(t+1) = A_{II}(I - \lambda_I^i(t))\kappa_I^i(t)\mathbb{E}\hat{e}_I^i(t) + A_{II}\lambda_I^i(t)\mathbb{E}\hat{E}_I^i(t) + A_I^{inei}\omega_I^i(t)\mathbb{E}\hat{e}_I^{inei}(t), \quad (3.19)$$

The variance can be computed as

$$\begin{aligned} &\mathbb{E}[(\hat{E}_I^i(t+1) - \mathbb{E}\hat{E}_I^i(t+1))(\hat{E}_I^i(t+1) - \mathbb{E}\hat{E}_I^i(t+1))^\top] \\ &= W_{1i}(t)\Gamma_{\hat{e}_I^i}(t)W_{1i}(t)^\top + W_{2i}(t)\Sigma_{v_{\hat{e}_I^i}}W_{2i}(t)^\top + W_{3i}(t)\Gamma_{\hat{e}_I^{inei}}(t)W_{3i}(t)^\top + \Sigma_{w_I}(t) \end{aligned} \quad (3.20)$$

being $\text{Cov}(\hat{\epsilon}_I^i(t), v_{\hat{\epsilon}_I^i}(t)) = 0$, $\text{Cov}(\hat{\epsilon}_I^i(t), w_I(t)) = 0$, $\text{Cov}(v_{\hat{\epsilon}_I^i}(t), w_I(t)) = 0$ and $\text{Cov}(\hat{\epsilon}_I^i(t), \hat{\epsilon}_I^{\text{inei}}(t)) = 0$ owing to Assumption 7, where

$$W_{1i}(t) = A_{II}[(I - \lambda_I^i(t))\kappa_I^i(t) + \lambda_I^i(t)Z_I^i], \quad (3.21)$$

$$W_{2i}(t) = A_{II}(I - \lambda_I^i(t))\eta_I^i(t), \quad (3.22)$$

$$W_{3i}(t) = A_I^{\text{inei}}\omega_I^i(t), \quad (3.23)$$

$$\Gamma_{\hat{\epsilon}_I^i}(t) = \mathbb{E}[(\hat{\epsilon}_I^i(t) - \mathbb{E}\hat{\epsilon}_I^i(t))(\hat{\epsilon}_I^i(t) - \mathbb{E}\hat{\epsilon}_I^i(t))^\top], \quad (3.24)$$

$$\Gamma_{\hat{\epsilon}_I^{\text{inei}}}(t) = \mathbb{E}[(\hat{\epsilon}_I^{\text{inei}}(t) - \mathbb{E}\hat{\epsilon}_I^{\text{inei}}(t))(\hat{\epsilon}_I^{\text{inei}}(t) - \mathbb{E}\hat{\epsilon}_I^{\text{inei}}(t))^\top], \quad (3.25)$$

Z_I^i denotes a $m_I \times m_I N_I^i$ block matrix, where only one block is identity matrix I_{m_I} , locating in the position corresponding to the i -th index in the neighboring set $\mathcal{N}_I^i$, and all the other blocks equal to $\mathbf{0}$. $\Sigma_{v_{\hat{\epsilon}_I^i}}$ is the measurement noise covariance matrix, including correlations between neighboring sensors.

Since these two objectives are convex in the decision variables (namely the filtering weights and the prediction parameters), then we can reduce the multi-objective optimization problem to a scalar case and pose the following Pareto optimization problem:

$$\min_{\kappa_I^i(t), \eta_I^i(t), \lambda_I^i(t), \omega_I^i(t)} \text{tr} [\rho_I^i(B_I^i)^2 + (1 - \rho_I^i)V_I^i] \quad (3.26a)$$

$$\text{s.t. } (\kappa_I^i(t) + \eta_I^i(t))\mathbf{1}_{N_I^i m_I} = \mathbf{1}_{m_I} \quad (3.26b)$$

$$\omega_I^i(t)\mathbf{1}_{p_I^i} = \mathbf{1}_{m_I^i} \quad (3.26c)$$

$$\text{convergence conditions Eq.(3.13)} \vee (3.14) \vee (3.15), \quad (3.26d)$$

where $0 \leq \rho_I^i \leq 1$ is the Pareto parameter, $B_I^i = \mathbb{E}\hat{E}_I^i(t+1)$ is the prediction error mean given in (3.19), $V_I^i = \mathbb{E}[(\hat{E}_I^i(t+1) - \mathbb{E}\hat{E}_I^i(t+1))(\hat{E}_I^i(t+1) - \mathbb{E}\hat{E}_I^i(t+1))^\top]$ is the variance of the prediction error given in (3.20). The constant conditions of problem (3.26) are given by Assumption 9; the convergence conditions are derived in Proposition 3.2.1, depending on the norm of matrix A_I^{Eq} . Based on the derivation of expected value

and variance of the prediction error, the cost function of problem (3.26) is

$$\begin{aligned}
& \text{tr} [\rho_I^i (B_I^i)^2 + (1 - \rho_I^i) V_I^i] \\
&= \text{tr} [A_{II} [(I - \lambda_I^i(t)) \kappa_I^i(t) + \lambda_I^i(t) Z_I^i] [\rho_I^i \mathbb{E} \hat{\epsilon}_I^i(t) \mathbb{E} \hat{\epsilon}_I^i(t)^\top + (1 - \rho_I^i) \Gamma_{\hat{\epsilon}_I^i}(t)] \\
&\quad \times [Z_I^{i\top} \lambda_I^i(t)^\top + \kappa_I^i(t)^\top (I - \lambda_I^i(t))^\top] A_{II}^\top \\
&\quad + A_I^{\text{inei}} \omega_I^i(t) [\rho_I^i \mathbb{E} \hat{\epsilon}_I^{\text{inei}}(t) \mathbb{E} \hat{\epsilon}_I^{\text{inei}}(t)^\top + (1 - \rho_I^i) \Gamma_{\hat{\epsilon}_I^{\text{inei}}}(t)] \omega_I^{i\top}(t) A_I^{\text{inei}\top} \\
&\quad + A_{II} \omega_I^{i\top}(t) A_I^{\text{inei}\top} [(I - \lambda_I^i(t)) \kappa_I^i(t) + \lambda_I^i(t) Z_I^i] \rho_I^i \mathbb{E} \hat{\epsilon}_I^i(t) \mathbb{E} \hat{\epsilon}_I^{\text{inei}}(t)^\top \omega_I^{i\top}(t) A_I^{\text{inei}\top} \\
&\quad + A_I^{\text{inei}} \omega_I^i(t) \rho_I^i \mathbb{E} \hat{\epsilon}_I^{\text{inei}}(t) \mathbb{E} \hat{\epsilon}_I^i(t)^\top [Z_I^{i\top} \lambda_I^i(t)^\top + \kappa_I^i(t)^\top (I - \lambda_I^i(t))^\top] A_{II}^\top \\
&\quad + A_{II} (I - \lambda_I^i(t)) \eta_I^i(t) [(1 - \rho_I^i) \Sigma_{v_{\hat{\epsilon}_I^i}}] \eta_I^i(t)^\top (I - \lambda_I^i(t))^\top A_{II}^\top + (1 - \rho_I^i) \Sigma_{w_I}(t)].
\end{aligned}$$

Note that the second convergence conditions in (3.13), (3.14), (3.15) are non convex expressions depending on three sets of decision variables, which further complicates the derivation of the solution especially for what concerns the use of necessary optimality conditions. Hence, we now analyze the optimization problem (3.26) in order to obtain a tractable problem that can be solved in an analytical way.

3.3.1 Approximation Approach to the Optimization

We briefly analyze the convergence conditions (3.13), (3.14), (3.15), since the absolute value would make the problem more difficult to solve in an analytical way, for tractability reasons, we use the more restrictive conditions

$$\begin{cases} \kappa_I^{i\top}(t) \mathbf{1}_{m_I} \geq \mathbf{0}_{N_I^i m_I} \\ \kappa_I^i(t) \mathbf{1}_{N_I^i m_I} < \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I} \end{cases} \quad (3.27)$$

instead of $\sum_{l=1}^{n_I} |k_I^{i,lr}| < \frac{1}{\|A_I^{Eg}\|_\infty}, \forall l$.

$$\begin{cases} \omega_I^{i\top}(t) \mathbf{1}_{m_I^i} \geq \mathbf{0}_{p_I^i} \\ \omega_I^i(t) \mathbf{1}_{p_I^i} < \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I^i} \end{cases} \quad (3.28)$$

instead of $\sum_{l=1}^{n_J} |\omega_{I,J}^{i,lr'}| < \frac{1}{\|A_I^{Eg}\|_\infty}, \forall l, J \in \mathcal{O}_I$.

We note that if $\sum_{l=1}^{n_I} |k_I^{i,l^r}| < \frac{1}{\|A_I^{Eg}\|_\infty}$ holds, then

$$\frac{-\frac{1}{\|A_I^{Eg}\|_\infty} + \sum_{l=1}^{n_I} |k_I^{i,l^r}|}{1 + \sum_{l=1}^{n_I} |k_I^{i,l^r}|} < 0.$$

Moreover, if $1/\|A_I^{Eg}\|_\infty > 1$, then

$$\frac{\frac{1}{\|A_I^{Eg}\|_\infty} + \sum_{l=1}^{n_I} |k_I^{i,l^r}|}{1 + \sum_{l=1}^{n_I} |k_I^{i,l^r}|} > 1.$$

If $1/\|A_I^{Eg}\|_\infty < 1$, then

$$0 < 1 - \frac{1 - \frac{1}{\|A_I^{Eg}\|_\infty}}{1 - \sum_{l=1}^{n_I} |k_I^{i,l^r}|} < 1.$$

Hence, we get the following sufficient convergence conditions, implying conditions (3.13), (3.14), (3.15): the conditions (3.27), (3.28) with

◦ if $1/\|A_I^{Eg}\|_\infty > 1$, then

$$0 \leq \lambda_I^i(t) \leq I_{m_I} \quad (3.29)$$

◦ if $1/\|A_I^{Eg}\|_\infty = 1$, then

$$0 \leq \lambda_I^i(t) < I_{m_I} \quad (3.30)$$

◦ if $1/\|A_I^{Eg}\|_\infty < 1$, then

$$0 \leq \lambda_I^i(t) < I_{m_I} \quad (3.31)$$

In the following analysis, we formulate an approximate optimization problem. For the sake of the notation simplicity, the dependence on the time is removed. To highlight the dependencies on the decisional variables κ_I^i , η_I^i , λ_I^i and ω_I^i , we rewrite the Pareto optimization problem (3.26) by introducing the following terms which are data of the problem or can be computed on-line using samples:

$$M_I^i(\rho_I^i) = (1 - \rho_I^i)\Gamma_{\hat{\epsilon}_I^i},$$

$$M_I^{inei}(\rho_I^i) = (1 - \rho_I^i)\Gamma_{\hat{\epsilon}_I^{inei}},$$

$$R_I^i(\rho_I^i) = \rho_I^i(\Upsilon_{I1}^i \mathbb{E}\hat{\epsilon}_I^i + \Upsilon_{I2}^i \mathbb{E}\hat{\epsilon}_I^{inei})(\Upsilon_{I1}^i \mathbb{E}\hat{\epsilon}_I^i + \Upsilon_{I2}^i \mathbb{E}\hat{\epsilon}_I^{inei})^\top,$$

$$S_I^i(\rho_I^i) = (1 - \rho_I^i)\Sigma_{v_{\hat{\epsilon}_I^i}},$$

with Υ_{I1}^i and Υ_{I2}^i are both column block matrices, denoted by $\Upsilon_{I1}^i = \text{col}(I_{m_I N_I^i}, \mathbf{0}_{p_I^i} \otimes \mathbf{1}_{m_I N_I^i}^\top)$ and $\Upsilon_{I2}^i = \text{col}(\mathbf{0}_{p_I^i}^\top \otimes \mathbf{1}_{m_I N_I^i}, I_{p_I^i})$, respectively. We obtain

$$\begin{aligned} \min_{\kappa_I^i, \eta_I^i, \lambda_I^i, \omega_I^i} \quad & \text{tr}[A_I^{\text{inei}} \omega_I^i M_I^{\text{inei}}(\rho_I^i) \omega_I^{i\top} A_I^{\text{inei}\top} + A_{II}(I - \lambda_I^i) \eta_I^i S_I^i(\rho_I^i) \eta_I^{i\top} (I - \lambda_I^i)^\top A_{II}^\top \\ & + A_{II}[(I - \lambda_I^i) \kappa_I^i + \lambda_I^i Z_I^i] M_I^i(\rho_I^i) [(I - \lambda_I^i) \kappa_I^i + \lambda_I^i Z_I^i]^\top A_{II}^\top \\ & + (A_{II}[(I - \lambda_I^i) \kappa_I^i + \lambda_I^i Z_I^i] \Upsilon_{I1}^{i\top} + A_I^{\text{inei}} \omega_I^i \Upsilon_{I2}^{i\top}) R_I^i(\rho_I^i) \\ & \times (A_{II}[(I - \lambda_I^i) \kappa_I^i + \lambda_I^i Z_I^i] \Upsilon_{I1}^{i\top} + A_I^{\text{inei}} \omega_I^i \Upsilon_{I2}^{i\top})^\top + (1 - \rho_i) \Sigma_w] \end{aligned} \quad (3.32a)$$

$$\text{s.t. } (\kappa_I^i + \eta_I^i) \mathbf{1}_{N_I^i m_I} = \mathbf{1}_{m_I} \quad (3.32b)$$

$$\omega_I^i \mathbf{1}_{p_I^i} = \mathbf{1}_{m_I^i} \quad (3.32c)$$

$$\kappa_I^{i\top} \mathbf{1}_{m_I} \geq \mathbf{0}_{N_I^i m_I} \quad (3.32d)$$

$$\kappa_I^i \mathbf{1}_{N_I^i m_I} < \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I} \quad (3.32e)$$

$$\omega_I^{i\top} \mathbf{1}_{m_I^i} \geq \mathbf{0}_{p_I^i} \quad (3.32f)$$

$$\omega_I^i \mathbf{1}_{p_I^i} < \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I^i} \quad (3.32g)$$

$$\lambda_I^i \mathbf{1}_{m_I} + \varphi \mathbf{1}_{m_I} \leq \mathbf{1}_{m_I} \quad (3.32h)$$

$$\lambda_I^i \mathbf{1}_{m_I} \geq \mathbf{0}_{m_I}, \quad (3.32i)$$

being φ a small positive constant, where $\varphi = 0$ when $1/\|A_I^{Eg}\|_\infty > 1$ or $\varphi \neq 0$ when $1/\|A_I^{Eg}\|_\infty \geq 1$. Note that problem (3.26) and problem (3.32) have the same objective function, but problem (3.32) is constrained by more restrictive conditions. This gives an optimization problem which is tractable. In fact, problem (3.32) is convex since the objective function has a quadratic form and M_I^i , M_I^{inei} , R_I^i and S_I^i are positive definite matrices. Coherently, in order to get the analytical solutions, we use Lagrangian duality to solve the optimization problem. We have the following result. Let introduce the dual variables $\xi_{I1}^i(t)$, $\xi_{I2}^i(t)$, $\xi_{I4}^i(t)$ and $\nu_{I1}^i(t)$, which are $m_I \times 1$ vectors, the $m_I N_I^i \times 1$ vector $\xi_{I3}^i(t)$, the $p_I^i \times 1$ vector $\xi_{I5}^i(t)$, and the $m_I^i \times 1$ vectors $\xi_{I6}^i(t)$ and $\nu_{I2}^i(t)$. To simplify the notation, we omit the dependence on time in the following lemma and proof.

Lemma 3.3.1 *Consider optimization problem (3.32). Let ϵ_{I1}^i and ϵ_{I2}^i be positive constants. Then, the optimal values of the primal $(\kappa_I^i, \eta_I^i, \lambda_I^i, \omega_I^i)$ and dual $(\xi_{I1}^i, \xi_{I2}^i, \xi_{I3}^i,$*

$\xi_{I4}^i, \xi_{I5}^i, \xi_{I6}^i, \nu_{I1}^i$ and ν_{I2}^i) variables satisfy the following equations:

$$(\kappa_I^i + \eta_I^i) \mathbf{1}_{N_I^i m_I} - \mathbf{1}_{m_I} = \mathbf{0}_{m_I}, \quad (3.33)$$

$$\omega_I^i \mathbf{1}_{p_I^i} - \mathbf{1}_{m_I^i} = \mathbf{0}_{m_I^i}, \quad (3.34)$$

$$\lambda_I^i \mathbf{1}_{m_I} + \varphi \mathbf{1}_{m_I} - \mathbf{1}_{m_I} \leq \mathbf{0}_{m_I} \quad (3.35)$$

$$-\lambda_I^i \mathbf{1}_{m_I} \leq \mathbf{0}_{m_I}, \quad (3.36)$$

$$\xi_{I1}^{i\top} (\lambda_I^i \mathbf{1}_{m_I} + \varphi \mathbf{1}_{m_I} - \mathbf{1}_{m_I}) = 0 \quad \xi_{I1}^i \geq 0, \quad (3.37)$$

$$\xi_{I2}^{i\top} (-\lambda_I^i \mathbf{1}_{m_I}) = 0 \quad \xi_{I2}^i \geq 0, \quad (3.38)$$

$$-\kappa_I^{i\top} \mathbf{1}_{m_I} \leq \mathbf{0}_{N_I^i m_I^i}, \quad (3.39)$$

$$\kappa_I^i \mathbf{1}_{N_I^i m_I} - \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I} + \epsilon_{I1}^i \leq \mathbf{0}_{m_I}, \quad (3.40)$$

$$\xi_{I3}^{i\top} (-\kappa_I^{i\top} \mathbf{1}_{m_I}) = 0 \quad \xi_{I3}^i \geq 0, \quad (3.41)$$

$$\xi_{I4}^{i\top} (\kappa_I^i \mathbf{1}_{N_I^i m_I} - \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I} + \epsilon_{I1}^i) = 0 \quad \xi_{I4}^i \geq 0, \quad (3.42)$$

$$-\omega_I^{i\top} \mathbf{1}_{m_I^i} \leq \mathbf{0}_{p_I^i} \quad (3.43)$$

$$\omega_I^i \mathbf{1}_{p_I^i} - \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I^i} + \epsilon_{I2}^i \leq \mathbf{0}_{m_I^i} \quad (3.44)$$

$$\xi_{I5}^{i\top} (-\omega_I^{i\top} \mathbf{1}_{m_I^i}) = 0 \quad \xi_{I5}^i \geq 0, \quad (3.45)$$

$$\xi_{I6}^{i\top} (\omega_I^i \mathbf{1}_{p_I^i} - \frac{1}{\|A_I^{Eg}\|_\infty} \mathbf{1}_{m_I^i} + \epsilon_{I2}^i) = 0 \quad \xi_{I6}^i \geq 0, \quad (3.46)$$

$$2[N_I^i(\rho_I^i)((I - \lambda_I^i)\kappa_I^i + \lambda_I^i Z_I^i)^\top A_{II}^\top A_{II}(I - \lambda_I^i) + Q_I^i(\rho_I^i)\omega_I^{i\top} A_I^{\text{inei}\top} A_{II}(I - \lambda_I^i)] \circ I_\kappa + [\mathbf{1}_{N_I^i m_I} \nu_{I1}^{i\top} + \mathbf{1}_{N_I^i m_I} \xi_{I4}^{i\top} - \xi_{I3}^i \mathbf{1}_{m_I}^\top] \circ I_\kappa = 0, \quad (3.47)$$

$$2[S_I^i(\rho_I^i)\eta_I^{i\top}(I - \lambda_I^i)^\top A_{II}^\top A_{II}(I - \lambda_I^i)] \circ I_\kappa + [\mathbf{1}_{N_I^i m_I} \nu_{I1}^{i\top}] \circ I_\kappa = 0, \quad (3.48)$$

$$2[\eta_I^i S_I^i(\rho_I^i)\eta_I^{i\top} + (Z_I^i - \kappa_I^i)N_I^i(\rho_I^i)(Z_I^i - \kappa_I^i)^\top] \lambda_I^i A_{II}^\top A_{II} + [(Z_I^i - \kappa_I^i)N_I^i(\rho_I^i)\kappa_I^{i\top} - \eta_I^i S_I^i(\rho_I^i)\eta_I^i] A_{II}^\top A_{II} + (Z_I^i - \kappa_I^i)Q_I^i(\rho_I^i)\omega_I^{i\top} A_I^{\text{inei}\top} A_{II}] \circ I_\lambda + [\mathbf{1}_{m_I} \xi_{I1}^{i\top} - \mathbf{1}_{m_I} \xi_{I2}^{i\top}] \circ I_\lambda = 0, \quad (3.49)$$

$$2[Q_I^{i\top}(\rho_I^i)((I - \lambda_I^i)\kappa_I^i + \lambda_I^i Z_I^i)^\top A_{II}^\top A_I^{\text{inei}} + N_I^{\text{inei}}(\rho_I^i)\omega_I^{i\top} A_I^{\text{inei}\top} A_I^{\text{inei}}] \circ I_\omega + [\mathbf{1}_{p_i} \nu_{I2}^{i\top} + \mathbf{1}_{p_i} \xi_{I6}^{i\top} - \xi_{I5}^i \mathbf{1}_{m_i}^\top] \circ I_\omega = 0 \quad (3.50)$$

where the $N_I^i m_I \times m_I$ block matrix $I_\kappa = \mathbf{1} \otimes I$, the $m_I \times m_I$ matrix $I_\lambda = I_{m_I}$, the $p_i \times m_i$ block matrix $I_\omega = \bigoplus_{i=1}^N (\mathbf{1}_{m_i} \otimes I_{m_i})$, $N_I^i(\rho_I^i) = M_I^i(\rho_I^i) + \Upsilon_{I1}^{i\top} R_I^i(\rho_I^i) \Upsilon_{I1}^i$, $N_I^{\text{inei}}(\rho_I^i) = M_I^{\text{inei}}(\rho_I^i) + \Upsilon_{I2}^{i\top} R_I^i(\rho_I^i) \Upsilon_{I2}^i$ and $Q_I^i(\rho_I^i) = \Upsilon_{I1}^{i\top} R_I^i(\rho_I^i) \Upsilon_{I2}^i$.

Proof: Problem (3.32) is convex. Therefore KKT conditions are both necessary and sufficient for optimality. Since κ_I^i , η_I^i , λ_I^i and ω_I^i are composed by diagonal blocks, the (3.47)–(3.50) are obtained by using matrix derivatives for diagonal matrices [109]: given matrices A , $M \succeq \mathbf{0}$ and diagonal matrix λ , then

$$\frac{d \operatorname{tr}(A \lambda M \lambda^\top A^\top)}{d \lambda} = 2(M \lambda A^\top A) \circ I_\lambda.$$

As an extension to this result, given block-diagonal matrix κ , positive semidefinite matrix M and matrix A , then

$$\frac{d \operatorname{tr}(A \kappa M \kappa^\top A^\top)}{d \kappa} = 2(M \kappa^\top A^\top A) \circ I_\kappa.$$

Given block-diagonal matrix ω , positive semidefinite matrix M and matrix A , then

$$\frac{d \operatorname{tr}(A \omega M \omega^\top A^\top)}{d \omega} = 2(M \omega^\top A^\top A) \circ I_\omega.$$

□

In the following we analyze the derived KKT conditions to obtain the optimal weight values.

3.3.2 The Optimal Weights for the Approximated Problem

In the previous section we formulated optimization problem (3.26) to minimize mean and variance of the local prediction error and we obtained some optimality conditions for the equivalent problem (3.32) with more conservative constraints. Here we analyze the KKT conditions to derive the optimal values for the decisional variables $\kappa_I^i(t)$, $\eta_I^i(t)$, $\lambda_I^i(t)$ and $\omega_I^i(t)$. Define the $m_I N_I^i \times 1$ vector $\kappa_I^{i\text{vec}}(t) = \kappa_I^{i\top}(t) \mathbf{1}_{m_I}$, collecting all the diagonals of each block in $\kappa_I^i(t)$ on a column vector. Similarly, we denote $\eta_I^{i\text{vec}}(t) = \eta_I^{i\top}(t) \mathbf{1}_{m_I}$, the $p_I^i \times 1$ vector $\omega_I^{i\text{vec}}(t) = \omega_I^{i\top}(t) \mathbf{1}_{m_I^i}$ and the $m_I \times 1$ vector $\lambda_I^{i\text{vec}}(t) = \lambda_I^{i\top}(t) \mathbf{1}_{m_I}$. Given a vector V , we denote $\text{diag}(V)$ the diagonal matrix whose diagonal entries are the elements of V , and $V = \text{diag}^{-1}(\text{diag}(V))$.

In the following we will omit the dependence on time and on the ρ_I^i Pareto parameter for the sake of notation simplicity. We now provide an analytical solution for problem (3.32). It is worth noting that only these formulas have to be implemented in each node to compute at each time step the optimal weights.

Proposition 3.3.1 *A solution for problem (3.32), for each node i in subsystem I , is given by $\xi_{I1}^i = 0$, $\xi_{I2}^i = 0$, $\xi_{I3}^i = 0$, $\xi_{I4}^i = 0$, $\xi_{I5}^i = 0$ and $\xi_{I6}^i = 0$,*

$$\kappa_I^{i\text{vec}} = -P_7^{-1}(P_2 P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} + I_\kappa \nu_{I1}^i + P_2 P_6 Z_\lambda^2 + Z_\lambda^1), \quad (3.51a)$$

$$\eta_I^{i\text{vec}} = -P_3^{-1} I_\kappa \nu_{I1}^i, \quad (3.51b)$$

$$\omega_I^{i\text{vec}} = P_6 P_2^\top \kappa_I^{i\text{vec}} + P_6 Z_\lambda^2 + P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i}, \quad (3.51c)$$

$$\begin{aligned} \lambda_I^{i\text{vec}} = & ((A_{II}^\top A_{II})^\top \circ (\eta_I^i S_I^i \eta_I^{i\top} + (Z_I^i - \kappa_I^i) N_I^i (Z_I^i - \kappa_I^i)^\top))^{-1} \\ & \times \text{diag}^{-1}([(\kappa_I^i - Z_I^i) N_I^i \kappa_I^{i\top} + \eta_I^i S_I^i \eta_I^{i\top}] A_{II}^\top A_{II} \\ & + (\kappa_I^i - Z_I^i) Q_I^i \omega_I^{i\top} A_I^{\text{inei}\top} A_{II}), \end{aligned} \quad (3.51d)$$

with

$$Z_\lambda^1 = ([N_I^i Z_I^{i\top} \lambda_I^{i\top} A_{II}^\top A_{II} (I - \lambda_I^i)] \circ I_\kappa) \mathbf{1}_{m_I},$$

$$Z_\lambda^2 = ([Q_I^{i\top} Z_I^{i\top} \lambda_I^{i\top} A_{II}^\top A_I^{\text{inei}}] \circ I_\omega) \mathbf{1}_{m_I^i},$$

$$\nu_{I1}^i = -P_8^{-1} (\mathbf{1}_{m_I} + I_\kappa^\top P_7^{-1} [P_2 P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} + P_2 P_6 Z_\lambda^2 + Z_\lambda^1]),$$

$$P_1 = 2\mathbf{D}_I^\top \circ N_I^i, P_2 = 2\mathbf{D}_{IJ}^\top \circ Q_I^i, P_3 = 2\mathbf{D}_I^\top \circ S_I^i, P_4 = 2\mathbf{D}_J^\top \circ N_I^{\text{inei}}, P_5 = I_\omega^\top P_4^{-1} I_\omega,$$

$$P_6 = P_4^{-1} I_\omega P_5^{-1} I_\omega^\top P_4^{-1} - P_4^{-1}, P_7 = P_1 + P_2 P_6 P_2^\top, P_8 = I_\kappa^\top P_7^{-1} I_\kappa + I_\kappa^\top P_3^{-1} I_\kappa.$$

where $D_I = (I - \lambda_I^i)^\top A_{II}^\top A_{II} (I - \lambda_I^i)$, $D_{IJ} = A_I^{\text{inei}\top} A_{II} (I - \lambda_I^i)$, $D_J = A_I^{\text{inei}\top} A_I^{\text{inei}}$, $\mathbf{D}_I = \mathbf{1}_{N_I^i} \mathbf{1}_{N_I^i}^\top \otimes D_I$, $\mathbf{D}_{IJ} = \mathbf{1}_{N_I^i}^\top \otimes (I_\omega D_{IJ})$ and $\mathbf{D}_J = I_\omega D_J I_\omega^\top$, if κ_I^i , ω_I^i , and λ_I^i so computed in addition satisfy (3.35), (3.36), (3.39), (3.40), (3.43), (3.44).

Proof: Consider matrices $M \in \mathbb{R}^{n \times m}$, $A \in \mathbb{R}^{m \times n}$ and vector $x \in \mathbb{R}^m$, we have the preliminary result in [95]:

$$\text{diag}^{-1}(M \text{diag}(x)A) = (A^\top \circ M)x.$$

Given $\xi_{I1}^i = 0$ and $\xi_{I2}^i = 0$, the Equation (3.49) has

$$\begin{aligned} & (2[(\eta_I^i S_I^i \eta_I^{i\top} + (Z_I^i - \kappa_I^i) N_I^i (Z_I^i - \kappa_I^i)^\top) \lambda_I^i A_{II}^\top A_{II}] \circ I_\lambda) \mathbf{1}_{m_I} \\ &= \text{diag}^{-1}(2[\eta_I^i S_I^i \eta_I^{i\top} + (Z_I^i - \kappa_I^i) N_I^i (Z_I^i - \kappa_I^i)^\top] \lambda_I^i A_{II}^\top A_{II}) \\ &= 2(A_{II}^\top A_{II})^\top \circ (\eta_I^i S_I^i \eta_I^{i\top} + (Z_I^i - \kappa_I^i) N_I^i (Z_I^i - \kappa_I^i)^\top) \lambda_I^{i\text{vec}} \\ &= (2[(\kappa_I^i - Z_I^i) N_I^i \kappa_I^{i\top} + \eta_I^i S_I^i \eta_I^{i\top}] A_{II}^\top A_{II}] \circ I_\lambda) \mathbf{1}_{m_I} \\ & \quad + (2[(\kappa_I^i - Z_I^i) Q_I^i \omega_I^{i\top} A_I^{i\text{nei}\top} A_{II}] \circ I_\lambda) \mathbf{1}_{m_I}. \end{aligned}$$

We have

$$\begin{aligned} \lambda_I^{i\text{vec}} &= ((A_{II}^\top A_{II})^\top \circ (\eta_I^i S_I^i \eta_I^{i\top} + (Z_I^i - \kappa_I^i) N_I^i (Z_I^i - \kappa_I^i)^\top))^{-1} \\ & \quad \times \text{diag}^{-1}([(\kappa_I^i - Z_I^i) N_I^i \kappa_I^{i\top} + \eta_I^i S_I^i \eta_I^{i\top}] A_{II}^\top A_{II} + (\kappa_I^i - Z_I^i) Q_I^i \omega_I^{i\top} A_I^{i\text{nei}\top} A_{II}). \end{aligned}$$

In (3.50) with conditions $\xi_{I5}^i = 0$ and $\xi_{I6}^i = 0$,

$$(2[N_I^{i\text{nei}} \omega_I^{i\top} D_J] \circ I_\omega) \mathbf{1}_{m_I^i} = \text{diag}^{-1}(2N_I^{i\text{nei}} \text{diag}(\omega_I^{i\text{vec}}) \mathbf{D}_J),$$

$$(2[Q_I^{i\top} \kappa_I^{i\top} D_{IJ}] \circ I_\omega) \mathbf{1}_{m_I^i} = \text{diag}^{-1}(2Q_I^{i\top} \text{diag}(\kappa_I^{i\text{vec}}) \mathbf{D}_{IJ}^\top).$$

Then we have

$$(2\mathbf{D}_J^\top \circ N_I^{i\text{reg}}) \omega_I^{i\text{vec}} + (2\mathbf{D}_{IJ} \circ Q_I^{i\top}) \kappa_I^{i\text{vec}} = -I_\omega \nu_{I2}^i - Z_\lambda^2. \quad (3.52)$$

The equality condition (3.33) is equivalent to

$$I_\omega^\top \omega_I^{i\text{vec}} = \mathbf{1}_{m_I^i}. \quad (3.53)$$

Using equations (3.53), (3.52), we have

$$\nu_{I2}^i = -P_5^{-1} \mathbf{1}_{m_I^i} - P_5^{-1} I_\omega^\top P_4^{-1} (P_2^\top \kappa_I^{i\text{vec}} + Z_\lambda^2),$$

$$\omega_I^{i\text{vec}} = P_6 P_2^\top \kappa_I^{i\text{vec}} + P_6 Z_\lambda^2 + P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i}.$$

With $\xi_{I3}^i = 0$ and $\xi_{I4}^i = 0$, the Equation (3.47) has

$$(2[N_I^i \kappa_I^{i\top} D_I] \circ I_\kappa) \mathbf{1}_{m_I} = \text{diag}^{-1}(2N_I^i \text{diag}(\kappa_I^{i\text{vec}}) \mathbf{D}_I),$$

$$(2[Q_I^i \omega_I^{i\top} D_{IJ}] \circ I_\kappa) \mathbf{1}_{m_I} = \text{diag}^{-1}(2Q_I^i \text{diag}(\omega_I^{i\text{vec}}) \mathbf{D}_{IJ}).$$

Hence we get

$$(2\mathbf{D}_I^\top \circ N_I^i) \kappa_I^{i\text{vec}} + (2\mathbf{D}_{IJ}^\top \circ Q_I^i) \omega_I^{i\text{vec}} = -I_\kappa \nu_{I1}^i - Z_\lambda^1. \quad (3.54)$$

By substituting $\omega_I^{i\text{vec}}$ in (3.54), we get

$$\kappa_I^{i\text{vec}} = -P_7^{-1}(P_2 P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} + I_\kappa \nu_{I1}^i + P_2 P_6 Z_\lambda^2 + Z_\lambda^1).$$

It is found that the Equation (3.48) satisfies

$$(2[S_I^i \eta_I^{i\top} D_I] \circ I_\kappa) \mathbf{1}_{m_I} = \text{diag}^{-1}(2S_I^i \text{diag}(\eta_I^{i\text{vec}}) \mathbf{D}_I),$$

$$(2\mathbf{D}_I^\top \circ S_I^i) \eta_I^{i\text{vec}} = - \left([\mathbf{1}_{N_I^i m_I} \nu_{I1}^{i\top}] \circ I_\kappa \right) \mathbf{1}_{m_I} = -I_\kappa \nu_{I1}^i.$$

$$\eta_I^{i\text{vec}} = -P_3^{-1} I_\kappa \nu_{I1}^i.$$

Based on the equality condition (3.34),

$$I_\kappa^\top (\kappa_I^{i\text{vec}} + \eta_I^{i\text{vec}}) = \mathbf{1}_{m_I}.$$

The dual variable ν_{I1}^i can be calculated

$$\nu_{I1}^i = -P_8^{-1}(\mathbf{1}_{m_I} + I_\kappa^\top P_7^{-1}[P_2 P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} + P_2 P_6 Z_\lambda^2 + Z_\lambda^1]).$$

Therefore, we have a solution of κ_I^i , η_I^i and ω_I^i depending on λ_I^i and vice versa. This is an optimal solution for the Pareto optimization problem (3.32), if, once computed the solutions, in addition they satisfy the constraints (3.35), (3.36), (3.39), (3.40), (3.43), (3.44). $\square$

The solution given by the previous proposition, is the optimal solution of problem (3.32), the approximated convex version of problem (3.26). In order to use the result in Prop. 3.3.1, we propose a computational method shown in Algorithm 3 to find optimal

values κ_I^{i*} , η_I^{i*} , ω_I^{i*} and λ_I^{i*} . At each step, we verify that the obtained values satisfy the convergence conditions (3.35), (3.36), (3.39), (3.40), (3.43), (3.44). Define a small positive constant ϵ_I^i , the computation is stopped until difference between current value of λ_I^i and the updated one is smaller than ϵ_I^i . Hence, based on the proof in Proposition 2.3.2, it is proved that Algorithm 3 can converge to the optimal solutions κ_I^{i*} , η_I^{i*} , ω_I^{i*} and λ_I^{i*} at each time step.

Algorithm 3 Optimal weights computation

Set $\lambda_I^{i+} = I_{m_I}/2$, ϵ_I^i

repeat

$\lambda_I^i = \lambda_I^{i+}$

Calculate $Z_\lambda^1, Z_\lambda^2, \mathbf{D}_I, \mathbf{D}_{IJ}, \mathbf{D}_J$.

Calculate $P_1, P_2, \dots, P_8$.

$\nu_{I1}^i = -P_8^{-1}(\mathbf{1}_{m_I} + I_\kappa^\top P_7^{-1}[P_2 P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} + P_2 P_6 Z_\lambda^2 + Z_\lambda^1])$

$\kappa_I^{i\text{vec}} = \mathbf{P}_K \left(P_7^{-1}(P_2 P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} + I_\kappa \nu_{I1}^i + P_2 P_6 Z_\lambda^2 + Z_\lambda^1) \right)$

$\eta_I^{i\text{vec}} = -P_3^{-1} I_\kappa \nu_{I1}^i$

$\omega_I^{i\text{vec}} = \mathbf{P}_\Omega \left(P_6 P_2^\top \kappa_I^{i\text{vec}} + P_6 Z_\lambda^2 + P_4^{-1} I_\omega P_5^{-1} \mathbf{1}_{m_I^i} \right)$

$\lambda_I^{i\text{vec}} = \mathbf{P}_\Lambda (((A_{II}^\top A_{II})^\top \circ (\eta_I^i S_I^i \eta_I^{i\top} + (Z_I^i - \kappa_I^i) N_I^i (Z_I^i - \kappa_I^i)^\top))^{-1} \times \text{diag}^{-1}([(\kappa_I^i - Z_I^i) N_I^i \kappa_I^{i\top} + \eta_I^i S_I^i \eta_I^{i\top}] A_{II}^\top A_{II} + (\kappa_I^i - Z_I^i) Q_I^i \omega_I^{i\top} A_I^{i\text{nei}\top} A_{II}))$

$\lambda_I^{i+} = \text{diag}(\lambda_I^{i\text{vec}})$

until $|\lambda_I^i - \lambda_I^{i+}| \leq \epsilon_I^i$

return $\kappa_I^i = I_\kappa^\top \circ (\mathbf{1}_{m_I} \otimes \kappa_I^{i\text{vec}\top})$, $\eta_I^i = I_\kappa^\top \circ (\mathbf{1}_{m_I} \otimes \eta_I^{i\text{vec}\top})$, $\omega_I^i = I_\omega^\top \circ (\mathbf{1}_{m_I^i} \otimes \omega_I^{i\text{vec}\top})$ and $\lambda_I^i = \lambda_I^{i+}$

3.4 Simulation Results

In this section, simulation results are presented to illustrate the distributed estimator and to compare the performances with the centralized Kalman filter and a partition-based estimation approach. We consider the four-tanks system introduced in [67], the schematic diagram of the monitoring architecture for the four-tank process is illustrated in Fig. 3.1. The nonlinear dynamics of the process can be described by

$$\begin{aligned}
 \frac{dh_1}{dt} &= -\frac{a_1}{A_1} \sqrt{2gh_1} + \frac{a_3}{A_1} \sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1 \\
 \frac{dh_2}{dt} &= -\frac{a_2}{A_2} \sqrt{2gh_2} + \frac{a_4}{A_2} \sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2 \\
 \frac{dh_3}{dt} &= -\frac{a_3}{A_3} \sqrt{2gh_3} + \frac{(1 - \gamma_2) k_2}{A_3} v_2 \\
 \frac{dh_4}{dt} &= -\frac{a_4}{A_4} \sqrt{2gh_4} + \frac{(1 - \gamma_1) k_1}{A_4} v_1,
 \end{aligned} \tag{3.55}$$

where A_i is the cross-section of Tank i , a_i the cross-section of the outlet hole, h_i the liquid level; v_1 and v_2 are the process inputs. The parameter pair $(\gamma_1, \gamma_2) = (0.7, 0.6)$ specifies the allotment ratio of flow to Tank 1 and Tank 2, respectively. The gravity acceleration $g[\text{cm/s}^2]$ is 981.

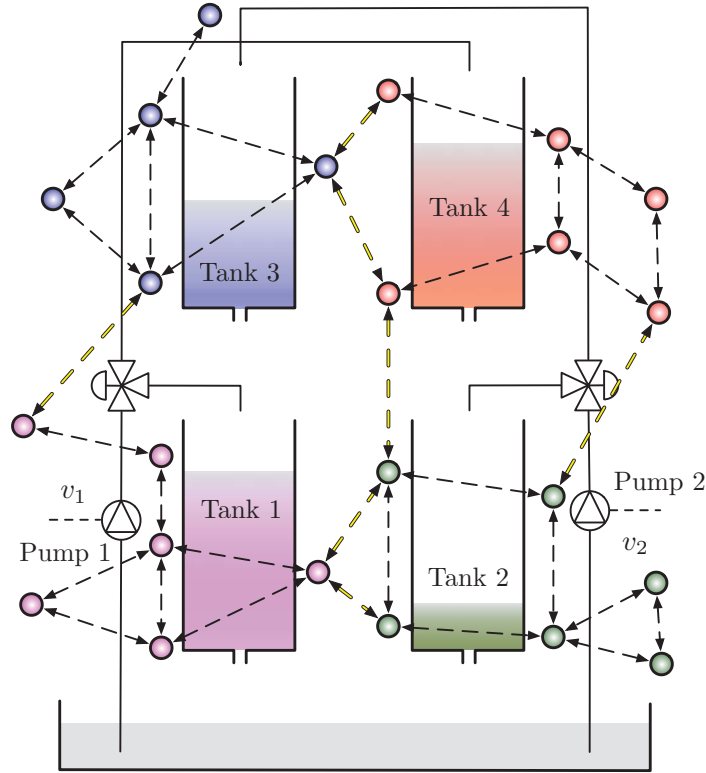


Fig. 3.1 The 4-tank system monitored by four sensor networks. Each tank is monitored by a set of sensors (having the same color as the corresponding tank level). The communication network is represented by dotted arrows (black arrows for local communication, yellow arrows for communication with sensors monitoring neighboring subsystems)

Table 3.1 : The parameters in the the four-tank process.

i	1	2	3	4
$A_i [\text{cm}^2]$	28	32	28	32
$a_i [\text{cm}^2]$	0.071	0.057	0.071	0.057
$h_i^0 [\text{cm}]$	12.4	12.7	1.8	1.4
$v_i^0 [\text{V}]$	3.00	3.00		
$k_i [\text{cm}^3/\text{V} \cdot \text{s}]$	3.33	3.35		

LQR control is applied to maintain the levels stable. The nonlinear system is linearized around the operation points $h_i^0, v_i^0, i \in (1, 2, 3, 4)$. The variables $x_i = h_i - h_i^0$ and $u_i = v_i - v_i^0$ respectively denote the states and inputs of the state-space model. Zero-order hold method is introduced to discretize the system with sample time $T_s = 1s$. By using the values given in Table 3.1, the discrete-time model of the process is given by

$$x(t+1) = \begin{bmatrix} 0.9842 & 0 & 0.0407 & 0 \\ 0 & 0.9890 & 0 & 0.0326 \\ 0 & 0 & 0.9590 & 0 \\ 0 & 0 & 0 & 0.9672 \end{bmatrix} x(t) + \begin{bmatrix} 0.0826 & 0.0010 \\ 0.0005 & 0.0625 \\ 0 & 0.0469 \\ 0.0307 & 0 \end{bmatrix} u(t) + \bar{w}, \quad (3.56)$$

where $\bar{w} = [-0.068; -0.059; 0.068; 0.059]^\top$ represents the linearization and discretization error found by averaging the difference between $x_i + h_i^0$ and real level h_i when the discrete-time model is working near operation points. Given $Q = I$ and $R = 2I$, the input $u(t)$ is generated so that state matrix A of the stable four-tank system is given by

$$A = \begin{bmatrix} 0.9430 & -0.0031 & 0.0262 & -0.0118 \\ -0.0036 & 0.9579 & 0.0121 & 0.0213 \\ -0.0025 & -0.0233 & 0.9500 & -0.0084 \\ -0.0153 & -0.0010 & -0.0053 & 0.9629 \end{bmatrix}.$$

The system can be decomposed in 4 interconnected subsystems and dynamics of each subsystem Σ_I is denoted by $A_{II} = A(I, I)$, $A_{IJ} = A(I, J)$ with $I, J = 1, 2, 3, 4$. We assume the modeling uncertainty ω_I has zero mean value and the system is initialized at $[4.81; 4.70; 1.0; 1.0]^\top$. The measurement noise v_i in each node i is an independent zero-mean Gaussian noise with $\sigma_{v_i}^2 = 0.0004I$ and $\text{SNR} = 33.9794$. We compare the following prediction methods:

E_{CKF} : Centralized Kalman predictor.

E_{DKF} : Distributed estimation approach as in [46], modified averaging the measurements from different sensors in each subsystem.

E_{p1} : The proposed distributed state predictor.

The centralized Kalman filter is considered as a benchmark.

A 40-nodes sensor network is considered. Each subsystem Σ_I is monitored by 10 sensors distributed randomly over a square area $\mathcal{A}_I$ with size 17. The graph $\mathcal{G}$ is acquired under the rule that two nodes are connected if their relative distance is less than $1.7\sqrt{17}$, if they belong to the same subsystem, and less than $3.4\sqrt{17}$ if

they belong to different subsystems. We compare the performance of the considered methods in different scenarios with different disturbance and measurement noises. The evaluated performance metric is the averaged mean square error of the predictions at all nodes. For each noise scenario, we run the experiment 80 times with a random network topology. Term MSE is denoted as the averaged mean square error value over 80 experiments over 40 nodes.

Fig. 3.2 and 3.3 shows MSE values of all methods with different measurement noise standard deviation. With regard to different standard deviation, it is presented that the proposed method always has better performance than the distributed estimation approach. Moreover, the performance of proposed method is closer to that of centralized Kalman filter when the disturbance standard deviation is lower.

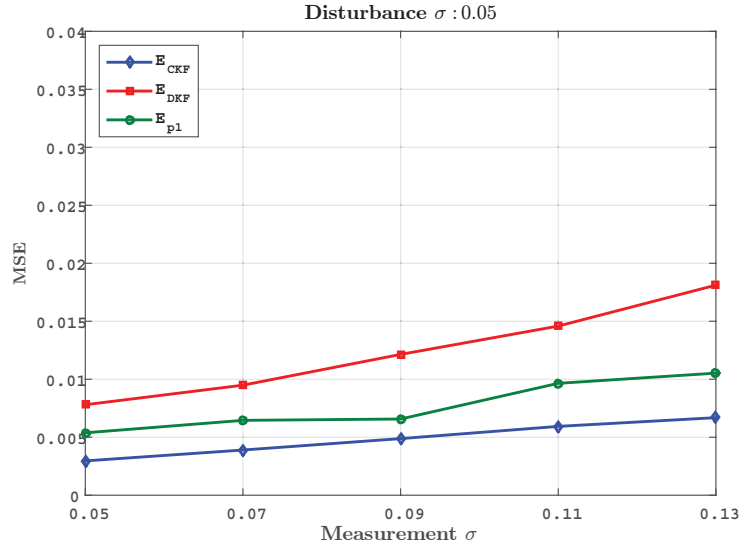


Fig. 3.2 MSE of 3 prediction methods with different measurement standard deviation. Disturbance standard deviation is 0.05.

3.5 Concluding Remarks

In this chapter, we propose a novel distributed estimation method using sensor networks for the monitoring of a distributed system. There are no assumptions on the topology of the communication network, apart from connectedness. And we do not require Gaussian distribution for the noises. Each subsystem is monitored by a sensor network, where each sensor node estimates the partitioned state of the distributed system in a multi-objective Pareto optimization framework. Hence, The filtering

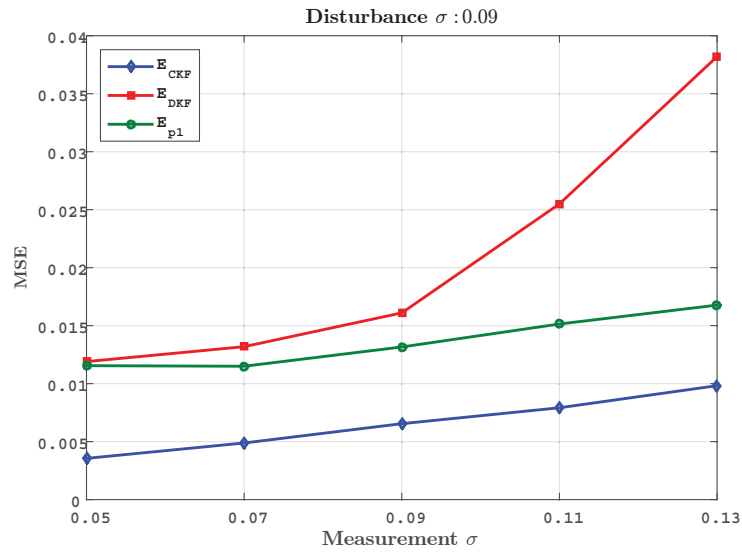


Fig. 3.3 MSE of 3 prediction methods with different measurement standard deviation. Disturbance standard deviation is 0.09.

weights and prediction parameters are jointly optimized at each time step to minimize both the mean and the variance of the prediction error.

Part II

FAULT DETECTION ALGORITHMS

Chapter 4

PROCESS FAULT DETECTION SCHEME USING SENSOR NETWORKS

In terms of the reliability of the monitoring architecture for a centralized or decentralized system, the architecture should be able to use a sensor network to detect process faults in the monitored system. Hence, in this chapter, we present a fault detection scheme for the nodes in a sensor network so that the fault detector at each sensor node can detect possible process faults occurred in the monitored system. The distributed state estimation method over a sensor network, proposed in Chapter 2, is applied to design the process fault detection scheme, where residual signals are defined and stochastic thresholds can be calculated according to a chosen maximum false alarm probability.

The advantage of using the distributed estimation method introduced in Chapter 2 is that it guarantees to minimize both mean and variance of the prediction error at each time step. Moreover, thanks to the way it is designed, the fault detector at each sensor node can compute mean and variance of the prediction error, thus allowing to obtain suitable thresholds for given residual signals with known confidence levels and guaranteed upper bound of the false alarm rate. Furthermore, since the distributed estimator in each sensor node does not require to know the distribution of process and measurements noises (only mean and variance), it can be applied to any noise distribution, although in the simulations of this chapter we consider the Gaussian case.

Unlike the results proposed in [16], we take into account the correlations existing between different state variables and process noise components, which is a more challenging framework and needs a much more complicated analysis. Each node in the sensor network acts as a local estimation and detection agent, where the local stochastic thresholds are updated at each time step. As the filtering step and model-based

prediction are coupled, the calculation of local stochastic thresholds needs information throughout the sensor network. Hence, a sophisticated data communication protocol is designed to avoid redundant information exchange among sensors in the network.

The chapter is organized as follows. Section 4.1 defines the process with stochastic uncertainty and potential faults. In Section 4.2, the distributed estimation method is briefly presented. Then the distributed fault detection algorithm is introduced in Section 4.3. Simulation results are given in Section 4.4. Finally, some concluding remarks in Section 4.5.

4.1 Problem Formulation

The process we want to monitor is modeled as

$$x(t+1) = Ax(t) + \bar{w}(t) + \tilde{w}(t) + \beta(t-t_0)\phi(x(t), t), \quad (4.1)$$

where t is the discrete time, $x \in \mathbb{R}^m$ denotes the state vector, $\bar{w} \in \mathbb{R}^m$ represents any known time-varying bias including possibly some known process disturbances and modeling errors, due to linearization as example, while $\tilde{w}(t)$ models uncertainties and process noises, and $\beta(t-t_0)\phi(x(t), t)$ represents the dynamics of a process fault occurring at the unknown time t_0 , with $\beta(t-t_0)$ being the time profile of the fault function.

Assumption 10 *We assume that $\tilde{w}(t)$ is a zero-mean process with covariance matrix $\Sigma_w(t)$.*

A sensor network, made of n nodes, monitors system (4.1) by taking measurements, so that, for each sensor node i , with $i = 1, \dots, n$, we have:

$$y_i(t) = x(t) + v_i(t), \quad (4.2)$$

where $y_i \in \mathbb{R}^m$ denotes the measurements vector taken by sensor i and $v_i \in \mathbb{R}^m$ is a zero-mean measurement noise, with Σ_{v_i} as covariance matrix. Each node of the sensor network observes the entire state vector and exchanges measurements and predictions with the neighboring nodes, which is slightly different from the algorithm proposed in Chapter 2, where each node can observe part of the entire state vector. However, both algorithms are implemented in a two-step approach to minimize the mean and variance of the prediction error in a Pareto optimization framework.

Assumption 11 *The measurement noise v_i and process noise $\tilde{w}$ are not correlated, for each $i = 1, \dots, n$.*

Assumption 12 *We assume that graph $\mathcal{G}$ is connected.*

Each node communicates with neighborings by means of a communication network modeled by an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where $\mathcal{N}_i = \{j \in \mathcal{S} : (j, i) \in \mathcal{E}\} \cup \{i\}$ is the set of neighbors of node $i \in \mathcal{S}$ plus the node itself. For well-posedness, the following assumption is clearly stated.

In the next section, we will briefly present the distributed state estimation method developed based on the algorithm introduced in Chapter 2.

4.2 Distributed State Estimation

We propose that each node i of the sensor network implements a two-step estimator: a *filtering* step and a *prediction* step. First, by communicating with neighboring nodes, the estimator filters the measurements noise by computing a linear combination of its own and neighboring measurements and predictions:

$$\bar{x}_i(t) = \sum_{j \in \mathcal{N}_i} k_{i,j}(t) \hat{x}_j(t) + \sum_{j \in \mathcal{N}_i} h_{i,j}(t) y_j(t), \quad (4.3)$$

where $k_{i,j}(t)$ and $h_{i,j}(t)$ are some properly designed time-varying filtering weights. The objective of this first step is for each sensor node to reduce its own measurement uncertainty, without the use of centralized coordination.

After the filtering step, each sensor node implements a classical model-based prediction, using filtered estimates (obtained from the previous step), instead of the observed measurements. The one-step-ahead prediction, considering the healthy dynamics, is

$$\hat{x}_i(t+1) = A\bar{x}_i(t) + \bar{w}(t) + \lambda'_i(t)(\hat{x}_i(t) - \bar{x}_i(t)), \quad (4.4)$$

where $\lambda'_i(t) \in \mathbb{R}^{m \times m}$ is a time-varying matrix collecting the properly designed prediction parameters. The term $\lambda'_i(t)(\hat{x}_i(t) - \bar{x}_i(t))$ represents a local correction of the previous prediction errors, taking into account the system dynamics. For tractability, we denote the filter parameter matrix $\lambda'_i(t) = A\lambda_i(t)$, where $\lambda_i(t)$ is the time varying diagonal matrix that we design. Filtering weights $k_{i,j}(t)$ and $h_{i,j}(t)$ and prediction parameters $\lambda_i(t)$ are computed at each time step by each node in a distributed way so to jointly

minimize the mean and variance of the prediction error $\hat{E}_i(t) = \hat{x}_i(t) - x(t)$, for all $i = 1, \dots, n$, in a Pareto optimization framework. For the sake of simplicity, we can rewrite (4.3) and (4.4) as

$$\begin{aligned}\bar{x}_i(t) &= \kappa_i(t)\hat{x}_i^{\text{reg}}(t) + \eta_i(t)y_i^{\text{reg}}(t) \\ \hat{x}_i(t+1) &= A\bar{x}_i(t) + \bar{w}(t) + A\lambda_i(t)(\hat{x}_i(t) - \bar{x}_i(t)),\end{aligned}\tag{4.5}$$

where $\hat{x}_i^{\text{reg}}$ and y_i^{reg} are two column vectors collecting the prediction vectors and the measurements vectors respectively available at node i , ordered according to their indexes: $\hat{x}_i^{\text{reg}} = (\hat{x}_{i_1}^\top, \dots, \hat{x}_{i_{N_i}}^\top)^\top$, $i_1 < \dots < i_{N_i}$, and $y_i^{\text{reg}} = (y_{i_1}^\top, \dots, y_{i_{N_i}}^\top)^\top$, $i_1 < \dots < i_{N_i}$, with $N_i = |\mathcal{N}_i|$ denoting the number of neighbors of i plus i itself. Moreover, $\kappa_i(t)$ and $\eta_i(t)$ are the time varying block matrices $\in \mathbb{R}^{m \times N_i m}$ collecting the diagonal matrices $k_{i,j}(t)$ and $h_{i,j}(t)$ with $j \in \mathcal{N}_i$, respectively.

Since the convergence property of the global prediction error mean in this chapter and in Chapter 2 is the same, we can directly use the conditions in Proposition 2.2.1 as the convergence conditions of the optimization problem in this chapter. Moreover, the convergence conditions are non convex expressions, a convex approximation of Pareto optimization problem in this chapter is given:

$$\min_{\kappa_i, \eta_i, \lambda_i} \text{tr} [\rho_i B_i^2 + (1 - \rho_i) V_i] \tag{4.6a}$$

$$\text{s.t. } (\kappa_i(t) + \eta_i(t)) \mathbf{1}_{m_{N_i}} = \mathbf{1}_m, \tag{4.6b}$$

$$\kappa_i^\top(t) \mathbf{1}_m \geq \mathbf{0}_{m_i} \tag{4.6c}$$

$$\kappa_i(t) \mathbf{1}_{m_i} < \frac{1}{\|A\|_\infty} \mathbf{1}_m \tag{4.6d}$$

$$\lambda_i(t) \mathbf{1}_m + \varphi \mathbf{1}_m \leq \mathbf{1}_m, \tag{4.6e}$$

$$\lambda_i(t) \mathbf{1}_m \geq \mathbf{0}_m, \tag{4.6f}$$

where $m_{N_i} = mN_i$, $0 \leq \rho_i \leq 1$ is the Pareto parameter, $B_i = \mathbb{E}\hat{E}_i(t+1)$ is the expected value of prediction error, $V_i = \mathbb{E}[(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))^\top]$ is the variance of prediction error. The equality constraints is the local conditions given in Assumption 5. φ a small positive constant, where the last two constraints are equivalent to $0 \leq \lambda_i \leq I$, if $\varphi = 0$, or $0 \leq \lambda_i < I$ if $\varphi \neq 0$, depending on matrix A norm, since $\lambda_i(t)$ is a diagonal matrix. The constraints represent some local conditions on the decisional variables to guarantee the asymptotic convergence of the prediction error mean.

Given a vector V , we denote $\text{diag}(V)$ the diagonal matrix whose diagonal entries are the elements of V , and $V = \text{diag}^{-1}(\text{diag}(V))$. We define the $m_{N_i} \times 1$ vector $\kappa_i^{\text{vec}}(t) = \kappa_i^\top(t) \mathbf{1}_m$, collecting all the decisional variables of the diagonals of $\kappa_i(t)$ on a column vector. Similarly, $\eta_i^{\text{vec}}(t) = \eta_i^\top(t) \mathbf{1}_m$, $\lambda_i^{\text{vec}}(t) = \lambda_i^\top(t) \mathbf{1}_m$ and $\nu_i^E(t) = [\mathbf{1}_{m_{N_i}} \nu_i^\top(t)] \circ (\mathbf{1} \otimes I) \mathbf{1}_m$, collecting N_i times $m \times 1$ vector ν_i , which is a Lagrangian dual variable. By analyzing KKT conditions, the optimal weights of the problem (4.6) are derived as

$$\kappa_i^{\text{vec}}(t) = - (2M_i(t, \rho_i) \circ \mathbf{D}_i(t))^{-1} \nu_i^E(t), \quad (4.7a)$$

$$\eta_i^{\text{vec}}(t) = - (2S_i(t, \rho_i) \circ \mathbf{D}_i(t))^{-1} \nu_i^E(t), \quad (4.7b)$$

$$\begin{aligned} \nu_i(t) = & -2(I_{m \times m_{N_i}}((M_i(t, \rho_i) \circ \mathbf{D}_i(t))^{-1} \\ & + (S_i(t, \rho_i) \circ \mathbf{D}_i(t))^{-1}) I_{m \times m_{N_i}}^\top)^{-1} \mathbf{1}_m, \end{aligned} \quad (4.7c)$$

$$\begin{aligned} \lambda_i^{\text{vec}}(t) = & [2(M_i^{\text{loc}}(t, \rho_i) + \kappa_i(t) M_i(t, \rho_i) \kappa_i^\top(t) \\ & + \eta_i(t) S_i(t, \rho_i) \eta_i^\top(t)) \circ (A^\top A)]^{-1} \\ & (\text{diag}^{-1}(2(\kappa_i(t) M_i(t, \rho_i) \kappa_i^\top(t) + \eta_i(t) S_i(t, \rho_i) \eta_i^\top(t)) A^\top A)), \end{aligned} \quad (4.7d)$$

with $\mathbf{D}_i(t) = \mathbf{1}_{N_i} \mathbf{1}_{N_i}^\top \otimes D_i(t)$ and $D_i(t) := (I - \lambda_i(t))^\top A^\top A (I - \lambda_i(t))$, where

$$M_i(t, \rho_i) = \rho_i \mathbb{E} \hat{\epsilon}_i(t) \mathbb{E} \hat{\epsilon}_i^\top(t) + (1 - \rho_i) \Gamma_{\hat{\epsilon}_i}(t),$$

$$M_i(t, \rho_i)^{\text{loc}} = \rho_i \mathbb{E} \hat{E}_i(t) \mathbb{E} \hat{E}_i^\top(t) + (1 - \rho_i) \Gamma_{\hat{E}_i}(t),$$

$$S_i(t, \rho_i) = (1 - \rho_i) \Sigma_{v_{\hat{\epsilon}_i}}(t),$$

where $\hat{\epsilon}_i$ collects the prediction error vectors $\hat{E}_j$, $j \in \mathcal{N}_i$, available at node i and $\Sigma_{v_{\hat{\epsilon}_i}}(t)$ is a block matrix collecting on the diagonal the measurement noise covariance matrices $\Sigma_{v_j}(t)$, $j \in \mathcal{N}_i$, related to the measurements communicated at node i , both ordered according to the indexes, and

$$\Gamma_{\hat{\epsilon}_i}(t) = \mathbb{E}[(\hat{\epsilon}_i(t) - \mathbb{E} \hat{\epsilon}_i(t))(\hat{\epsilon}_i(t) - \mathbb{E} \hat{\epsilon}_i(t))^\top]$$

The Pareto parameter ρ_i is locally set.

In Chapter 2, a computational method is proposed so that each node can compute locally at each time step the filtering and prediction weights. Moreover, it is possible to locally compute the estimates $\hat{m}_{\epsilon_i}$ and $\hat{\Gamma}_i$ respectively of the expected value $\mathbb{E} \hat{\epsilon}_i$ and variance $\Gamma_{\hat{\epsilon}_i}$ using at each time step the new samples of local measurements y_i^{reg} and state predictions $\hat{x}_i^{\text{reg}}$ (see Algorithm 4 for details about the estimation from samples).

4.3 Distributed Fault Detection

In the previous section, we have introduced the one-step ahead prediction method. In this section, a distributed fault detection scheme is designed by analyzing the dynamics of a stochastic residual. Hence, some stochastic thresholds are computed for the generated residual.

4.3.1 Local Residual Analysis

In this subsection we introduce the residual signal and we analyze its stochastic features. The local residual is defined as $r_i(t) = \bar{x}_i(t) - \hat{x}_i(t)$. Its dynamics can be described as:

$$\begin{aligned} r_i(t+1) &= \bar{x}_i(t+1) - \hat{x}_i(t+1) \\ &= \bar{x}_i(t+1) - x(t+1) + x(t+1) - \hat{x}_i(t+1) \\ &= \lambda_i(t)Ar_i(t) + \bar{e}_i(t+1) - A\bar{e}_i(t) + \tilde{w}(t), \end{aligned} \quad (4.8)$$

where $\bar{e}_i(t) = \bar{x}_i(t) - x(t)$ is the local filtering error. We define

$$\chi_i(t) = \bar{e}_i(t) - A\bar{e}_i(t-1) + \tilde{w}(t-1),$$

which is the stochastic part of the residual $r_i(t)$ and it is a stochastic variable at time t . We then analyze the stochastic features of the residual. The mean of χ_i is given by

$$\mu_{\chi_i}(t) = \mathbb{E}\chi_i(t) = \mathbb{E}\bar{e}_i(t) - A\mathbb{E}\bar{e}_i(t-1). \quad (4.9)$$

Since, using the definition, the filtering error $\bar{e}_i(t)$ can be computed as

$$\bar{e}_i(t) = \kappa_i(t)\hat{e}_i(t) + \eta_i(t)v_{\epsilon_i}(t), \quad (4.10)$$

where v_{ϵ_i} collects the measurement noise vectors v_j , $j \in \mathcal{N}_i$, then we can derive the needed expression of the expected value for the local estimation

$$\mathbb{E}\bar{e}_i(t) = \kappa_i(t)\mathbb{E}\hat{e}_i(t), \quad (4.11)$$

where the following is needed

$$\begin{aligned} \mathbb{E}\hat{E}_i(t+1) &= A(I - \lambda_i(t))\kappa_i(t)\mathbb{E}\hat{e}_i(t) + A\lambda_i(t)\mathbb{E}\hat{E}_i(t) \\ &= A[(I - \lambda_i(t))\kappa_i(t) + \lambda_i^0(t)]\mathbb{E}\hat{e}_i(t), \end{aligned} \quad (4.12)$$

where λ_i^0 is a block matrix $\in \mathbb{R}^{m \times N_i m}$ with only one-zero component λ_i , following the same index as that of node i in the set $\mathcal{N}_i$.

We now analyze the variance of the stochastic variable χ_i . It is worth noting that the local filtering error $\bar{e}_i(t)$ and process noise $\tilde{w}(t-1)$ are not independent as the error $\bar{e}_i(t)$ in (4.10) depends on the prediction errors $\hat{e}_i(t)$, which depend on the process noise $\tilde{w}(t-1)$, as for each node i the prediction error dynamics can be described by

$$\hat{E}_i(t+1) = A(I - \lambda_i(t))\kappa_i(t)\hat{e}_i(t) + A\lambda_i(t)\hat{E}_i(t) + A(I - \lambda_i(t))\eta_i(t)v_{\epsilon_i}(t) - \tilde{w}(t). \quad (4.13)$$

Moreover, the autocorrelation function of the stochastic process $\bar{e}_i$ at lag 1 is not null. Given Assumption 11, the variance of $\chi_i(t)$ is denoted as

$$\begin{aligned} \sigma_{\chi_i}^2(t) = & \Gamma_{\bar{e}_i}(t) + A\Gamma_{\bar{e}_i}(t-1)A^\top + \Sigma_w(t-1) - Cov(\bar{e}_i(t), \bar{e}_i(t-1))A^\top \\ & - (\mathbf{1}_{N_i}^\top \otimes \Sigma_w(t-1))\kappa_i^\top - ACov(\bar{e}_i(t-1), \bar{e}_i(t)) - \kappa_i(\mathbf{1}_{N_i} \otimes \Sigma_w(t-1)), \end{aligned} \quad (4.14)$$

where $\Gamma_{\bar{e}_i}(t) = \mathbb{E}[(\bar{e}_i(t) - \mathbb{E}\bar{e}_i(t))(\bar{e}_i(t) - \mathbb{E}\bar{e}_i(t))^\top]$ denotes the variance of local filtering error and can be computed as

$$\Gamma_{\bar{e}_i}(t) = \kappa_i(t)\Gamma_{\hat{e}_i}(t)\kappa_i(t)^\top + \eta_i(t)\Sigma_{v_{\epsilon_i}}\eta_i(t)^\top, \quad (4.15)$$

where $\Sigma_{v_{\epsilon_i}}$ is the measurement noise covariance matrix. The local autocorrelation matrix is defined as

$$Cov(\bar{e}_i(t), \bar{e}_i(t-1)) = \mathbb{E}[(\bar{e}_i(t) - \mathbb{E}\bar{e}_i(t))(\bar{e}_i(t-1) - \mathbb{E}\bar{e}_i(t-1))^\top]. \quad (4.16)$$

In the next subsection, we explain how to compute it locally.

4.3.2 Local Autocorrelation Matrix

$Cov(\bar{e}_i(t), \bar{e}_i(t-1))$ is the i^{th} diagonal component of global covariance matrix $Cov(\bar{e}(t), \bar{e}(t-1))$. To facilitate the analysis, the distributed estimation formulation can be expressed in a vector form that includes all the nodes in the sensor network:

$$\begin{aligned} \bar{x}(t) &= K(t)\hat{x}(t) + H(t)y(t) \\ \hat{x}(t+1) &= A_E\bar{x}(t) + \bar{w}_E(t) + \lambda'(t)(\hat{x}(t) - \bar{x}(t)) \\ y(t) &= x_E(t) + v(t), \end{aligned} \quad (4.17)$$

where $\bar{x}$, $\hat{x}$, y and v are column vectors $\in \mathbb{R}^{mn \times 1}$ collecting the local vectors $\bar{x}_i$, $\hat{x}_i$, y_i and v_i , respectively, with $i = 1, \dots, n$; A_E is a diagonal block matrix, with each block on the diagonal equal to A ; $\bar{w}_E$ and x_E are both column vectors of appropriate length, where the process disturbance vector $\bar{w}$ and x are repeated n times, respectively. $K(t)$ and $H(t)$ are block matrices, where each $(i, j)^{\text{th}}$ block, with $i, j = 1, \dots, n$, collects on the diagonal the weights with which the j^{th} sensor weights measurements or estimates components developed by the j^{th} node, if they are neighbors; the block is a null matrix if i and j are not neighboring nodes. $\lambda'(t) = A_E \lambda(t)$ is a diagonal block matrix, collecting the matrices λ'_i , $i = 1, \dots, n$, on the diagonal.

Therefore, the global filtering error $\bar{e}(t) = \bar{x}(t) - x_E(t)$ and the global prediction error $\hat{E}(t) = \hat{x}(t) - x_E(t)$ can be computed as follows:

$$\begin{aligned} \bar{e}(t) &= K(t)\hat{x}(t) + H(t)y(t) - x_E(t) \\ &= K(t)\hat{x}(t) + H(t)(x_E(t) + v(t)) - x_E(t) + K(t)x_E(t) - K(t)x_E(t) \\ &= K(t)\hat{E}(t) + H(t)v(t), \end{aligned} \quad (4.18)$$

and

$$\begin{aligned} \hat{E}(t+1) &= A_E \bar{x}(t) + \bar{w}_E(t) + \lambda'(t)(\hat{x}(t) - \bar{x}(t)) - A_E x_E(t) - \bar{w}_E(t) - \tilde{w}_E(t) \\ &= A_E [(I - \lambda(t))K(t) + \lambda(t)] \hat{E}(t) + A_E (I - \lambda(t))H(t)v(t) - \tilde{w}_E(t), \end{aligned} \quad (4.19)$$

with $\tilde{w}_E$ is a column vector of appropriate length, where the process disturbance vector $\tilde{w}$ is repeated n times, and the first constraint of problem (4.6) is used. Let us compute mean and variance of the global filtering and prediction errors. The expected value expressions are

$$\mathbb{E}\bar{e}(t) = K(t)\mathbb{E}\hat{E}(t), \quad (4.20)$$

$$\mathbb{E}\hat{E}(t+1) = A_E [(I - \lambda(t))K(t) + \lambda(t)] \mathbb{E}\hat{E}(t). \quad (4.21)$$

Given the local variance defined in (4.13)

$$\begin{aligned} &\mathbb{E}[(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))(\hat{E}_i(t+1) - \mathbb{E}\hat{E}_i(t+1))^{\top}] \\ &= W_i^{(1)}(t)\Gamma_{\hat{\epsilon}_i}(t)W_i^{(1)}(t)^{\top} + W_i^{(2)}(t)\Sigma_{v_{\hat{\epsilon}_i}}W_i^{(2)}(t)^{\top} + \Sigma_w(t), \end{aligned} \quad (4.22)$$

where

$$W_i^{(1)}(t) = A[(I - \lambda_i(t))\kappa_i(t) + \lambda_i^0(t)],$$

$$W_i^{(2)}(t) = A(I - \lambda_i(t))\eta_i(t).$$

The global variance matrix can be computed as

$$\begin{aligned}\Gamma^{\hat{E}}(t+1) &\triangleq \mathbb{E} \left[(\hat{E}(t+1) - \mathbb{E}\hat{E}(t+1))(\hat{E}(t+1) - \mathbb{E}\hat{E}(t+1))^{\top} \right] \\ &= W^{(1)}(t)\Gamma^{\hat{E}}(t)W^{(1)}(t)^{\top} + W^{(2)}(t)\Sigma^v(t)W^{(2)}(t)^{\top} + \Sigma_{w_E}(t),\end{aligned}\quad (4.23)$$

with

$$\begin{aligned}W^{(1)}(t) &= A_E [(I - \lambda(t))K(t) + \lambda(t)], \\ W^{(2)}(t) &= A_E (I - \lambda(t))H(t), \\ \Sigma^v(t) &= \mathbb{E} [(v(t) - \mathbb{E}v(t))(v(t) - \mathbb{E}v(t))^{\top}], \\ \Sigma_{w_E}(t) &= \mathbb{E} [\tilde{w}_E(t) - \mathbb{E}\tilde{w}_E(t)]^2,\end{aligned}$$

where $W^{(1)}(t)$, $W^{(2)}(t)$, $\Sigma^v(t)$ and $\Sigma_{w_E}(t)$ are $n \times n$ block matrices, being $\text{Cov}(\hat{E}(t), v(t)) = 0$, $\text{Cov}(\hat{E}(t), \tilde{w}_E(t)) = 0$ and $\text{Cov}(v(t), \tilde{w}_E(t)) = 0$, using Assumption 11. Given (4.18) – (4.21), we have that

$$\begin{aligned}\bar{e}(t) - \mathbb{E}\bar{e}(t) &= K(t)(\hat{E}(t) - \mathbb{E}\hat{E}(t)) + H(t)v(t), \\ \hat{E}(t) - \mathbb{E}\hat{E}(t) &= W^{(1)}(t-1)(\hat{E}(t-1) - \mathbb{E}\hat{E}(t-1)) + W^{(2)}(t-1)v(t-1) - \tilde{w}_E(t-1).\end{aligned}$$

Hence, the global covariance matrix, collecting on the diagonal the matrices (4.16) needed to compute (4.14), is described by

$$\text{Cov}(\bar{e}(t), \bar{e}(t-1)) = K(t)W^{(1)}(t-1)\Gamma^{\hat{E}}(t-1)K(t-1)^{\top} + K(t)W^{(2)}(t-1)\Sigma^v(t-1)H(t-1)^{\top}.$$

We now explain how to compute the local component (4.16) from the previous expression. Let us define two block matrices $P(t) = W^{(1)}(t)\Gamma^{\hat{E}}(t)$ and $Q(t) = W^{(2)}(t)\Sigma^v(t)$. The $(i, j)^{\text{th}}$ block of each matrix can be derived by

$$\begin{aligned}P_{i,j}(t) &= \sum_{l \in \mathcal{S}} W_{i,l}^{(1)}(t)\Gamma_{l,j}^{\hat{E}}(t) \\ &= \sum_{l \in \mathcal{N}_i} W_{i,l}^{(1)}(t)\Gamma_{l,j}^{\hat{E}}(t)\end{aligned}\quad (4.24)$$

and

$$\begin{aligned}Q_{i,j}(t) &= \sum_{l \in \mathcal{S}} W_{i,l}^{(2)}(t)\Sigma_{l,j}^v(t) \\ &= W_{i,j}^{(2)}(t)\Sigma_{jj}^v(t)\end{aligned}\quad (4.25)$$

where the terms $W^{(1)}$ and $W^{(2)}$ are derived by

$$\begin{cases} W_{i,l}^{(1)}(t) = A((I - \lambda_i(t))k_{i,l}(t) + \lambda_i(t)) & i = l \\ W_{i,l}^{(1)}(t) = A((I - \lambda_i(t))k_{i,l}(t)) & \text{otherwise} \end{cases} \quad (4.26)$$

and

$$W_{i,l}^{(2)}(t) = A(I - \lambda_i(t))h_{i,l}(t), \quad (4.27)$$

where λ_i is the i^{th} diagonal block of matrix λ , $k_{i,l}$ and $h_{i,l}$ are the $(i, l)^{\text{th}}$ blocks of matrix K and H , respectively. Finally, $\Gamma_{l,j}^{\hat{E}}$ and $\Sigma_{l,j}^v$ are the $(l, j)^{\text{th}}$ blocks of matrix $\Gamma^{\hat{E}}$ and matrix Σ^v , respectively. Based on the definition in (4.23), $\Gamma_{l,j}^{\hat{E}}(t)$ is given by

$$\Gamma_{l,j}^{\hat{E}}(t) = \mathbb{E} \left[(\hat{E}_l(t) - \mathbb{E}\hat{E}_l(t))(\hat{E}_j(t) - \mathbb{E}\hat{E}_j(t))^{\top} \right].$$

Therefore, the local covariance $Cov(\bar{e}_i(t), \bar{e}_i(t-1))$ is derived by

$$\begin{aligned} Cov(\bar{e}_i(t), \bar{e}_i(t-1)) &= \sum_{j' \in \mathcal{N}_i} \sum_{j \in \mathcal{N}_i} k_{i,j}(t) P_{j,j'}(t-1) k_{i,j'}(t-1)^{\top} \\ &\quad + \sum_{j' \in \mathcal{N}_i} \sum_{j \in \mathcal{N}_i} k_{i,j}(t) Q_{j,j'}(t-1) h_{i,j'}(t-1)^{\top}. \end{aligned} \quad (4.28)$$

The local covariance $Cov(\bar{e}_i(t), \bar{e}_i(t-1))$ in (4.14) should be available at time t in node i . The communication protocol is assumed that at each time step the information can not be sent to the neighbors of neighbors.

The terms $k_{i,j}(t)$, $k_{i,j'}(t-1)^{\top}$ and $h_{i,j'}(t-1)^{\top}$ are available in node i as they are components of the locally computed block matrix $\kappa_i(t)$, $\kappa_i(t-1)$ and $h_i(t-1)$, respectively. The block matrices $P_{j,j'}(t-1)$ and $Q_{j,j'}(t-1)$ should be locally calculated in node j and communicated to node i at latest at time t . Consider matrix $Q_{j,j'}(t-1)$ defined in (4.25), since $W_{j,j'}^{(2)}(t-1)$ is a null matrix when $j' \notin \mathcal{N}_j$, it is just necessary for node j to import the measurement covariance matrix $\Sigma_{j',j'}^v(t-1)$ from the neighboring nodes $j' \in \{\mathcal{N}_j \cap \mathcal{N}_i\}$ for computing matrix $Q_{j,j'}(t-1)$. On the other hand, for what concerns $P_{j,j'}(t-1)$ defined in (4.24), matrix $W_{j,l}^{(1)}(t-1)$, $l \in \mathcal{N}_j$, can be calculated locally by node j using the local filtering weights and prediction parameters.

However, matrix $\Gamma_{l,j'}^{\hat{E}}(t-1)$, $l \in \mathcal{N}_j$, does not belong to the local variance matrix $\Gamma_{\hat{e}_j}(t-1)$ when $j' \in \mathcal{N}_i \setminus \{\mathcal{N}_i \cap \mathcal{N}_j\}$, as the prediction $\hat{x}_{j'}(t-1)$ and measurement $y_{j'}(t-1)$ are not available at time $t-1$ in node j . Fortunately, the prediction $\hat{x}_{j'}(t-1)$ and measurement $y_{j'}(t-1)$ are available in node i at time $t-1$ because $j' \in \mathcal{N}_i$. In order to estimate matrix $\Gamma_{l,j'}^{\hat{E}}(t-1)$ at time t in node j , the prediction $\hat{x}_{j'}(t-1)$ and

measurement $y_{j'}(t-1)$ should be sent from node i to node j at time t . Hence, the matrix $P_{j,j'}(t-1)$ is available in node j at time t for computing the local covariance.

Then we define the following set, denoting the neighbours of neighbours of node j :

$$\mathcal{N}_j^{\text{all}} = \bigcup_{l \in \mathcal{N}_j} \mathcal{N}_l.$$

New column vectors $\hat{x}_j^{\text{all}}$ and y_j^{all} are introduced, which collect predictions and measurements respectively available at node j , ordered by the indexes: $\hat{x}_j^{\text{all}} = (\hat{x}_{j_1}^\top, \dots, \hat{x}_{j_{N_j^{\text{all}}}}^\top)^\top$ and $y_j^{\text{all}} = (y_{j_1}^\top, \dots, y_{j_{N_j^{\text{all}}}}^\top)^\top$, $j_1 < \dots < j_{N_j^{\text{all}}}$, with $N_j^{\text{all}} = |\mathcal{N}_j^{\text{all}}|$.

It is worth noting that neither we need a new communication network, nor we require a faster communication rate. Hence, at node j the vector $\hat{\epsilon}_j^{\text{all}} = \hat{x}_j^{\text{all}} - y_j^{\text{all}}$ can be computed collecting the prediction error vectors $\hat{E}_{j'}$ with $j' \in \mathcal{N}_j^{\text{all}}$. Then, the local estimates of the expected value $\mathbb{E}\hat{\epsilon}_j^{\text{all}}$ and of the variance $\Gamma_{\hat{\epsilon}_j^{\text{all}}}$, denoted by $\hat{m}_{\epsilon_j}^{\text{all}}$ and $\hat{\Gamma}_j^{\text{all}}$ respectively, can be calculated at node j based on Algorithm 4 (see Subsection 4.3.4 for a description). The estimate of matrix $\Gamma_{\hat{E}_{j'}}^{\hat{E}}(t-1)$, needed at node j to compute $P_{j,j'}(t-1)$ for (4.28), is acquired from $\hat{\Gamma}_j^{\text{all}}(t-1)$.

Remark 4.3.1 In order to have $\hat{x}_j^{\text{all}}$ and y_j^{all} available in node j we owe to a two-step collecting approach: predictions and measurements related to any node $j' \in \mathcal{N}_j^{\text{all}} \setminus \mathcal{N}_j$ can be collected from node $i \in \mathcal{N}_{j'} \cap \mathcal{N}_j$. Therefore, at each time step t all the variables can be computed just using locally communicated information.

4.3.3 The Fault Detection Thresholds

In this section, the objective is to design some time-varying thresholds that bound the residual when the system is healthy with a guaranteed probability. The distribution of the stochastic variable χ_i is unknown. However, the bounds for the residual signal can be designed based on the Chebyshev's inequality:

$$\Pr \left(|\chi_i^{(k)} - \mu_{\chi_i}^{(k)}| \leq \alpha \sigma_{\chi_i}^{(k)} \right) \geq 1 - \frac{1}{\alpha^2}, \quad (4.29)$$

where $\chi_i^{(k)}$ and $\mu_{\chi_i}^{(k)}$ are the k^{th} component of vectors χ_i and μ_{χ_i} , defined in (4.9), respectively; $\sigma_{\chi_i}^{(k)}$ is the square root of the k -th diagonal component of matrix $\sigma_{\chi_i}^2$ in (4.14). It is possible to choose the parameter α so that the inequalities given below hold with a certain probability.

$$\mu_{\chi_i}^{(k)} - \alpha \sigma_{\chi_i}^{(k)} \leq \chi_i^{(k)} \leq \mu_{\chi_i}^{(k)} + \alpha \sigma_{\chi_i}^{(k)}. \quad (4.30)$$

Given the dynamics of the residual in (4.8), we introduce time-varying upper and lower thresholds for the residual $r_i(t+1)$:

$$\begin{aligned}\bar{r}_i^{(k)+}(t+1) &= \lambda_i(t)\bar{r}_i^{(k)+}(t) + \mu_{\chi_i}^{(k)}(t+1) + \alpha\sigma_{\chi_i}^{(k)}(t+1) \\ \bar{r}_i^{(k)-}(t+1) &= \lambda_i(t)\bar{r}_i^{(k)-}(t) + \mu_{\chi_i}^{(k)}(t+1) - \alpha\sigma_{\chi_i}^{(k)}(t+1),\end{aligned}\quad (4.31)$$

where $\bar{r}_i^{(k)+}$ and $\bar{r}_i^{(k)-}$ can be initialized at time $t = 1$ by choosing proper values satisfying the inequality:

$$\bar{r}_i^{(k)-}(1) \leq r_i^{(k)}(1) \leq \bar{r}_i^{(k)+}(1).$$

Hence, there exists a ‘tube’ between the two thresholds where the residual lies (in a probabilistic sense) in healthy conditions:

$$\bar{r}_i^{(k)-}(t) \leq r_i^{(k)}(t) \leq \bar{r}_i^{(k)+}(t). \quad (4.32)$$

Thanks to the way they are designed, the proposed thresholds guarantee that the false-alarms rate is lower than $\frac{1}{\alpha^2}$. According to (4.32), a fault is *detected* by a node $i \in \{1, \dots, n\}$ when at least one component $r_i^{(k)}$ of the residual $r_i^{(k)}$ exceeds the corresponding thresholds $\bar{r}_i^{(k)+}$ or $\bar{r}_i^{(k)-}$. The α parameter can be set according to the wanted false alarms probability, taking into account the trade-off existing between robustness and sensitivity of the thresholds. Less conservative thresholds can be designed if we assume the noises $\tilde{w}$ and v_i are normally distributed.

4.3.4 The Algorithm

The proposed distributed fault detection method is illustrated by Algorithm 4.

In this chapter the Pareto parameter ρ is set to 0.5. In Chapter 2, it is explained how to alternatively compute it at each step in order to reach the knee-point of the Pareto optimization. $\hat{m}_{\bar{e}_i}$ denotes the local estimate of $\mathbb{E}\bar{e}_i$. Roman numbers indicate the steps where information communication is required. Points I and III require the communication from neighboring nodes in $\mathcal{N}_i$ to node i of the covariance matrices Σ_{v_j} with $j \in \mathcal{N}_i^{\text{all}}$ and $j \in \mathcal{N}_i$, respectively. Point II indicates the collection of predictions and measurements components from neighboring nodes. Step IV can be implemented by acquiring additive predictions from $\hat{x}_j^{\text{reg}}$ and measurements y_j^{reg} from neighboring nodes $j \in \mathcal{N}_i$. Points V and VI show the block matrices that are sent to neighboring nodes in $\mathcal{N}_i$. Therefore, all the information transmission occurs by means of the communication network described by $\mathcal{G}$.

Algorithm 4 Process Fault detection algorithm for node i

$t := 0, N_i := |\mathcal{N}_i|, N_i^{\text{all}} := |\mathcal{N}_i^{\text{all}}|, \rho_i := 0.5$
 $\hat{m}_{\epsilon_i}^{\text{all}}(0) := \mathbf{0}$
 $\hat{\Gamma}_i^{\text{all}}(0) := \Sigma_{v_{\epsilon_i}^{\text{all}}}$ ▷ I
 Extract $\hat{m}_{\epsilon_i}(0)$ from $\hat{m}_{\epsilon_i}^{\text{all}}(0)$
 Extract $\hat{\Gamma}_i(0)$ from $\hat{\Gamma}_i^{\text{all}}(0)$
 $\bar{x}_i(0) := y_i(0)$
 $\hat{x}_i(1) := A\bar{x}_i(0)$
repeat
 $t := t + 1$
 Collect $\mathbf{x}_i^{\text{reg}}(t) := (\hat{x}_{i_1}(t), \dots, \hat{x}_{i_{N_i}}(t))^{\top}$ and $\mathbf{y}_i^{\text{reg}}(t) := (y_{i_1}(t), \dots, y_{i_{N_i}}(t))^{\top}$ where $\{i_1, \dots, i_{N_i}\} \in \mathcal{N}_i$
 ▷ II
 $\hat{\epsilon}_i(t) := \frac{\mathbf{x}_i^{\text{reg}}}{1+\nu} - \frac{\nu \mathbf{1}^{\top} \mathbf{x}_i^{\text{reg}} + (1+\nu) \mathbf{1}^{\top} \mathbf{y}_i^{\text{reg}}}{N_i(1+2\nu)(1+\nu)} \mathbf{1}$
 $\hat{m}_{\epsilon_i}(t) := \frac{t-1}{t} \hat{m}_{\epsilon_i}(t-1) + \frac{1}{t} \hat{\epsilon}_i(t)$
 $\hat{\Gamma}_i(t) := \frac{t-1}{t} \hat{\Gamma}_i(t-1) + \frac{1}{t} (\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))(\hat{\epsilon}_i(t) - \hat{m}_{\epsilon_i}(t))^{\top}$
 $S_i := (1 - \rho_i) \Sigma_{v_{\epsilon_i}}(t)$ ▷ III
 $M_i := \rho_i (\hat{m}_{\epsilon_i}(t) \hat{m}_{\epsilon_i}^{\top}(t)) + (1 - \rho_i) \hat{\Gamma}_i(t)$
 Compute $\kappa_i(t), \eta_i(t), \lambda_i(t)$ as in (4.7)
 $\bar{x}_i(t) = \kappa_i(t) \mathbf{x}_i^{\text{reg}}(t) + \eta_i(t) \mathbf{y}_i^{\text{reg}}(t)$
 $\hat{x}_i(t+1) = A\bar{x}_i(t) + \bar{w}(t) + \lambda_i(t)(A\hat{x}_i(t) - A\bar{x}_i(t))$
 $r_i(t) := \bar{x}_i(t) - \hat{x}_i(t)$
 $\hat{m}_{\bar{\epsilon}_i}(t) := \kappa_i(t) \hat{m}_{\epsilon_i}(t)$
 $\Gamma_{\bar{\epsilon}_i}(t) := \kappa_i(t) \hat{\Gamma}_i(t) \kappa_i(t)^{\top} + \eta_i(t) \Sigma_{v_{\bar{\epsilon}_i}} \eta_i(t)^{\top}$
 if $t = 1$ **then**
 $\bar{r}_i^+(1) := r_i(1) + 2|r_i(1)|$
 $\bar{r}_i^-(1) := r_i(1) - 2|r_i(1)|$
 else
 Collect $\mathbf{x}_i^{\text{all}}(t-1) := (\hat{x}_{i_1}(t-1), \dots, \hat{x}_{i_{N_i^{\text{all}}}}(t-1))^{\top}$ and $\mathbf{y}_i^{\text{all}}(t-1) := (y_{i_1}(t-1), \dots, y_{i_{N_i^{\text{all}}}}(t-1))^{\top}$
 where $\{i_1, \dots, i_{N_i^{\text{all}}}\} \in \mathcal{N}_i^{\text{all}}$ ▷ IV
 $\hat{\epsilon}_i^{\text{all}}(t-1) := \frac{\mathbf{x}_i^{\text{all}}(t-1)}{1+\nu} - \frac{\nu \mathbf{1}^{\top} \mathbf{x}_i^{\text{all}}(t-1) + (1+\nu) \mathbf{1}^{\top} \mathbf{y}_i^{\text{all}}(t-1)}{N_i^{\text{all}}(1+2\nu)(1+\nu)} \mathbf{1}$
 $\hat{m}_{\epsilon_i}^{\text{all}}(t-1) := \frac{t-2}{t-1} \hat{m}_{\epsilon_i}^{\text{all}}(t-2) + \frac{1}{t-1} \hat{\epsilon}_i^{\text{all}}(t-1)$
 $\hat{\Gamma}_i^{\text{all}}(t-1) := \frac{t-2}{t-1} \hat{\Gamma}_i^{\text{all}}(t-2) + \frac{1}{t-1} (\hat{\epsilon}_i^{\text{all}}(t-1) - \hat{m}_{\epsilon_i}^{\text{all}}(t-1))(\hat{\epsilon}_i^{\text{all}}(t-1) - \hat{m}_{\epsilon_i}^{\text{all}}(t-1))^{\top}$
 Compute $P_{i,i'}(t-1)$ with $i' \in \mathcal{N}_i^{\text{all}}$ as in (4.24) and send to $j \in \mathcal{N}_i$ ▷ V
 Compute $Q_{i,i'}(t-1)$ with $i' \in \mathcal{N}_i^{\text{all}}$ as in (4.25) and send to $j \in \mathcal{N}_i$ ▷ VI
 Collect $P_{j,j'}(t-1), Q_{j,j'}(t-1)$ with $j, j' \in \mathcal{N}_i$
 Compute $\sigma_{\chi_i}^2(t)$ as in (4.14)
 $\mu_{\chi_i}(t) = \hat{m}_{\bar{\epsilon}_i}(t) - A\hat{m}_{\bar{\epsilon}_i}(t-1)$
 $\bar{r}_i^+(t) := \lambda_i(t-1) \bar{r}_i^+(t-1) + \mu_{\chi_i}(t) + \alpha \sigma_{\chi_i}(t)$
 $\bar{r}_i^-(t) := \lambda_i(t-1) \bar{r}_i^-(t-1) + \mu_{\chi_i}(t) - \alpha \sigma_{\chi_i}(t)$
 end if
until $\exists k : r_i^{(k)}(t) > r_i^{(k)+}(t) \vee r_i^{(k)}(t) < r_i^{(k)-}(t)$
 A fault is detected at time t .

4.4 Simulation Results

In this section, some simulation results are presented to illustrate the fault detection performance. We consider a 15-node sensor network monitoring a four-tank system introduced in [67] and Chapter 3. The schematic diagram of the four-tank process is illustrated in Fig. 4.1. The nonlinear dynamics of the process can be described by

$$\begin{aligned}
 \frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1 \\
 \frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2 \\
 \frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2 \\
 \frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1,
 \end{aligned} \tag{4.33}$$

where A_i is the cross-section of Tank i , a_i the cross-section of the outlet hole, h_i the liquid level; v_1 and v_2 are the process inputs. The parameter pair $(\gamma_1, \gamma_2) = (0.7, 0.6)$ specifies the allotment ratio of flow to Tank 1 and Tank 2, respectively. The gravity acceleration $g[\text{cm/s}^2]$ is 981. The sensors in the sensor network can measure all the liquid levels in the process. We linearize and discretize it so to obtain the linear system described by the matrix:

$$A = \begin{bmatrix} 0.9430 & -0.0031 & 0.0262 & -0.0118 \\ -0.0036 & 0.9579 & 0.0121 & 0.0213 \\ -0.0025 & -0.0233 & 0.9500 & -0.0084 \\ -0.0153 & -0.0010 & -0.0053 & 0.9629 \end{bmatrix}.$$

The process noise $\tilde{w}(t)$ is a zero-mean Gaussian noise with $\sigma_{\tilde{w}}^2 = 0.0001\text{I}$. The initial state is $x(0) = [4.81; 4.70; 1.0; 1.0]^\top$. The measurement noise v_i in each node i is an independent zero-mean Gaussian noise with $\sigma_{v_i}^2 = 0.0004\text{I}$ and $\text{SNR} = 33.9794$. The 15-nodes sensor network is obtained by distributing the nodes randomly over a squared area of size $n/2$. The graph $\mathcal{G}$ is acquired under the rule that two nodes are connected if their relative distance is less than $1.7\sqrt{n}$. The parameter α is chosen equal to 1. Assuming that a leakage occurs at the bottom of Tank 3 at time $t_0 = 150\text{s}$, which

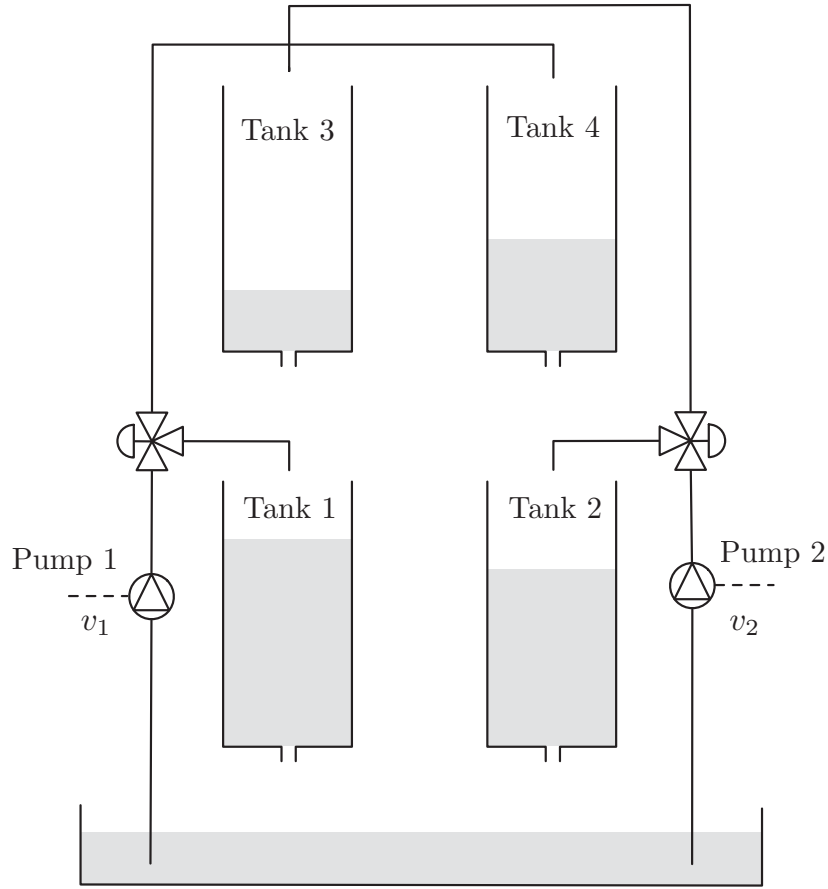


Fig. 4.1 The schematic diagram of the four-tank process.

means a_3 in (4.33) increases of a $\Delta a_3 = 0.034\text{cm}^2$. Then the fault matrix Φ is given by

$$\Phi = \begin{bmatrix} 0 & 0 & 0.0188 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0190 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Fig. 4.2 and Fig. 4.3 show the estimation performance and residuals and thresholds in the state components affected by the fault. We can see that the fault is detected at 153s in the third and first state components respectively. It only takes around 3 seconds to detect the process fault in each node. The sensitivity with respect to the fault magnitude is then analyzed. We run 101 experiments, where Δa_3 is assigned evenly every 0.02 in the interval $[0.1, 2.1]\text{mm}^2$. The detection results with the measurement noise such that the Signal over Noise Ratio $\text{SNR} = 33.9794$ are shown in Fig. 4.4.

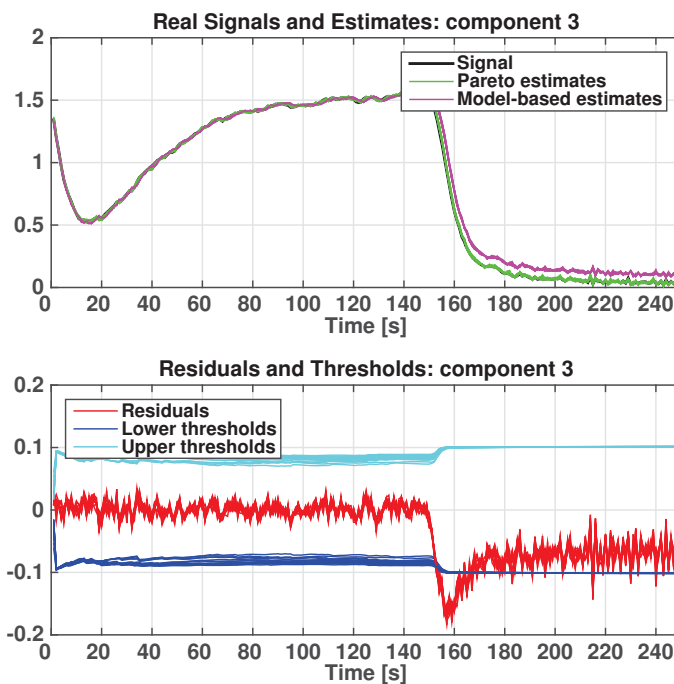


Fig. 4.2 Estimates, Residuals and Thresholds for state component 3 .

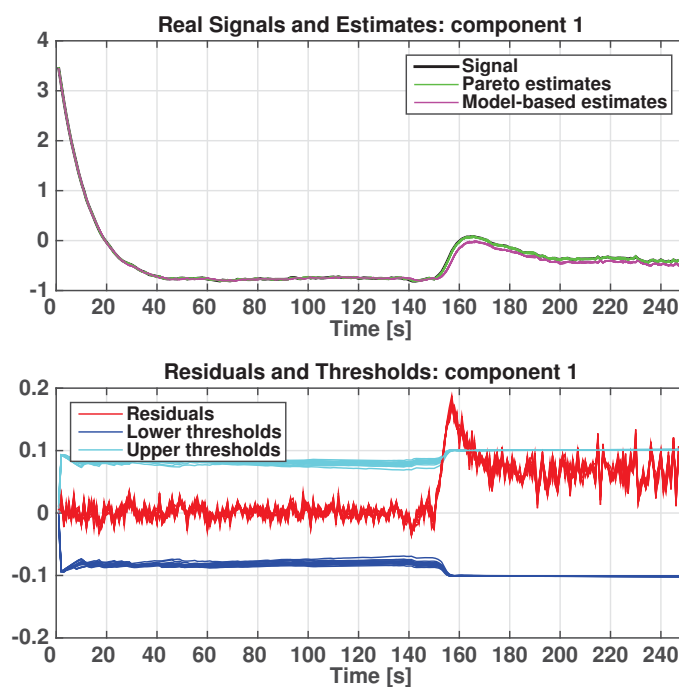


Fig. 4.3 Estimates, Residuals and Thresholds for state component 1 .

We can see that the detection time decreases when the cross-section of the leakage increases, reaching then a minimum time of detection. The false-alarms rate is denoted by

$$\text{FAR} = \frac{\text{number of false alarms}}{101 \times n \times m} \times 100\%.$$

The detection results with measurement noise so that $\text{SNR} = 30.4576$ and three different α are illustrated in Fig. 4.5, Fig. 4.6 , Fig. 4.7. Since the measurements are affected by a higher noise, we can see in this scenario the presence of false-alarms and FAR is smaller while fault detection time becomes longer when the chosen α is larger.

Table 4.1 : FAR with various α and SNR .

α	0.8	1.0	1.2
SNR = 33.98	0%	0%	0%
SNR = 32.04	0.173%	0.08%	0.03%
SNR = 30.46	2.343%	0.776%	0.693%

Given three different SNR and α , the false alarm rate is presented in Table 4.1, which verifies the conclusion above. This confirms that the α parameter has to be chosen according to the noise features of the process and the wanted false-alarms rate. However, when the α parameter is too large, the thresholds would be far more conservative, which leads to misdetections. The misdetection rate is denoted by

$$\text{MIS} = \frac{\text{number of misdetections}}{101 \times n \times m} \times 100\%.$$

Table 4.2 : MIS with various α and SNR .

α	2.0	2.2	2.4
SNR = 33.98	76.44%	98.27%	99.86%
SNR = 32.04	73.51%	97.95%	99.61%
SNR = 30.46	69.46%	95.37%	98.92%

Given three different SNR and α , the misdetection rate is presented in Table 4.2, which shows that the larger α parameter is chosen, the more misdetections would occur.

4.5 Concluding Remarks

In this chapter we have extended the distributed estimation method proposed in Chapter 2 to present a process fault detection algorithm in the monitoring architecture for a centralized or decentralized system. The distributed estimation method in the proposed fault detection algorithm can compute mean and variance of the prediction error, thus allowing to obtain suitable thresholds for some suitable residual signals with known confidence levels and guaranteed upper bound of false alarm rate. And the algorithm needs no assumption on the distribution of process and measurements noises (only mean and variance). A sophisticated data communication protocol is designed to avoid redundant information exchange among sensors in the network when calculating the local stochastic thresholds. Apart from the process faults in monitored systems, the monitoring architecture should also be able to detect any sensor faults occurred in sensor network. In the next chapter, the centralized, decentralized, and distributed sensor fault detection schemes will be introduced for the sensor nodes in the monitoring architecture for a distributed system.

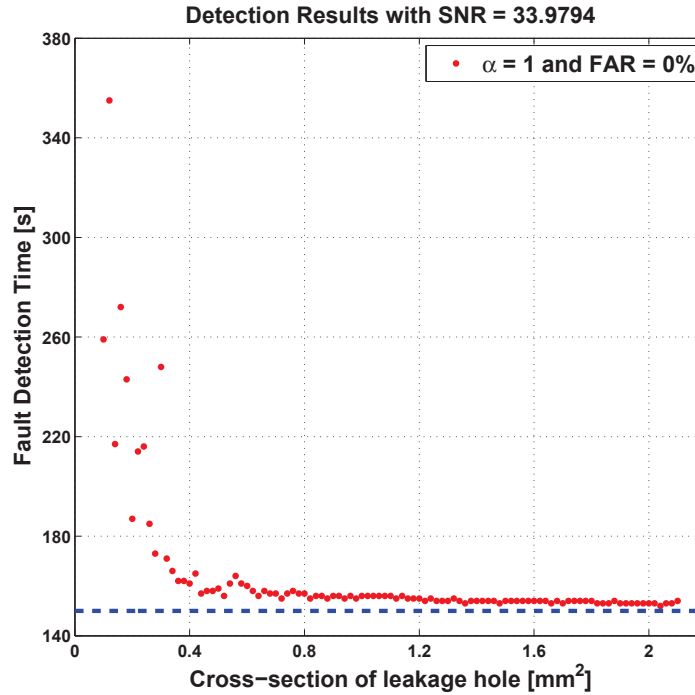


Fig. 4.4 Detection time with various leakage size

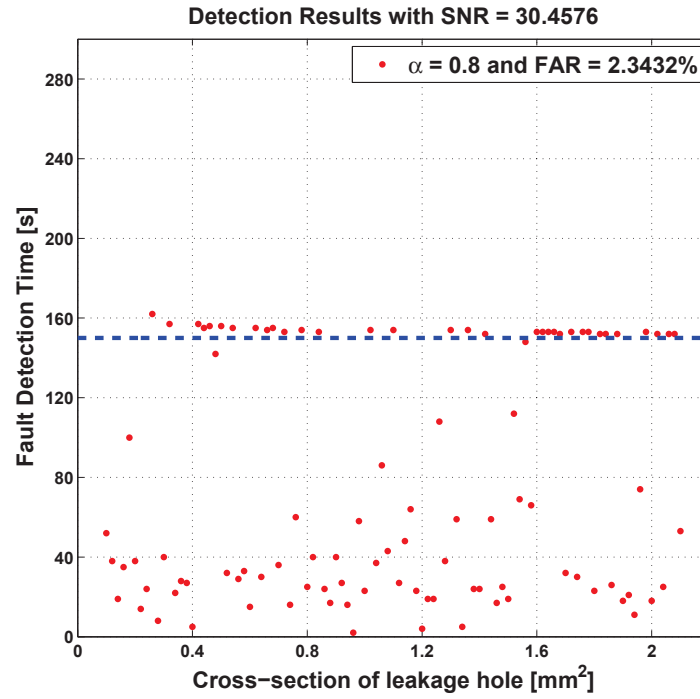


Fig. 4.5 Detection time with various leakage size and chosen $\alpha = 0.8$

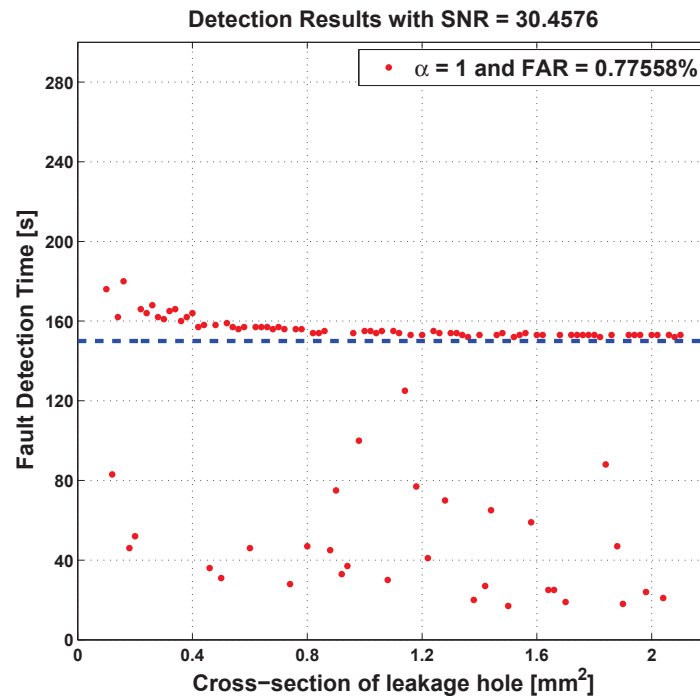


Fig. 4.6 Detection time with various leakage size and chosen $\alpha = 1.0$

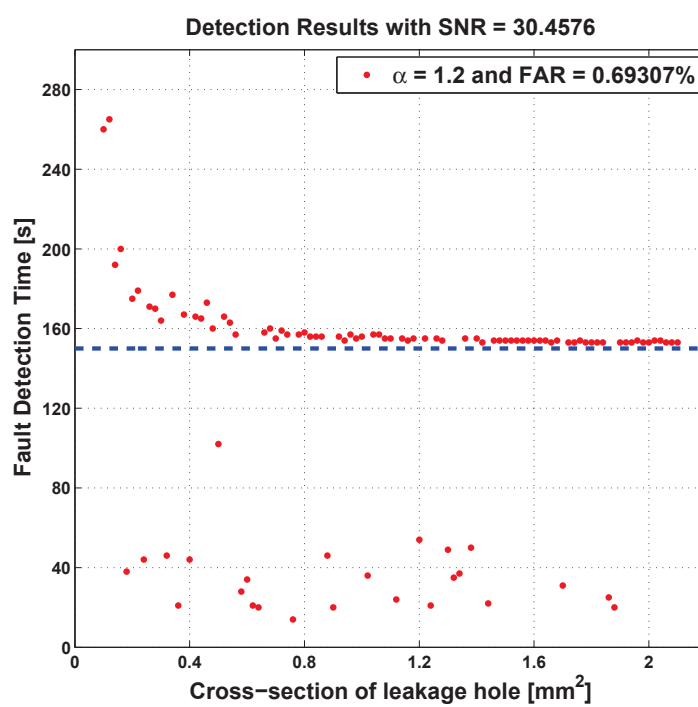


Fig. 4.7 Detection time with various leakage size and chosen $\alpha = 1.2$

Chapter 5

SENSOR FAULT DETECTION SCHEMES IN THREE ARCHITECTURES

In the previous chapter, a fault detection scheme is introduced so that the monitoring architecture can detect possible process faults in the monitored system. In terms of the reliability of the monitoring architecture, apart from the capability to detect possible process faults in the monitored system, the architecture should also be able to detect any sensor faults in the sensor networks. In this chapter, we present the centralized, decentralized, and distributed sensor fault detection schemes in a discrete-time framework. These three architectures can be implemented at each sensor node to detect possible sensor faults occurred in the sensor networks.

In each sensor node, the model-based detector estimates the states monitored by this sensor node. Meanwhile, the adaptive thresholds are calculated by the sensor node at each time step; therefore, the sensor fault is detected when the difference between estimate and the real measurement exceeds the designed threshold. For the centralized, decentralized, or distributed architecture of the detection scheme, the mathematical model used in each detector is different. Unlike the centralized scheme, where each sensor node uses a global mathematical model of the monitored system, the decentralized and distributed schemes both estimate the subsystem's states by using the mathematical model of the subsystem. Moreover, the model-based detector in the distributed architecture can also interact with the detectors in the sensor nodes, which are monitoring neighboring subsystems.

A benchmark simulation on sensor fault detection for an experimental continuous stirred tank heater (CSTH) pilot plant is proposed to compare the performance of all three architectures. All these schemes take advantage of the adaptive approximation approach introduced in [122] to learn the system of CSTH during the initial stage of

benchmark simulation. Then the upgraded model can be used to develop three sensor fault detection schemes without time-consuming efforts on system modeling. And the computer algorithm pseudo-codes are also introduced in this chapter. The work in this chapter has been partially supported by the project 'Increased Functionality Energy-autonomous Sensor Networks for Self-monitoring Industrial Environments' granted by ABB Research.

The chapter is organized as follows. In Section 5.1, we introduce the tank heater plant dynamics in CSTH. In Section 5.2, the centralized sensor fault detection method is presented in a discrete-time framework. The decentralized sensor fault detection method is demonstrated in Section 5.3. Then, the distributed sensor fault detection method is proposed in Section 5.4. Finally, some simulation results are illustrated in Section 5.5 and concluding remarks are given in Section 5.6.

5.1 Tank Heater Plant Dynamics

Thornhill, et al. [142] defined a benchmark problem consisting of a continuous stirred tank heater pilot plant shown in Fig. 5.1 below. In the following the proposed sensor drift fault is evaluated in this benchmark which is particularly challenging as it is based on a first-principle model (of course affected by severe uncertainties).

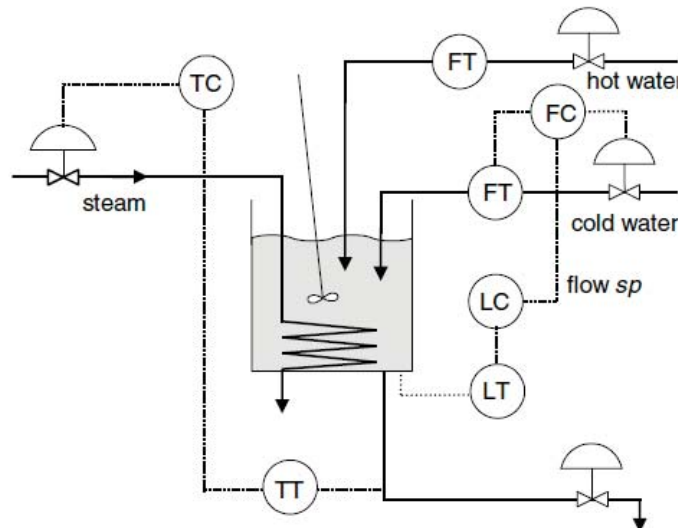


Fig. 5.1 The continuous stirred tank heater pilot plant

5.1.1 The Stirred Tank Heater Process Model

In the stirred tank experimental rig, hot and cold water are mixed, heated further using steam through a heating coil and drained from the tank through a long pipe. This continuous stirred tank heater pilot plant is located at the University of Alberta, Canada. The temperature in the tank is assumed the same as the outflow temperature. The cold and hot water to the tank are pressurized with a pump, and the hot water comes from the boiler heated by the campus steam source, which also supplies the steam to the heater. The temperature of cold and hot water feeds are set to 50°C and 24°C respectively. Control valves in the plant have pneumatic actuators, the seat and stem sets are chosen to suit the range of control. Flow instruments are orifice plates with differential pressure transmitters, and the level instrument is also a differential pressure measurement. The temperature instrument is a type J metal sheathed thermocouple inserted into the outflow pipe. Two kinds of balances are considered in the heater, namely volumetric and heat balances. Then, we have

$$\begin{aligned}\frac{dV(x)}{dt} &= f_{cw} + f_{hw} - f_{out}(x) \\ \frac{dH}{dt} &= W_{st} + h_{hw}\rho_{hw}f_{hw} + h_{cw}\rho_{cw}f_{cw} - h_{out}\rho_{out}f_{out}(x),\end{aligned}\tag{5.1}$$

where x is the level, V is the volume of the water, f_{hw} is the hot water flow into the tank, f_{cw} is the cold water flow into the tank, f_{out} is the outflow from the tank, H is the total enthalpy in the tank, h_{hw} is the specific enthalpy of hot water feed, h_{cw} is the specific enthalpy of cold water feed, h_{out} is the specific enthalpy of water leaving the tank, ρ_{hw} is the density of incoming hot water, ρ_{cw} is the density of incoming cold water, and ρ_{out} is the density of water leaving the tank. The variables denoted with underline can be measured by sensors in this plant. W_{st} is the heat inflow from steam. This heat inflow depends on the steam valve setting, and the relationship between inflow heat and steam valve position can be determined empirically from steady state running at a range of steam valve settings. These two balances have a relationship shown in equation below:

$$h_{out} = \frac{H}{V\rho_{out}}.\tag{5.2}$$

The manual outflow valve is fixed at 50% as a standard operation condition (see Fig. 5.1). The empirical expression below is shown:

$$f_{out} = (0.1013 \times \sqrt{(55 + x)} + 0.0237) \times 10^{-4}.\tag{5.3}$$

For the operation condition with only cold water and steam, the inputs to the heater are electronic signals *steam valve/mA* and *cold water valve/mA* that go to the steam and cold water valves. The outputs are measurements from the level, cold water flow and temperature, such as *level/mA*, *cold water flow/mA* and *temp/mA*. So the following relationships should be represented in the calibration models:

- $\text{level/mA} \Leftrightarrow \text{level/cm} \Leftrightarrow \text{volume/liter}$
- $\text{steam valve/mA} \Leftrightarrow \text{heat inflow/mA} \Leftrightarrow \text{temp}/^{\circ}\text{C} \Leftrightarrow \text{temp/mA}$
- $\text{cold water valve/mA} \Leftrightarrow \text{cold water flow/litre.s}^{-1} \Leftrightarrow \text{cold water flow/mA}$

In addition, for the cold water valve model, there a 1s delay is present which can be represented as

$$M(s) = \frac{e^{-s}}{3.8s + 1} OP(s), \quad (5.4)$$

where $M(s)$ is the Laplace transform of the true *cold water valve/mA* and $OP(s)$ is the Laplace transform of the electronics input *cold water valve/mA*. These calibration models can be determined by measurement at several points in the range and be stored in suitable the piece-wise linear look-up tables.

5.1.2 Pilot Plant Simulation

The simulation model of the tank heater has three inputs, namely cold water valve, hot water valve and steam valve. The outputs are the electronics measurements from the level, cold water flow, hot water flow, and temperature instruments. The PI controllers are designed so as to make sure that outputs measurements level and temperature can track the reference variables in agreement with given tracking performance specifications. In this chapter, we consider the operating condition with cold water, hot water, and steam feeds. The closed-loop control diagram is shown in Fig. 5.2. In the scheme shown in Fig. 5.2, two reference variables are present, namely, the level set point and temperature set point, which can be tracked reasonably well thanks to the designed PI controllers. It is worth noting that in this operating scenario, we only control the cold water valve and the steam valve so that the level and outflow temperature can track their respective references. Specifically, the control input cold water valve is based on the error of level measurement and its reference; the control input steam valve is based on the error of temperature measurement and its reference. Concerning the tank heater, there are three states characterizing the open loop system dynamics, namely the tank volume V , the cold water valve position M and the total enthalpy H . Most of

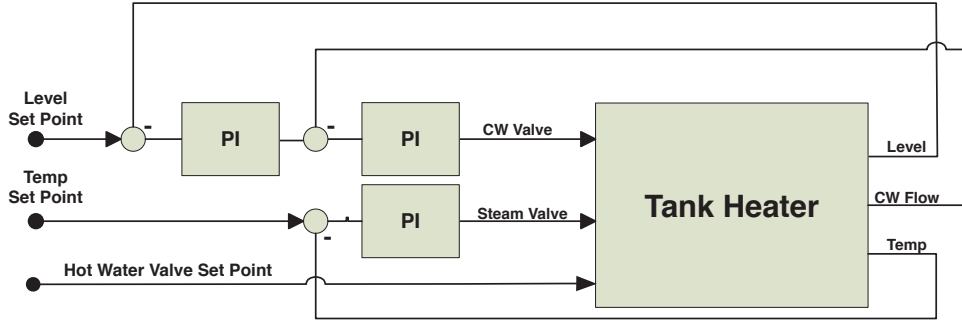


Fig. 5.2 Closed-loop control diagram in the benchmark

simulation approaches in the literature are presented in the form of a state-space model. The operation condition has been linearized with the stirred tank heater operating with both cold and hot water feed. The linearized model of the tank heater is given by

$$\begin{aligned}\dot{x} &= A_c x + B_c u \\ y &= C x\end{aligned}$$

and

$$A_c = \begin{bmatrix} -3.7313 \times 10^{-3} & 1.5789 \times 10^{-6} & 0 \\ 0 & -2.6316 \times 10^{-1} & 0 \\ 4.1580 \times 10^3 & 1.5842 \times 10^{-1} & -2.7316 \times 10^{-2} \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0 & 0 & 4.2900 \times 10^{-5} \\ 1 & 0 & 0 \\ 0 & 6.4000 \times 10^{-1} & 8.8712 \end{bmatrix}$$

$$C = \begin{bmatrix} 2690 & 0 & 0 \\ 0 & 1.5132 \times 10^{-1} & 0 \\ -1979.2 & 1.36 & 1.1226 \times 10^{-2} \end{bmatrix}$$

where x_1 the tank volume; x_2 the output of the integrator in the valve transfer function in (5.4) and x_3 the total enthalpy in the tank; u_1 is the cold water valve position in mA; u_2 the steam valve position in mA; u_3 the hot water valve position in mA; y_1 the level measurement in mA; y_2 the cold water flow measurement in mA; y_3 temperature measurement in mA.

5.2 Centralized Sensor Fault Detection Scheme

The sensor fault detection scheme illustrated in Fig. 5.3 is achieved by analytical redundancy generated by model-based estimation of states and parameters. The state-space model used in this scheme is linearized at a particular operation condition, which means that there exists modeling uncertainty. Meanwhile, a deterministic oscillatory disturbance to the cold water flow, a stochastic disturbance to the level, and temperature measurement noise cannot be neglected. Hence, robustness in a fault detection approach should be given priority in designing the fault detection scheme.

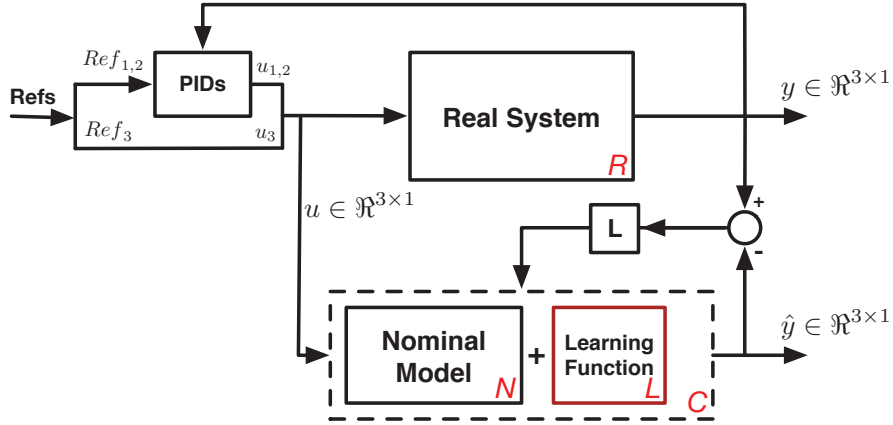


Fig. 5.3 Centralized sensor fault detection scheme for the sensors in the benchmark

Consider a class of uncertain system R described by

$$\begin{aligned}\dot{x} &= A_c x + B_c u + \eta_c(x, u, t) \\ y &= Cx + d(x, y, t) + \beta(t - T_0)f(t),\end{aligned}$$

where $x \in \mathbb{R}^3$ is the system state vector, $u \in \mathbb{R}^3$ is the input vector, $y \in \mathbb{R}^3$ is the output vector, $\eta_c : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^+ \mapsto \mathbb{R}^3$ is the system modeling uncertainty, and $d : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^+ \mapsto \mathbb{R}^3$ is the output disturbance. The term β is a step function characterizing time-profile of the sensor fault function. The possible fault $f(t)$ is an incipient drift on the measurement in one sensor:

$$f(t) = M(1 - e^{-0.5(t-T_0)}),$$

where T_0 is the occurrence of sensor fault, M is the magnitude of sensor fault.

Assumption 13 *The system state and input signal belong to the compact and bounded region $\mathcal{D} \subset \mathbb{R}^3 \times \mathbb{R}^3$ before and after the occurrence of a fault, that is, $(x(t), u(t)) \in \mathcal{D}$, $\forall t \geq 0$.*

Hence, the nominal model of this real system in discrete-time is given by

$$\begin{aligned} x_N(t+1) &= Ax_N(t) + Bu(t) \\ y_N(t) &= Cx_N(t). \end{aligned}$$

Remark 5.2.1 The nominal model is activated where initial values are all zero, which means this linearized model has unknown and negligible modeling uncertainty. The initial values of real system are unknown and volatile once any reference is modified. Hence a more conservative upper bound on the modeling uncertainty is needed to avoid false alarm, leading to a poor fault detectability. However, the application of adaptive approximation method can help learn the nonlinear modeling uncertainty during the transient period of the pilot plant so that an updated nominal model can be used in sensor fault detection scheme.

5.2.1 Learning Approach

The approach is supposed to learn the unknown modeling uncertainty on-line for a while. The real system in discrete-time is given by

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + \eta(x, u, t) \\ y(t) &= Cx(t) + d(x, u, t). \end{aligned}$$

We consider linearly parameterized approximators to learn the modeling uncertainty, then we have

$$\eta(x, u, t) = f(x, u, t)\theta^* + \epsilon_\eta(x, u, t),$$

where $\epsilon_\eta(x, u, t)$ is the minimum approximation error and optimal parameter θ^* is defined by

$$\theta^* = \arg \left\{ \min_{\theta \in \Theta} \left\{ \sup |\eta(x, u, t) - f(x, u, t)\theta| \right\} \right\}.$$

The following assumption is required for the adaptive approximation approach.

Assumption 14 *During the time interval $[0, T_L]$, there are no sensor fault.*

Assumption 15 *The pair (A, C) is observable.*

For $t > T_L$, learning is no longer active and the parameters of the approximation function are frozen at time T_L . This allows to update the local nominal model to be used for fault detection. The adaptive estimator is given by

$$\begin{aligned}\hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + f(\hat{x}(t))\hat{\theta}(t) + L(y(t) - \hat{y}(t)) + \Omega(t+1)\Sigma(t) \\ \Omega(t+1) &= A^0\Omega(t) + f(\hat{x}(t)), \quad A^0 \triangleq A - LC \\ \hat{\theta}(t+1) &= \mathcal{P}_\Theta \left\{ \hat{\theta}(t) + \Gamma(C\Omega(t))^\top (y(t) - \hat{y}(t)) \right\} \\ \hat{y}(t) &= C\hat{x}(t).\end{aligned}\tag{5.5}$$

In the following, details are given concerning the computation of each term in the adaptive estimator.

- Each component $j, j = \{1, 2, 3\}$ in the vector $f(\hat{x})\hat{\theta} \in \mathbb{R}^{3 \times 1}$ represents a sum of radial basis functions with weight parameters to build up function approximations to learn on line the modeling uncertainty.

$$f_j(\hat{x}_j)\hat{\theta}_j(t) = \sum_{i=1}^{n_j} \hat{\theta}_{j_i}(t) e^{-\frac{(\hat{x}_j(t) - c_{j_i})^2}{\sigma_j^2}}, j = \{1, 2, 3\}, \tag{5.6}$$

where $f_j(\hat{x}_j) \in \mathbb{R}^{1 \times n_j}$ is a group of radial basis functions, $\hat{\theta}_j \in \mathbb{R}^{n_j}$ is the weighted parameter vector and there are n_j centers evenly distributed in the region c_j , width σ_j^2 is defined as the variance of c_j . The total number of centers in approximation function $f(\hat{x})\hat{\theta}$ is $n = \sum_{j=1}^3 n_j, j = \{1, 2, 3\}$.

- The term $\mathcal{P}_\Theta$ is a projection operator restricting $\hat{\theta}$ within a compact set Θ . $\Omega \in \mathbb{R}^{3 \times n}$ is the filter version of $f(\hat{x})$ and $\Gamma \in \mathbb{R}^{n \times n}$ is a symmetric, positive definite learning rate matrix. For the sake of simplicity, we assume Θ to be a origin-centered hyper-sphere with radius M_Θ ,

$$\mathcal{P}_\Theta \left\{ \hat{\theta}(t) \right\} = \begin{cases} \hat{\theta}(t) & \text{if } |\hat{\theta}(t)| \leq M_\Theta \\ \frac{M_\Theta \hat{\theta}(t)}{|\hat{\theta}(t)|} & \text{if } |\hat{\theta}(t)| > M_\Theta. \end{cases} \tag{5.7}$$

- The term $\Omega\Sigma$ is added to compensate the estimation error because of $\hat{\theta}$.

$$\Sigma(t) = \begin{cases} \Gamma(C\Omega(t))^\top (y(t) - \hat{y}(t)) & \text{if } |\hat{\theta}(t)| \leq M_\Theta \\ \frac{M_\Theta \hat{\theta}(t)}{|\hat{\theta}(t)|} - \hat{\theta}(t) & \text{if } |\hat{\theta}(t)| > M_\Theta. \end{cases} \tag{5.8}$$

Concerning the configuration values for the estimator, we have: $n_1 = 15, n_2 = 25, n_3 = 25$. and $c_1 \in [0.0014, 0.0056], c_2 \in [-8, 28], c_3 \in [300, 2700]$. For each component, $M_{\Theta_1} = 0.08, M_{\Theta_1} = 1.5, M_{\Theta_1} = 82$. and $\Gamma_1 = 0.01\mathcal{I}, \Gamma_2 = 20\mathcal{I}, \Gamma_3 = 800\mathcal{I}$.

The stability of the adaptive nonlinear estimation scheme is analyzed in the following analysis. We define the state estimation error $\epsilon_x(t) = x(t) - \hat{x}(t)$ and weight parameter estimation error $\epsilon_\theta(t) = \hat{\theta}(t) - \theta^*$. The state estimation error dynamics are described by

$$\epsilon_x(t+1) = A^0 \epsilon_x(t) + f(\hat{x})(\theta^* - \hat{\theta}(t)) - \Omega(t+1)\Sigma(t) + \epsilon,$$

where $\epsilon = (f(x) - f(\hat{x}))\theta^* + \epsilon_\eta - Ld(x, u, t)$. Given $f(\hat{x}) = \Omega(t+1) - A^0\Omega(t)$.

$$\epsilon_x(t+1) = A^0(\epsilon_x(t) + \Omega(t)\epsilon_\theta(t)) - \Omega(t+1)(\hat{\theta}(t) - \theta^* + \hat{\theta}(t+1) - \hat{\theta}(t)) + \epsilon.$$

Let $\tilde{\epsilon}_x(t) = \epsilon_x(t) + \Omega(t)\epsilon_\theta(t)$, then we have

$$\tilde{\epsilon}_x(t+1) = A^0 \tilde{\epsilon}_x(t) + \epsilon.$$

Since the minimum approximation error ϵ_η and Ω are bounded functions, and because of the use of weighted parameter projection in $\hat{\theta}$, it proves that the state estimation error is bounded as well.

Several criteria can be used to stop the learning during healthy mode of behavior. As an example and without loss of generality, the following cost function $J_j(t)$ has been designed:

$$J_j(t) = \frac{1}{n_j} \sum_{i=1}^{n_j} \sqrt{\frac{1}{t} \sum_{p=1}^t (\hat{\theta}_{j_i}(p) - \mu)^2}, j = \{1, 2, 3\} \quad (5.9)$$

where

$$\mu = \frac{1}{t} \sum_{p=1}^t \hat{\theta}_{j_i}(p).$$

Learning is stopped at suitable time T_L once a steady decrease of $J_j(t)$ is noticed. Practically, a suitable and small enough constant ϵ_j is given so that the learning approach is stopped at time T_L when the following inequality holds:

$$|J_j(t) - J_j(t-1)| \leq \epsilon_j.$$

In the simulation trials, we set $\epsilon_1 = 2.7148 \times 10^{-8}$, $\epsilon_2 = 2.2268 \times 10^{-5}$ and $\epsilon_3 = 3.6571 \times 10^{-4}$. After T_L , the updated nominal model is given by

$$\begin{aligned} x_N(t+1) &= Ax_N(t) + Bu(t) + f(x_N(t))\hat{\theta}(T_L) \\ y_N(t) &= Cx_N(t). \end{aligned}$$

Remark 5.2.2 Compared with the linearized model, the updated nominal model has an extra nonlinear functional approximator. However, $\hat{\theta}$ cannot reach zero because of noise in output measurements and this approximator can partially compensate the modeling uncertainty so that the nominal model is improved considerably.

5.2.2 Centralized Sensor Fault Detector

Based on the updated nominal model, the centralized fault detection estimator is given by

$$\begin{aligned} \hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + f(\hat{x}(t))\hat{\theta}(T_L) + L_2(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t). \end{aligned} \tag{5.10}$$

The *residual*:

$$r_j(t) = |y_j(t) - \hat{y}_j(t) + C_j(A_1)^t \hat{x}(T_L)|, \text{ with } A_1 = A - L_2C. \tag{5.11}$$

The *threshold*:

$$\mu_j(t) = \bar{d}_2 + k_j \lambda_j^t \bar{\chi}_j + \sum_{i=0}^{t-1} k_j \lambda_j^{t-1-i} (\bar{\eta}_2 + \|L_2\|_2 \bar{d}_2), \tag{5.12}$$

where $\eta_2 = \eta - f(\hat{x})\hat{\theta}(T_L)$, $|\eta_2| \leq \bar{\eta}_2, \forall t \geq T_L$, and $|d_2| \leq \bar{d}_2, \forall t \geq T_L$; k_{j_2} and λ_{j_2} are chosen positive constants so that $\|C_j(A_1)^t\| \leq k_j \lambda_j^t, \forall t \geq 0$.

Hence, a fault is *detected* when at least one *residual* $r_j(t)$ exceeds its corresponding *threshold* $\mu_j(t)$.

The constants $\bar{d}_j$, $\bar{\chi}_j$, $\bar{\eta}_j$ depend on the physical characteristics of the original state variables and on the modeling uncertainty. These critical scalars are estimated by off-line simulations and possible a-priori knowledge on the monitored system.

$\bar{d}_1 = 0.1, \bar{d}_2 = 0.2, \bar{d}_3 = 0.37; \bar{\eta}_1 = 0.000217, \bar{\eta}_2 = 0.15, \bar{\eta}_3 = 1.3; \text{ and } \bar{\chi}_1 = 0.0038, \bar{\chi}_2 = 14.2, \bar{\chi}_3 = 586.$

To sum up, the overall flow-chart of the computations carried out in the local fault detector are given in the next page.

Remark 5.2.3 The centralized sensor fault detection scheme is activated at time T_L ; therefore, the upper bounds on modeling uncertainty and output disturbance are only needed after time T_L . Based on the historical data under the healthy condition, the constant upper bounds $\bar{\eta}_2$ and $\bar{d}_2$ can be found by testing the detection estimator several times.

Algorithm 5 Centralized Fault Detection Algorithm

Learning = ON
 Detection = OFF
 Initialize the estimate $\hat{x}_j(0) = 0, j = \{1, 2, 3\}$
 Initialize the estimate $\hat{\theta}_{j_i}(0) = 0, i = \{1, \dots, n_j\}$
 Initialize the estimate $\Omega(0) = \mathbf{0}, \mathbf{0} \in \mathbb{R}^{3 \times n}$
 Measurement $y(0)$ is acquired
 Input $u(0)$ is acquired
 Compute the cost function $J_j(0)$ in (5.9)
 Compute the approximation function $f(\hat{x}(0))\hat{\theta}(0)$ in (5.5)
 Compute the estimates $\Omega(1), \hat{x}(1), \hat{\theta}(1)$, and $\hat{y}(1)$ in (5.5)
 Set $t = 1$
while Learning = ON **do** ▷ Learning Approach
 Compute the cost function $J_j(t)$ in (5.9)
 if $\bigvee_{j=1}^3 (|J_j(t) - J_j(t-1)| > \epsilon_j)$ **then**
 Measurement $y_j(t)$ is acquired
 Input $u(t)$ is acquired
 Compute the approximation function $f(\hat{x}(t))\hat{\theta}(t)$ in (5.5)
 Compute the estimates $\Omega(t+1), \hat{x}(t+1), \hat{\theta}(t+1)$, and $\hat{y}(t+1)$ in (5.5)
 $t = t + 1$
 else
 Learning Approach is stopped
 Learning = OFF
 Detection = ON
 $T_L = t$ ▷ Stop Learning Time
 end if
end while
while Detection = ON **do** ▷ Centralized Sensor Fault Detection
 Measurement $y(t)$ is acquired
 Compute the residual $r_j(t)$ in (5.11)
 Compute the threshold $\mu_j(t)$ in (5.12)
 Compare $r_j(t)$ with $\mu_j(t)$
 if $\bigvee_{j=1}^3 (r_j(t) \geq \mu_j(t))$ **then**
 A sensor fault is detected
 Detection = OFF
 $T_D = t$ ▷ Fault Detection Time
 else
 Input $u(t)$ is acquired
 Compute the estimates $\hat{x}(t+1)$ and $\hat{y}(t+1)$ in (5.10)
 $t = t + 1$
 end if
end while

5.3 Decentralized Sensor Fault Detection Scheme

Let us refer to Fig. 5.4 where the part above the dashed-line represents the nonlinear system to be monitored against sensor faults and the part below the dashed-line denotes the decentralized fault detection scheme. In Fig. 5.4, the part above the dashed line

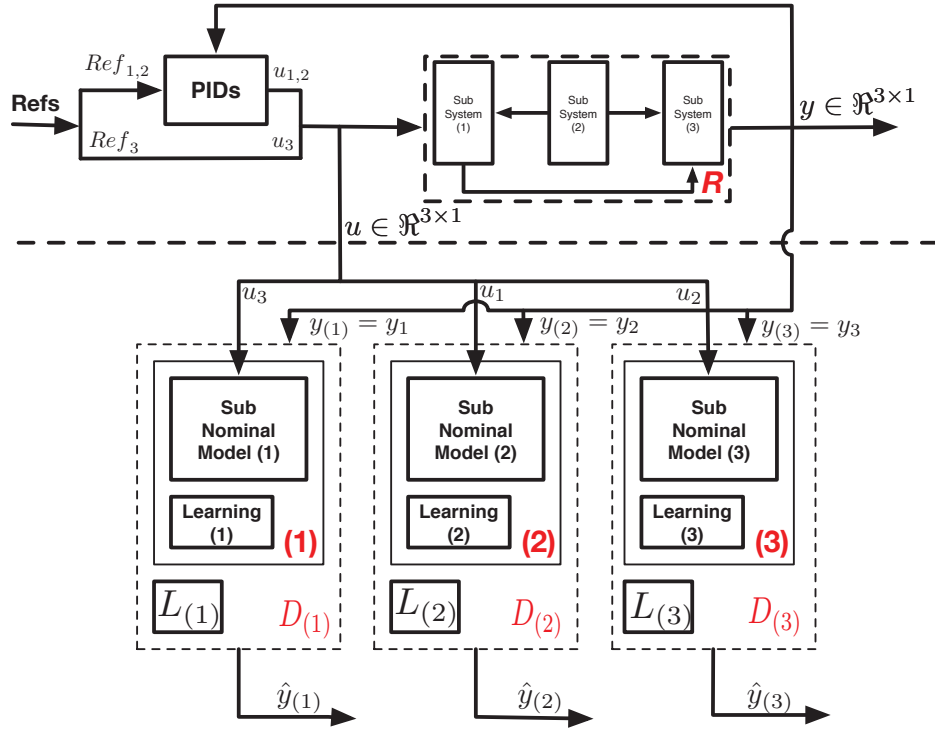


Fig. 5.4 Decentralized fault detection scheme without information exchange among Subsystems (1), (2), and (3). The component in each subsystem is given: $Sub_{(1)} = \{x_1\}$, $Sub_{(2)} = \{x_2\}$, $Sub_{(3)} = \{x_3\}$.

schematically depicts the true controlled system to be diagnosed. In particular, the three dynamically-coupled nonlinear subsystems are enhanced as well as the PID-based multivariable control system. The accessible variables are the vector of control variables u generated by the closed-loop controller and the vector of output variables y provided by the three available sensors. After discretization, the discrete-time dynamics of each subsystem in the controlled system R takes on the form

$$\begin{aligned} x_{\star}(t+1) &= A_{\star}x_{\star}(t) + B_{\star}u_{\star}(t) + \phi_{\star}(\bar{x}_{\star}(t), \bar{u}_{\star}(t)) + \eta_{\star}(t) \\ y_{\star}(t) &= C_{\star}x_{\star}(t) + \varphi_{\star}(\bar{x}_{\star}(t)) + d_{\star}(t) + \beta(t - T_0)\psi_{\star}(M, \alpha, T_0, t), \end{aligned} \quad (5.13)$$

where the subscript $\star$ is a short-hand for subscripts (1), (2), or (3) associated with the respective original state variable $x_\star \in \mathbb{R}$, control variable $u_\star \in \mathbb{R}$ and output variable $y_\star \in \mathbb{R}$. Moreover, we recall that:

- Terms $\phi_\star$ and $\varphi_\star$ denote the interconnection functions between subsystem $\star$ and its neighbouring subsystems.
- Variables $\bar{x}_\star$ and $\bar{u}_\star$ denote the states and inputs of the neighboring subsystems that affect subsystem $\star$.
- Terms $\eta_\star$ and $d_\star$ characterize modelling uncertainty and output disturbance, respectively.
- Function $\psi_\star$ is the sensor fault occurring at unknown time T_0 , represented by $M(1 - e^{-\alpha(t-T_0)})$.
- Term β is a step function characterizing time-profile of the sensor fault function.

The part depicted below the dashed line in Fig. 5.4 describes the decentralized fault detection methodology. More specifically, a decentralized decomposition is addressed in which a local fault detector is associated with each sensor. This local fault detector is only allowed to access the local measurement provided by the sensor and the input variables. In this configuration, **no information is exchanged among the three local fault detectors**.

The nominal linearized model exploited to detect faults at a local level is given by:

$$\begin{aligned} x_{N\star}(t+1) &= A_\star x_{N\star}(t) + B_\star u_\star(t) \\ y_{N\star}(t) &= C_\star x_{N\star}(t). \end{aligned} \tag{5.14}$$

Then, we describe the actual computations carried out in each local fault detector. Without loss of generality, we consider the case of subsystem (1) (that is, in the following we have $\star = (1)$). The local architecture is shown in Fig. 5.5. As already mentioned, only locally available information is used (in this case, the output provided by sensor (1)). As can be seen in Fig. 5.5, three main modules compose the local fault detector: $\mathcal{L}_\star$, $\mathcal{S}_\star$, and $\mathcal{D}_\star$. In the following subsections, the main computational steps involved in each of these blocks will be addressed.

5.3.1 Learning Block ($\mathcal{L}_\star$)

In this block, an adaptive state estimator equipped with a local learning module is implemented. The inputs to this block are the variables $u_\star(t)$ and $y_\star(t)$ taken from

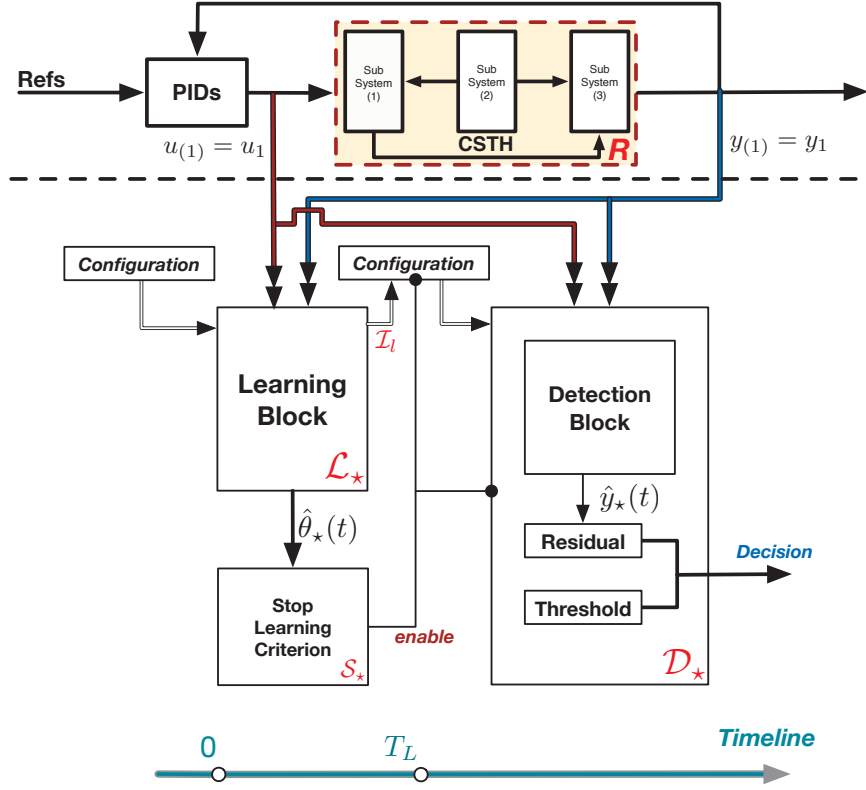


Fig. 5.5 General architecture of decentralized sensor fault detection scheme for Sub-system (1). The timeline is shown enhancing the fact that learning is only activated during the time-horizon $(0, T_L)$ in which it is assumed that the system is not faulty.

the monitored system R , while the outputs are $\hat{\theta}_*(t)$ to the block $\mathcal{S}_*$ and the variable $\mathcal{I}_l = \{\hat{x}_*(t), \hat{\theta}_*(t)\}$ to the block $\mathcal{D}_*$. Let us first recall Assumption 14, at each discrete-time instant, the computations carried out are described by the following relationships:

$$\begin{aligned}
 \hat{x}_*(t+1) &= A_*\hat{x}_*(t) + B_*u_*(t) + f_*(\hat{x}_*(t))\hat{\theta}_*(t) + L_*(y_*(t) - \hat{y}_*(t)) + \Omega_*(t+1)\Sigma_*(t) \\
 \Omega_*(t+1) &= A_*^0\Omega_*(t) + f_*(\hat{x}_*(t)), \quad A_*^0 \triangleq A_* - L_*C_* \\
 \hat{\theta}_*(t+1) &= \mathcal{P}_{\Theta_*} \left\{ \hat{\theta}_*(t) + \Gamma_*(C_*\Omega_*(t))^\top (y_*(t) - \hat{y}_*(t)) \right\} \\
 \hat{y}_*(t) &= C_*\hat{x}_*(t).
 \end{aligned} \tag{5.15}$$

In the following, details are given concerning the computation of each term in (5.15).

- The term $f_*(\hat{x}_*)\hat{\theta}_*$ represents a sum of radial basis functions with weight parameters $\hat{\theta}_*$ to build up function approximations to partially learn on line the modeling uncertainty.

$$f_*(\hat{x}_*(t))\hat{\theta}_*(t) = \sum_{i=1}^{n_*} \hat{\theta}_{*i}(t) e^{-\frac{(\hat{x}_*(t) - c_{*i})^2}{\sigma_*^2}}, \quad (5.16)$$

where $f_*(\hat{x}_*) \in \mathbb{R}^{1 \times n_*}$ is a group of radial basis functions, $\hat{\theta}_* \in \mathbb{R}^{n_*}$ is the weighted parameter vector and there are n_* centers evenly distributed in the region c_* , width σ_*^2 is defined as the variance of c_* .

- The term $\mathcal{P}_{\Theta_*}$ is a projection operator restricting $\hat{\theta}_*$ within a compact set Θ_* . $\Omega_* \in \mathbb{R}^{1 \times n_*}$ is the filter version of $f_*(\hat{x}_*)$ and $\Gamma_* \in \mathbb{R}^{n_* \times n_*}$ is a symmetric, positive definite learning rate matrix. For the sake of simplicity, we assume Θ_* to be a origin-centered hyper-sphere with radius M_{Θ_*} ,

$$\mathcal{P}_{\Theta_*} \left\{ \hat{\theta}_*(t) \right\} = \begin{cases} \hat{\theta}_*(t) & \text{if } |\hat{\theta}_*(t)| \leq M_{\Theta_*} \\ \frac{M_{\Theta_*} \hat{\theta}_*(t)}{|\hat{\theta}_*(t)|} & \text{if } |\hat{\theta}_*(t)| > M_{\Theta_*}. \end{cases} \quad (5.17)$$

- The term $\Omega_* \Sigma_*$ is added to compensate the estimation error because of $\hat{\theta}_*$.

$$\Sigma_*(t) = \begin{cases} \Gamma_*(C_* \Omega_*(t))^T (y_*(t) - \hat{y}_*(t)) & \text{if } |\hat{\theta}_*(t)| \leq M_{\Theta_*} \\ \frac{M_{\Theta_*} \hat{\theta}_*(t)}{|\hat{\theta}_*(t)|} - \hat{\theta}_*(t) & \text{if } |\hat{\theta}_*(t)| > M_{\Theta_*}. \end{cases} \quad (5.18)$$

Concerning the specific configuration values for Subsystems (1), (2), and (3), we have: $\hat{x}_*(0)$, $\hat{\theta}_*(0)$ and $\Omega_*(0)$ are all set to zero. $n_1 = 15, n_2 = 25, n_3 = 25$. and $c_1 \in [0.0014, 0.0056]$, $c_2 \in [-8, 28]$, $c_3 \in [300, 2700]$. $M_{\Theta_{(1)}} = 0.08$, $M_{\Theta_{(2)}} = 1.5$, $M_{\Theta_{(3)}} = 82$. and $\Gamma_{(1)} = 0.01\mathcal{I}$, $\Gamma_{(2)} = 20\mathcal{I}$, $\Gamma_{(3)} = 800\mathcal{I}$, where $\mathcal{I}$ is the identity matrix. The stability of the adaptive approximators can be proved by the similar analysis introduced in Section 5.2.1.

5.3.2 Stop Learning Block ($\mathcal{S}_*$)

The input to this block is $\hat{\theta}_*(t)$ coming from block $\mathcal{L}_*$. The output is a trigger signal *enable* to activate the sensor fault detection scheme $\mathcal{D}_*$ at time T_L when the stopping criterion is satisfied. For $t \geq T_L$, learning is no longer active and the parameters of the approximation function $f_*(\hat{x}_*)\hat{\theta}_*$ are frozen at time T_L . This allows to *update* the local nominal model to be used for fault detection, as will be discussed in the next

subsection. Several criteria can be used to stop the learning during healthy mode of behavior. As an example and without loss of generality, the following cost function $J_\star(t)$ has been defined:

$$J_\star(t) = \frac{1}{n_\star} \sum_{i=1}^{n_\star} \sqrt{\frac{1}{t} \sum_{p=1}^t (\hat{\theta}_{\star_i}(p) - \bar{\theta}_\star)^2}, \quad (5.19)$$

where

$$\bar{\theta}_\star = \frac{1}{t} \sum_{p=1}^t \hat{\theta}_{\star_i}(p).$$

Setting a small enough constant $\epsilon_\star$, the learning is stopped when

$$|J_\star(t) - J_\star(t-1)| \leq \epsilon_\star.$$

In the simulation trials, we set $\epsilon_{(1)} = 1.1919 \times 10^{-7}$, $\epsilon_{(2)} = 1.9498 \times 10^{-5}$, $\epsilon_{(3)} = 2.0726 \times 10^{-4}$, thus getting $T_L = 300$.

5.3.3 Local Sensor Fault Detection Block ($\mathcal{D}_\star$)

As previously mentioned, the computation in this block is activated for $t \geq T_L$. The inputs to this block are the variables $u_\star(t)$ and $y_\star(t)$ taken from the monitored system R . As the parameter vector of the learning function is now frozen at the value $\hat{\theta}_\star(T_L)$, then (5.15) takes on the following simpler form:

$$\begin{aligned} \hat{x}_\star(t+1) &= A_\star \hat{x}_\star(t) + B_\star u_\star(t) + f_\star(\hat{x}_\star(t)) \hat{\theta}_\star(T_L) + LD_\star(y_\star(t) - \hat{y}_\star(t)) \\ \hat{y}_\star(t) &= C_\star \hat{x}_\star(t). \end{aligned} \quad (5.20)$$

The availability of $\hat{x}_\star$ and $\hat{y}_\star$ allow to compute the following *residual* for the local fault detector:

$$r_\star(t) = |y_\star(t) - \hat{y}_\star(t) + C_\star(A_\star^1)^t \hat{x}_\star(T_L)|, \quad (5.21)$$

where

$$A_\star^1 = A_\star - LD_\star C_\star.$$

The above *residual* is compared at each discrete-time instant with the following *threshold*:

$$\mu_\star(t) = \bar{d}_\star + k_\star \lambda_\star^t \bar{\chi}_\star + \sum_{i=0}^{t-1} k_\star \lambda_\star^{t-1-i} \left(\bar{\eta}_\star + \|LD_\star\|_2 \bar{d}_\star \right), \quad (5.22)$$

where the constants $k_\star > 0$ and $0 < \lambda_\star < 1$ are chosen so that the following inequality holds for any iteration t :

$$\left\| C_\star (A_\star^1)^t \right\| \leq k_\star \lambda_\star^t.$$

The constants $\bar{d}_\star$, $\bar{\chi}_\star$, $\bar{\eta}_\star$ depend on the physical characteristics of the original state variables and on the modeling uncertainty. These critical scalars are estimated by off-line simulations and possible a-priori knowledge on the monitored system. Then we have: $\bar{d}_{(1)} = 0.1$, $\bar{d}_{(2)} = 0.2$, $\bar{d}_{(3)} = 4.05$, $\bar{\eta}_{(1)} = 0.067$, $\bar{\eta}_{(2)} = 0.342$, $\bar{\eta}_{(3)} = 9.85$, and $\bar{\chi}_{(1)} = 0.0038$, $\bar{\chi}_{(2)} = 14.2$, $\bar{\chi}_{(3)} = 586$. To sum up, the overall flow-chart of the computations carried out in the local fault detector are given in the next page.

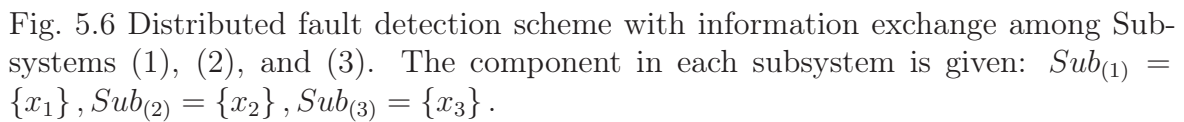
Algorithm 6 Decentralized Fault Detection Algorithm for the Local Fault Detector $\star$

```

Learning = ON
Detection = OFF
Initialize the estimate  $\hat{x}_\star(0) = 0$ 
Initialize the estimate  $\hat{\theta}_\star(0) = 0$ 
Initialize the estimate  $\Omega_\star(0) = 0$ 
Measurement  $y_\star(0)$  is acquired
Input  $u_\star(0)$  is acquired
Compute the cost function  $J_\star(0)$  in (5.19)
Compute the approximation function  $f_\star(\hat{x}_\star(0))\hat{\theta}_\star(0)$  in (5.16)
Compute the estimates  $\Omega_\star(1)$ ,  $\hat{x}_\star(1)$ ,  $\hat{\theta}_\star(1)$ , and  $\hat{y}_\star(1)$  in (5.15)
Set  $t = 1$ 
while Learning = ON do ▷ Learning
    Compute the cost function  $J_\star(t)$  in (5.19)
    if  $|J_\star(t) - J_\star(t-1)| > \epsilon_\star$  then
        Measurement  $y_\star(t)$  is acquired
        Input  $u_\star(t)$  is acquired
        Compute the approximation function  $f_\star(\hat{x}_\star(t))\hat{\theta}_\star(t)$  in (5.16)
        Compute the estimates  $\Omega_\star(t+1)$ ,  $\hat{x}_\star(t+1)$ ,  $\hat{\theta}_\star(t+1)$ , and  $\hat{y}_\star(t+1)$  in (5.15)
         $t = t + 1$ 
    else
        Learning Approach is stopped
        Learning = OFF
        Detection = ON
         $T_L = t$  ▷ Stop Learning
    end if
end while
while Detection = ON do ▷ Local Sensor Fault Detection
    Measurement  $y_\star(t)$  is acquired
    Compute the residual  $r_\star(t)$  in (6.6)
    Compute the threshold  $\mu_\star(t)$  in (5.22)
    Compare  $r_\star(t)$  with  $\mu_\star(t)$ 
    if  $r_\star(t) \geq \mu_\star(t)$  then
        A sensor fault is detected
        Detection = OFF
         $T_D = t$  ▷ Fault Detection Time
    else
        Input  $u_\star(t)$  is acquired
        Compute the estimates  $\hat{x}_\star(t+1)$  and  $\hat{y}_\star(t+1)$  in (5.20)
         $t = t + 1$ 
    end if
end while

```

In Fig. 5.6, the part above the dashed line still schematically show the true controlled system to be diagnosed. The part depicted below the dashed line illustrates the structure of the distributed fault detection methodology. However, unlike the decentralized decomposition addressed in Fig. 5.4, the local fault detectors are connected by some communication channels so that **information can be exchanged among the three local fault detectors**. Compared with the nominal model (5.14), the new nominal


$$\begin{aligned} x_{N\star}(t+1) &= A_{\star}x_{N\star}(t) + B_{\star}u_{\star}(t) + \phi_{\star}(\bar{x}_{N\star}(t), \bar{u}_{\star}(t)) \\ y_{N\star}(t) &= C_{\star}x_{N\star}(t) + \varphi_{\star}(\bar{x}_{N\star}(t)), \end{aligned} \quad (5.23)$$

Remark 5.4.1 The learning mechanism shown in previous section is only supposed to approximate the modeling uncertainty $\phi_\star + \eta_\star$. However, this mechanism cannot

ensure robustness unless output uncertainty, denoted by $\varphi_\star + d_\star$, is small enough. The discrepancy between the true system (5.13) and the nominal linearized models (5.23) (quantified by the terms $\phi_\star(\bar{x}_\star, \bar{u}_\star) - \phi_\star(\hat{x}_\star, \bar{u}_\star) + \eta_\star$ and $\varphi_\star(\bar{x}_\star) - \varphi_\star(\hat{x}_\star) + d_\star$) becomes less significant owing to the estimation of interconnection functions. This on-line compensation to the discrepancy in output can improve the learning performance, thus making the performance rather similar to the one achievable in the fully-centralised case.

Then, we describe the actual computations carried out in each local fault detector based on the new fault detection scheme in Fig. 5.6. Without loss of generality, we consider the case of subsystem (3) (that is, in the following we have $\star = (3)$). The local architecture is shown in Fig. 5.7, where the learning block and detection block can both use the information from neighbouring subsystems.

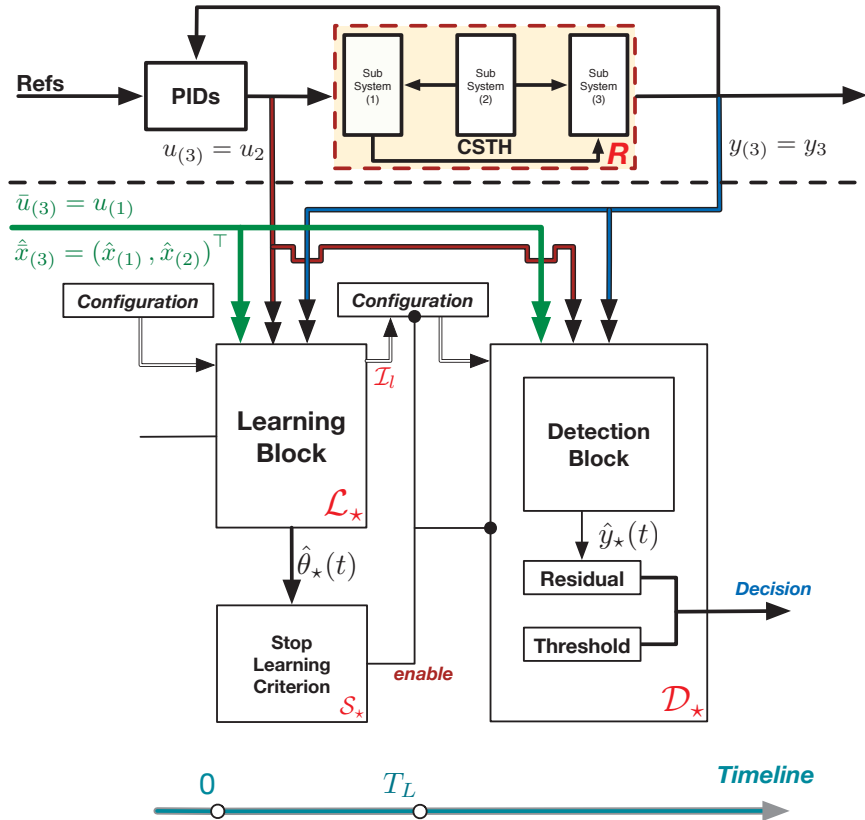


Fig. 5.7 General architecture of distributed sensor fault detection scheme for Subsystem (3). The timeline is shown enhancing the fact that learning is only activated during the time-horizon $(0, T_L)$ in which it is assumed that the system is not faulty.

5.4.1 Learning Block ($\mathcal{L}_\star$)

The inputs to this block are the variables $u_\star(t)$ and $y_\star(t)$ taken from the monitored system R and the **exchanged information** $\bar{u}_\star$ and $\hat{x}_\star$ transferred from the neighboring subsystems. While the outputs are $\hat{\theta}_\star(t)$ to the block $\mathcal{S}_\star$ and the variable $\mathcal{I}_l = \{\hat{x}_\star(t), \hat{\theta}_\star(t)\}$ to the block *Configuration* for $\mathcal{D}_\star$.

The learning block has more inputs due to the information exchange. Compared with the learning approach in Section 5.3.1, a new relationship used to carry out the computations, at each discrete-time instant, is introduced below, and the difference between this relationship and the previous one is highlighted in red font.

$$\begin{aligned}
 \hat{x}_\star(t+1) &= A_\star \hat{x}_\star(t) + B_\star u_\star(t) + \phi_\star(\hat{x}_\star(t), \bar{u}_\star(t)) + f_\star(\hat{x}_\star(t)) \hat{\theta}_\star(t) \\
 &\quad + L_\star(y_\star(t) - \hat{y}_\star(t)) + \Omega_\star(t+1) \Sigma_\star(t) \\
 \Omega_\star(t+1) &= A_\star^0 \Omega_\star(t) + f_\star(\hat{x}_\star(t)), \quad A_\star^0 \triangleq A_\star - L_\star C_\star \\
 \hat{\theta}_\star(t+1) &= \mathcal{P}_{\Theta_\star} \left\{ \hat{\theta}_\star(t) + \Gamma_\star (C_\star \Omega_\star(t))^\top (y_\star(t) - \hat{y}_\star(t)) \right\} \\
 \hat{y}_\star(t) &= C_\star \hat{x}_\star(t) + \varphi_\star(\hat{x}_\star(t)),
 \end{aligned} \tag{5.24}$$

where the details on the computations of each term in (5.24) are the same as the ones shown in (5.16), (6.2) and (5.18). Concerning the specific configuration values for Subsystems (1), (2) and (3), we have: $\hat{x}_\star(0)$, $\hat{\theta}_\star(0)$ and $\Omega_\star(0)$ are all set to zero. $n_1 = 15, n_2 = 25, n_3 = 25$. and $c_1 \in [0.0014, 0.0056], c_2 \in [-8, 28], c_3 \in [300, 2700]$. $M_{\Theta_{(1)}} = 0.08, M_{\Theta_{(2)}} = 1.5, M_{\Theta_{(3)}} = 82$. and $\Gamma_{(1)} = 0.01\mathcal{I}, \Gamma_{(2)} = 20\mathcal{I}, \Gamma_{(3)} = 800\mathcal{I}$, where $\mathcal{I}$ is the identity matrix. The stability of the adaptive approximators can be proved by the similar analysis introduced in Section 5.2.1.

5.4.2 Stop Learning Block ($\mathcal{S}_\star$)

Clearly the stop learning block has the same layout as the one described in Section 5.3.2. However, in the simulation trials, we need a different small enough constant for subsystem (3). Hence, these small enough constants are given by $\epsilon_{(1)} = 1.1919 \times 10^{-7}, \epsilon_{(2)} = 1.9498 \times 10^{-5}, \epsilon_{(3)} = 4.6428 \times 10^{-4}$.

5.4.3 Local Sensor Fault Detection Block ($\mathcal{D}_\star$)

As previously mentioned, the computation in this block is activated for $t \geq T_L$. The inputs to this block are the variables $u_\star(t)$ and $y_\star(t)$ taken from the monitored system R and the **exchanged information** $\bar{u}_\star$ and $\hat{x}_\star$ transferred from the neighboring

subsystems. As the parameter vector of the learning function is now frozen at the value $\hat{\theta}_*(T_L)$, then (5.24) takes on the following simpler form below, and the difference with respect to (5.20) is highlighted by red.

$$\begin{aligned}\hat{x}_*(t+1) &= A_*\hat{x}_*(t) + B_*u_*(t) + \phi_*(\hat{x}_*(t), \bar{u}_*(t)) + f_*(\hat{x}_*(t))\hat{\theta}_*(T_L) + LD_*(y_*(t) - \hat{y}_*(t)) \\ \hat{y}_*(t) &= C_*\hat{x}_*(t) + \varphi_*(\hat{x}_*(t)).\end{aligned}\quad (5.25)$$

The availability of $\hat{x}_*$ and $\hat{y}_*$ allow to compute the following *residual* for the local fault detector:

$$r_*(t) = |y_*(t) - \hat{y}_*(t) + C_*(A_*^1)^t \hat{x}_*(T_L)|, \quad (5.26)$$

where

$$A_*^1 = A_* - LD_*C_*.$$

The above *residual* is compared at each discrete-time instant with the following *threshold*:

$$\mu_*(t) = \bar{d}_*^C + k_*\lambda_*^t\bar{\chi}_* + \sum_{i=0}^{t-1} k_*\lambda_*^{t-1-i}(\bar{\eta}_*^C + \|LD_*\|_2 \bar{d}_*^C), \quad (5.27)$$

where the constants $k_* > 0$ and $0 < \lambda_* < 1$ are chosen so that the following inequality holds for any iteration t :

$$\|C_*(A_*^1)^t\| \leq k_*\lambda_*^t.$$

The constants $\bar{d}_*^C$, $\bar{\chi}_*$, $\bar{\eta}_*^C$ depend on the physical characteristics of the original state variables and on the modeling uncertainty. These critical scalars are estimated by off-line simulations and possible a-priori knowledge on the monitored system. Then we have: $\bar{d}_{(1)}^C = 0.1$, $\bar{d}_{(2)}^C = 0.2$, $\bar{d}_{(3)}^C = 2.16$, $\bar{\eta}_{(1)}^C = 0.0972$, $\bar{\eta}_{(2)}^C = 0.341$, $\bar{\eta}_{(3)}^C = 3.33$, and $\bar{\chi}_{(1)} = 0.0038$, $\bar{\chi}_{(2)} = 14.2$, $\bar{\chi}_{(3)} = 586$. To sum up, the overall flow-chart of the computations carried out in the local fault detector are given in the next page, where the code concerning the new information exchange parts is highlighted in red font.

Algorithm 7 Distributed Fault Detection Algorithm for the Local Fault Detector $\star$

Learning = ON
 Detection = OFF
 Initialize the estimate $\hat{x}_\star(0) = 0$
 Initialize the estimate $\hat{\hat{x}}_\star(0) = 0$
 Initialize the estimate $\hat{\theta}_\star(0) = 0$
 Initialize the estimate $\Omega_\star(0) = 0$
 Measurement $y_\star(0)$ is acquired
 Input $u_\star(0)$ and $\bar{u}_\star(0)$ are acquired
 Compute the cost function $J_\star(0)$ in (5.19)
 Compute the approximation function $f_\star(\hat{x}_\star(0))\hat{\theta}_\star(0)$ in (5.16)
 Compute the estimates $\Omega_\star(1)$, $\hat{x}_\star(1)$, $\hat{\theta}_\star(1)$, and $\hat{y}_\star(1)$ in (5.24)
 Set $t = 1$
while Learning = ON **do** ▷ Learning
 Compute the cost function $J_\star(t)$ in (5.19)
 if $|J_\star(t) - J_\star(t-1)| > \epsilon_\star$ **then**
 Measurement $y_\star(t)$ is acquired
 Exchange information $\hat{\hat{x}}_\star(t)$ is acquired
 Input $u_\star(t)$ and $\bar{u}_\star(t)$ are acquired
 Compute the approximation function $f_\star(\hat{x}_\star(t))\hat{\theta}_\star(t)$ in (5.16)
 Compute the estimates $\Omega_\star(t+1)$, $\hat{x}_\star(t+1)$, $\hat{\theta}_\star(t+1)$, and $\hat{y}_\star(t+1)$ in (5.24)
 $t = t + 1$
 else
 Learning Approach is stopped
 Learning = OFF
 Detection = ON
 $T_L = t$ ▷ Stop Learning
 end if
end while
while Detection = ON **do** ▷ Local Sensor Fault Detection
 Measurement $y_\star(t)$ is acquired
 Compute the residual $r_\star(t)$ in (5.26)
 Compute the threshold $\mu_\star(t)$ in (5.27)
 Compare $r_\star(t)$ with $\mu_\star(t)$
 if $r_\star(t) \geq \mu_\star(t)$ **then**
 A sensor fault is detected
 Detection = OFF
 $T_D = t$ ▷ Fault Detection Time
 else
 Exchange information $\hat{\hat{x}}_\star(t)$ is acquired
 Input $u_\star(t)$ and $\bar{u}_\star(t)$ are acquired
 Compute the estimates $\hat{x}_\star(t+1)$ and $\hat{y}_\star(t+1)$ in (5.25)
 $t = t + 1$
 end if
end while

5.5 Simulation Outcomes

The centralized, decentralized, and distributed sensor fault detection schemes are evaluated on the Simulink model in Fig. 5.8. The model block **Benchmark_f** (used

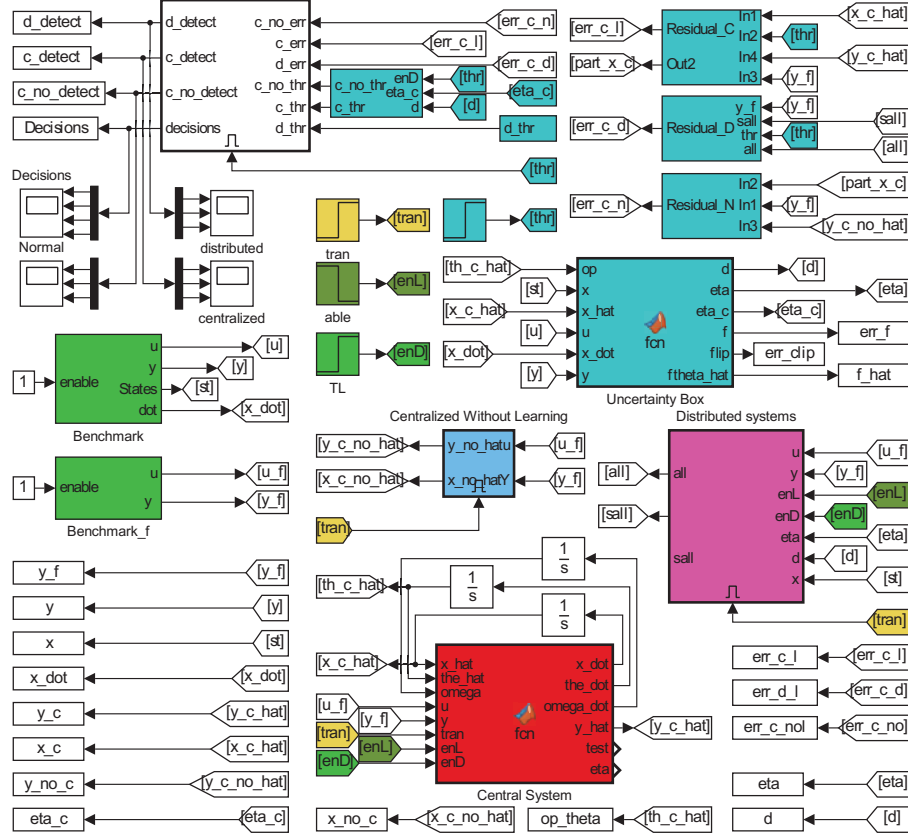


Fig. 5.8 General Simulink model of centralized, decentralized, and distributed sensor fault detection schemes

as a "surrogate" of the real system) represents the nonlinear continuous-time stirred tank heater pilot plant. The details of this block can be found in Fig. 5.9, where the pilot plant model is controlled by three PID controllers. The possible fault of any sensor is a drift on the measurement. In this simulation, the case of a single drift fault occurring in one sensor has been considered and tested. The designed three scenarios are listed below.

- One sensor drift occurs at time 350s in y_1 , $M = 2.5$, $\alpha = 0.5$.
- One sensor drift occurs at time 350s in y_2 , $M = 2.0$, $\alpha = 0.5$.
- One sensor drift occurs at time 350s in y_3 , $M = 6.0$, $\alpha = 0.5$.

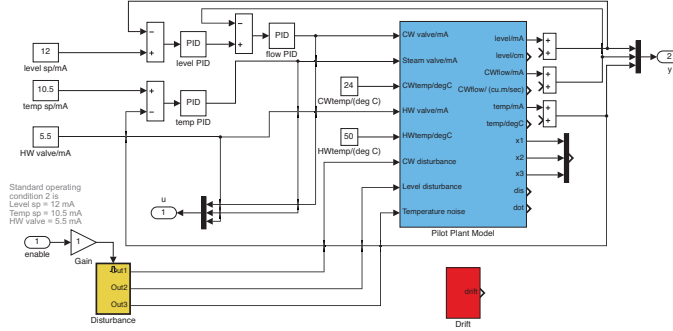


Fig. 5.9 Simulink Block **Benchmark_f**: Stirred tank heater pilot plant

5.5.1 Learning Performance

Before analyzing the learning performance on output disturbance and modeling uncertainties, the cost function J is illustrated in Fig. 5.10. It is found that all the

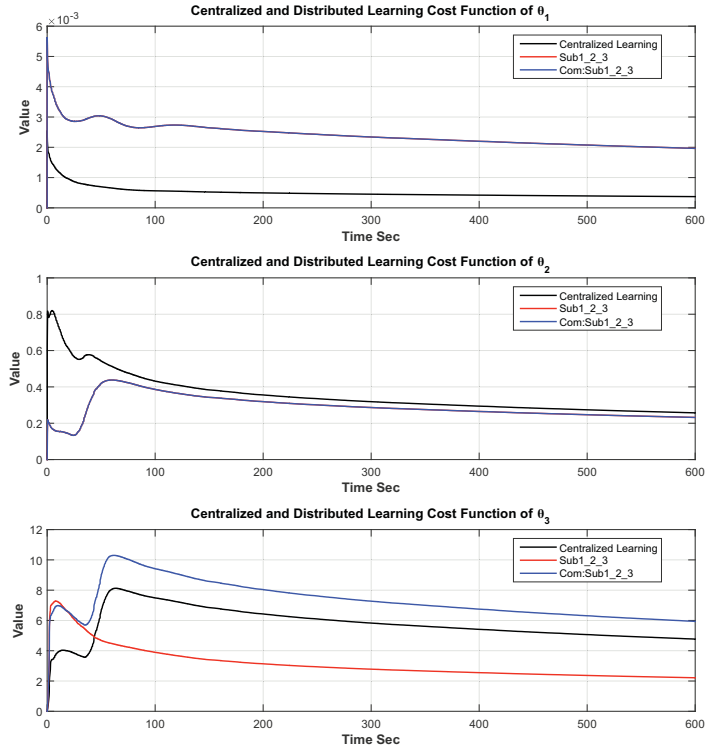


Fig. 5.10 Centralized, decentralized, and distributed learning cost functions

cost function drop steadily after 250s, based on the given small enough constant ϵ , the stop learning time T_L can be chosen as 300s.

Then the learning performance on output disturbance is shown in Fig. 5.11. Due to the different nominal models used in the learning architecture of these three fault

detection schemes, the components of output disturbance are various. The output disturbances are expressed by the functions listed below.

- Centralized Case: d , which is the global output disturbance.
- Decentralized Case: $d_\star^L = d_\star + \varphi_\star(\bar{x}_\star)$.
- Distributed Case: $d_\star^{LC} = d_\star + \varphi_\star(\bar{x}_\star) - \varphi_\star(\hat{\bar{x}}_\star)$.

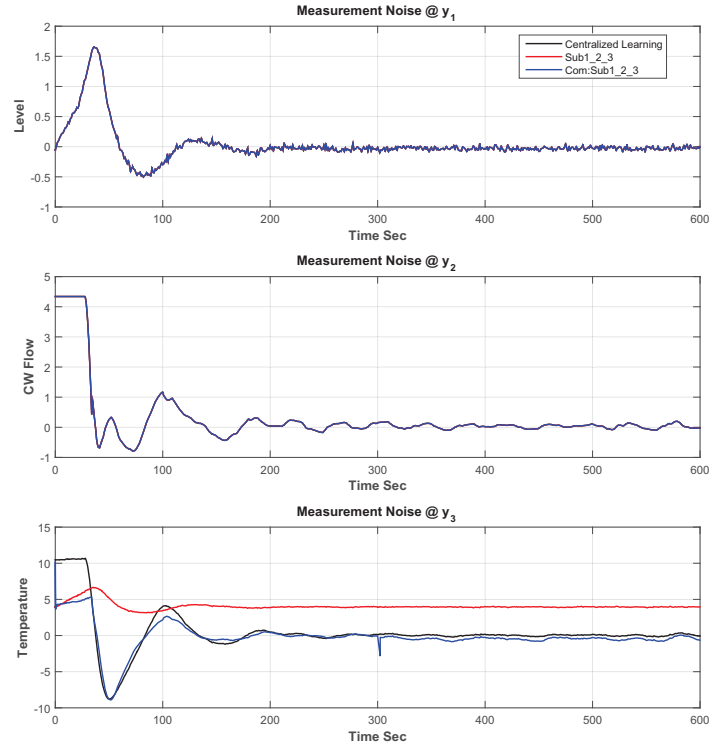


Fig. 5.11 Learning performance on output disturbance

In Fig. 5.11, it is found that the output disturbance on 1st and 2nd outputs are around the same; the local decentralized and distributed estimators for 3rd output have a significantly larger disturbance because 3rd output is influenced by all the states in the real system. However, if subsystem is allowed to communicate with its neighboring subsystems, the output disturbance on 3rd output becomes much smaller than that in the decentralized scheme without communication.

The difference between the real uncertainties and learnt uncertainties is shown in Fig. 5.12. And the functions listed below can help to distinguish the real and learnt uncertainties.

- Centralized Case: Real is η , which is the global modeling uncertainty; Learnt is $f(\hat{x})\hat{\theta}(T_L)$.

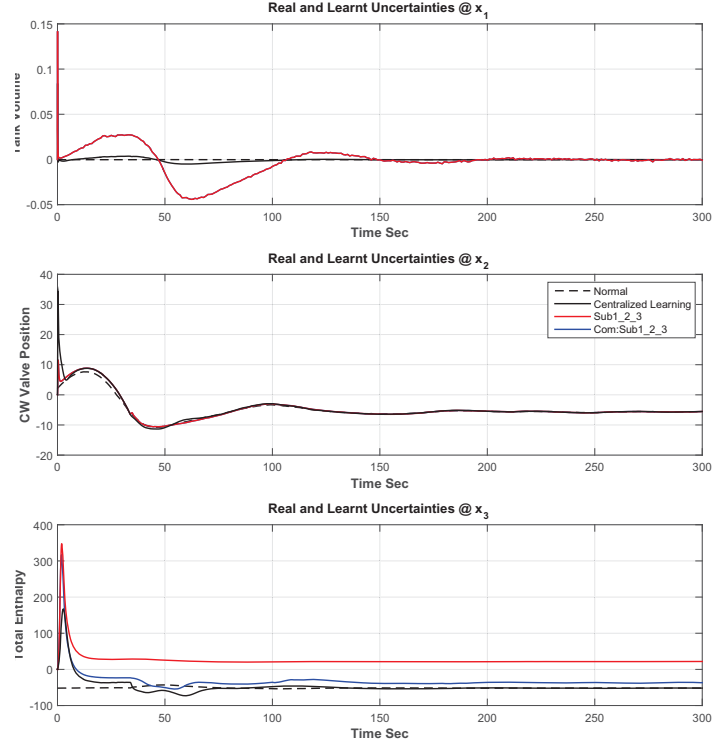


Fig. 5.12 Learning performance on real and learnt modeling uncertainties

- Decentralized Case: Real is η_* ; Learnt is $f_*(\hat{x}_*)\hat{\theta}_*(T_L)$.
- Distributed Case: Real is η_* ; Learnt is $f_*(\hat{x}_*)\hat{\theta}_*(T_L)$.

In Fig. 5.13, it is found that the learnt uncertainties by centralized learning approach are can almost track the real uncertainty after 200s. In the decentralized scheme, the interconnection functions which are omitted from local distributed estimators become part of the modeling uncertainty. Particularly, no access to interconnection functions in 3rd output may cause a failure of learning. The local agents in the distributed scheme are offered communication channels, only the estimation error of interconnection functions become part of the modeling uncertainty, which can significantly improve the learning performance, especially in 3rd output.

Finally, the updated modeling uncertainties are illustrated in Fig. 5.14. And the functions of updated modeling uncertainties are listed below.

- Centralized Case: $\eta_C = \eta - f(\hat{x})\hat{\theta}(T_L)$.
- Decentralized Case: $\eta_*^L = \phi_*(\bar{x}_*, \bar{u}_*) + \eta_* - f_*(\hat{x}_*)\hat{\theta}_*(T_L)$.
- Distributed Case: $\eta_*^{LC} = \phi_*(\bar{x}_*, \bar{u}_*) - \phi_*(\hat{x}_*, \bar{u}_*) + \eta_* - f_*(\hat{x}_*)\hat{\theta}_*(T_L)$.

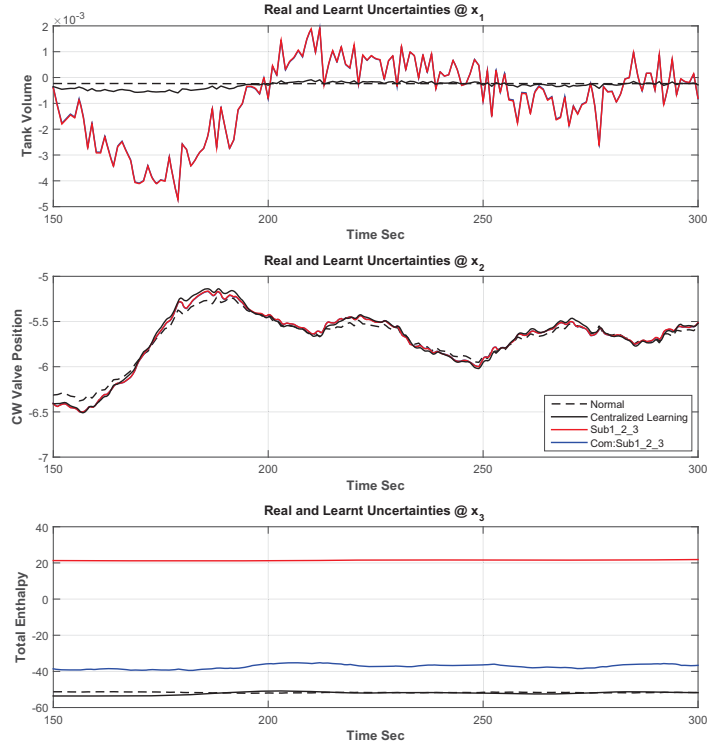


Fig. 5.13 Details of the learning performance on real and learnt modeling uncertainties

In Fig. 5.14, compared with the performance of centralized learning approach, the results from decentralized and distributed approaches are always worse. But the modeling uncertainties in 1st and 2nd states can be neutralized significantly at least. In these two cases, the modeling uncertainty can be partially compensated by the learnt function. However, there exists a considerable improvement in 3rd state only when the communication between subsystems is available. A (possibly) conservative upper bound should be used for designing thresholds. By using the historical data on inputs and outputs under the healthy condition, we can test the local detection estimator several times to find proper upper bounds on the modified joint modeling uncertainty and joint output disturbance, respectively.

5.5.2 Sensor Fault Detection Scheme

Scenario 1: One sensor drift occurs at time 350s in y_1 , $M = 2.5$, $\alpha = 0.5$. The results are shown in Fig. 5.15–5.17. Fault detected at 370.5s, 356.2s and 355.9s respectively. The drift sensor fault in 1st output can be compensated quickly because of the cascaded PIDs. The decentralized nominal models still have significant model uncertainties in the dynamics of 3rd state; therefore, the designed threshold for 3rd

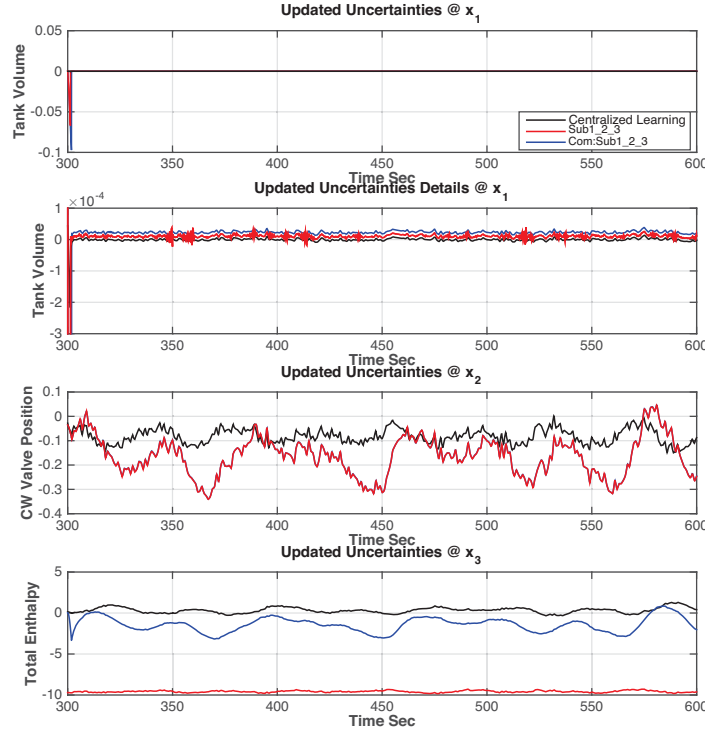


Fig. 5.14 Learning performance on updated modeling uncertainties

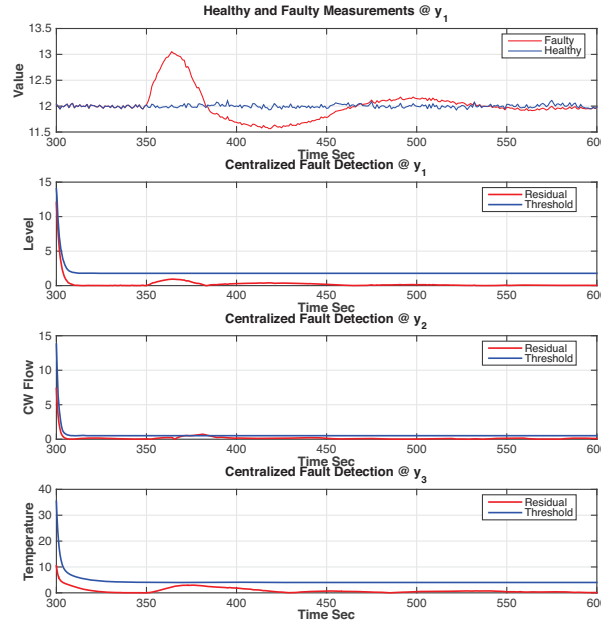
residual is considerably conservative.

Scenario 2: One sensor drift occurs at time 350s in y_2 , $M = 2.0$, $\alpha = 0.5$.

The results are shown in Fig. 5.18–5.20. Fault detected at 358.3s, 360.2s and 359.9s respectively. The centralized, decentralized, and distributed fault detection schemes have nearly the same performance when a sensor fault occurring in 2nd output. In the distributed schemes, with the access to neighboring subsystem estimates, the threshold for 3rd residual become better than the decentralized case.

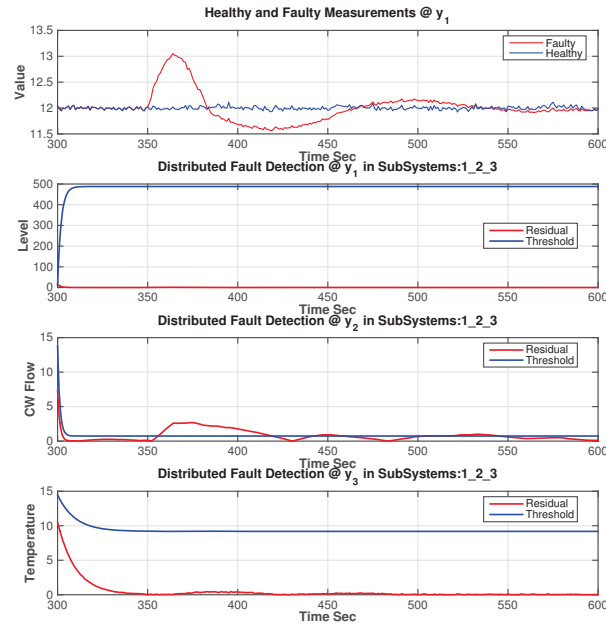
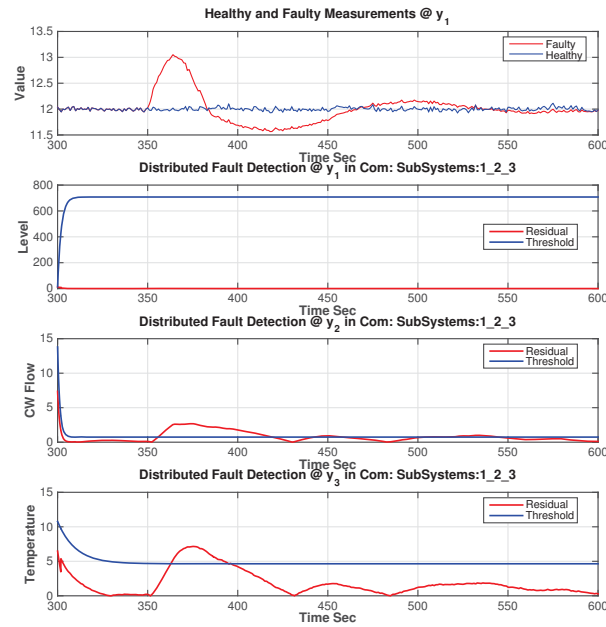
Scenario 3: One sensor drift occurs at time 350s in y_3 , $M = 6.0$, $\alpha = 0.5$.

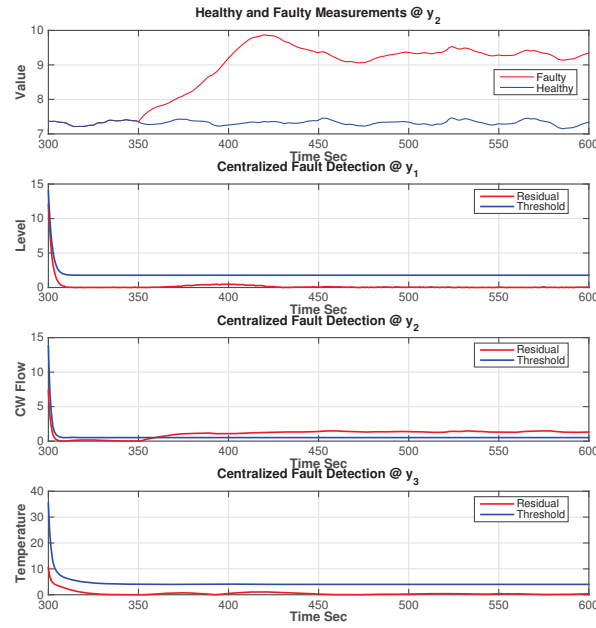
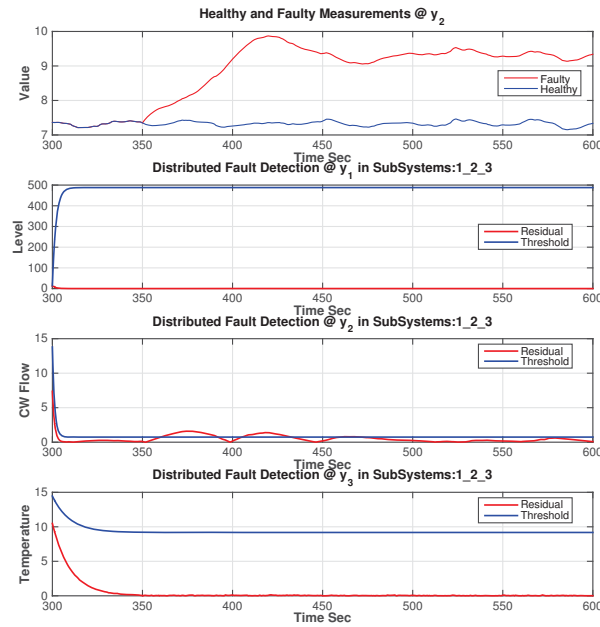
The results are shown in Fig. 5.21–5.23. Fault detected at 385.1s, missed and 394.8s respectively. Any fault in 3rd output will not propagate back to other states of system dynamics. The decentralized detection scheme misses the sensor fault occurring on 3rd output due to the conservative threshold. With the access to neighbor subsystem estimates, the distributed scheme can avoid the modeling uncertainty caused by interconnection functions. Hence, a less conservative threshold can be designed to solve "missed detection" problem.

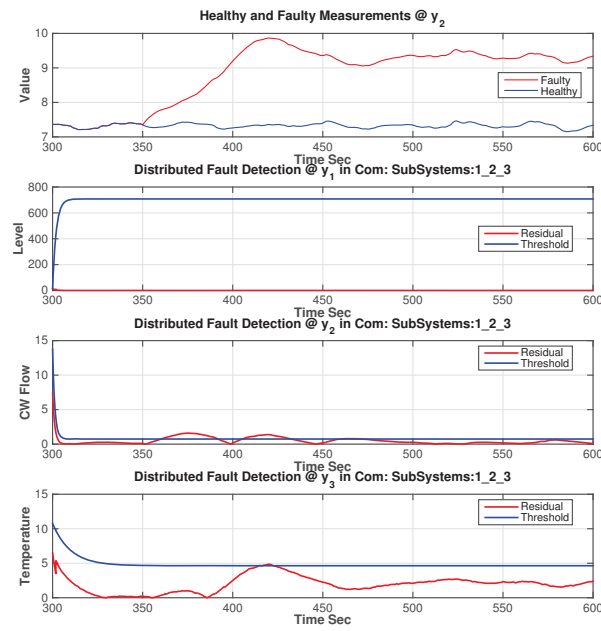
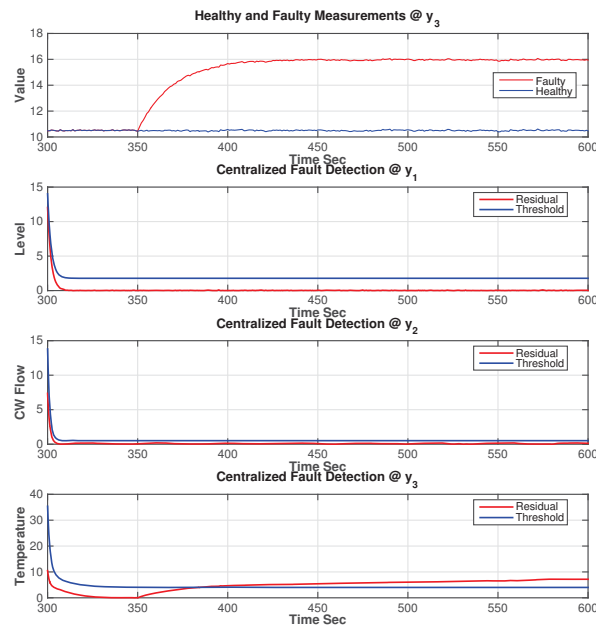
Fig. 5.15 Centralized sensor detection scheme in y_1

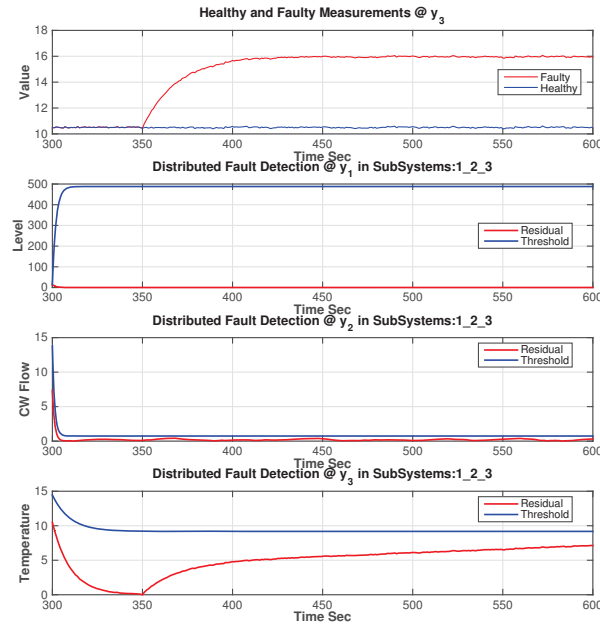
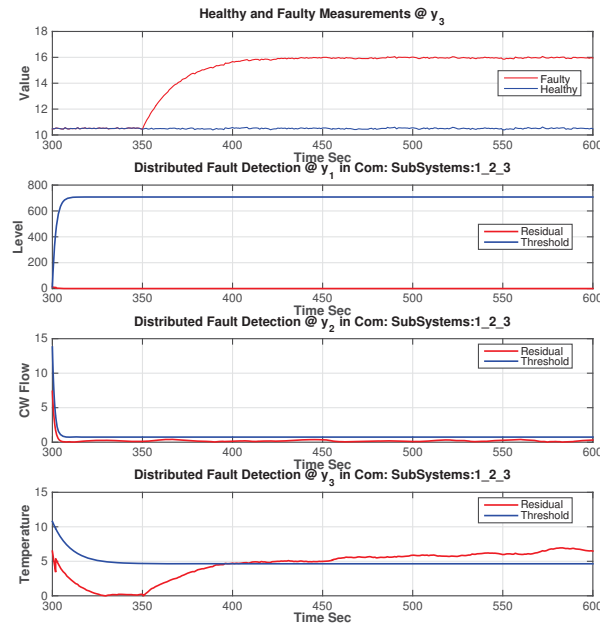
5.6 Concluding Remarks

In this chapter, we present the centralized, decentralized, and distributed sensor fault detection algorithms, which can be implemented at each sensor node to detect possible sensor faults in the sensor networks. The computer pseudo-codes of all three architectures are introduced to offer the user-friendly algorithms for the industry clients. And the detection performance is compared by an industrial benchmark simulation in a continuous stirred tank heater (CSTH) pilot plant. In the next chapter, the sensor fault detectability analysis is presented that each sensor node can evaluate its fault detector by a novel off-line assessment criterion.

Fig. 5.16 Decentralized sensor detection scheme in y_1 Fig. 5.17 Distributed sensor detection in y_1

Fig. 5.18 Centralized sensor detection scheme in y_2 Fig. 5.19 Decentralized sensor detection scheme in y_2

Fig. 5.20 Distributed sensor detection scheme in y_2 Fig. 5.21 Centralized sensor detection scheme in y_3

Fig. 5.22 Decentralized sensor detection scheme in y_3 Fig. 5.23 Distributed sensor detection scheme in y_3

Chapter 6

DRIFT SENSOR FAULT DETECTABILITY ANALYSIS

In this chapter, a further analysis on the sensor fault detectability is proposed for the fault detector in the sensor nodes monitoring a centralized system or a subsystem of the decentralized system. Hence, each sensor node is able to evaluate the fault detector quantitatively. Unlike the robustness achieved by modeling uncertainty bounds and adaptive algorithm, detectability is influenced by modeling uncertainty and sensor faults, which is relatively difficult to be derived explicitly. The existing fault detectability analysis can be found in [148], [143], [161]. Moreover, Polycarpou et al. [114] investigated a rigorous detectability analysis for system fault diagnosis scheme with full-state measurement. The objective of this chapter is to provide an elaborate fault detectability and detection time interval analysis for a sensor fault detection scheme, concerning a class of nonlinear uncertain systems with a potential abrupt or incipient sensor fault. The assumptions are introduced on constant modeling uncertainty bounds and the slowest evolution rate of incipient sensor fault. The main contributions in this chapter are the off-line assessment criterion for sensor fault detectability and analysis of detection time interval. Owing to the derivation of adaptive detection thresholds and estimation errors, the set of detectable sensor faults can be characterized before activating the corresponding sensor fault detection scheme. Moreover, the upper bound on detection time interval can be analytically outlined and a relationship between the magnitude of sensor fault and the upper bound is acquired as well.

The chapter is organized as follows. Section 6.1 firstly defines the class of nonlinear systems and potential sensor faults, then the design of sensor fault detection scheme, including the definition of detection time interval, is described. In Section 6.2, the sets

of detectable sensor fault are characterized under two different conditions. Followed by the detection time interval analysis is presented in Section 6.3. In Section 6.4, the analytical results are illustrated by a simulation example. Finally, Section 6.5 has some conclusion remarks.

6.1 Problem Formulation

In this section, a class of nonlinear uncertain systems is introduced and the family of sensor faults dealt with is defined. Moreover, the fault detection scheme is briefly described.

6.1.1 Nominal Plant, Uncertainty and Fault Representation

We consider the formulation given in [159]. Specifically, we address a MIMO nonlinear uncertain system of the form

$$\begin{aligned}\dot{x} &= Ax + \gamma(y, u) + \eta(x, u, t) \\ y &= Cx + d(x, u, t) + \beta(t - T_0)F\phi(\theta, \alpha, T_0, t),\end{aligned}\tag{6.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector and $y \in \mathbb{R}^p$ is the output vector. The functions $\gamma : \mathbb{R}^p \times \mathbb{R}^m \mapsto \mathbb{R}^n$, $\eta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \mapsto \mathbb{R}^n$, and $d : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \mapsto \mathbb{R}^p$ are smooth vector fields, and (A, C) is a detectable pair. η and d denote modeling uncertainty and/or disturbances acting on the state equation and the output equations, respectively.

The term $\beta(t - T_0)F\phi(\theta, \alpha, T_0, t)$ characterizes the way in which the sensor fault affects the output equation in (6.1). Specifically, $\phi(\theta, \alpha, T_0, t)$ models the nonlinear sensor fault function and $\beta(t - T_0)$ is a step function of the form

$$\beta(t - T_0) = \begin{cases} 0 & t < T_0 \\ 1 & t \geq T_0, \end{cases}$$

where T_0 is unknown fault occurrence time. This step function allows to characterize in explicit form the beginning of the fault effect in the output equation. It is worth noting that in the literature (see [161]) the term $\beta(t - T_0)$ is used to describe abrupt and incipient faults in the same formulation. In this chapter, this is not necessary because of the specific kind of sensor faults considered – drift sensor faults – for which the distinction between abrupt and incipient faults is not significant. To model a drift

in the s -th sensor yielding the s -th component of the output vector y , we have:

$$F\phi(\theta^s, \alpha^s, T_0, t) = \theta^s(1 - e^{-\alpha^s(t-T_0)}),$$

where θ^s is the unknown sensor saturation level in s^{th} sensor and α^s is the unknown evolution rate of the drift fault. In this chapter, for the sake of simplicity the following assumptions are introduced.

Assumption 16 *Only one component of the output vector y is affected by a drift sensor fault.*

Assumption 16 implies that F belongs to a class of p possible vectors $\{F^1, F^2, \dots, F^p\}$, where, for any vector $F^s, s = 1, \dots, p$, only the s^{th} component F_s^s of vector F^s is equal to 1 whereas all other components are zero.

Assumption 17 *The system states and input signals belong to a compact region $\mathcal{D} \subset \mathbb{R}^n \times \mathbb{R}^m$ before and after the occurrence of a fault, that is, $(x(t), u(t)) \in \mathcal{D}, \forall t \geq 0$.*

Assumption 18 *The modeling uncertainties, represented by η and d in (6.1), are unstructured and unknown nonlinear function of x, u and t , but bounded by given constants $\bar{\eta}$ and $\bar{d}$, i.e.,*

$$\begin{aligned} |\eta(x, u, t)| &\leq \bar{\eta}, \quad |d(x, u, t)| \leq \bar{d} \\ \forall (x, u) &\in \mathcal{D}, \forall t \geq 0. \end{aligned} \tag{6.2}$$

Assumption 19 *The s -th sensor fault evolution rate α^s is constant and unknown, but satisfies*

$$\alpha^s \geq \underline{\alpha}^s > 0, \quad \forall t \geq 0,$$

where the lower bounds $\underline{\alpha}^s, s = 1, \dots, p$ on the drift fault evolution rate are assumed to be known.

In Fig. 6.1, an example of drift fault is given to gain further insight on the problem formulation and on Assumption 19.

Remark 6.1.1 Assumption 17 is a well-posedness assumption guaranteeing the boundedness of system states and inputs before and after the occurrence of a sensor fault (fault accommodation is out of the scope of this chapter). Assumption 18 provides uniform bounding constants on the modeling uncertainty terms $\eta(x, u, t)$ and $d(x, u, t)$.

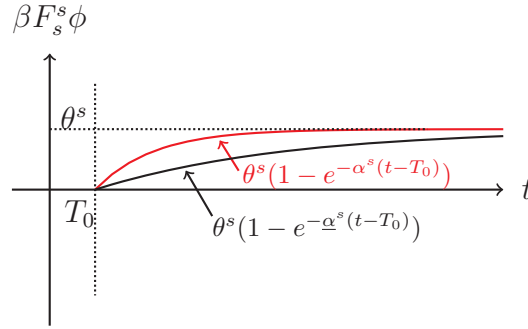


Fig. 6.1 Time-evolution of a drift sensor fault (red line) and the corresponding lower bound (black line).

Time-varying bounds have been considered in the literature (see, for example, [160], [161], [159]). However, for the fault-detectability analysis provided in the sequel, uniform bounds on the uncertainties are needed. In Assumption 19, the availability of a lower bound $\underline{\alpha}^s$ on drift evolution rate is supposed to be available and it will be exploited in the design of the detection threshold. Clearly, the more conservative (that is smaller) $\underline{\alpha}^s$, the more difficult a timely detection is in the presence of modeling uncertainty.

6.1.2 Fault Detection Scheme

The fault detection scheme is based on the fault detection estimator given in [159] and that is briefly recalled in this subsection for the readers' convenience:

$$\begin{aligned}\dot{\hat{x}}^0 &= A\hat{x}^0 + L(\hat{y}^0 - y) + \gamma(y, u), \quad \hat{x}^0(0) = 0 \\ \hat{y}^0 &= C\hat{x}^0\end{aligned}$$

where $\hat{x}^0$ and $\hat{y}^0$ denote the estimated state and output vectors, respectively, L is a suitable $n \times p$ design gain matrix, and $\epsilon_x^0 \triangleq x - \hat{x}^0$ denotes the state estimation error. Then

$$\dot{\epsilon}_x^0 = A_0\epsilon_x^0 + Ld(x, u, t) + \eta(x, u, t), \quad t < T_0 \quad (6.3)$$

where the gain L is chosen such that the matrix $A_0 \triangleq (A + LC)$ is Hurwitz and diagonalizable. As shown in [159], each component $\epsilon_{y_j}^0 = y_j - \hat{y}_j^0$, $j = 1, \dots, p$ of the output estimation error is given by

$$\epsilon_{y_j}^0 = C_j\epsilon_x^0 + d_j(x, u, t),$$

where C_j is the j^{th} row vector of matrix C and d_j is the j^{th} component of the measurement uncertainty d . Then

$$\epsilon_{y_j}^0(t) = \int_0^t C_j e^{A_0(t-\tau)} (Ld(x, u, \tau) + \eta(x, u, \tau)) d\tau + d_j(x, u, t) + C_j e^{A_0 t} \epsilon_x^0(0). \quad (6.4)$$

and hence

$$|\epsilon_{y_j}^0(t)| < \int_0^t k_j e^{-\lambda_j(t-\tau)} |Ld(x, u, \tau) + \eta(x, u, \tau)| d\tau + |d_j(x, u, t)| + k_j \bar{x} e^{-\lambda_j t}, \quad (6.5)$$

where k_j and λ_j are positive constant chosen such that $|C_j e^{A_0 t}| < k_j e^{-\lambda_j t}$, $\forall t > 0$ and $\bar{x}$ is an upper bound for $|x|$, that is $|x| \leq \bar{x}, \forall x \in \chi$.

A fault is *detected* when at least one component of the output estimation error $|\epsilon_{y_j}^0(t)|$ reaches its corresponding threshold $\bar{\epsilon}_{y_j}^0(t)$ defined as

$$\bar{\epsilon}_{y_j}^0(t) \triangleq \int_0^t k_j e^{-\lambda_j(t-\tau)} (||L||_2 \bar{d} + \bar{\eta}) d\tau + \bar{d}_j + k_j \bar{x} e^{-\lambda_j t}, \quad (6.6)$$

and the detection time T_d is given by

$$T_d \triangleq \min_j \{T_j^d\}, \quad (6.7)$$

where

$$T_j^d = \inf \{t : |\epsilon_{y_j}^0(t)| \geq \bar{\epsilon}_{y_j}^0(t)\}.$$

Remark 6.1.2 As is well known, the above fault detection scheme has nice robustness properties in that threshold (6.6) guarantees *no false-positive alarms* before the fault occurrence time thanks to the use of the known upper bounds on $\eta(x, u, t)$ and $d(x, u, t)$. However, robustness is obtained at the price of conservativeness and deteriorated detectability properties. In the next section, this important issue is addressed by an enhanced detectability analysis and a modification of the detection thresholds.

6.2 Enhanced Fault Detectability Analysis

In the literature (see, for instance, [114, 143, 148]), the fault detectability analysis characterizes the set of detectable faults in an implicit way, where this set depends on unknown quantities like, for example, the fault occurrence time T_0 . The following Theorem 6.2.1 characterizes in explicit way the set of detectable sensor faults before that is computable off-line before the activation of the fault detection scheme.

Theorem 6.2.1 a) Assume that a sensor drift fault occurs at an unknown time T_0 in s -th sensor, $s \in \{1, \dots, p\}$. Define

$$\underline{\theta}^s \triangleq \min_{j \in \mathbb{Z}^s} \frac{2[\frac{k_j}{\lambda_j}(\|L\|_2 \bar{d} + \bar{\eta}) + \bar{d}_j]}{|-C_j A_0^{-1} L F^s + F_j^s|}, \quad (6.8)$$

where

$$\mathbb{Z}^s \triangleq \{j : |-C_j A_0^{-1} L F^s + F_j^s| \neq 0\}.$$

If the drift sensor fault satisfies the inequality satisfies $|\theta^s| > \underline{\theta}^s$, then it is detectable in finite time.

b) Assume that an arbitrary type of sensor fault $\phi_{ar}^s(T_0, t)$ occurs at an unknown time T_0 in the s -th sensor, $s \in \{1, \dots, p\}$. If the sensor fault satisfies

$$|\phi_{ar}^s(T_0, t)| > \underline{\theta}_{non}^s (1 - e^{-\alpha^s(t-T_0)}), \forall t \geq T_0, \quad (6.9)$$

where

$$\mathbb{Z}_{non}^s = \mathbb{Z}_{\Delta}^s \cap \mathbb{Z}^s,$$

$$\mathbb{Z}_{\Delta}^s = \mathbb{Z}_{ne}^s \cup \mathbb{Z}_{po}^s,$$

$$\mathbb{Z}_{ne}^s = \{j : |j - s| C_j e^{A_0 t} L F^s < 0, \forall t > 0\},$$

$$\mathbb{Z}_{po}^s = \{j : C_j e^{A_0 t} L F^s > 0, \forall t > 0\}$$

and

$$\underline{\theta}_{non}^s \triangleq \min_{j \in \mathbb{Z}_{non}^s} \frac{2[\frac{k_j}{\lambda_j}(\|L\|_2 \bar{d} + \bar{\eta}) + \bar{d}_j]}{|-C_j A_0^{-1} L F^s + F_j^s|},$$

with $\mathbb{Z}_{non}^s$ being not empty. Then, the sensor fault in s -th sensor is detectable in finite time. $\square$

Proof: a): Define a time $T_j^s \triangleq t_j^s + T_0$. At the time instant $T_j^s \geq T_0$, the j^{th} component of the output estimation error is

$$\begin{aligned} \epsilon_{y_j}^0(T_j^s) = & \int_0^{T_j^s} C_j e^{-A_0(\tau-T_j^s)} (\eta + Ld) d\tau + d_j(T_j^s) + \int_{T_0}^{T_j^s} C_j e^{-A_0(\tau-T_j^s)} L F^s \phi(\tau) d\tau \\ & + C_j e^{A_0 T_j^s} x(0) + F_j^s \phi(T_j^s). \end{aligned}$$

By the triangle inequality, the output estimation error is split as follows:

$$\begin{aligned} \left| \epsilon_{y_j}^0(T_j^s) \right| \geq & \left| \int_{T_0}^{T_j^s} C_j e^{-A_0(\tau-T_j^s)} L F^s \phi(\tau) d\tau + F_j^s \phi(T_j^s) \right| \\ & - \left(\left| \int_0^{T_j^s} C_j e^{-A_0(\tau-T_j^s)} (\eta + Ld) d\tau \right| + |d_j(T_j^s)| + |C_j e^{A_0 T_j^s} x(0)| \right). \end{aligned} \quad (6.10)$$

Letting $\omega = \tau - T_0$, we use the definition of threshold in (6.6) and get

$$\begin{aligned} \left| \epsilon_{y_j}^0(T_j^s) \right| > & \left| \int_0^{t_j^s} C_j e^{-A_0(\omega-t_j^s)} L F^s (1 - e^{-\alpha^s \omega}) d\omega \right. \\ & \left. + F_j^s (1 - e^{-\alpha^s t_j^s}) \right| \cdot |\theta^s| - \bar{\epsilon}_{y_j}^0(T_j^s). \end{aligned} \quad (6.11)$$

As θ^s is an unknown constant parameter, we consider $g_j^s(t_j^s)$ as a time-evolution of *estimation error* due to the occurrence of a sensor fault.

$$g_j^s(t_j^s) \triangleq \int_0^{t_j^s} C_j e^{-A_0(\omega-t_j^s)} L F^s (1 - e^{-\alpha^s \omega}) d\omega + F_j^s (1 - e^{-\alpha^s t_j^s}), \quad (6.12)$$

where A_0 can be factorized as

$$A_0 = V(\oplus_{i=1}^n \lambda_{c_i}) V^{-1}.$$

If the equation (6.13) for any j holds at certain moment t_j^s , then $\left| \epsilon_{y_j}^0(T_j^s) \right| > \bar{\epsilon}_{y_j}^0(T_j^s)$ and the drift fault in s^{th} sensor can be detected.

$$\left| g_j^s(t_j^s) \right| \cdot |\theta^s| = 2\bar{\epsilon}_{y_j}^0(T_j^s). \quad (6.13)$$

However, $|g_j^s(t_j^s)|$ is a nonlinear time-varying function with unknown evolution rate α^s , which cannot be analyzed explicitly. Consider the worst detectability scenario:

$$\left| g_j^s(t_j^s) \right| \cdot |\theta^s| < 2\bar{\epsilon}_{y_j}^0(T_j^s), \forall t_j^s > 0. \quad (6.14)$$

The corresponding fault detectability sufficient condition can be given by

$$\left| g_j^s(\infty) \right| \cdot |\theta^s| > 2\bar{\epsilon}_{y_j}^0(\infty). \quad (6.15)$$

So the set of detectable sensor drift faults can be characterized by a proper lower bound on θ^s without any on-line data, such as output measurements and estimates. After

some algebraic operations, the fault detection threshold (6.6) is given by

$$\bar{\epsilon}_{y_j}^0(T_j^s) = \frac{k_j}{\lambda_j} (\|L\|_2 \bar{d} + \bar{\eta}) (1 - e^{-\lambda_j T_j^s}) + \bar{d}_j + k_j \bar{x} e^{-\lambda_j T_j^s}. \quad (6.16)$$

We have

$$\lim_{t_j^s \rightarrow \infty} 2\bar{\epsilon}_{y_j}^0(T_j^s) = 2\left[\frac{k_j}{\lambda_j} (\|L\|_2 \bar{d} + \bar{\eta}) + \bar{d}_j\right]. \quad (6.17)$$

Then the convergence of $|g_j^s(t_j^s)|$ is analyzed in the following part.

$$g_j^s(t_j^s) = \int_0^{t_j^s} C_j V G_1(t_j^s, \omega) V^{-1} L F^s d\omega + F_j^s (1 - e^{-\alpha^s t_j^s}),$$

where

$$G_1(t_j^s, \omega) = \oplus_{i=1}^n \varphi_i(t_j^s, \omega)$$

and for $i = 1, \dots, n$,

$$\varphi_i(t_j^s, \omega) = e^{\lambda_{c_i} t_j^s} (e^{-\lambda_{c_i} \omega} - e^{-(\lambda_{c_i} + \alpha^s) \omega}).$$

We firstly assume that $\lambda_{c_i} \neq -\alpha^s$, after some algebraic operations, we get a new equation with $\xi_i(t_j^s)$,

$$g_j^s(t_j^s) = C_j V G_2(t_j^s) V^{-1} L F^s + F_j^s (1 - e^{-\alpha^s t_j^s}), \quad (6.18)$$

where

$$G_2(t_j^s) = \oplus_{i=1}^n \xi_i(t_j^s)$$

and for $i = 1, \dots, n$,

$$\xi_i(t_j^s) = \frac{1}{\lambda_{c_i} (\lambda_{c_i} + \alpha^s)} (\lambda_{c_i} e^{-\alpha^s t_j^s} + \alpha^s e^{\lambda_{c_i} t_j^s}) - \frac{1}{\lambda_{c_i}}. \quad (6.19)$$

Finally $\xi_i(t_j^s)$ converges to a positive constant shown below.

$$\lim_{t_j^s \rightarrow \infty} \xi_i(t_j^s) = -\frac{1}{\lambda_{c_i}} \quad (6.20)$$

Then consider the special scenario that $\lambda_{c_i} = -\alpha^s$ and we obtain an alternative equation (6.21) with $\xi_i'(t_j^s)$.

$$g_j^s(t_j^s) = C_j V G_3(t_j^s) V^{-1} L F^s + F_j^s (1 - e^{-\alpha^s t_j^s}), \quad (6.21)$$

where

$$G_3(t_j^s) = \oplus_{i=1}^n \xi_i'(t_j^s)$$

and

$$\xi_i'(t_j^s) = \frac{1}{\lambda_{c_i}} (e^{\lambda_{c_i} t_j^s} - 1) - t_j^s e^{\lambda_{c_i} t_j^s}, i = 1, \dots, n, \quad (6.22)$$

$$\lim_{t_j^s \rightarrow \infty} \xi_i'(t_j^s) = -\frac{1}{\lambda_{c_i}}. \quad (6.23)$$

In conclusion, no matter any eigenvalue λ_{c_i} is chosen, the time-varying parts of the equation (6.18) or (6.21) will converge to some known constants after a sufficiently long time.

Given

$$G_4 = \oplus_{i=1}^n \left(-\frac{1}{\lambda_{c_i}}\right),$$

we have

$$\begin{aligned} \lim_{t_j^s \rightarrow \infty} |g_j^s(t_j^s)| &= |C_j V G_4 V^{-1} L F^s + F_j^s| \\ &= |-C_j A_0^{-1} L F^s + F_j^s| \end{aligned} \quad (6.24)$$

If the norm of sensor fault saturation level θ^s is large than the ratio between limit values of $|g_j^s(t_j^s)|$ and $2\bar{\epsilon}_{y_j}^0(T_j^s)$, the equation (6.13) will certainly hold in a finite-time. Based on the fault detection logic in 6.1.2, a fault is detected as long as one component of the output estimation error $|\epsilon_{y_j}^0(t)|$ exceeds its corresponding threshold $\bar{\epsilon}_{y_j}^0(t)$. Hence, among all the valid ratio for different index j , the smallest one can be regarded as the lower bound on $|\theta^s|$. In conclusion, if the condition (6.8) is satisfied, there exists at least one index j that the equation (6.13) holds and the sensor drift can be certainly detected in finite time, thus concluding the proof of Part a).

b): Part a) of Theorem 6.2.1 represents a set of detectable sensor drift fault $\phi(\theta^s)$. In case of a non empty set $\mathbb{Z}_{non}^s$, a more general type of detectable sensor fault can be characterized. Consider an arbitrary type of sensor fault ϕ_{ar}^s , the inequality (6.11) for component j^* is replaced by

$$\left| \epsilon_{y_{j^*}}^0(T_{j^*}^s) \right| > E_{j^*}^s(t_{j^*}^s) - \bar{\epsilon}_{y_{j^*}}^0(T_{j^*}^s), \quad (6.25)$$

where

$$E_{j^*}^s(t_{j^*}^s) \triangleq \left| \int_0^{t_{j^*}^s} F_{j^*}^s \phi_{ar}^s(T_0, T_0 + t_{j^*}^s) + C_{j^*} e^{-A_0(\omega - t_{j^*}^s)} L F^s \phi_{ar}^s(T_0, T_0 + \omega) d\omega \right| \quad (6.26)$$

and

$$j^* \triangleq \arg \min_{j \in \mathbb{Z}_{non}^s} \frac{2[\frac{k_j}{\lambda_j}(\|L\|_2 \bar{d} + \bar{\eta}) + \bar{d}_j]}{|-C_j A_0^{-1} L F^s + F_j^s|}. \quad (6.27)$$

Clearly $j^* \in \mathbb{Z}_{\Delta}^s$; therefore, the term $C_{j^*} e^{A_0 t} L F^s$ will not change sign at any time. If the condition (6.9) is satisfied,

$$E_{j^*}^s(t_{j^*}^s) > \left| \int_0^{t_{j^*}^s} C_{j^*} e^{-A_0(\omega - t_{j^*}^s)} L F^s (1 - e^{-\alpha^s \omega}) d\omega + F_{j^*}^s (1 - e^{-\alpha^s t_{j^*}^s}) \right| \cdot \underline{\theta}_{non}^s \quad (6.28)$$

Based on the proof of a), $|g_{j^*}^s(\infty)| \cdot \underline{\theta}_{non}^s = 2\bar{\epsilon}_{j^*}^0(\infty)$. Hence,

$$\begin{aligned} \left| \bar{\epsilon}_{y_{j^*}}^0(\infty) \right| &> E_{j^*}^s(\infty) - \bar{\epsilon}_{y_{j^*}}^0(\infty) \\ &> |g_{j^*}^s(\infty)| \cdot \underline{\theta}_{non}^s - \bar{\epsilon}_{y_{j^*}}^0(\infty) \\ &= \bar{\epsilon}_{y_{j^*}}^0(\infty) \end{aligned} \quad (6.29)$$

In conclusion, if the condition (6.9) is satisfied, there exists one index $j^* \in \mathbb{Z}_{non}^s$ that the sensor fault ϕ_{ar}^s can be certainly detected in finite-time, thus concluding the proof. $\square$

Remark 6.2.1 As the terms T_0 and α^s are both unknown, the worst-case detectability scenario is considered to derive the fault detectability condition in Theorem 6.2.1. Although this sufficient condition is fairly conservative, it is useful to characterize an off-line class of detectable sensor faults. And the condition in Theorem 6.2.1 a) and b) is pictorially shown in the following two figures, respectively.

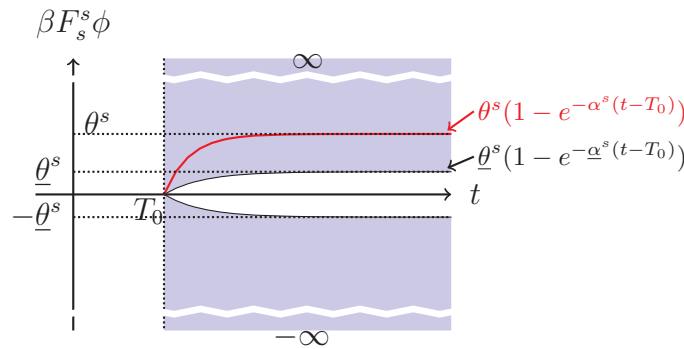


Fig. 6.2 Time-evolution of a sensor fault (red curve) and the class of detectable faults in s^{th} sensor (purple shade).

6.3 Detection Time Interval Analysis

$$t_d^s = T_d^s - T_0.$$
$$t_d^s = \min_j \{T_{d_j}^s\} - T_0, \quad (6.30)$$

where $T_{d_j}^s = \inf\{T_j^s : |\epsilon_{y_j}^0(T_j^s)| \geq \bar{\epsilon}_{y_j}^0(T_j^s)\}$, $j \in \{1, \dots, p\}$. The detectability of a sensor diagnosis approach is not only evaluated by the range of detectable sensor faults, but

classified by the duration of detection time interval as well. It is fairly necessary to detect an abrupt sensor fault rapidly in case the system malfunctions instantly. However, no matter how slow a sensor fault deteriorates, a quick fault detection is beneficial to avoid system failures due to fault propagation. There exists few analytical analysis of detection time interval in previous literatures; therefore, the following theorem defines an analytical upper bound on the detection time interval of sensor faults and reveals a relationship between the upper bound and the set of detectable sensor fault.

Theorem 6.3.1 *Assume that a sensor fault occurs at an unknown time T_0 in s^{th} sensor, $s \in \{1, \dots, p\}$. If there exists a non empty set $\mathbb{Z}_\Delta^s$, then consider a choice of $k_j^s, \lambda_j^s \in \mathbb{R}^+$. such that for $j \in \mathbb{Z}_\Delta^s$*

$$\mathbb{Z}_d^s \triangleq \{(k_j^s, \lambda_j^s) : |C_j e^{A_0 t} L F^s| > k_j^s e^{-\lambda_j^s t}, \forall t > 0\}. \quad (6.31)$$

a) The upper bound on detection time interval t_d^s is given by

$$\bar{t}_d^s = \min_{j \in \mathbb{Z}_\Delta^s} \{\bar{t}_{d_j}^s\}, \quad (6.32)$$

where

$$\bar{t}_{d_j}^s = \inf_{t_j^s \in \mathbb{R}^+} \{t_j^s : \underline{g}_j^s(t_j^s) \cdot |\theta^s| = 2\epsilon_{y_j}^0(t_j^s + T_0'), (k_j^s, \lambda_j^s) \in \mathbb{Z}_d^s\},$$

$$\underline{g}_j^s = \int_0^{t_j^s} k_j^s e^{-\lambda_j^s(t_j^s - \omega)} (1 - e^{-\alpha^s \omega}) d\omega + F_j^s (1 - e^{-\alpha^s t_j^s}),$$

and

$$T_0' = \begin{cases} 0 & \text{if } \frac{k_j}{\lambda_j} (\|L\|_2 \bar{d} + \bar{\eta}) \leq k_j \bar{x} \\ \infty & \text{otherwise.} \end{cases}$$

b) The upper bound $\bar{t}_d^s$ decreases monotonically as the norm of sensor fault saturation level $|\theta^s|$ increases. $\square$

Proof: a): Define a time $T_j^s \triangleq t_j^s + T_0$, the term t_j^s can be regarded as time interval. According to Theorem 6.2.1, assuming there exists a non empty set $\mathbb{Z}_\Delta^s$ for index j , the term $C_j e^{A_0 t} L F^s$ does not change sign and a pair (k_j^s, λ_j^s) satisfying the condition in (6.31) is chosen for each $j \in \mathbb{Z}_\Delta^s$. Hence we have

$$|C_j e^{A_0 t_j^s} L F^s| > k_j^s e^{-\lambda_j^s t_j^s}, \forall t_j^s > 0$$

$$|g_j^s(t_j^s)| > \underline{g}_j^s(t_j^s), \quad \forall t_j^s > 0. \quad (6.33)$$

As the occurrence time of sensor fault T_0 is unknown, time T_0' is introduced to update (6.16). Hence, $\forall t_j^s > 0$ and $\forall T_0 > 0$. we have

$$\bar{\epsilon}_{y_j}^0(T_j^s) \leq \bar{\epsilon}_{y_j}^0(t_j^s + T_0') \quad (6.34)$$

From (6.11), (6.12), (6.33) and (6.34), the norm of output estimation error at time $\bar{t}_{d_j}^s + T_0$ is given by

$$|\epsilon_{y_j}^0(\bar{t}_{d_j}^s + T_0)| > \underline{g}_j^s(\bar{t}_{d_j}^s) \cdot |\theta^s| - \bar{\epsilon}_{y_j}^0(\bar{t}_{d_j}^s + T_0'). \quad (6.35)$$

Assume the condition in (6.32) is satisfied, we have

$$\begin{aligned} \underline{g}_j^s(\bar{t}_{d_j}^s) \cdot |\theta^s| &= 2\bar{\epsilon}_{y_j}^0(\bar{t}_{d_j}^s + T_0'), \\ |\epsilon_{y_j}^0(\bar{t}_{d_j}^s + T_0)| &> \bar{\epsilon}_{y_j}^0(\bar{t}_{d_j}^s + T_0). \end{aligned} \quad (6.36)$$

Based on the definition in (6.30), time $T_{d_j}^s$ is the first moment when the j^{th} residual reaches its corresponding threshold. However, it is clear that at time $\bar{t}_{d_j}^s + T_0$ the j^{th} residual already passed its corresponding threshold. So we get

$$\bar{t}_{d_j}^s + T_0 > T_{d_j}^s, \quad \forall j \in \mathbb{Z}_{\Delta}^s$$

$$\min_{j \in \mathbb{Z}_{\Delta}^s} \{\bar{t}_{d_j}^s\} + T_0 > \min_{j \in \mathbb{Z}_{\Delta}^s} \{T_{d_j}^s\} \geq \min_{j \in \{1, \dots, p\}} \{T_{d_j}^s\}$$

So

$$\bar{t}_d^s > t_d^s.$$

In conclusion, given a pair $(k_j^s, \lambda_j^s) \in \mathbb{Z}_{\Delta}^s$, if the condition (6.32) is satisfied, the upper bound on detection time interval t_d^s is given by $\bar{t}_d^s$.

b): Part a) of Theorem 6.3.1 characterizes the upper bound on detection time interval t_d^s in an implicit way, which means the upper bound cannot be calculated unless a new assumption on the lower bound of sensor fault saturation level is introduced. However, a relationship between the upper bound and the sensor fault saturation level

can be properly analyzed. By evaluating $\underline{g}_j^s$ we obtain

$$\begin{aligned}\underline{g}_j^s &= \int_0^{t_j^s} k_j^s e^{-\lambda_j^s(t_j^s - \omega)} (1 - e^{-\underline{\alpha}^s \omega}) d\omega + F_j^s (1 - e^{-\underline{\alpha}^s t_j^s}) \\ &= k_j^s e^{-\lambda_j^s t_j^s} \int_0^{t_j^s} e^{\lambda_j^s \omega} (1 - e^{-\underline{\alpha}^s \omega}) d\omega + F_j^s (1 - e^{-\underline{\alpha}^s t_j^s}) \\ &= \frac{k_j^s}{\lambda_j^s} (1 - e^{-\lambda_j^s t_j^s}) + \frac{k_j^s}{-\underline{\alpha}^s + \lambda_j^s} (e^{-\lambda_j^s t_j^s} - e^{-\underline{\alpha}^s t_j^s}) + F_j^s (1 - e^{-\underline{\alpha}^s t_j^s})\end{aligned}$$

Taking the derivative of $\underline{g}_j^s$ with respect to t_j^s , we have

$$\begin{aligned}\frac{d\underline{g}_j^s}{dt_j^s} &= k_j^s e^{-\lambda_j^s t_j^s} + \frac{k_j^s}{-\underline{\alpha}^s + \lambda_j^s} (-\lambda_j^s e^{-\lambda_j^s t_j^s} + \underline{\alpha}^s e^{-\underline{\alpha}^s t_j^s}) + F_j^s \underline{\alpha}^s e^{-\underline{\alpha}^s t_j^s} \\ &= \frac{e^{-\underline{\alpha}^s t_j^s} - e^{-\lambda_j^s t_j^s}}{-\underline{\alpha}^s + \lambda_j^s} \cdot k_j^s \underline{\alpha}^s + F_j^s \underline{\alpha}^s e^{-\underline{\alpha}^s t_j^s}\end{aligned}$$

Given $\frac{e^{-\underline{\alpha}^s t_j^s} - e^{-\lambda_j^s t_j^s}}{-\underline{\alpha}^s + \lambda_j^s} > 0, \forall t_j^s > 0$, therefore $\frac{d\underline{g}_j^s}{dt_j^s} > 0$ and $\underline{g}_j^s(t_j^s)$ is monotonically increasing. Considering $\bar{\epsilon}_{y_j}^0(t_j^s + T_0')$, we get

$$\begin{cases} \text{monotonically decreasing} & \text{if } \frac{k_j}{\lambda_j} (\|L\|_2 \bar{d} + \bar{\eta}) \leq k_j \bar{x} \\ \text{constant} & \text{otherwise} \end{cases}$$

Hence, function $\frac{2\bar{\epsilon}_{y_j}^0(t_j^s + T_0')}{\underline{g}_j^s(t_j^s)}$ is monotonically decreasing. It is clear that all the solutions $\bar{t}_{d_j}^s$ satisfy the equation $|\theta^s| = \frac{2\bar{\epsilon}_{y_j}^0(t_j^s + T_0')}{\underline{g}_j^s(t_j^s)}$; therefore, the upper bound $\bar{t}_d^s$ decreases monotonically as the norm of sensor fault saturation level $|\theta^s|$ increases. $\square$

Remark 6.3.1 Unlike Theorem 6.2.1, which describes an off-line assessment approach, the upper bound shown above cannot be calculated independently due to few information on parameter $|\theta^s|$. However, Theorem 6.2.1 introduces a conservative lower bound of sensor fault saturation level, whose sensor fault can be detected at least after a sufficiently long time. Based on the result in Theorem 6.3.1, if the lower bound $\underline{\theta}_{non}^s$ increases, the upper bound $\bar{t}_d^s$ will drops from a sufficient long time to a fairly short time interval.

6.4 Simulation Results

In this section, a nonlinear uncertain system is introduced to illustrate the analytical analysis of sensor fault detectability of the fault detection scheme presented by Zhang et al. [159]. Consider a single-link robotic arm with a revolute elastic joint rotating in a vertical plane, the motion equations are given by

$$\begin{aligned} J_l \ddot{q}_1 + F_l \dot{q}_1 + k(q_1 - q_2) + mgl \sin q_1 &= 0 \\ J_m \ddot{q}_2 + F_m \dot{q}_2 - k(q_1 - q_2) &= u, \end{aligned}$$

where q_1 and q_2 are the link displacement and the rotor displacement, respectively. All the parameters are positive constants. The control u is the torque delivered by the motor. By choosing $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$, $x_4 = \dot{q}_2$, and assuming that x_1 , x_3 and x_4 are measurable, the nonlinear model can be derived as the form in (6.1),

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k}{J_l} & \frac{-F_l}{J_l} & \frac{k}{J_l} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_m} & 0 & \frac{-k}{J_m} & \frac{-F_m}{J_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-mgl}{J_l} \sin x_1 \\ 0 \\ \frac{u}{J_m} \end{bmatrix} + \begin{bmatrix} 0 \\ \eta \\ 0 \\ 0 \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ d_2 \\ d_3 \end{bmatrix} + \beta(t - T_0)F\phi, \end{aligned}$$

where the parameters used in the simulation are given: $k = 2$, $F_m = 1$, $F_l = 0.5$, $J_m = 1$, $J_l = 2$, $m = 4$, $g = 9.8$ and $l = 0.5$. Compared with the motion equations, the state-space model is amended by considering the modeling uncertainty and sensor fault. The system states are bounded by $[0.5; 0.2; 5; 2]$. The modeling uncertainty $\eta = \frac{0.45mgl}{J_l} \sin x_1$ is assumed to be up to 5% error in the mass of the link. The modeling uncertainty in the output is defined to be: $d_2 = 0.01 \sin 5t$ and $d_3 = 0.01 \sin 2t$. Hence, the upper bounds on modeling uncertainty are given by $\bar{\eta} = \frac{0.05mgl}{J_l} |\sin 0.5|$, $\bar{d}_2 = \bar{d}_3 = 0.01$. A fault detection estimator in Section 6.1.2 is constructed with the input given by $u = 8 \sin \frac{t}{3}$. The initial conditions of system states are all chosen to be zero and the poles of A_0 are located at -2.1 , -1.5 , -1.7 , and -1.9 . Hence, the chosen constants are given by $k_1 = 3.35$, $\lambda_1 = 1.259$, $k_1 = 1.80$, $\lambda_2 = 1.49$, and $k_3 = 3.03$, $\lambda_3 = 2.09$.

Assuming an ideal drift sensor fault $\phi = 0.7(1 - e^{-0.04(t-20)})$ occurs in Sensor 3 at time $20s$, we have $\mathbb{Z}^3 = \{1, 3\}$, $\underline{\theta}^3 = 0.5478$ and $\underline{\alpha}^3 = 0.02$. Fig. 6.4 shows the set of

detectable idea drift sensor fault and an ideal drift sensor fault in this set is detected at time $T_d^3 = T_{d_3}^3 = 31.47s$.

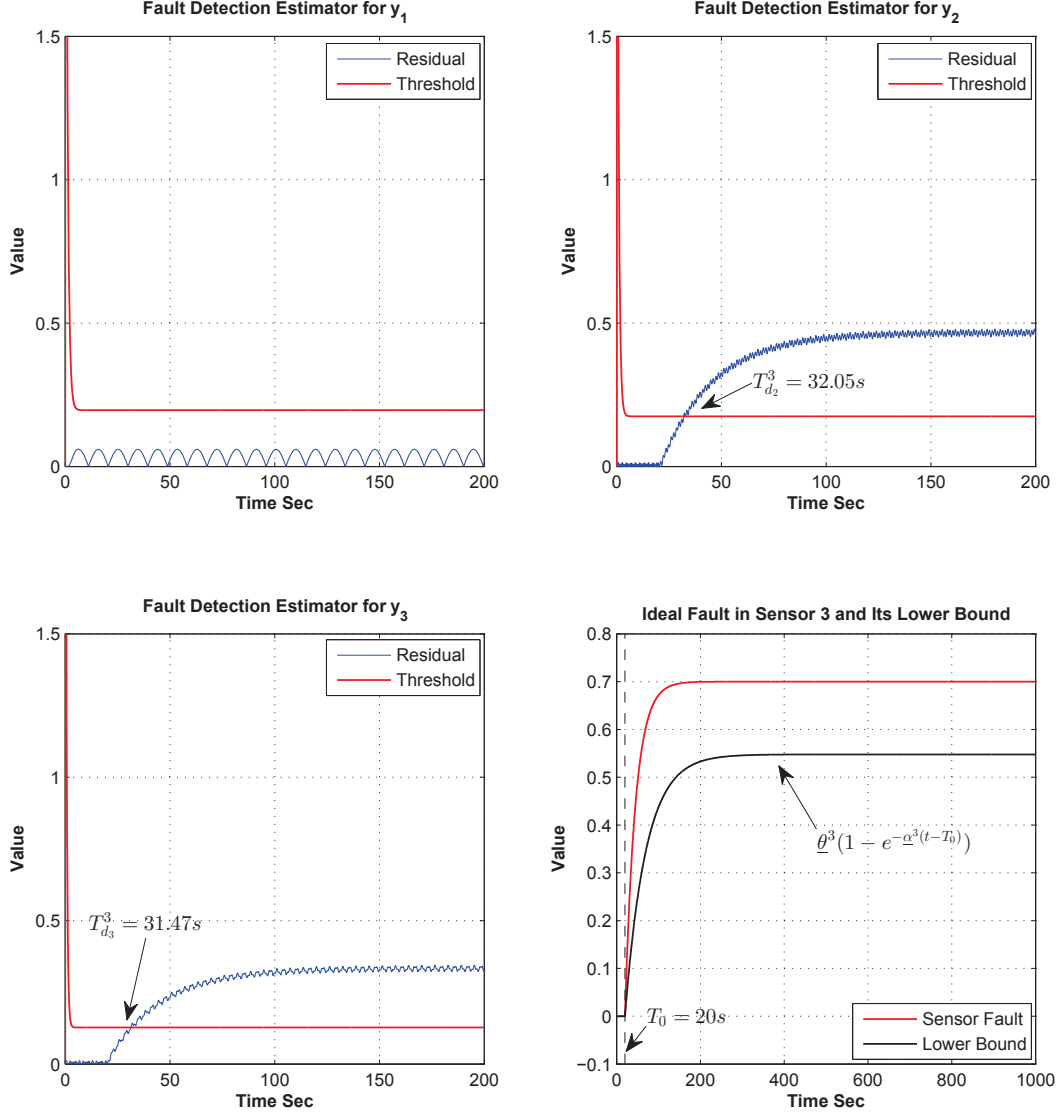


Fig. 6.4 Ideal drift sensor fault detected in Sensor 3.

Consider an arbitrary drift sensor fault $\phi_{ar} = 0.5(1 - e^{-0.03(t-20)}) + 0.05 \sin 0.05t + \omega$ occurring in Sensor 2 at time $20s$, where ω is stochastic noise. We get $\mathbb{Z}^2 = \{1, 3\}$, $\mathbb{Z}_{\Delta}^2 = \{3\}$, $\mathbb{Z}_{non}^2 = \{3\}$, $\theta_{non}^2 = 0.3317$ and $\alpha^2 = 0.01$. Fig. 6.5 illustrates the set of detectable arbitrary drift sensor fault and an arbitrary drift sensor in this set fault is detected at time $T_d^2 = T_{d_3}^2 = 26.82s$.

There exists non empty sets $\mathbb{Z}_{\Delta}^1 = \{3\}$, $\mathbb{Z}_{\Delta}^2 = \{3\}$ and $\mathbb{Z}_{\Delta}^3 = \{2\}$. Moreover, the corresponding constant pairs are given by $(k_3^1, \lambda_3^1) = (2, 2.105)$, $(k_3^2, \lambda_3^2) = (2, 2.105)$ and $(k_2^3, \lambda_2^3) = (1, 1.51)$. We consider all three cases in which an ideal sensor fault

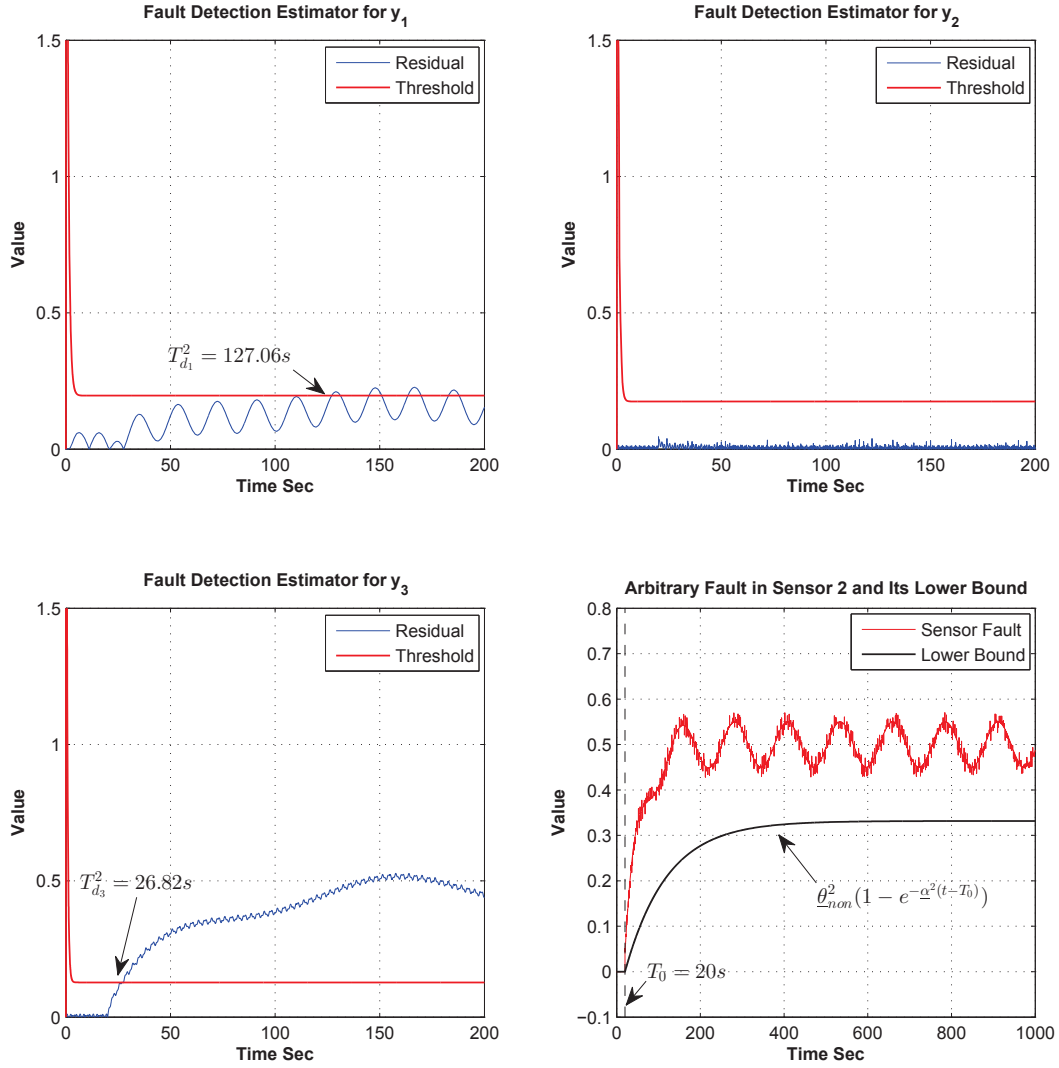


Fig. 6.5 Arbitrary drift sensor fault in Sensor 2 .

occurs in each sensor owing to the existence of non empty set $\mathbb{Z}_{\Delta}^s$ for any s . Given a group of various sensor fault saturation level θ^s , the upper bound on detection time interval can be calculated and illustrated in Fig. 6.6. For any sensor, there always exists a limit that the saturation level of a sensor fault is too small so that the sensor fault cannot be detected. The calculation of upper bound is based on the same set $\mathbb{Z}_{\Delta}^s$ used in b) of Theorem 6.2.1; therefore, each sufficient upper bound curve converges to its corresponding θ_{non}^s . Then we apply ideal sensor faults with a wider range of θ^s occurring at random chosen time in the sensor fault detection scheme, it is clear that the actual fault detection time interval curve is bounded by the red curve in Fig. 6.6.

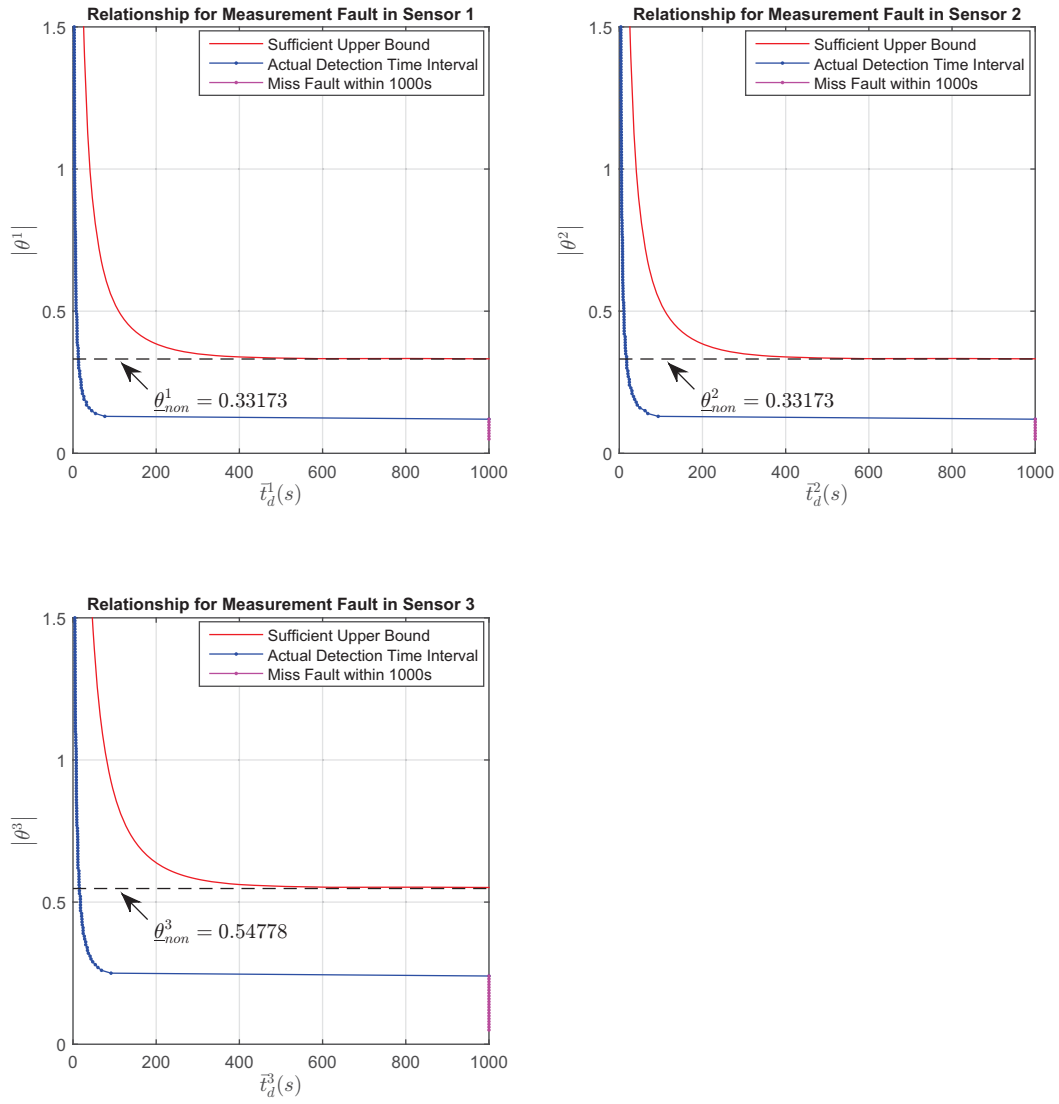


Fig. 6.6 Relationship between upper bound $\bar{t}_d^s$ and norm of sensor fault saturation level $|\theta^s|$.

6.5 Concluding Remarks

The automated fault diagnosis approaches attract significant industrial interest owing to their considerable economic benefits. Firstly, automated health monitoring systems using sensor networks are highly needed by industry as manually health monitoring for large-scale system is no longer affordable. And the deterioration of control systems due to a failure costs a lot. Sensor fault detection is one of the vital objectives in industry as any fault in sensor networks may cause a failure in controllers or a malfunction in neighboring systems, respectively.

In this chapter, the off-line assessment of fault detectability and analysis of detection time interval upper bounds are described. A conservative set of detectable sensor fault is characterized without consideration given to the detection time. Based on the relationship between fault detectability and modeling efforts, a flexible sensor fault diagnosis is available by applying various modeling efforts. The further analysis on fault detection time interval can evaluate the capability of fast fault detection as well.

Chapter 7

CONCLUSIONS AND FUTURE PROSPECTS

7.1 Concluding Remarks

Sensor networks play a vital role to gather and refine information from a large amount of raw data in the physical operations, especially in the monitoring of modern systems, such as water management, industrial processes, traffic management, and smart cities. The monitoring architecture is a long-standing research topic and has been developed in various aspects by a wide variety of techniques. In this thesis, we mainly focus on the tracking precision and reliability of the monitoring architecture, where two distributed state estimation algorithms are proposed to track the modern systems in centralized, decentralized, or distributed structure. Meanwhile, various fault detection algorithms are introduced to enable the sensor networks to detect both process faults in the monitored systems and sensor faults in the sensor networks. The thesis provides a comprehensive review of the literature on two research fields. First part is about the distributed estimation approaches, such as distributed Kalman filter and distributed state estimation based on MSE analysis. Second part is about the fault detection methods, such as process fault detection, distributed fault detection in sensor networks, and sensor fault detection.

This thesis proposes a monitoring architecture, where two distributed state estimation algorithms are implemented by every node in the sensor networks. The algorithms both take advantage of the two-step strategy, widely proposed in the distributed Kalman filter methods; and the optimization frameworks. Hence, the local estimator in each sensor node consists of a filtering step – which uses a weighted combination of neighboring sensors information – and a model-based state predictor of the moni-

tored system. The filtering weights and prediction parameters are jointly optimized to minimize both the mean and variance of the prediction error in a multi-objective Pareto optimization framework at each time step. Unlike the distributed Kalman filter methods, where only the asymptotic convergence is achieved, the proposed methods can also provide good instantaneous performance so that the unwanted transient bias and variance of the estimation error can be avoided at each time step. On the other hand, the two-step strategy enables the sensor networks to fuse measurements and estimates at the same time step also to predict the monitored variables. Moreover, in the monitoring architecture for the centralized or decentralized system, each sensor network can use the algorithm proposed in Chapter 2 to monitor a centralized system or a subsystem of the decentralized system. However, in the architecture for distributed systems, another distributed estimation algorithm is introduced in Chapter 3, which can coordinate several sensor networks to monitor their corresponding subsystems of the distributed system.

The proposed monitoring architecture in this thesis is able to track the system's state while simultaneously detecting the faults occurred in either the systems or the sensor networks. There exists a broad range of literature dealing with the problem of fault detection for sensor network, whereas the use of sensor networks for detecting process faults is a relatively rare field of research. Hence, the distributed state estimation algorithm in a sensor network is applied to design process fault detection algorithms for a centralized or decentralized system, where suitable stochastic thresholds are designed, allowing to set the parameters so as to guarantee a maximum false alarms probability. Thanks to the support of the research project granted by ABB Research, the centralized, decentralized, and distributed sensor fault detection methods are presented in a discrete-time framework to test the performance in an experimental continuous stirred tank heater pilot plant. A benchmark simulation is presented to compare the performance of all three architectures. And the computer algorithm pseudo-codes are also introduced. Finally, a further analysis on the sensor fault detectability is proposed for the fault detection scheme in the monitoring architecture of centralized or decentralized systems, which offers the explicit off-line assessment of fault detectability and detection time interval.

The performance of the proposed distributed estimation methods has been tested and compared with other existing approaches via extensive numerical simulations. The effectiveness of presented fault detection methods is evaluated by extensive numerical and industrial benchmark simulations.

7.2 Future Work

This section is devoted to highlighting possible extensions of the proposed approaches. These extensions can be developed in two aspects: more comprehensive theories and potential applications. From a theoretical point of view, the distributed process fault detection scheme is addressed only in the monitoring architecture for centralized and decentralized systems. Further research efforts can focus on the distributed process fault detection algorithm over multiple sensor networks. Due to the intensive interaction among sensor nodes inside the sensor network and throughout neighboring sensor networks, a sophisticated communication protocol over sensor networks is essential to reduce the communication volume at each time step. Another aspect where the proposed distributed estimation algorithms should be improved is the monitoring architecture with adaptive approximation capability. Hence, each sensor node in the architecture is able to learn the nonlinear uncertainties in the monitored systems or the interconnection uncertainties in the distributed systems. These enhanced distributed estimation methods over sensor networks can significantly expand the range of applicable modern systems to be monitored. Furthermore, in this thesis, the topology of sensor networks in the proposed distributed estimation algorithms is given and fixed. Nodes in the networks are arbitrarily placed in the deployment region; therefore, the graph representing the communication links between the nodes is usually unstructured. Since the changing topology of sensor networks is due to the node mobility and/or link failure, apart from the sensor fault detection algorithms in the monitoring architecture, future research should explore the robust distributed estimation algorithms in greater details so that the architecture can work in sensor networks with dynamically changing topology.

In terms of the potential application of the proposed algorithms, in cooperation with ABB research, we have proposed three sensor fault detection architectures for an experimental continuous stirred tank heater pilot plant. The proposed distributed estimation algorithms can be applied to a more complex system for future research, such as the Tennessee Eastman Process, illustrated in Fig. 7.1. The Tennessee Eastman Process benchmark problem is discussed in [43, 123], in where the whole process is decomposed into four subsystems to be controlled. The challenging of this process is that the dynamic model of the systems are difficult to derived in real-world industrial situations by analytical laws. But even so, in academic field, researchers have proposed a number of promising methods to determine the model by suitable input-output system identification procedures. Then this distributed process can be tracked by the proposed monitoring architecture. Another interesting research topic could be to refine



the decomposition of the whole process so that the detectability of the process fault detection methods can be improved, for instance, the extent of the class of detectable faults.

References

- [1] Abdelgawad, A. and Bayoumi, M. (2011). Low-power distributed kalman filter for wireless sensor networks. *EURASIP Journal on Embedded Systems*, 2011:3.
- [2] Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3):325–349.
- [3] Alippi, C. and Galperti, C. (2008). An adaptive system for optimal solar energy harvesting in wireless sensor network nodes. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(6):1742–1750.
- [4] Alriksson, P. and Rantzer, A. (2006). Distributed Kalman filtering using weighted averaging. In *17th Int. Symposium on Mathematical Theory of Networks and Systems*, pages 1–60.
- [5] Arablouei, R., Doğançay, K., Werner, S., and Huang, Y.-F. (2016). On the asymptotic bias of the diffusion-based distributed pareto optimization. *Signal Processing*.
- [6] Baheti, R. and Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12:161–166.
- [7] Bamieh, B., Paganini, F., and Dahleh, M. A. (2002). Distributed control of spatially invariant systems. *IEEE Transactions on automatic control*, 47(7):1091–1107.
- [8] Battistelli, G., Benavoli, A., and Chisci, L. (2012). Data-driven communication for state estimation with sensor networks. *Automatica*, 48(5):926–935.
- [9] Battistelli, G. and Chisci, L. (2014). Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50(3):707–718.
- [10] Bauer, P. H., Premaratne, K., and Duran, J. (1993). A necessary and sufficient condition for robust asymptotic stability of time-variant discrete systems. *IEEE Transactions on Automatic Control*, 38(9):1427–1430.
- [11] Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2003). *Diagnosis and Fault Tolerant Control*. Springer, Berlin.
- [12] Boem, F. (2013). Distributed methods for estimation and fault diagnosis: the case of large-scale networked systems. In *PhD Thesis*. Università degli studi di Trieste.

- [13] Boem, F., Ferrari, R., Parisini, T., and Polycarpou, M. (2011a). Distributed fault detection and isolation of continuous-time nonlinear systems. *European Journal of Control*, 5-6:603–620.
- [14] Boem, F., Ferrari, R. M., Parisini, T., and Polycarpou, M. M. (2011b). A distributed fault detection methodology for a class of large-scale uncertain input-output discrete-time nonlinear systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 897–902. IEEE.
- [15] Boem, F., Xu, Y., Fischione, C., and Parisini, T. (2012). A distributed estimation method for sensor networks based on Pareto optimization. In *51th IEEE Conference on Decision and Control*, pages 775–781.
- [16] Boem, F., Xu, Y., Fischione, C., and Parisini, T. (2013). Distributed fault detection using sensor networks and Pareto estimation. In *European Control Conference*, pages 932–937.
- [17] Boem, F., Xu, Y., Fischione, C., and Parisini, T. (2015). A distributed Pareto-optimal dynamic estimation method. In *European Control Conference*, pages 3678–3685.
- [18] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- [19] Caballero-Águila, R., García-Garrido, I., and Linares-Pérez, J. (2014). Information fusion algorithms for state estimation in multi-sensor systems with correlated missing measurements. *Applied Mathematics and Computation*, 226:548–563.
- [20] Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2007). Distributed Kalman filtering using consensus strategies. In *46th IEEE Conference on Decision and Control*, pages 5486–5491.
- [21] Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008). Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633.
- [22] Cattivelli, F. S., Lopes, C. G., and Sayed, A. H. (2008a). Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, 56(5):1865–1877.
- [23] Cattivelli, F. S., Lopes, C. G., and Sayed, A. H. (2008b). Diffusion strategies for distributed kalman filtering: formulation and performance analysis. *Proceedings of the Cognitive Information Processing*, pages 36–41.
- [24] Cattivelli, F. S. and Sayed, A. H. (2010a). Diffusion lms strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035–1048.
- [25] Cattivelli, F. S. and Sayed, A. H. (2010b). Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084.

- [26] Cattivelli, F. S. and Sayed, A. H. (2011a). Analysis of spatial and incremental lms processing for distributed estimation. *IEEE Transactions on Signal Processing*, 59(4):1465–1480.
- [27] Cattivelli, F. S. and Sayed, A. H. (2011b). Distributed detection over adaptive networks using diffusion adaptation. *IEEE Transactions on Signal Processing*, 59(5):1917–1932.
- [28] Chen, J., Kher, S., and Somani, A. (2006). Distributed fault detection of wireless sensor networks. In *Workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pages 65–72. ACM.
- [29] Chen, J. and Sayed, A. H. (2013). Distributed pareto optimization via diffusion strategies. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):205–220.
- [30] Chen, J. C., Yao, K., and Hudson, R. E. (2002). Source localization and beam-forming. *IEEE Signal Processing Magazine*, 19(2):30–39.
- [31] Cheng, Q., Varshney, P., Michels, J., and Belcastro, C. (2009). Distributed fault detection with correlated decision fusion. *Aerospace and Electronic Systems, IEEE Transactions on*, 45(4):1448–1465.
- [32] Christofides, P. D., Scattolini, R., de la Pena, D. M., and Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41.
- [33] Cortés, J. (2009). Distributed kriged kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827.
- [34] Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2002). Coverage control for mobile sensing networks. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1327–1332. IEEE.
- [35] D’Andrea, R. and Dullerud, G. E. (2003). Distributed control design for spatially interconnected systems. *IEEE Transactions on automatic control*, 48(9):1478–1495.
- [36] Davis, M. H. (1977). *Linear estimation and stochastic control*. London.
- [37] De Angelis, A. and Fischione, C. (2011). A distributed information fusion method for localization based on pareto optimization. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–8. IEEE.
- [38] Delouille, V., Neelamani, R., and Baraniuk, R. G. (2006). Robust distributed estimation using the embedded subgraphs algorithm. *IEEE Transactions on Signal Processing*, 54:2998–3010.
- [39] Demetriou, M. A. and Polycarpou, M. M. (1998). Incipient fault diagnosis of dynamical systems using online approximators. *IEEE Transactions on Automatic Control*, 43:1612–1617.

- [40] Ding, D., Wang, Z., Dong, H., and Shu, H. (2012). Distributed H_{∞} state estimation with stochastic parameters and nonlinearities through sensor networks: the finite-horizon case. *Automatica*, 48(8):1575–1585.
- [41] Ding, D., Wang, Z., and Shen, B. (2014). Recent advances on distributed filtering for stochastic systems over sensor networks. *International Journal of General Systems*, 43(3-4):372–386.
- [42] Dong, H., Wang, Z., Lam, J., and Gao, H. (2014). Distributed filtering in sensor networks with randomly occurring saturations and successive packet dropouts. *International Journal of Robust and Nonlinear Control*, 24(12):1743–1759.
- [43] Downs, J. and Vogel, E. (1993). Industrial challenge problems in process control a plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3):245 – 255.
- [44] Estrin, D., Girod, L., Pottie, G., and Srivastava, M. (2001). Instrumenting the world with wireless sensor networks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP'01). 2001 IEEE International Conference on*, volume 4, pages 2033–2036. IEEE.
- [45] Fagiolini, A., Visibelli, E. M., and Bicchi, A. (2008). Logical consensus for distributed network agreement. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 5250–5255. IEEE.
- [46] Farina, M. and Carli, R. (2015). Plug and play partition-based state estimation based on kalman filter. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3155–3160.
- [47] Farina, M., Ferrari-Trecate, G., and Scattolini, R. (2010a). Distributed moving horizon estimation for linear constrained systems. *IEEE Transactions on Automatic Control*, 55(11):2462 –2475.
- [48] Farina, M., Ferrari-Trecate, G., and Scattolini, R. (2010b). Moving-horizon partition-based state estimation of large-scale systems. *Automatica*, 46(5):910 – 918.
- [49] Fax, J. A. and Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control*, 49(9):1465–1476.
- [50] Ferrari, R., Parisini, T., and Polycarpou, M. (2012a). Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach. *IEEE Transactions on Automatic Control*, 57(2):275 –290.
- [51] Ferrari, R. M. G., Parisini, T., and Polycarpou, M. M. (2009). Distributed fault diagnosis with overlapping decompositions: an adaptive approximation approach. *IEEE Transactions on Automatic Control*, 54:794–799.
- [52] Ferrari, R. M. G., Parisini, T., and Polycarpou, M. M. (2012b). Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach. *IEEE Transactions on Automatic Control*, 57:275–290.

- [53] Franco, E., Magni, L., Parisini, T., Polycarpou, M. M., and Raimondo, D. M. (2008). Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach. *IEEE Transactions on Automatic Control*, 53(1):324–338.
- [54] Franco, E., Olfati-Saber, R., Parisini, T., and Polycarpou, M. (2006). Distributed fault diagnosis using sensor networks and consensus-based filters. In *45th IEEE Conference on Decision and Control*, pages 386–391.
- [55] Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26:459–474.
- [56] Frank, P. M. (1992). Principles of model-based fault detection. *Annual Review in Automatic Programming*, 17:213–220.
- [57] Fudenberg, D. and Tirole, J. (1983). Multiple nash equilibria, focal points, and pareto optimality, game theory.
- [58] Garin, F. and Schenato, L. (2010). A survey on distributed estimation and control applications using linear consensus algorithms. In *Networked Control Systems*, volume 406, pages 75–107. Springer.
- [59] Gertler, J. J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York.
- [60] Guibas, L. J. (2002). Sensing, tracking and reasoning with relations. *IEEE Signal Processing Magazine*, 19(2):73–85.
- [61] Hendeby, G. and Gustafsson, F. (2005). Fundamental filtering limitations in linear non-gaussian systems. In *16th IFAC World Congress*, Prague, Czech Republic.
- [62] Hu, J. and Hu, X. (2010). Nonlinear filtering in target tracking using cooperative mobile sensors. *Automatica*, 46(12):2041–2046.
- [63] Hui, Q. and Haddad, W. M. (2008). Distributed nonlinear control algorithms for network consensus. *Automatica*, 44(9):2375–2381.
- [64] Isermann, R. (1984). Process fault detection based on modeling and estimation methods. A survey. *Automatica*, 20:387–404.
- [65] Isermann, R. (2006). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, Berlin.
- [66] Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001.
- [67] Johansson, K. (2000). The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *Control Systems Technology, IEEE Transactions on*, 8(3):456–465.

- [68] Jourdan, D. and de Weck, O. (2004). Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. In *IEEE 59th Vehicular Technology Conference*, volume 5, pages 2466 – 2470.
- [69] Kadu, S. C., Bhushan, M., and Gudi, R. (2008). Optimal sensor network design for multirate systems. *Journal of Process Control*, 18(6):594 – 609.
- [70] Kaplan, L. M., Le, Q., and Molnar, N. (2001). Maximum likelihood methods for bearings-only target localization. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 5, pages 3001–3004. IEEE.
- [71] Kar, S., Hug, G., Mohammadi, J., and Moura, J. M. F. (2014). Distributed state estimation and energy management in smart grids: A consensus + innovations approach. *IEEE Journal of Selected Topics in Signal Processing*, 8(6):1022–1038.
- [72] Kar, S. and Moura, J. M. (2011). Gossip and distributed kalman filtering: weak consensus under weak detectability. *IEEE Transactions on Signal Processing*, 59(4):1766–1784.
- [73] Keliris, C., Polycarpou, M. M., and Parisini, T. (2013). A distributed fault detection filtering approach for a class of interconnected continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 58:2032–2047.
- [74] Khan, U. A. and Moura, J. M. (2008). Distributing the kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, 56(10):4919–4935.
- [75] Kim, K.-D. and Kumar, P. (2013). An overview and some challenges in cyber-physical systems. *Journal of the Indian Institute of Science*, 93(3):341–352.
- [76] Koushanfar, F., Potkonjak, M., and Sangiovanni-Vincentelli, A. (2003). On-line fault detection of sensor measurements. In *Sensors, 2003. Proceedings of IEEE*, volume 2, pages 974–979. IEEE.
- [77] Lagarias, J., Reeds, J., Wright, M., and Wright, P. (1998). Convergence properties of the nelder-mead simplex method in low dimensions. *CiteSeerX - Scientific Literature Digital Library and Search Engine* [<http://citeseerx.ist.psu.edu/oai2/>] (United States).
- [78] Larson, E., Parker, B.E., J., and Clark, B. (2002). Model-based sensor and actuator fault detection and isolation. In *American Control Conference, 2002. Proceedings of the 2002*, volume 5, pages 4215–4219 vol.5.
- [79] Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M., and Davis, R. E. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74.
- [80] Lesser, V., Ortiz Jr, C. L., and Tambe, M. (2012). *Distributed sensor networks: A multiagent perspective*, volume 9. Springer Science & Business Media.
- [81] Li, D., Wong, K. D., Hu, Y. H., and Sayeed, A. M. (2002). Detection, classification, and tracking of targets. *IEEE signal processing magazine*, 19(2):17–29.

- [82] Li, J. and AlRegib, G. (2007). Rate-constrained distributed estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 55(5):1634–1643.
- [83] Li, W., Gui, W., Xie, Y., and Ding, S. X. (2009). Decentralized fault detection system design for large-scale interconnected system. *IFAC Proceedings Volumes*, 42(8):816–821.
- [84] Liang, J., Wang, Z., and Liu, X. (2011). Distributed state estimation for discrete-time sensor networks with randomly varying nonlinearities and missing measurements. *IEEE Transactions on Neural Networks*, 22(3):486–496.
- [85] Liang, J., Wang, Z., Shen, B., and Liu, X. (2012). Distributed state estimation in sensor networks with randomly occurring nonlinearities subject to time delays. *ACM Transactions on Sensor Networks (TOSN)*, 9(1):4.
- [86] Liu, J., Chu, M., and Reich, J. E. (2007). Multitarget tracking in distributed sensor networks. *IEEE Signal Processing Magazine*, 24(3):36–46.
- [87] Liu, J., de la Peña, D. M., and Christofides, P. D. (2010). Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica*, 46(1):52–61.
- [88] Lopes, C. G. and Sayed, A. H. (2007). Incremental adaptive strategies over distributed networks. *IEEE Transactions on Signal Processing*, 55(8):4064–4077.
- [89] Lopes, C. G. and Sayed, A. H. (2008). Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136.
- [90] Luo, H., Liu, Y., and Das, S. K. (2007). Routing correlated data in wireless sensor networks: A survey. *IEEE Network*, 21(6):40–47.
- [91] Mahmoud, M. S. and Khalid, H. M. (2013). Distributed Kalman filtering: a bibliographic review. *IET Control Theory & Applications*, 7(4).
- [92] Medeiros, H., Park, J., and Kak, A. (2008). Distributed object tracking using a cluster-based kalman filter in wireless camera networks. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):448–463.
- [93] Meng, X. and Chen, T. (2014a). Optimality and stability of event triggered consensus state estimation for wireless sensor networks. In *American Control Conference*, pages 3565–3570.
- [94] Meng, X. and Chen, T. (2014b). Optimality and stability of event triggered consensus state estimation for wireless sensor networks. In *American Control Conference*, pages 3565–3570.
- [95] Minka, T. P. (2001). Old and new matrix algebra useful for statistics.
- [96] Mosallae, M. and Salahshoor, K. (2008). Sensor fault detection using adaptive modified extended kalman filter based on data fusion technique. In *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*, pages 513–518.

- [97] Murray, R. O. S. R. M. (2003). Consensus protocols for networks of dynamic agents. In *Proceedings of the 2003 American Controls Conference*.
- [98] Nandi, M., Nayak, A., Roy, B., and Sarkar, S. (2015). Hypothesis testing and decision theoretic approach for fault detection in wireless sensor networks. *International Journal of Parallel, Emergent and Distributed Systems*, 30(4):262–285.
- [99] Nedic, A. and Bertsekas, D. P. (2001). Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138.
- [100] Oka, A. and Lampe, L. (2008). Energy efficient distributed filtering with wireless sensor networks. *IEEE Transactions on Signal Processing*, 56(5):2062–2075.
- [101] Olfati-Saber, R. (2005). Distributed Kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control, and European Control Conference*, pages 8179–8184.
- [102] Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420.
- [103] Olfati-Saber, R. (2007). Distributed Kalman filtering for sensor networks. In *46th IEEE Conference on Decision and Control*, pages 5492–5498.
- [104] Olfati-Saber, R. (2009). Kalman-consensus filter: Optimality, stability, and performance. In *48th IEEE Conference on Decision and Control, and 28th Chinese Control Conference*, pages 7036–7042.
- [105] Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.
- [106] Olfati-Saber, R. and Jalalkamali, P. (2012). Coupled distributed estimation and control for mobile sensor networks. *IEEE Transactions on Automatic Control*, 57(10):2609–2614.
- [107] Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533.
- [108] Parisini, T., Polycarpou, M., Sanguineti, M., and Vemuri, A. (1997). Robust parametric and non-parametric fault diagnosis in nonlinear input-output systems. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 5, pages 4481–4482 vol.5.
- [109] Petersen, K. B. and Pedersen, M. S. (2006). The matrix cookbook.
- [110] Polycarpou, M. and Ioannou, P. (1996). A robust adaptive nonlinear control design. *Automatica*, 32(3):423 – 427.
- [111] Polycarpou, M. M. (1996). Stable adaptive neural control scheme for nonlinear systems. *Automatic Control, IEEE Transactions on*, 41(3):447–451.

- [112] Polycarpou, M. M. and Helmicki, A. J. (1995). Automated fault detection and accommodation: a learning systems approach. *IEEE Transactions on Systems, Man and Cybernetics*, 25:1447–1458.
- [113] Polycarpou, M. M. and Ioannou, P. A. (1992). Neural networks as on-line approximators of nonlinear systems. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 7–12 vol.1.
- [114] Polycarpou, M. M. and Trunov, A. B. (2000). Learning approach to nonlinear fault diagnosis: detectability analysis. *IEEE Transactions on Automatic Control*, 45:806–812.
- [115] Polycarpou, M. M. and Vemuri, A. T. (1998). Learning approach to fault tolerant control: an overview. In *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings*, pages 157–162.
- [116] Rabbat, M. G. and Nowak, R. D. (2004). Decentralized source localization and tracking [wireless sensor networks]. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*, volume 3, pages iii–921. IEEE.
- [117] Raffard, R. L., Tomlin, C. J., and Boyd, S. P. (2004). Distributed optimization for cooperative agents: Application to formation flight. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 3, pages 2453–2459. IEEE.
- [118] Rajagopalan, R. and Varshney, P. K. (2006). Data aggregation techniques in sensor networks: A survey.
- [119] Ren, W., Beard, R. W., and Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1859–1864. IEEE.
- [120] Ren, W. et al. (2007). Information consensus in multivehicle cooperative control.
- [121] Reppa, V., Polycarpou, M. M., and Panayiotou, C. G. (2012). Distributed sensor fault detection and isolation for nonlinear uncertain systems. *IFAC Proceedings Volumes*, 45(20):1077–1082.
- [122] Reppa, V., Polycarpou, M. M., and Panayiotou, C. G. (2014). Adaptive approximation for multiple sensor fault detection and isolation of nonlinear uncertain systems. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):137–153.
- [123] Ricker, N. L. (1996). Decentralized control of the tennessee eastman challenge process. *Journal of Process Control*, 6(4):205 – 221.
- [124] Rivero, S., Farina, M., Scattolini, R., and Ferrari-Trecate, G. (2013a). Plug-and-play distributed state estimation for linear systems. In *52nd IEEE Conference on Decision and Control*, pages 4889–4894. IEEE.

- [125] Rivero, S., Rubini, D., and Ferrari-Trecate, G. (2013b). Distributed bounded-error state estimation for partitioned systems based on practical robust positive invariance. In *Control Conference (ECC), 2013 European*, pages 2633–2638.
- [126] Sanislav, T. and Miclea, L. (2012). Cyber-physical systems-concept, challenges and research areas. *Journal of Control Engineering and Applied Informatics*, 14(2):28–33.
- [127] Schizas, I. D., Ribeiro, A., and Giannakis, G. B. (2008). Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals. *IEEE Transactions on Signal Processing*, 56(1):350–364.
- [128] Serpedin, E., Li, H., Dogandzić, A., Dai, H., and Cota, P. (2008). Distributed signal processing techniques for wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2008.
- [129] Sharma, G. and Mazumdar, R. R. (2008). A case for hybrid sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(5):1121–1132.
- [130] Shen, B., Wang, Z., and Hung, Y. (2010). Distributed H_{∞} -consensus filtering in sensor networks with multiple missing measurements: the finite-horizon case. *Automatica*, 46(10):1682–1688.
- [131] Shen, X., Song, E., Zhu, Y., and Luo, Y. (2009). Globally optimal distributed kalman fusion with local out-of-sequence-measurement updates. *IEEE Transactions on Automatic Control*, 54(8):1928–1934.
- [132] Sinopoli, B., Sharp, C., Schenato, L., Schaffert, S., and Sastry, S. S. (2003). Distributed control applications within sensor networks. *Proceedings of the IEEE*, 91(8):1235–1246.
- [133] Smith, D. and Singh, S. (2006). Approaches to multisensor data fusion in target tracking: A survey. *IEEE transactions on knowledge and data engineering*, 18(12):1696–1710.
- [134] Spanos, D., Olfati-Saber, R., and Murray, R. (2005a). Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *4th International Symposium on Information Processing in Sensor Networks*, pages 133 – 139.
- [135] Spanos, D. P., Olfati-Saber, R., and Murray, R. M. (2005b). Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*. Citeseer.
- [136] Speranzon, A., Fischione, C., Johansson, K., and Sangiovanni-Vincentelli, A. (2008). A distributed minimum variance estimator for sensor networks. *IEEE Journal on Selected Areas in Communications*, 26(4):609–621.
- [137] Speranzon, A., Fischione, C., and Johansson, K. H. (2006). Distributed and collaborative estimation over wireless sensor networks. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1025–1030. IEEE.

- [138] Stanković, M. S., Stanković, S. S., and Stipanović, D. M. (2008). Consensus based multi-agent control structures. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4364–4369. IEEE.
- [139] Stanković, S. S., Stanković, M. S., and Stipanović, D. M. (2009). Consensus based overlapping decentralized estimation with missing observations and communication faults. *Automatica*, 45(6):1397–1406.
- [140] Stipanović, D. M., Inalhan, G., Teo, R., and Tomlin, C. J. (2004). Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica*, 40(8):1285–1296.
- [141] Takahashi, N., Yamada, I., and Sayed, A. (2009). Diffusion adaptive networks with changing topologies. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, (ICASSP'06)*, pages 2845–2849.
- [142] Thornhill, N. F., Patwardhan, S. C., and Shah, S. L. (2008). A continuous stirred tank heater simulation model with applications. *Journal of process control*, 18:347–360.
- [143] Trunov, A. B. and Polycarpou, M. M. (2000). Automated fault diagnosis in nonlinear multivariable systems using a learning methodology. *IEEE Transactions on Neural Networks*, 11:91–101.
- [144] Ugrinovskii, V. (2011). Distributed robust filtering with H_{∞} consensus of estimates. *Automatica*, 47(1):1–13.
- [145] Vemuri, A. T. (1999). Diagnosis of sensor bias faults. In *Proc. American Control Conference*, volume 1, pages 460–464. IEEE.
- [146] Vemuri, A. T. (2001). Sensor bias fault diagnosis in a class of nonlinear systems. *IEEE Transactions on Automatic Control*, 46:949–954.
- [147] Vemuri, A. T. and Polycarpou, M. M. (1996). On-line approximation methods for robust fault detection. In *Proc. 13th IFAC World Congr.*, pages 319–324.
- [148] Vemuri, A. T. and Polycarpou, M. M. (1997). Robust nonlinear fault diagnosis in input-output systems. *International Journal of Control*, 68:343–360.
- [149] Vemuri, A. T. and Polycarpou, M. M. (1998). On the use of on-line approximators for sensor fault diagnosis. In *Proc. American Control Conference*, volume 5, pages 2857–2861. IEEE.
- [150] Wang, H., Huang, Z. J., and Daley, S. (1997). On the use of adaptive updating rules for actuator and sensor fault diagnosis. *Automatica*, 33:217–225.
- [151] Wang, T., Chang, L., and Chen, P. (2009). A collaborative sensor-fault detection scheme for robust distributed estimation in sensor networks. *Communications, IEEE Transactions on*, 57(10):3045–3058.
- [152] Xiao, L., Boyd, S., and Kim, S.-J. (2007). Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46.

- [153] Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 63–70. IEEE.
- [154] Yang, W., Chen, G., Wang, X., and Shi, L. (2014). Stochastic sensor activation for distributed state estimation over a sensor network. *Automatica*, 50(8):2070–2076.
- [155] Yi-Ming, W., Wei, D., and Hao, Y. (2012). Distributed H_{∞} filtering with consensus strategies in sensor networks: considering consensus tracking error. *Acta Automatica Sinica*, 38(7):1211–1217.
- [156] Zeadally, S. and Jabeur, N. (2015). *Cyber-Physical System Design with Sensor Networking Technologies*. IET.
- [157] Zhang, D., Yu, L., Song, H., and Wang, Q.-G. (2013). Distributed H_{∞} filtering for sensor networks with switching topology. *International Journal of Systems Science*, 44(11):2104–2118.
- [158] Zhang, X. (2010). Decentralized fault detection for a class of large-scale nonlinear uncertain systems. In *Proceedings of the 2010 American Control Conference*, pages 5650–5655. IEEE.
- [159] Zhang, X., Parisini, T., and Polycarpou, M. M. (2005). Sensor bias fault isolation in a class of nonlinear systems. *IEEE Transactions on Automatic Control*, 50:370–376.
- [160] Zhang, X., Polycarpou, M. M., and Parisini, T. (2001). Robust fault isolation for a class of non-linear input-output systems. *International Journal of Control*, 74:1295–1310.
- [161] Zhang, X., Polycarpou, M. M., and Parisini, T. (2002). A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Transactions on Automatic Control*, 47:576–593.
- [162] Zhao, F., Liu, J., Liu, J., Guibas, L., and Reich, J. (2003). Collaborative signal and information processing: an information-directed approach. *Proceedings of the IEEE*, 91(8):1199–1209.
- [163] Zhao, F., Shin, J., and Reich, J. (2002). Information-driven dynamic sensor collaboration. *IEEE Signal processing magazine*, 19(2):61–72.
- [164] Zhou, Y., Boem, F., Fischione, C., and Parisini, T. (2016). Distributed fault detection with sensor networks using pareto-optimal dynamic estimation method. In *Proc. European Control Conference, (to appear)*, Aalborg.
- [165] Zhou, Y., Parisini, T., and Polycarpou, M. M. (2015). Detection of drift sensor faults in a class of nonlinear uncertain systems. In *54th IEEE Conference on Decision and Control (CDC)*, pages 3169–3174.