UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXV CICLO DEL DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

Posto di dottorato attivato grazie al contributo di DANIELI AUTOMATION S.P.A.

Distributed Methods for Estimation and Fault Diagnosis: the case of Large-scale Networked Systems

Settore scientifico-disciplinare **ING-INF/04**

DOTTORANDA
Francesca Boem

COORDINATORE
Chiar.mo Prof. Walter Ukovich

TUTORE E RELATORE
Chiar.mo Prof. Thomas Parisini

CORRELATORE
Chiar.mo Prof. Marios M. Polycarpou

CORRELATORE
Ing. Ph.D. Riccardo M. G. Ferrari

ANNO ACCADEMICO 2011/2012

Preface

This thesis deals with the problem of the monitoring of modern complex systems. In fact, nowadays reliability is a key requirement in the systems design. While fault diagnosis architectures and estimation methods have been extensively studied for centralized systems, the interest towards distributed, networked, large-scale and complex systems, such as Cyber-Physical Systems and Systems-of-Systems, has grown in the recent years. Therefore, the design of distributed methods for estimation and fault diagnosis specifically for these kinds of systems is an emergent topic.

The system being monitored is modeled as the interconnection of several subsystems and a divide et impera approach allowing overlapping decomposition is used. The local diagnostic decision is made on the basis of the knowledge of the local subsystem dynamic model and of an adaptive approximation of the uncertain interconnection with neighboring subsystems.

The goal of this work is to present some recent results, considering different frameworks and facing some of the issues emerging when dealing with the implementation of monitoring architectures for real use-cases. The target is to integrate all the aspects of the monitoring process in a comprehensive architecture.

Following a logical order, first of all we design the measurements acquisition task by proposing a distributed estimator for sensor networks, able to filter measurements so that both the variance and the mean of the estimation error is minimized by means of a Pareto optimization problem.

Besides, a synchronization method is proposed in order to consider the case of multi-rate systems and to compensate delays in the communication network between sensors and diagnosers.

In fact, one of the problems when dealing with distributed, large-scale or networked systems and therefore with a communication network, is inevitably the presence of stochastic delays and packet dropouts, that degrade performance and could be a source of instability.

We propose therefore a delay compensation strategy, able to manage delays and packet losses in the communication network between the Local Fault Diagnosers, and develop a novel consensus-based estimator with time-varying weights, allowing to improve detectability and isolability skills in the case of variables shared among more than one subsystem. In the con-

sensus protocol, at each step each agent uses only the information given by the communication link and the agent which are more reliable at that time. The convergence of the proposed estimator is demonstrated without any assumption on the communication network topology and analytical conditions for detectability and isolability are derived, showing that the novel consensus-based estimator improves diagnosis performance.

For the sake of completeness, the monitoring architecture is studied and adapted to different frameworks: the fault detection and isolation methodology is extended for continuous-time systems and the case where the state is only partially measurable is considered for discrete-time and continuous-time systems.

Acknowledgments

I really wish to thank my advisor, prof. Thomas Parisini, that supported and guided me. I could learn a lot from him: both from the scientific point of view and from the personal one. He motivated me during these years, being a model for me as far as the professional life. He permitted my attendance to several PhD courses, seminars and conferences. In this way I could learn a lot and I had the opportunity to grow up and know some very interesting people sharing with me the enthusiasm for scientifical research.

I also want to thank my co-supervisor Prof. Marios Polycarpou, for the help and suggestions and for the big opportunity I had to work with him.

My gratitude goes also to Danieli Automation S.p.A., that contributed to make this research possible, to all the people of the R&D laboratory and in particular to Riccardo Ferrari. I would like to thank him for the time, the help and the advice he gave me. I have had the opportunity to cooperate with a big company and to learn more about many practical issues.

I want to acknowledge Prof. Carlo Fischione: it has been a great experience to cooperate and work with him.

A special thank goes to Felice Andrea Pellegrino and Gianfranco Fenu, for the help they give me in all situations and the opportunity to work on different interesting research fields.

Moreover, I'd like to thank the Automation Lab group at the Trieste University and the Italian control community: it is great to work in a so fruitful, dynamic and networked community. I especially want to express my gratitude to some PhD students that I consider friends: Sergio, Laura, Chiara, Stefano, Paola, Andrea, Giuseppe, Giulio, among others.

Finally, my deep thankfulness goes to my family that supported me in all these years and to Alessandro: with him everything is easier.

Contents

List of Figures

Chapter 1

Introduction

In the present world, everything is connected and networked. In this sense, modern systems are getting more and more complex: several different elements work to obtain common or competitive goals, interacting with each other and influencing their behavior. The term *network*, describing a collection of nodes and links, nowadays has become of common use, thanks to our extensive reliance on networks for our everyday life and our job and, in the scientific research, for the design, the study and the analysis of complex systems. Influences between individuals or systems are not limited to certain local areas since distances between people and objects are overcome thanks to novel communication and transportation systems. The exchange of information is simple and quick by means of a lot of common technological tools: smartphones and tablets permit to be always connected to the rest of the world; social networks allow to exchange opinions, pictures, documents and other with a click; also the business is on-line: almost everything can be bought using Internet, it is possible to order an item, to pay it, often also to check its status and to know where it is before delivery. The relationship between the real physical environment and the cyber world is always tighter. These changes have involved also the industrial world: let us consider as example large-scale industrial processes, where a lot of elements interact with each other and that can be monitored in a remote way. It is not possible to consider single systems without considering the influences from other systems (physical or computational), human interaction and the external world. In fact, emerging applications are not just large-scale and complex: they are also characterized by decentralized, distributed, networked compositions of heterogeneous and (semi)autonomous elements [1]. The scientific research is oriented to consider systems with a novel approach in order to provide integration between the different levels constituting a modern system, which deal with physical, computational, control and communication tasks. That's why we talk about decentralized, distributed, networked systems, Cyber Physical Systems (CPS) [2] and Systems of Systems (SoS) [1].

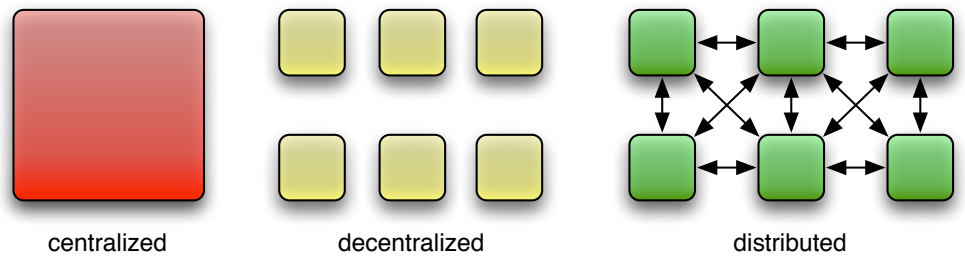

Figure 1.1: Pictorial representation of a centralized, a decentralized and a distributed system.

The first two terms will be described in detail in the following (see Figure 1.1). As regards CPS, the term cyber-physical systems refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities [2], expanding the capabilities of the physical world through computation, communication, and control. On the other hand, a SoS can be considered as a composition, made of components that are themselves systems, which is characterized by two properties that the whole must possess for it [3]: operational and managerial independence of components. This means that the component systems fulfill their own purposes and continue to operate to fulfill those purposes even if disassembled from the overall system; besides, the component systems are managed (at least in part) for their own purposes rather than the purposes of the whole. There are some differences and some common points in the definitions of these kinds of systems and some systems can belong to more than one set when trying to classify them. Often the descriptions of these systems refer to similar concepts, but observed from different points of view. In this work, we consider the common requirements emerging from these systems and in general we will refer to them as complex systems. More specifically, in the next section we will consider which are the common needs that these typologies of systems require. Examples of these systems are: air and road transportation and traffic management, power grids, smart grids, healthcare systems, water management, complex industrial processes, critical infrastructures, integrated supply chains, collaborative enterprise systems, smart homes and cities, and others.

1.1 Motivations

Let us consider therefore the needs and the issues emerging when dealing with this kind of systems. First of all, the need for integration: different elements/levels are present both in the structure itself of these systems and in the tools we use to analyze them. The scientific research is partitioned into

isolated disciplines (let us consider as example sensors, communications and networking, control theory, computer science, software engineering, mathematics) and the tools and formalisms used are very different; anyway, it is necessary to provide integration between the different levels composing physical and computational systems, which are correlated in modern systems. The present work addresses the need for integration by proposing a comprehensive architecture, where all the parts of complex distributed systems are considered: the physical environment, the sensor level, the diagnosers layer and the communication networks. Moreover, the modularity of the architecture allows to partially consider dynamics in the evolving structure of these complex systems: in particular, the proposed architecture can manage a dynamic structure of the sensor layer. This approach can be even more beneficial in the case of the new paradigm of system of systems, where systems may be added or removed and the expectation is that the overall system should continue to operate optimally [4]. In fact, systems of systems require the easy interconnection and interoperability of multiple systems each including one or more control systems, composed by several components (sensors, actuators, decision algorithms, being implemented in hardware and/or software).

Secondly, reliability, safety and security are crucial requirements that frameworks, algorithms, methods, and tools have to satisfy when considering heterogeneous cooperating elements that interact in a complex, coupled physical environment operating over different spatial and temporal scales [2]. In fact, one of the defining features characterizing CPS is that they are networked, at multiple and different scales, and complex. Therefore, when considering large-scale systems (this can be interpreted more in a logical than necessarily in a spatial sense, that is systems with a large number of state components), it is worth noting that increased scale can imply a proportionate increase in risk: failures in a low-level component may have a small impact and may be managed; on the other hand, failures at high level can have bigger consequences for individuals, societies, system owners, operators and the environment. The costs for validation and verification of software and systems are high when dealing with safety-critical systems (let us consider as example the aviation industry, the medical applications, the automotive or energy systems). Therefore, new methods, algorithms and tools are needed for building high-confidence systems and infrastructures. The development and increasing interest for this kind of systems relies on a renewed emphasis on monitoring, fault detection and diagnosis, and fault-tolerant control. Overdesign and physical redundancy is not always a tractable solution for complex systems where interoperability is needed. Therefore, new methods, algorithms and tools are needed for building high-confidence systems and to improve the trustworthiness that is lacking in many of today's infrastructures. Since it is not possible to avoid all the components failures, it is necessary to develop methodologies and tools specifically for this kind of

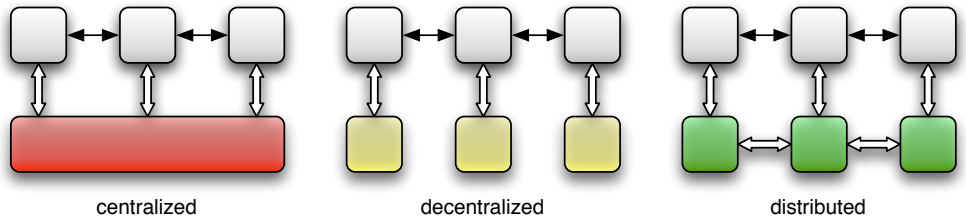

Figure 1.2: Pictorial representation of a centralized (red), a decentralized (yellow) and a distributed (green) architecture applied to a distributed system (white). Physical interaction between subsystems is represented by black arrows, while white thick arrows represent communication and measuring channels.

systems in order to ensure safe and reliable operations also in the case of component faults, that is, systems have to be able to continue providing their service or at least a degraded version of their service (*fault tolerant* system).

Remembering that one of the characterizing features of CPS is that they are networked and often large scale, we propose a distributed scheme. The need for a distributed architecture is justified by the drawbacks of centralized fault diagnosis architectures when dealing with actual large-scale, distributed complex systems. In fact, the alarms related to a Fault Detection and Isolation (FDI) architecture are useful only if they are provided in real-time, so to leave the larger possible amount of time for fault accommodation before the fault event may lead to a failure. The task of computing in real-time all the estimations needed by a fault detection and isolation scheme in a large-scale system may be limited by the amount of computation power available at the centralized computation node. Moreover, if the measurements from the actual system are not taken from sensors directly wired to the computation node, also the available bandwidth of the communication network to gather all the needed measurements to the place where they are processed may be limited. Besides the issue of being large-scale, the unfeasibility or inappropriateness of a centralized monitoring architecture may be due to another feature of the system, that is the characteristic of being *distributed*. This term means that systems structure can be analyzed as being constituted by multiple subsystems that interact with neighboring subsystems. This differs from the term *decentralized*, that refers to systems structure made of multiple subsystems that do not interact with each other, and of course it is in contrast with the term *centralized*, where a subdivision in distinct subsystems is not possible, as every part of the system interacts with every other one. The difference between the concepts of centralized, decentralized and distributed systems can be easily understood by observing

Fig. 1.1 and 1.2 [5].

Furthermore there are cases in which centralized architectures, even if feasible, would be undesirable because they would suffer from robustness, scalability and security issues. In the literature, a typical solution to this kind of problems is to adopt a *divide et impera* approach [6, 7], where a complex problem is *decomposed* into smaller subproblems simpler enough to be solved with the existing computation and communication infrastructures. The easiest way to apply such an approach is through a decentralized architecture, which has anyway the disadvantage of ignoring interactions between neighboring subsystems (see Fig. 1.2). On the other hand, the more general solution is the implementation of a distributed architecture. Therefore, in this work we will adopt a distributed architecture, as the one proposed in [8] and [5], where as many computing nodes as subsystems are employed, connected by a communication network that matches the physical interconnections of the subsystems with their neighbors.

1.2 State of the art

In the previous section we have seen that reliability is a fundamental requirement for modern systems. Reliability can be defined as the ability of a system to perform its intended function over a given period of time [9]. A change in the behavior of a system, or part of it, from the behavior that was set at design time represents a *fault*, while, when we refer to a *failure*, we mean the inability of the system to perform its function, and it can be due to the effects of a fault.

1.2.1 Methods Classification

There are many methods to address the possible presence of a fault. The simpler is *physical redundancy*, that is the fact that critical components of the system are replicated in a greater number than what is strictly necessary. This is effective but implies a highly expensive solution and can be justified only for critical, potentially life-threatening systems. Let us think about aviation applications as example. Another more affordable approach consists in the use of *analytic redundancy* [9], that is, the redundancy is not obtained by having multiple physical copies of critical components, but providing one or more mathematical models of the healthy system behavior. This choice implies the implementation of a procedure needed to reduce the effects of a fault in order to have a fault-tolerant system. The main steps of the monitoring process are: *detection* of a fault, *isolation* and *identification* of the fault and fault *accommodation* or reconfiguration of the system. Fault detection consists in understanding whether a fault has occurred and when it has occurred, while the isolation task refers to pinpointing the type of fault and its location. Fault identification is an extra step that is carried on after

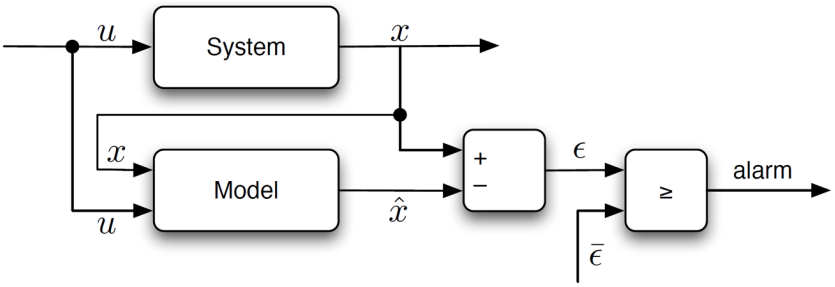

Figure 1.3: A scheme for model-based fault detection. The difference between the measurable system state x and the model state estimate $\hat{x}$ is compared to the detection threshold $\bar{\epsilon}$.

isolation in order to quantify the extent to which a fault is present. The fault accommodation addresses the problem of how the system actively responds to the fault: after a successful fault diagnosis, the controller parameters must be adjusted to accommodate changed plant dynamics in order to prevent failure at the system level.

A classification of fault diagnosis methods distinguishes between model-based and signal-based approaches. In signal-based techniques, known features of signals, such as spectral components or statistical features are compared to nominal ones [10, 11]. These methods require some knowledge of previous behavior of the system during healthy operation: that's why they are classified into the wider class of process history fault diagnosis approaches [12]. On the other hand, the model-based approach, which was born during the 1970s thanks to the seminal works of Beard, Jones and Clark [13, 14, 15] among others (see the survey papers [11, 16, 17, 18]), is based on the use of a mathematical model of the healthy behavior of the process that must be monitored. The basic idea is: by using the model it is possible to compute some estimates of the measured variables and, by comparing the estimations to the actual measurements, a deviation due to a fault can be detected [9]. The difference between measurements and estimates can be used as a residual, which ideally should be zero when no faults are acting in the system (see Figure 1.3). The residuals are then compared to suitable thresholds by detection and isolation logics in order to provide a fault decision regarding the health of the system.

An obvious problem in the practical implementation of model-based FDI schemes is that deriving a good mathematical model of an actual engineering system may be a challenging task. A line of research tried to overcome this problem by using qualitative models, where only qualitative information, such as sign or trend of measured variables, are used [19]. A more successful

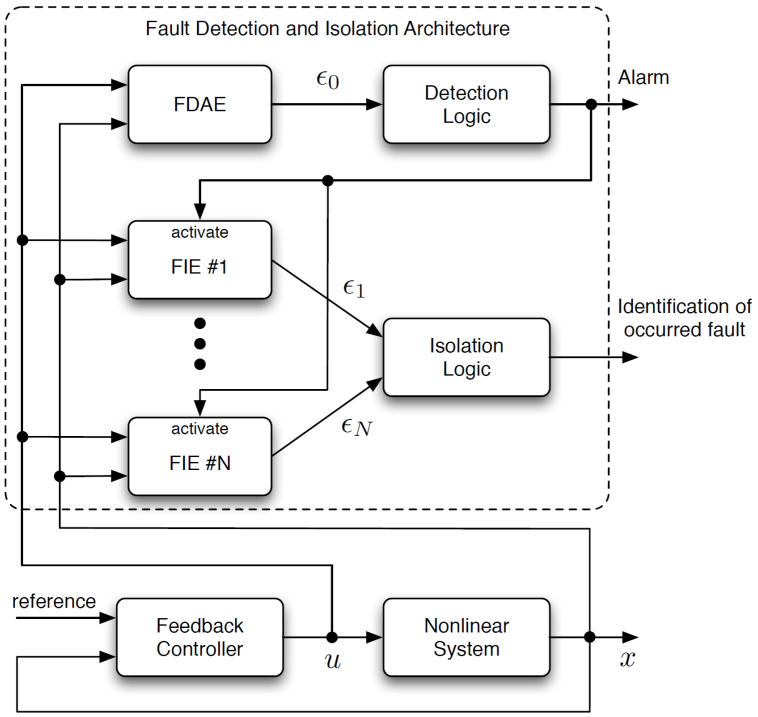

Figure 1.4: The GOS scheme on which the proposed FDI architecture is based.

approach, anyway, is based on the use of adaptive on-line approximators, such as neural networks as example, to learn on-line the unknown or uncertain parts of the system dynamical model or the fault model [20, 21, 22].

In the model-based FDI literature, two different schemes were developed: the Dedicated Observer Scheme (DOS) proposed by Clark [15], and the Generalized Observer Scheme (GOS) presented by Frank (see [17] and the references by the same author therein). A representation of a GOS scheme is shown in Figure 1.4. As far as the isolation task, in both schemes as many residuals as the number of possible faults are generated. The difference is that in the DOS scheme, each residual is sensitive to only a single fault, while in the GOS, each residual is sensitive to every but one fault. The DOS scheme is appealing as it can isolate also concurrent faults, but it cannot always be designed. Instead the GOS can be always applied, but can isolate only non-concurrent faults.

1.2.2 Distributed systems

Although many results exist for centralized architectures, the development of distributed monitoring schemes specifically for distributed systems has begun in the very recent years for what concerns discrete-time or continuous-time systems [23, 24, 25, 26, 27, 8, 28, 29].

The study of control, cooperation, estimation problems for distributed and large–scale systems is not a completely new field, and recently there has been significant research activity in this direction (see, among many others, [30, 31, 32, 33, 34, 35, 36] and the references cited therein). Some notable application examples of large-scale distributed systems are: large distribution and communication networks control and analysis, such as water distribution networks [37] and data networks [38], coupled nonlinear systems synchronization [39, 40], formation keeping and rendez-vous of unmanned vehicles [41, 42, 43, 36], satellites [44, 45, 46, 47], and robots [48, 49, 50, 51, 52, 53], transportation systems analysis (such as airplane formation and air traffic management [31, 54], and Automatic Highway Systems (AHS) [55, 56]), collective behavior [57, 58, 59, 60] and the analysis and synthesis of social [61, 62, 63, 64], biological [57, 65, 59, 60, 63, 66], robotic [67, 33, 40, 68, 69] or software [70] networks. Another interesting field of research regarding distributed systems comprises sensor networks [71, 72, 73, 74], consensus problems [75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85] and the distributed estimation topic ([86, 87, 88, 89, 90, 91, 92, 93, 94]). In fact, because of the increasing scientific interest for distributed, networked and large scale systems, distributed estimation algorithms have become extremely useful tools: due to the limited power, computation and communication capabilities of typical sensor networks, it is important for each node to be able to estimate the values of some signals, without the help of a centralized structure, in particular when monitoring large-scale systems or environments.

Common element of all these examples is of course that they are distributed and/or large scale systems, with usually very complex global dynamics. Decentralized control methods suited to these systems were proposed since at least the 1970s, as described in the seminal paper [95], the well known book by Šiljak [96] and in the survey work of Sandell [97]. Since then, there have been many enhancements in the design and analysis of decentralized and, later, distributed control and estimation schemes. Much less research activity there has been in the design of fault diagnosis schemes specifically for distributed and large-scale systems. It is true that a lot of effective works have been proposed for the development of distributed fault diagnosis algorithms suited to discrete event systems (see, for instance, [98, 99, 100, 101, 102, 103, 104]), especially in the Computer Science literature where the problem of fault diagnosis for multi-processor systems is of great importance [105, 106, 107, 108, 109, 101, 110]. A notable contribution in the field of decentralized hybrid systems fault diagnosis is [111].

An interesting scheme for the nonlinear fault detection of spacecraft formations, though it is neither distributed nor decentralized, was presented in [112], and similar results can be found in [113, 114]. Furthermore, analysis of fault scenarios and effects in distributed systems were presented in [115, 116]. But, as far as distributed discrete-time or continuous-time systems are concerned, only qualitative fault diagnosis schemes were attempted very recently [117, 118, 23, 24], or quantitative methods that were formulated for linear systems only [119, 120, 25, 26], with [28], [121] and [122] being some of the few contributions on decentralized/distributed fault detection for large-scale nonlinear systems. Similar considerations are made for the Input-Output case: although there exist several papers dealing with centralized fault diagnosis schemes for input-output systems ([123, 124, 125, 126, 127]), in the literature the contributions addressing distributed schemes for input-output nonlinear systems are few ([128], dealing with discrete-time systems, and [121]).

Concerning Cyber-Physical Systems, in the literature many contributions deal with the description of the technical challenges and design and modeling issues that need to be addressed in order to interface with these modern systems, the technological impact deriving by CPS and the requirements emerging by them ([2] and [129, 130, 131, 132, 133, 134]). As regards reliability, safety and security, some recent works deal with the topic of the detection of attacks against process control systems [135] and cyber-physical attacks in power networks [136], [137], [138], [139]. In [140] and [141] distributed schemes to detect and isolate the attacks on networked control systems using observers are developed. In [141] applications to power networks and robotic formations are presented. All these papers assume the system to be linear.

An interesting solution for distributed fault diagnosis can be devised by exploiting sensor networks. Some works exist addressing the problem of fault diagnosis of sensor networks, such as sensor fault detecting, packet losses and energy consumption monitoring (see [142, 143, 144] and references cited therein), but fewer are the works using sensor networks as a tool for dynamical systems monitoring. Classical methods for quantitative fault diagnosis in the state of the art deal with the use of model-based analytical redundancy techniques and a lot of these works require the centralized collection of the information obtained from the sensor devices. Some exceptions are [24, 26, 29] and other works dealing with discrete-event systems ([98, 102, 104]). Even if methods for distributed estimation already exist (see [85] for a survey and [86, 87] as examples), the links between distributed estimation and distributed monitoring are still lacking. An exception is [145], where a distributed fault detection and isolation technique is designed, relying on decentralized Kalman state-estimation method. In fact, as regard as distributed estimation, an important branch of research is the one represented by distributed Kalman filters ([89]) and their combination with the

diffusion mechanism ([88, 90]).

1.3 Main contributions

In this work, the results presented in [5] and [29], where a distributed fault detection and isolation methodology for nonlinear uncertain large-scale discrete-time dynamical systems is designed, are extended in order to face some of the issues emerging when implementing distributed monitoring architectures in real large-scale networked/distributed systems. The goal is to integrate all the aspects of the monitoring process (measurement, communication, estimation, detection, isolation...) in a comprehensive architecture, able to satisfy the requirements of this kind of systems.

The monitored system is modeled as the interconnection of several subsystems, each one supervised by a single Local Fault Diagnoser (LFD). The local fault decision is based on the knowledge of the local subsystem dynamic model and of an adaptive approximation of the interconnection function with neighboring subsystems. Since subsystems are allowed to overlap, a specially-designed consensus-based estimator is derived in order to improve the detectability of faults affecting variables shared among more than one subsystem. In fact, in order to design high-confidence systems, it is not reasonable to assume exact knowledge of the components and of their interconnections. That's why we consider uncertain systems and we face the uncertainty issue using an adaptive approximation of the interconnection function and of the fault function.

Moreover, we consider the following aspects that can limit the performance of a monitoring architecture:

- measurement noise: a distributed filtering method is proposed, so that both the variance and the mean of the estimation error are minimized by means of a Pareto optimization problem [146];
- non-synchronized measurements and multi-rate systems: a re - synchronization method is proposed;
- communication delays and packet dropouts: a delay compensation strategy is adopted.

The first point is implemented by introducing a sensor networks layer between the physical environment and the diagnosers level. This fact allows the decoupling of the physical and the sensing/computation topology [129], bringing some advantages, such as scalability and resilience of the diagnosis architecture itself.

Anyway, the introduction of the sensor networks layer between the physical world and the diagnosers, and the fact that different sensor networks

may have different sample rates and are not synchronized together, leads to an important issue that must be solved. Since a typical diagnoser receives measurements from different sensor networks, these measurements may not be synchronized: not only they may be received by the diagnoser at different times, but they may have been *taken* at different time instants. As pointed out by Lee ([131] and [130]), the big issue of Cyber-physical systems, and of modern systems in general, is *concurrency*: physical processes are intrinsically concurrent and their coupling with the computing environment requires composition of the cyber processes with the physical ones. The events in the cyber and in the physical levels occur together, but they are not synchronized nor predictable. A work focusing on the synchronization issue for CPS is [147], where a formal complexity-reducing architectural pattern [148] is defined for distributed protocols in multi-rate asynchronous systems. Also in [149], the synchronization problem is considered, where Loosely Time-Triggered Architectures (LTTA) are analyzed in order to relax some strict requirements on synchronization imposed by Time-Triggered Architectures (TTA). In this work, multi-rate, variable sampling systems are considered and a solution to the synchronization issue is proposed, basing on a model-based re-synchronization mechanism to be implemented by each diagnoser.

Moreover, since we deal with distributed, large-scale or networked systems and therefore with communication networks, one issue that has to be considered is inevitably the presence of stochastic delays and packet dropouts, that degrade performance and could be a source of instability. This kind of issue have been considered in several works proposing control architectures, while it is a novel feature in the research regarding FDI schemes. The problem of designing networked control systems managing delays has recently attracted considerable research efforts, as, for instance, [150], [151], [152] and [153]. Since network-induced delays and data packet losses are inherently random and time-varying, they have been modeled in various probabilistic ways ([154], [155], [156]). The problem of delays is even more important when not only the considered system, but also the controllers are distributed: [157] analyzes the problem of cooperative control of a team of distributed agents; in [144] a distributed model predictive control scheme for non linear systems is designed, where the controllers coordinate their actions, taking asynchronous measurements and delays into account. The papers facing fault diagnosis problem for large-scale systems that consider the problem of delays usually deal with centralized fault detection schemes (see, as example, [158], [159], [160] and [161], in which FDI schemes for networked systems are analyzed). An exception is the case of [162] and the references cited therein, which deal with discrete-event systems. In the present work, a delay-compensation strategy is designed.

The thesis is organized as follows: in Part I, the proposed distributed monitoring architecture is described in detail. Specifically:

- Chapter 2 analyzes the structure of the whole architecture, giving a preliminary introduction to the comprehensive architecture, which is analyzed in a more rigorous way in the following chapters;

- in Chapter 3, the physical layer is introduced, by presenting the considered model and the system decomposition we employed;

- Chapter 4 deals with the sensor layer, by describing issues and advantages deriving by its introduction and proposing a distributed estimation method in order to filter measurements noise. The time-varying filter weights allow to minimize at the same time both mean and variance of the estimation error;

- Chapter 5 analyzes the re-synchronization scheme;

- in Chapter 6, the Distributed Fault Detection and Isolation (DFDI) architecture is designed and the delay-compensation strategy is introduced. Fault detectability and isolability conditions are derived and the convergence of the estimation error is proved;

- Chapter 7: the simulation results are presented and discussed.

In Part II, other DFDI frameworks are presented, considering the continuous-time case (Chapter 8) and the case of not completely measurable state for the discrete-time context in Chapter 9 and in a continuous-time framework in Chapter 10.

Finally, in Chapter 11 some concluding remarks are given and future developments are discussed.

1.4 Publications

The research presented in this thesis has been extensively published in archival journals and presented at international conferences. The list is given in the following:

- F. Boem, R. M. G. Ferrari, and T. Parisini, "Distributed Fault Detection and Isolation of Continuous-Time Nonlinear Systems", European Journal of Control, pp.603- 620, Vol. 17, 2011.

- F. Boem, R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "A Distributed Fault Detection Methodology for a Class of Large-scale Uncertain Input-output Discrete-Time Nonlinear Systems". In Proc. 50^{th} IEEE Conf. on Decision and Control and European Control Conference, Orlando, Florida, pp.897-902, 2011.

- F. Boem, R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed Fault Diagnosis for Input-Output Continuous-Time Nonlinear Systems". In Proc. Safeprocess Conference, Mexico City, Mexico, pp.1089- 1094, 2012.

- F. Boem, Y. Xu, C. Fischione, and T. Parisini, "A Distributed Estimation Method for Sensor Networks Based on Pareto Optimization". In Proc. 51^{st} IEEE Conf. on Decision and Control, Maui, Hawaii, pp.775-781, 2012.

- F. Boem, R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed Fault Detection for Uncertain Nonlinear Systems: a Network Delay Compensation Strategy". In Proc. American Control Conference, Washington, 2013.

- F. Boem, Y. Xu, C. Fischione, and T. Parisini, "Distributed Fault Detection using Sensor Networks and Pareto Estimation", European Control Conference, Zurich, 2013.

- F. Boem, R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed Fault Diagnosis for Continuous-Time Nonlinear Systems: the Input-Output case", Annual Reviews in Control (to appear), 2013.

- F. Boem, R. M. G. Ferrari, C. Keliris, T. Parisini, and M. M. Polycarpou, "A Distributed Fault Diagnosis Approach for Large-scale Cyber-Physical Systems", IEEE Transactions on Automatic Control, (submitted).

Part I

The distributed monitoring architecture

Chapter 2

The monitoring architecture

In this chapter, we describe the structure of the proposed monitoring architecture, composed by three layers (see Figure 2.1). The architecture itself can be considered a *cyber-physical* system:

- the presence of the *physical* system, which is monitored in order to detect faults,
- the sensor networks, which have a physical part that interact with the physical environment and a *cyber* part, able to take measurements m of the state variables x, make model-free estimations $\tilde{x}$ and exchange information with each other and with the diagnosers level,
- the diagnosers, which are *cyber*-systems, able to make model-based estimation and exchange information with each other.

In the following, objectives and features of each layer are described, while the analytical details and results are derived and analyzed in the following chapters for each layer.

2.1 The physical layer

The physical layer represents the system that has to be diagnosed for faults. In general, it could be also a computational or a cyber-physical system, but it is modeled by a discrete-time or continuous-time system. Let us consider a large-scale physical system. It can be represented as an uncertain non-linear continuous-time system, which we call the *monolithic* system:

$$\mathscr{S} : \dot{x}(t) = f(x(t), u(t)) + \eta(x(t), u(t), t) + \beta(t - T_0)\phi(x(t), u(t)), \quad (2.1)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$ are the state and the control input of the system, $f : \mathbb{R}^n \times \mathbb{R}^p \mapsto \mathbb{R}^n$ represents the nominal healthy dynamics, η is

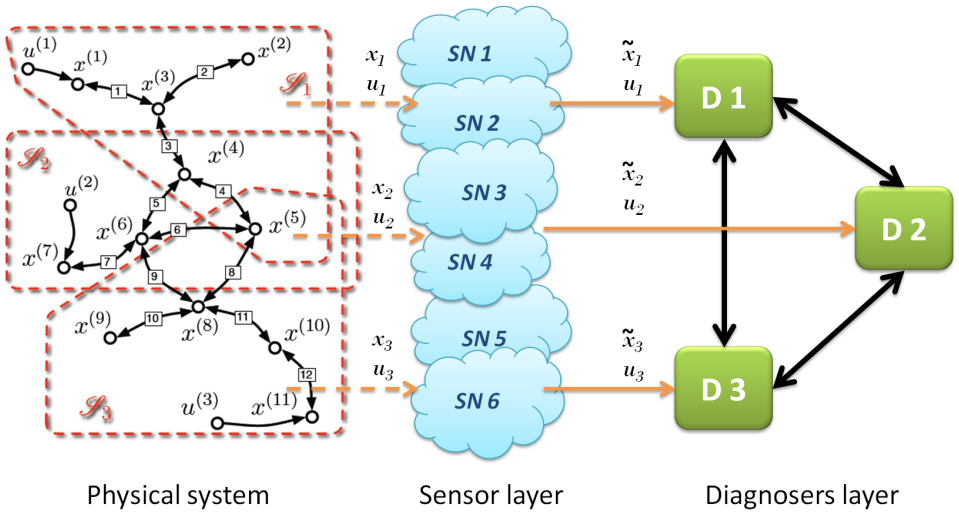

Figure 2.1: Fault diagnosis architecture.

the uncertainty function, including modeling errors and disturbances, and $\beta(t - T_0)\phi$ represents the dynamics of the system due to the presence of a fault: $\beta(t - T_0)$ describes the time-profile of the fault, modeling either incipient or abrupt faults, while ϕ is the unknown fault function, which is null in the case of healthy dynamics.

Using a "divide et impera" approach, as in [29], the monolithic system is decomposed into some subsystems and the influences between subsystems are considered. In Figure 2.1, on the left, the physical layer is represented using its *structural graph* (see Definition 3.1.2). In fact, the structure of a dynamical system, which is a way to describe how the different parts of the system interact with each other, can be represented through a directed graph or digraph [96]. It is constituted by as many nodes as the state and input components: an oriented edge exists between a node a and a node b if a appears in the dynamic equation of b. The fact that the edge is oriented preserves the causality information. These intuitive definitions are formalized in Chapter 3.1. More details can be found in [5].

The analytical features of the physical system, some assumptions and the decomposition are described in detail in Chapter 3.

2.2 The sensor layer

Between the physical layer and the diagnosers, we assume the existence of an intermediate layer containing one or more sensor networks. Each sensor network is made of similar sensors that can measure one or more variables of the physical large scale system, but are not limited to a given subsystem. The reasons for the addition of this layer are:

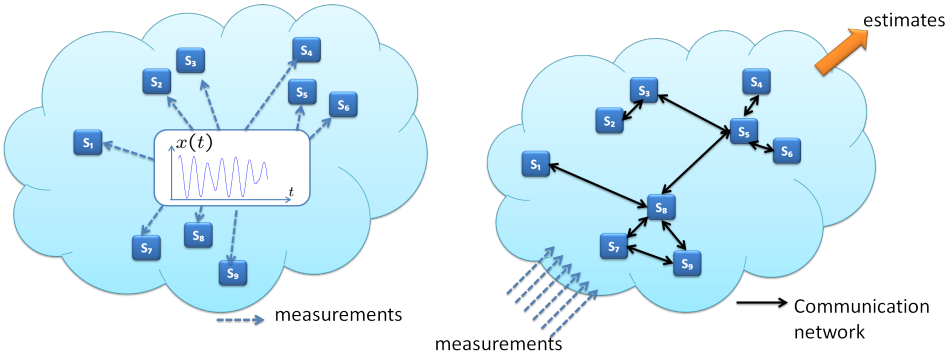

Figure 2.2: A sensor network. The measurement task and the filtering task.

- different sensor networks do not need to be synchronized together by assumption, and do not need to share the same sampling frequency;
- the sensing topology is decoupled from the physical topology, as a given sensor network can measure variables belonging to different subsystems of the monolithic system, each one monitored by a different diagnoser;
- the sensor network itself is more scalable, and resilient to sensor failures, as sensors can be removed or added at any time;
- the introduction of a sensor network allows the use of the distributed estimation technique developed in [163], thus allowing to reduce the measurement error and to define less conservative fault detection thresholds.

Nevertheless, we assume that all the sensors in each sensor network are synchronized together (the problem of clock synchronization in a sensor network is not new, and was solved for instance in [164], [165], [166], [167] and some standard protocols exist). In the r-th sensor network, each sensor device i, with $i = 1, \ldots, N_r$ and N_r the number of sensors in the r-th network, can measure one (the k-th for example) or more components of the system state with a certain measurement noise:

$$m_i^{(k)}(t) = x^{(k)}(t) + w_i^{(k)}(t), \tag{2.2}$$

where $m_i^{(k)}$ is the k-th component of the measurements vector obtained at node i and $w_i^{(k)}$ is the measurement noise (assumed to be zero-mean). Each sensor communicates with its neighboring nodes in its sensor network by means of a communication network, in order to implement kind of a consensus protocol. This can be seen in Fig. 2.2.

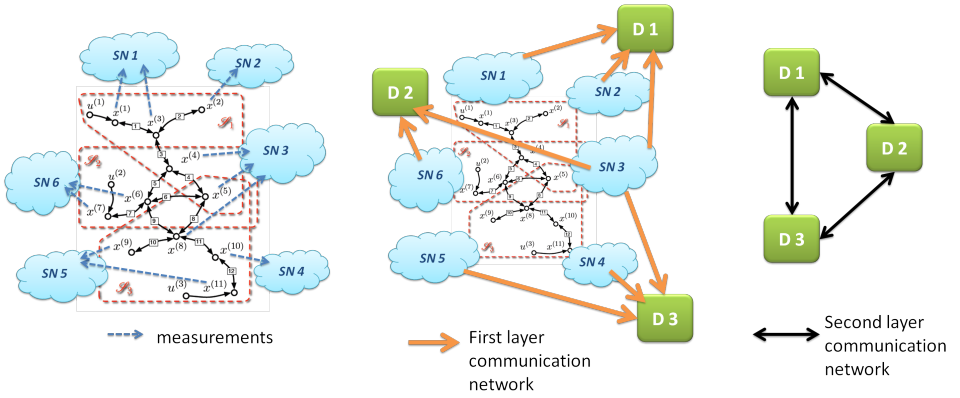

Figure 2.3: Measurements acquisition. From the physical system to the diagnosers, by means of the sensor layer.

We suppose that sensors in each sensor network are synchronized and have the same time-step. Therefore, for estimation purposes, each sensor network works on a discrete-time framework, where measurements are sampled at certain time instants. Different sensor networks may be not synchronized with each other and can have different time-steps. Each sensor in a sensor network filters the measurement noise by communicating with neighboring nodes and computes an estimate $\tilde{x}_i$ of the value of the measured variable x by taking a linear combination of neighboring estimates and measurements. The time-varying weights of the filter are designed in [163] as the solution of a Pareto optimization problem that minimizes both the bias and variance of the estimation errors. It is important to remark that this level does not require the knowledge of the model of the system. At each time step, as we will see in Chapter 4, it is possible to compute mean and variance of the estimation error $e_i^{(k)}(t) = \tilde{x}_i^{(k)}(t) - x^{(k)}(t)$ and so, thanks to the filtering, we can reduce the measurement error w_i and provide a stochastic value of this novel measurement error, the estimation error, which we denote $e(t)$, collecting the components in a column vector. Each sensor network communicates its filtered measurements to some higher level agents, the Local Fault Diagnosers (LFDs). This happens thanks to a communication network that we call "first layer communication network" (see the second picture of Fig. 2.3). This network can induce delays, for instance because of collision between different sensor networks trying to communicate at the same time. We suppose there exists a distributed protocol to determine which sensor inside each sensor network will communicate to diagnosers.

The introduction of the sensor layer and therefore the decoupling of the two topologies, the physical and the sensing/computation topology, may bring some advantages, such as scalability and resilience of the diagnosis

architecture itself.

Summing up, the goal of the sensor layer is threefold:

- the measurement task;
- the filtering estimation task, implementing a kind of consensus protocol;
- the synchronization task inside each sensor network.

In Chapter 4, we will describe with some detail the tasks of the sensor network layer, designing the distributed estimation method and deriving some analytical results. We assume that the sensors label their measurements with a Time Stamp (TS) [168], specifying the time instant at which the measurements were taken. This is an important point, since we deal with communication networks where delays and packet dropouts may happen, because it allows the diagnosers to know the age of the received information. This topic will be clarified in Chapter 5, where the synchronization problem is addressed, and in Section 6.2.2, where a delay compensation strategy is proposed.

2.3 Diagnosers level

The diagnosers level is composed by some agents, the Local Fault Diagnosers, which are designed for fault diagnosis purposes in order to monitor the physical system. We assume to have N LFDs, one for each subsystem of the decomposition of the monolithic system. It is important to remark that the sensor layer is decoupled from the physical and the monitoring layer since each sensor network can measure one or more variables from one or more subsystems. Sensor networks communicate the estimates of the measured variables to the diagnosers. Therefore, the diagnosers can see part of the physical system, what we call a subsystem or at least a part of it at a certain time instant. In fact, it is possible that some measurements are temporarily not received due to communication network problems or sensor failures.

Each diagnoser receives the filtered measurements from a part of the model, a subsystem, and knows the local model, which represents the dynamics of the measured/received variables. Therefore, we suppose that each diagnoser knows the local model of the subsystem it is monitoring.

The diagnosers have mainly two tasks: first, they collect some measurements from sensor networks and they re-organize these measurements in order to use them for the second goal: fault detection and isolation. In the next subsection, we will deal with the synchronization issue, while the fault detection and isolation problem will be addressed in Chapter 6.

2.3.1 Synchronization goal

It is worth noting that the introduction of the sensor networks (SNs) layer between the physical world and the diagnosers, and the fact that the SNs may have different sample rates and may be not synchronized together, leads to an important issue that must be solved. In fact, a typical diagnoser receives the measurements of the local variables of its subsystem from different sensor networks by means of the "first layer communication network" (see Fig. 2.3). This means, then, that these measurements are not synchronized and can be affected by different transmission delays: not only they may be received by the diagnoser at different times, but they may have been *taken* at different time instants. This poses a problem in using them for implementing a model-based fault diagnosis scheme, as we usually assume that all the components of the system input and state (or output) vectors refer to the same time instant.

The proposed solution to this issue is based on a re-synchronization mechanism to be implemented by each diagnoser, formalized in Chapter 5, that is briefly illustrated in the example reported in Fig. 2.4 in which we imagine that there is a single fault diagnoser, receiving the measurements it needs by three different sensor networks. For fault diagnosis purposes, the diagnoser computes its residuals and thresholds only at discrete time instants t, $t+1$, ... This is shown in Fig. 2.4 by plotting a clock signal that represents the rate at which the fault diagnosis task is executed by the diagnoser. Let us consider, for instance, the time instant t. At this time, the fault diagnoser has to compute an estimate that refers to time $t+1$, but, unfortunately, the value of all the needed variables at time t is not yet available due to delays, computation and transmission time. Instead, the latest time at which the variables were sampled by the sensor networks are t_s^1, t_s^2 and t_s^3 and they were received by the diagnoser at time t_a^1, t_a^2 and t_a^3. Note that the diagnoser, thanks to the use of the Time Stamps produced by the sensors, knows the values of the sample time instants t_s^1, t_s^2 and t_s^3, corresponding to the instants when the measurements were taken. The re-synchronization mechanism, that is formalized in the following, takes these filtered measurements, projects them forward to the time instant t, by using a model of the system dynamics, and labels the projected measurements with a novel time stamp, the "virtual Time Stamp", as if measurements were taken at time t. At this point, the diagnoser can use the projected measurements as they were produced by a "virtual" sensor that is synchronized with its fault diagnosis task. It is important to remark that two different kinds of time stamps are used in the proposed architecture: the first, which we call simply "Time Stamp", is the one created by the sensors and used in the first layer communication network; on the other hand, the "virtual Time Stamp" is the one added by diagnosers after the re-synchronization task and communicated in

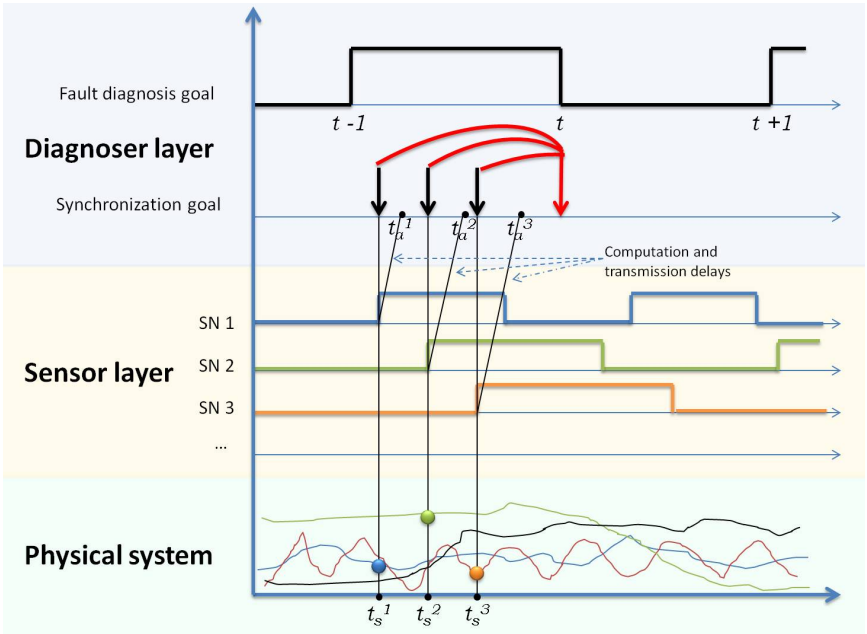

Figure 2.4: Synchronization procedure. We assume there is one diagnoser whose local model depends on three variables, received from three different sensor networks. We plot the clock signals of each layer involved.

the second layer communication network between diagnosers.

Moreover, the diagnosers may be involved in the clock synchronization task between sensor networks and diagnosers and inside sensor networks, acting as masters in a clock synchronization protocol. Different sensor networks may be not synchronized with each other and can have different time-steps. We propose to adopt the IEEE 1588-2002 standard, officially entitled "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", published in 2002 and describing a hierarchical master-slave architecture for clock distribution. Within each sensor network, a synchronization protocol is applied. We propose that each diagnoser is elected as a synchronization master for the sensor networks that communicate with it. Similarly, at the higher level, the diagnosers are synchronized thanks to the same master-slave approach. In this way, all the devices of the monitoring architecture can share the same clock and the use of Time Stamps can be valid. The hierarchical architecture provided by the standard permits to avoid any assumption on the topology of the sensor networks and of the diagnosers.

2.3.2 Diagnosis Goal

The Distributed Fault Detection and Isolation architecture is based on the framework portrayed in [29]. The DFDI tasks are assigned to a network of N *agents*, the Local Fault Diagnosers, each one monitoring a single subsystem. Each LFD is allowed to communicate with neighboring LFDs in order to collaborate on the diagnosis of system components that may be shared between different subsystems. The inter-LFD communication is carried over a packet-switched network, which we call the "second level communication network" (see the right side of Figure 2.3), subject to packet delays and dropouts. In order to manage network delays, the data-packets are Time Stamped (with the "virtual Time Stamp"), so that they contain the information on the time instant the filtered measurements refer to. In this layer we assume to have perfect clock synchronization between the diagnosers. We suppose that, for the FDI task, the diagnosers know also a discrete-time model of the monitored subsystem. Such diagnosers, each of which has a different view on the system, implement consensus techniques in order to reach a common overall fault decision. The *local fault decisions*, regarding the mode of behavior (healthy or one among the possible faulty modes) of the subsystems, are gathered by a higher level agent, which is referred to as *Global Fault Diagnoser* (GFD). The task of the GFD is to coordinate the LFDs and formulate a *global fault decision* about the health of the global system (Fig. 2.5).

Consistently with the fault isolation formulation given in [22], to make the isolation possible, it is assumed that for the global system there exists a *global fault set* collecting all the possible nonlinear fault functions. Because of the decomposition, the introduction of the global fault set leads to the existence, for each subsystem, of a *local fault set* containing $N_{\mathcal{F}_I}$ known types of possible nonlinear fault functions. Each LFD thus relies on $N_{\mathcal{F}_I}+1$ nonlinear adaptive estimators of the local state x_I, with $I \in \{1,\dots,N\}$. The first estimator, called *Fault Detection Approximation Estimator* (FDAE), is based on the local nominal model and is used for fault detection. The remaining $N_{\mathcal{F}_I}$ estimators, called *Fault Isolation Estimators* (FIE), are used to determine which of the possible $N_{\mathcal{F}_I}$ fault in the set $\mathcal{F}_I$ has occurred. The FDAE is the only estimator that is activated by each LFD under normal operating conditions (that is from time $t = 0$ until a fault occurs and is detected). After a fault is detected by any of the N LFDs, the GFD switches each LFD from fault detection to fault isolation operating mode: each LFD activates its own bank of FIEs in order to locally isolate the occurred fault, by employing kind of a *Generalized Observer Scheme* (GOS), (see [17, 169] and Section 1.2.1). The local fault decisions are communicated to the GFD, allowing it to determine which one of the faults in the global set, if any, affects the system. The DFDI architecture will be analyzed in detail in

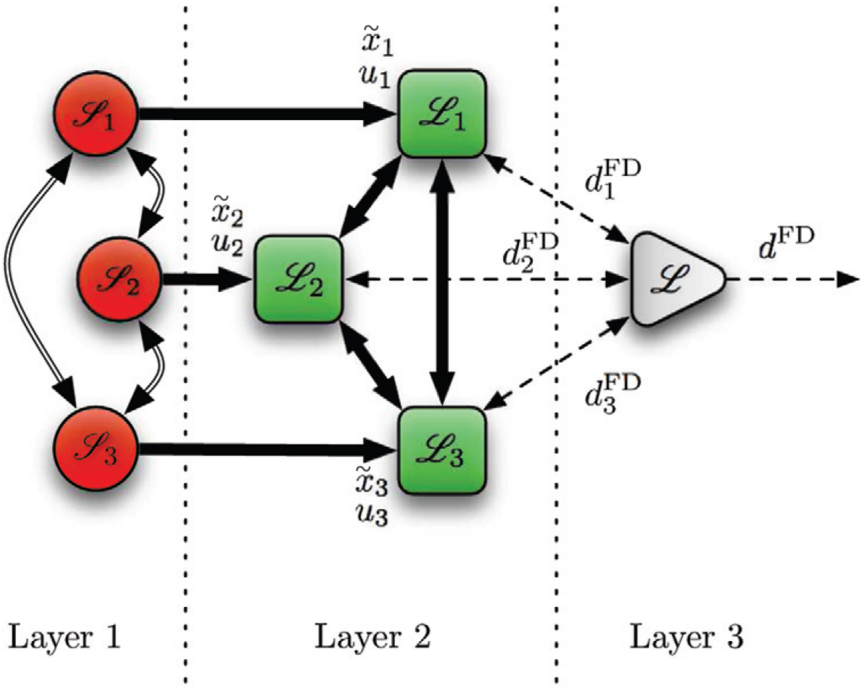

Figure 2.5: A scheme of the proposed DFDI architecture. In this example, in the first layer three subsystems ($\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$) are considered and the arrows represent physical interactions. In the middle layer the local fault diagnosers $\mathscr{L}_I$ are rendered as squares. The arrows from the corresponding subsystem symbolize the transmission of the filtered measurements of local variables by means of the sensor networks, while the arrows between the diagnosers account for information exchange between them. In the third layer, the global diagnoser $\mathscr{L}$ communicates with LFDs in order to formulate a global fault decision d^{FD}. These information exchanges are rendered with dashed arrows because they are sporadic and event-driven.

Chapter 6.

2.4 Concluding remarks

In this chapter, the structure of the monitoring architecture has been briefly illustrated. Goals and tasks of each layer (physical system, sensor layer and diagnosers layer) have been introduced, providing an intuitive picture of the whole architecture. In the following, the proposed approach will be formalized and analytical details and results will be derived for each layer.

Chapter 3

The physical system

In this chapter we introduce the physical layer and the decomposition of the monolithic system. First of all, it is useful to define some intuitive concepts: by the expression "system" or "physical system" we denote the "entity" whose the state should be controlled, monitored or estimated. On the other hand, by "architecture" we will mean a combination of hardware and software used to implement and execute the control or estimation task. Finally, by "subsystem" we mean a logical or a physical portion of a system. After that, we can remind some already expressed concepts and we can give the following qualitative definitions (see also Section 1.1, Fig. 1.1), partly inspired by [96, 170]:

- a system or architecture is distributed if it can be considered as being constituted by a number of subsystems, so that the behavior of any single subsystem is influenced by its own state, and by the state of a (possibly small) subset of all the other subsystems;
- a system or architecture is decentralized if it can be considered as being constituted by a number of subsystems, so that the behavior of any single subsystem is influenced only by its local state, without any interaction with other subsystems.

Let us therefore consider a large-scale distributed system. The dynamics of the physical system can be described by the following model, representing the *monolithic system* $\mathscr{S}$. For the sake of generality, we consider non-linear uncertain systems:

$$\mathscr{S} \ : \ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\eta}(\mathbf{x}, \mathbf{u}, t) + \beta(t - T_0)\boldsymbol{\phi}(\mathbf{x}, \mathbf{u}), \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ and $\mathbf{u} \in \mathbb{R}^{\mathbf{p}}$ denote[1] the state and input vectors, respectively, and $\mathbf{f} : \mathbb{R}^{\mathbf{n}} \times \mathbb{R}^{\mathbf{p}} \mapsto \mathbb{R}^{\mathbf{n}}$ represents the *nominal healthy dynamics*. The

[1] Here and in the rest of the work the use of bold letters indicates that a given quantity is related to the monolithic system.

function $\boldsymbol{\eta} : \mathbb{R}^{\mathbf{n}} \times \mathbb{R}^{\mathbf{p}} \times \mathbb{R}^{+} \mapsto \mathbb{R}^{\mathbf{n}}$ models the uncertainty in the state equation and includes external disturbances as well as modeling errors. The term $\beta(t - T_0)\boldsymbol{\phi}(\mathbf{x}, \mathbf{u})$ represents the deviation in the system dynamics due to a fault: $\beta(t - T_0)$ characterizes the time profile of a fault that occurs at some *unknown* instant T_0, and $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u})$ denotes the nonlinear fault function. This formulation (first introduced in [171]) allows both additive and multiplicative faults (since $\boldsymbol{\phi}$ is a function of $\boldsymbol{x}$ and $\boldsymbol{u}$), as well as more general nonlinear faults. The fault time profile $\beta(t - T_0)$ models *incipient* faults characterized by a decaying exponential time-profile [22]

$$\beta(t - T_0) = \begin{cases} 0 & \text{if} \quad t < T_0 \\ 1 - e^{-\alpha(t - T_0)} & \text{if} \quad t \geq T_0 \end{cases}, \tag{3.2}$$

where $\alpha > 0$ denotes the unknown fault-evolution rate. Small values of α characterize slowly developing faults. For large values of α, the time profile β approaches a step function (the case of an "abrupt" fault time-profile can be obtained as $\alpha \to \infty$ in (3.2)). It is worth noting that the fault time profile given by (3.2) only reflects the developing rate of the fault, while all its other basic features are captured by the function $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u})$, which describes the changes in the dynamics due to the fault.

As already seen in the introduction chapter, when analyzing distributed, large-scale, networked complex systems, the use of centralized architectures may be not possible nor desirable. Moreover, the use of decentralized architectures, even if simpler, would not consider the influences between different subsystems. Therefore, we decided to use a distributed architecture. In this work, as in [29], a *divide et impera* approach will be adopted, in order to decompose the monolithic system $\mathscr{S}$ into N subsystems[2] $\mathscr{S}_I$, $I = 1, \ldots, N$, each characterized by a *local* state vector $x_I \in \mathbb{R}^{n_I}$ and monitored by one local *agent*.

3.1 The decomposition

The decomposition of the monolithic system $\mathscr{S}$ is based on decomposing its structural graph [172, 96, 173].

First of all, the system *structure* is defined using graph theory [174]. Let us consider the following definitions:

Definition 3.1.1: The *structure* $\Sigma_{\mathscr{S}}$ of a dynamical system $\mathscr{S}$ having a state vector $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ and an input vector $\mathbf{u} \in \mathbb{R}^{\mathbf{p}}$ is the set of ordered pairs

$$\begin{aligned} \Sigma_{\mathscr{S}} \triangleq & \{(\mathbf{x^{(i)}}, \mathbf{x^{(j)}}) : \mathbf{i}, \mathbf{j} \in \{\mathbf{1}, \ldots, \mathbf{n}\}, \text{"}\mathbf{x^{(i)}} \text{ affects } \mathbf{x^{(j)}}\text{"}\} \\ & \cup \{(\mathbf{u^{(i)}}, \mathbf{x^{(j)}}) : \mathbf{i} \in \{\mathbf{1}, \ldots, \mathbf{p}\}, \mathbf{j} \in \{\mathbf{1}, \ldots, \mathbf{n}\}, \text{"}\mathbf{u^{(i)}} \text{ affects } \mathbf{x^{(j)}}\text{"}\}. \end{aligned}$$

[2] In the paper, a capital-case index will denote a specific sub-system.

Definition 3.1.2: The *structural graph* [96] of a dynamical system $\mathscr{S}$, having a state vector $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ and an input vector $\mathbf{u} \in \mathbb{R}^{\mathbf{p}}$, is the directed graph (*digraph*) $\mathcal{G} \triangleq \{\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G}\}$ having the node set $\mathcal{N}_\mathcal{G} \triangleq \{\mathbf{x^{(i)}} : \mathbf{i} \in \{\mathbf{1}, \ldots, \mathbf{n}\}\} \cup \{\mathbf{u^{(i)}} : \mathbf{i} \in \{\mathbf{1}, \ldots, \mathbf{p}\}\}$ and the system structure $\Sigma_\mathscr{S}$ as the arc set, that is $\mathcal{E}_\mathcal{G} = \Sigma_\mathscr{S}$.

Definition 3.1.3: A *decomposition* $\mathcal{D}$ of dimension N of the large-scale system $\mathscr{S}$ is a multiset $\mathcal{D} \triangleq \{\mathscr{S}_1, ..., \mathscr{S}_N\}$ made of N subsystems, defined through a multiset $\{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$ of index sets, such that for each $I \in \{1, \ldots, N\}$ the following holds:

1. $\mathcal{I}_I \neq \emptyset$;
2. $\mathcal{I}_I^{(j)} \leq \mathbf{n}$, for each $j \in \{1, \ldots, n_I\}$;
3. the subdigraph of $\mathcal{G}$ induced by $\mathcal{I}_I$ must be weakly connected, that is, each component of x_I must act on or must be acted on by at least another component of x_I;
4. $\bigcup_{I=1}^{N} \mathcal{I}_I = \{\mathbf{1}, ..., \mathbf{n}\}$.

In this last definition, the first point prevents the definition of trivial empty subsystems, the second is necessary for well-posedness, while the third point avoids that resulting subsystems have isolated state components. Finally, the fourth point requires that the decomposition covers the whole original monolithic system. It is worth noting that the above decomposition does not require that for any two subsystems $\mathcal{I}_I \cap \mathcal{I}_J = \emptyset$, $I, J \in \{1, \ldots, N\}$. This means that it is possible for a state component of $\mathscr{S}$ to be assigned to more than one subsystems, thus being "shared" and giving rise to *overlapping decompositions*.

It is possible therefore to define the decomposition of the monolithic system: to decompose a monolithic system $\mathscr{S}$ having a state vector $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$, an input vector $\mathbf{u} \in \mathbb{R}^{\mathbf{p}}$ and a structural graph $\mathcal{G} = (\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, we define $N \geq 1$ subsystems $\mathscr{S}_I$, with $I \in \{1, \ldots, N\}$, each one having a *local state* vector $x_I \in \mathbb{R}^{n_I}$ and a *local input* vector $u_I \in \mathbb{R}^{p_I}$. These local vectors are composed by the components of the monolithic system vectors $\mathbf{x}$ and $\mathbf{u}$, ordered basing on the sets of indexes $\mathcal{I}_I \triangleq (\mathcal{I}_I^{(1)}, \ldots, \mathcal{I}_I^{(n_I)})$, called *extraction index set* [27, 175, 8].

Definition 3.1.4: The *local state* $x_I \in \mathbb{R}^{n_I}$ and the *local input* $u_I \in \mathbb{R}^{p_I}$ of a dynamical subsystem $\mathscr{S}_I$, arising from the decomposition of a monolithic system $\mathscr{S}$, are respectively the vectors $x_I \triangleq \mathrm{col}(\mathbf{x^{(j)}} : \mathbf{j} \in \mathcal{I}_I)$ and $u_I \triangleq \mathrm{col}(\mathbf{u^{(k)}} : (\mathbf{u^{(k)}}, \mathbf{x^{(j)}}) \in \mathcal{E}_\mathcal{G}, \mathbf{j} \in \mathcal{I}_I, \mathbf{k} = \mathbf{1}, \ldots, \mathbf{p})$, where $\mathcal{I}_I$ is the extraction index set of the I–th subsystem.

According to Definition 3.1.4, the local input contains all the input components that affect at least one component of the local state vector. In this way, the structural graph of the I–th subsystem can be easily defined as the subgraph $\mathcal{G}_I$ induced on $\mathcal{G}$ by the subset made of all the components of x_I together with those of u_I.

Since the decomposition is assumed to be possibly overlapped, some components of $\mathbf{x}$ belong to more than one subsystems. It is therefore necessary to define the concepts of *shared state variable* and *overlap index set*.

Definition 3.1.5: A shared state variable $\mathbf{x}^{(\mathbf{s})}$ is a component of $\mathbf{x}$ such that $\mathbf{s} \in \mathcal{I}_I \cap \mathcal{I}_J$, for some $I,\ J \in \{1, \dots N\}$, $I \neq J$ and a given decomposition $\mathcal{D}$ of cardinality N.

Definition 3.1.6: The *overlap index set* of subsystems sharing a variable $\mathbf{x}^{(\mathbf{s})}$ is the set $\mathcal{O}_{\mathbf{s}} \triangleq \{I : \mathbf{s} \in \mathcal{I}_I\}$, whose cardinality is $N_{\mathbf{s}} \triangleq |\mathcal{O}_{\mathbf{s}}|$.

In the following, the notation $x_I^{(s_I)}$, with $x_I^{(s_I)} \equiv \mathbf{x}^{(\mathbf{s})}$, is used to denote the fact that the $\mathbf{s}$–th state component of the original large–scale system, after the decomposition became the s_I–th of the I–th subsystem, $I \in \mathcal{O}_{\mathbf{s}}$.

At this point, let us consider the interactions between different subsystems: the external variables influencing the dynamics of local state components of subsystem $\mathscr{S}_I$ are defined as the *interconnection variables* z_I.

Definition 3.1.7: The *interconnection variables* vector $z_I \in \mathbb{R}^{p_I}$, $(p_I \leq \mathbf{n} - n_I)$ of the subsystem $\mathscr{S}_I$ is the vector $z_I \triangleq \mathrm{col}(\mathbf{x}^{(\mathbf{k})} : (\mathbf{x}^{(\mathbf{k})}, \mathbf{x}^{(\mathbf{j})}) \in \mathcal{E}_{\mathcal{G}},\ \mathbf{j} \in \mathcal{I}_I,\ \mathbf{k} \in \{\mathbf{1}, \dots, \mathbf{n}\})$.

The set of subsystems acting on a given subsystem $\mathscr{S}_I$ through the interconnection vector z_I is the *neighbors index set* $\mathcal{V}_I$.

Definition 3.1.8: The *neighbors index set* of a subsystem $\mathscr{S}_I$ is the set $\mathcal{V}_I \triangleq \{K : \exists\, (\mathbf{x}^{(\mathbf{k})}, \mathbf{x}^{(\mathbf{j})}) \in \mathcal{E}_{\mathcal{G}},\ \mathbf{k} \in \mathcal{I}_K,\ \mathbf{j} \in \mathcal{I}_I, K \in \{1, \dots, N\} \setminus \{I\}\}$.

The decomposition task for nonlinear systems is not a trivial problem: unlike linear systems, for which powerful model decomposition techniques and descriptions exist (see as example [32, 176]), for nonlinear systems, in general, it is not possible to devise an additive decomposition into purely local and purely interconnection terms.

In this work, the following general decomposition, as in [96], is considered:

$$\mathscr{S}_I :\ \dot{x_I} = f_I(x_I, u_I) + g_I(x_I, z_I, u_I) + \beta(t - T_0)\phi_I(x_I, z_I, u_I)\,, \qquad (3.3)$$

where $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \mapsto \mathbb{R}^{n_I}$ is the *local nominal* function and $g_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{p_I} \mapsto \mathbb{R}^{n_I}$ represents the *interconnection function*, where the effects of the local uncertainty term η_I has also been incorporated, with $\eta_I \triangleq \mathrm{col}(\boldsymbol{\eta}^{(\mathbf{j})} : \mathbf{j} \in \mathcal{I}_I)$. Moreover, $u_I \in \mathbb{R}^{p_I}$, $(p_I \leq \mathbf{p})$ is the *local input* (see Definition

3.1.4), $z_I \in \mathbb{R}^{p_I}$, $(p_I \leq \mathbf{n} - n_I)$ is the vector of *interconnection variables* (see Definition 3.1.7), and $\phi_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{p_I} \mapsto \mathbb{R}^{n_I}$ is the *local fault function.*

3.1.1 Example

To gain some insight into the afore-described decomposition approach, consider the example illustrated in Fig. 3.1 [5], where a specific decomposition of a system $\mathscr{S}$ into three overlapping subsystems $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$ is considered. The example of Fig. 3.1 corresponds to the dynamics of a 11–tank system, which will be described in Chapter 7 for simulations.

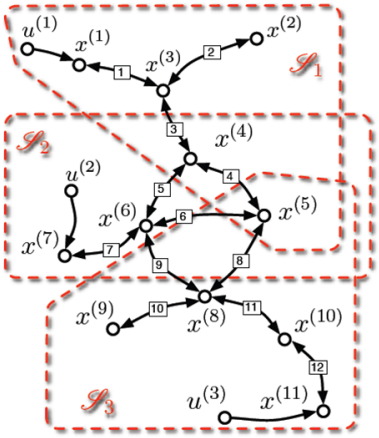

Figure 3.1: Example of decomposition of a system $\mathscr{S}$ into three overlapping subsystems $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$.

The decomposition shown in this example is such that:

$$x_1 = [\boldsymbol{x^{(1)}}, \boldsymbol{x^{(2)}}, \boldsymbol{x^{(3)}}, \boldsymbol{x^{(4)}}, \boldsymbol{x^{(5)}}]^\top,$$
$$x_2 = [\boldsymbol{x^{(4)}}, \boldsymbol{x^{(5)}}, \boldsymbol{x^{(6)}}, \boldsymbol{x^{(7)}}]^\top$$

and

$$x_3 = [\boldsymbol{x^{(5)}}, \boldsymbol{x^{(8)}}, \boldsymbol{x^{(9)}}, \boldsymbol{x^{(10)}}, \boldsymbol{x^{(11)}}]^\top$$

are the local states, $u_1 = \boldsymbol{u^{(1)}}$, $u_2 = \boldsymbol{u^{(2)}}$ and $u_3 = \boldsymbol{u^{(3)}}$ are the local inputs, $z_1 = [\boldsymbol{x^{(6)}}, \boldsymbol{x^{(8)}}]^\top$, $z_2 = [\boldsymbol{x^{(3)}}, \boldsymbol{x^{(8)}}]^\top$ and $z_3 = [\boldsymbol{x^{(4)}}, \boldsymbol{x^{(6)}}]^\top$ are the interconnection variables. Furthermore, $\boldsymbol{x^{(4)}} \equiv x_1^{(4)} \equiv x_2^{(1)}$ and $\boldsymbol{x^{(5)}} \equiv x_1^{(5)} \equiv x_2^{(2)} \equiv x_3^{(1)}$ are shared variables with $\mathcal{O}_{\mathbf{4}} = \{1, 2\}$ and $\mathcal{O}_{\mathbf{5}} = \{1, 2, 3\}$.

3.2 Concluding remarks

In this Chapter some modeling issues have been addressed. More specifically, the model of the physical system has been presented and the system

decomposition problem, introduced to overcome the limits due to the use of a centralized architecture when dealing with large-scale systems, has been described in detail. A divide et impera approach is used, in order to decompose the (possible large-scale) original monitoring problem into a number of smaller problems, easier to solve. In the next chapter, the sensor layer will be analyzed. We suppose that a set of sensor networks measures the state variables of the physical system. Each sensor, by communicating with neighboring sensors and by implementing a kind of consensus protocol, filters measurements noise. The distributed estimation method, able to minimize at the same time both the bias and the variance of the estimation error, will be described in detail and some analytical and simulation results will be presented.

Chapter 4

The sensor layer

In this chapter we describe the functionalities of the sensor layer, examining in depth the distributed estimation method implemented by the sensor networks. As already explained, the main goals of sensor networks are to measure state variables, to filter measurement noise and to communicate the estimated variables to the diagnosers. The introduction of the sensor layer between the physical layer and the diagnosers one allows to decouple the two topologies, improving the robustness, reliability and scalability of the monitoring architecture. Some sensors may be removed or added without implying any change in the monitoring scheme. Let us now analyze the filtering method, first proposed in [146]. It can be implemented in a distributed way by the sensor nodes and minimizes at the same time both the bias and the variance of the estimation error, by solving a multi-objective optimization problem in a Pareto framework.

4.1 The distributed estimation problem

Let us consider R sensor networks. Each r-th sensor network, with $r = 1, \dots, R$, is constituted by $N_r > 1$ sensor nodes. Each sensor takes a measurement of one (or more) common scalar signals $x^{(k)}(t)$, affected by additive noise:

$$m_i^{(k)}(t) = x^{(k)}(t) + w_i^{(k)}(t), \quad i = 1, \dots, N_r,$$

where $x^{(k)}(t)$ represents the k-th state component of the monolithic system, with $k = 1, \dots, n$, and $w_i^{(k)}(t)$ is a zero-mean white noise. Collecting the variables of all sensors in vectors, it is possible to write:

$$m^{(k)}(t) = x^{(k)}(t)\mathbb{1} + w^{(k)}(t).$$

Here and in the following, by $\mathbb{1}$ and I we denote the vector $(1, \dots, 1)^\top$ and the identity matrix, respectively. We assume the covariance matrix Σ_k of $w^{(k)}(t)$ to be diagonal: here, for the sake of simplicity and without loss of

generality, we choose $\Sigma_k = \sigma_k^2 I$, but the following results hold also in the more general and realistic case $\Sigma_k = \text{diag}([\sigma_{k1}^2, ..., \sigma_{kN_r}^2])$.

As in [86], the communication network is modeled as an undirected graph $\mathcal{G}^k = (\mathcal{V}^k, \mathcal{E}^k)$, where $\mathcal{N}_i^k = \left\{ j \in \mathcal{V}^k : (j,i) \in \mathcal{E}^k \right\} \cup \{i\}$ is the set of neighbors of node $i \in \mathcal{V}^k$ plus the node itself. It is assumed that there are no message losses in the communication network between sensors.

Each node i computes an estimate $\tilde{x}_i^{(k)}(t)$ of the signal $x^{(k)}(t)$, taking a linear combination of neighboring estimates and measurements:

$$\tilde{x}_i^{(k)}(t) = \sum_{j \in \mathcal{N}_i^k} l_{k,i,j}(t) \tilde{x}_j^{(k)}(t-1) + \sum_{j \in \mathcal{N}_i^k} h_{k,i,j}(t) m_j^{(k)}(t).$$

In vector notation, the above becomes

$$\tilde{x}^{(k)}(t) = L_k(t)\tilde{x}^{(k)}(t-1) + H_k(t)m^{(k)}(t), \tag{4.1}$$

where L_k and H_k can be seen as the adjacency matrices of two graphs with time-varying weights. The algorithm is initialized with $\tilde{x}_j^{(k)}(0) = m_i^{(k)}(0),\ j \in \mathcal{N}_i^k$.

The estimation error $e^{(k)}(t) = \tilde{x}^{(k)}(t) - x^{(k)}(t)\mathbb{1}$ can be computed:

$$\begin{aligned} e^{(k)}(t) =& L_k(t)e^{(k)}(t-1) + x^{(k)}(t)(L_k(t) + H_k(t) - I)\mathbb{1} \\ & - \delta^{(k)}(t)L_k(t)\mathbb{1} + H_k(t)w^{(k)}(t), \end{aligned} \tag{4.2}$$

where $\delta^{(k)}(t) = x^{(k)}(t) - x^{(k)}(t-1)$. Moreover, the expected value of the estimation error dynamics with respect to the stochastic variable $w^{(k)}(t)$ is given by

$$\mathbb{E}e^{(k)}(t) = L_k(t)\mathbb{E}e^{(k)}(t-1) + x^{(k)}(t)(L_k(t) + H_k(t) - I)\mathbb{1} - \delta^{(k)}(t)L_k(t)\mathbb{1}\,. \tag{4.3}$$

Now, we introduce the following

Assumption 4.1.1: We assume that $(L_k(t) + H_k(t))\mathbb{1} = \mathbb{1}$.

Assumption 4.1.1 is needed to guarantee the convergence properties of the centralized estimation error, that are derived in [86] (see section B) and that hold likewise in our case: if, in addition, $H_k(t)\mathbb{1} = \mathbb{1}$, then, the expected value of the estimation error converges to 0 and so the estimation error is unbiased; otherwise, if $x^{(k)}(t)$ is slowly varying (that is, $\delta^{(k)}(t)$ is bounded), then also $\left\|\mathbb{E}e^{(k)}(t)\right\|$ tends to a bounded value in the neighborhood of the origin. In this way, if we consider the i-th node, the expected value of the estimation error can be computed in a distributed way as:

$$\mathbb{E}e_i^{(k)}(t) = \kappa_{i,k}^\top(t)\mathbb{E}\varepsilon_{i,k}(t-1) - \kappa_{i,k}^\top(t)\delta^{(k)}(t)\mathbb{1}, \tag{4.4}$$

where $\varepsilon_{i,k}(t)$ collects the estimation errors available at node i for the k-th state component, ordered according to their indexes:

$$\varepsilon_{i,k} = (e_{i_1}^{(k)}, \dots, e_{i_{M_i^k}}^{(k)})^\top, i_1 < \dots < i_{M_i^k},$$

with $M_i^k = \left|\mathcal{N}_i^k\right|$ the number of neighbors of node i plus i itself (where by $|\cdot|$ we denote cardinality), and we introduced $\kappa_{i,k}^\top(t)$ and $\eta_{i,k}^\top(t)$ that correspond to the non–zero elements of the i-th row of matrices $L_k(t)$ and $H_k(t)$ respectively. Note that Assumption 4.1.1 is equivalent to require

$$(\kappa_{i,k}(t) + \eta_{i,k}(t))^\top \mathbb{1} = 1,$$

that can be computed in a distributed way. Now, we are able to compute the variance of the estimation error:

$$\mathbb{E}(e_i^{(k)}(t) - \mathbb{E}e_i^{(k)}(t))^2 = \kappa_{i,k}^\top(t)\Gamma_{i,k}(t-1)\kappa_{i,k}(t) + \sigma_k^2\eta_i^\top(t)\eta_{i,k}(t), \qquad (4.5)$$

where $\Gamma_{i,k}(t) = \mathbb{E}(\varepsilon_{i,k}(t) - \mathbb{E}\varepsilon_{i,k}(t))(\varepsilon_{i,k}(t) - \mathbb{E}\varepsilon_{i,k}(t))^\top$ is the error covariance matrix. When considering the noise covariance matrix

$$\Sigma_k = \mathrm{diag}([\sigma_{\mathrm{k,1}}^2, ..., \sigma_{\mathrm{k,N_r}}^2]),$$

it is sufficient to replace σ_k^2 with an appropriate defined local matrix $Q_i^k = \mathrm{diag}([\sigma_{\mathrm{k,i_1}}^2, ..., \sigma_{\mathrm{k,i_{M_i}}}^2])$.

We want to determine $\kappa_{i,k}^\top(t)$ and $\eta_{i,k}^\top(t)$ at each step, simultaneously minimizing both the variance of the estimation error and the bias. We remark that this approach, proposed in [146], differs substantially from the one proposed in [86] because in the latter paper only the variance of the estimation error is minimized. This is important since the estimation error can be affected also by bias term when dealing with decentralized scenarios, such as sensor networks. We propose to formulate the problem as a Pareto Optimization problem:

$$\begin{aligned} \min_{\kappa_{i,k}^\top, \eta_{i,k}^\top} \quad & (1-\rho_{i,k})V_{i,k} + \rho_{i,k}B_{i,k}^2 \\ \text{s.t. } & (\kappa_{\mathrm{i,k}}(\mathrm{t}) + \eta_{\mathrm{i,k}}(\mathrm{t}))^\top \mathbb{1} = 1, \end{aligned} \qquad (4.6)$$

where $0 \le \rho_{i,k} \le 1$, $B_{i,k} = \mathbb{E}e_i^{(k)}(t)$ is the bias term of the estimation error and $V_{i,k} = \mathbb{E}(e_i^{(k)}(t) - \mathbb{E}e_i^{(k)}(t))^2$ is the variance term. This problem can be solved in a distributed way by each node i. It can be rewritten as:

$$\begin{aligned} \min_{\kappa_{i,k}^\top, \eta_{i,k}^\top} \quad & \kappa_{i,k}^\top(t)\Theta_{i,k}(t)\kappa_{i,k}(t) + (1-\rho_{i,k})\sigma_k^2\eta_i^\top(t)\eta_{i,k}(t) \\ \text{s.t. } & (\kappa_{\mathrm{i,k}}(\mathrm{t}) + \eta_{\mathrm{i,k}}(\mathrm{t}))^\top \mathbb{1} = 1, \end{aligned} \qquad (4.7)$$

where

$$\Theta_{i,k}(t) =(1-\rho_{i,k})\Gamma_{i,k}(t-1)+\rho_{i,k}\Lambda_{i,k}(t),$$

with $\Lambda_{i,k}(t) = (\mathbb{E}\varepsilon_{i,k}(t-1)-\delta^{(k)}(t)\mathbb{1})(\mathbb{E}\varepsilon_{i,k}(t-1)-\delta^{(k)}(t)\mathbb{1})^\top$. The problem is convex since the cost function and the constraint are convex: $\Gamma_{i,k}(t-1)$ is positive definite since it represents the covariance matrix, $\Lambda_{i,k}(t)$ is positive semi-definite, so the linear combination of these two matrices with non-negative coefficients is positive definite, as well. In addition, Slater's conditions are satisfied, so strong duality holds [177]. We derived optimal values of $\kappa_{i,k}^\top(t)$ and $\eta_{i,k}^\top(t)$ for a fixed $\rho_{i,k}$.

Proposition 4.1.1: For a given positive definite matrix $\Theta_{i,k}(t)$, the solution to the optimization problem is:

$$\kappa_{i,k}(t) = \frac{(1-\rho_{i,k})\sigma_k^2\Theta_{i,k}^{-1}\mathbb{1}}{(1-\rho_{i,k})\sigma_k^2\mathbb{1}^\top\Theta_{i,k}^{-1}\mathbb{1}+M_i^k}, \tag{4.8}$$

$$\eta_{i,k}(t) = \frac{\mathbb{1}}{(1-\rho_{i,k})\sigma_k^2\mathbb{1}^\top\Theta_{i,k}^{-1}\mathbb{1}+M_i^k}. \tag{4.9}$$

Proof: Since the problem is convex and Slater's condition holds, the Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient for optimality:

$$(\kappa_{i,k}^*+\eta_{i,k}^*)^\top\mathbb{1}-1=0,$$

$$2\Theta_{i,k}\kappa_{i,k}^*+\nu_{i,k}^*\mathbb{1}=0,$$

$$2\sigma^2(1-\rho_{i,k})\eta_{i,k}^*+\nu_{i,k}^*\mathbb{1}=0,$$

where $(\kappa_{i,k}^*, \eta_{i,k}^*)$ are the primal optimal points and $\nu_{i,k}^*$ is the dual optimal variable. The last two KKT conditions derive from $\nabla_{\kappa_{i,k}}L(\kappa_{i,k},\eta_{i,k},\nu_{i,k})$ and $\nabla_{\eta_{i,k}}L(\kappa_{i,k},\eta_{i,k},\nu_{i,k})$ with L being the Lagrangian form

$$L(\kappa_{i,k},\eta_{i,k},\nu_{i,k}) = \kappa_{i,k}^\top\Theta_{i,k}\kappa_{i,k}+(1-\rho_{i,k})\sigma_k^2\eta_i^\top\eta_{i,k}+\nu_{i,k}((\kappa_{i,k}+\eta_{i,k})^\top\mathbb{1}-1).$$

It is possible to provide the solution in a closed form, simply by solving the system of equations defined by the KKT conditions. ■

4.2 Choice of the Pareto parameter

In the literature, the best value of $\rho_{i,k}$ is determined by building the Pareto trade–off curve and selecting the "knee–point" of this curve, that is, choosing $\rho_{i,k}^*$ such that $B_{i,k}$ and $V_{i,k}$, computed with the values $\kappa_{i,k}^{\top *}(\rho_{i,k}^*)$ and $\eta_{i,k}^{\top *}(\rho_{i,k}^*)$, are $V_{i,k} = B_{i,k}^2$. It is worth noting that also these values are functions of

$\rho_{i,k}$. The desired condition can be obtained by solving the following further problem:

$$\min_{\rho_{i,k}}(V_{i,k}(\kappa_{i,k}^{\top *}(\rho_{i,k}),\eta_{i,k}^{\top *}(\rho_{i,k}))-B_{i,k}^2(\kappa_{i,k}^{\top *}(\rho_{i,k}),\eta_{i,k}^{\top *}(\rho_{i,k})))^2.$$

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, [178], [179]). We tested different approaches for the definition of the Pareto parameter. Specifically, we chose to compute it locally, using the Nelder-Mead simplex algorithm as described in [180], to minimize the cost function $(1-\rho)V_{i,k}+\rho B_{i,k}^2$ with the values of $B_{i,k}$ and $V_{i,k}$ obtained at the previous step. 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 $\Theta_{i,k}^{-1}$.

In this connection, it is worth noting that, by setting the Pareto parameter as $\rho_{i,k}=1$, it turns out that only the bias is minimized and the optimal cost function is given by

$$B_{i,k}^2=0,\qquad V_{i,k}=\frac{\sigma_k^2}{M_i^k}. \tag{4.10}$$

It is worth noting that the variance becomes smaller with the increasing of the number of neighboring nodes. On the other hand, by setting $\rho_{i,k}=0$, the variance is minimized and we obtain

$$\begin{aligned}B_{i,k}^2&=(\frac{\sigma_k^2\Gamma_i^{-1}\mathbb{1}}{\sigma_k^2\mathbb{1}^\top\Gamma_{i,k}^{-1}\mathbb{1}+M_i^k})^\top\Lambda_{i,k}(\frac{\sigma_k^2\Gamma_i^{-1}\mathbb{1}}{\sigma_k^2\mathbb{1}^\top\Gamma_{i,k}^{-1}\mathbb{1}+M_i^k}),\\ V_{i,k}&=(\frac{\sigma_k^2\Gamma_i^{-1}\mathbb{1}}{\sigma_k^2\mathbb{1}^\top\Gamma_{i,k}^{-1}\mathbb{1}+M_i^k})^\top\Gamma_{i,k}(\frac{\sigma_k^2\Gamma_i^{-1}\mathbb{1}}{\sigma_k^2\mathbb{1}^\top\Gamma_{i,k}^{-1}\mathbb{1}+M_i^k})\\ &\quad+\sigma_k^2(\frac{\mathbb{1}}{\sigma_k^2\mathbb{1}^\top\Gamma_{i,k}^{-1}\mathbb{1}+M_i^k})^2.\end{aligned} \tag{4.11}$$

Pareto parameter can be set depending on the required features. In the next Section, it will be shown how to define a bound on the bias by appropriately setting this parameter.

4.3 Bounds on the bias

As demonstrated in [86], the size of the cumulative bias can be kept small with respect to the signal to track by defining a proper value of $\gamma_{\max}(L_k(t))$, which denotes the largest singular value of matrix $L_k(t)$. This means that it is possible to bound the bias by defining the following global constraint:

$$\gamma_{\max}(L_k(t))\leq f(\Delta^{(k)},\Upsilon_k), \tag{4.12}$$

where Υ_k denotes the Signal-to-Noise Ratio $\Upsilon_k = SNR/N_r$, with $SNR = P_s/P_b$, P_s denotes the average power of the signal $x^{(k)}$ and P_b the desired power of the biases of the average of the estimates, $\Delta^{(k)}$ is a bound on the derivative of the signal, that is $\left|\delta^{(k)}\right| \leq \Delta^{(k)}$, and

$$f(\Delta^{(k)}, \Upsilon_k) = \frac{\sqrt{\Upsilon_k}}{\sqrt{\Upsilon_k} + \Delta^{(k)}}.$$

In [86], it is shown that the global constraint in Eq. (4.12) holds when the following local constraint holds:

$$\|\kappa_{i,k}\|^2 \leq \psi_{i,k},$$

where $\psi_{i,k} > 0$ is a suitable constant scalar that can be computed locally (for more details, see [86]). This new constraint ensures the stability of the estimation error even if it leads to a distributed solution which is in general different from the centralized one. Problem (4.7) can be reformulated, taking into account the bound on the bias, as follows:

$$\begin{aligned} \min_{\kappa_{i,k}^\top, \eta_{i,k}^\top} \quad & \kappa_{i,k}^\top(t)\Theta_{i,k}(t)\kappa_{i,k}(t) + (1-\rho_{i,k})\sigma_k^2\eta_{i,k}^\top(t)\eta_{i,k}(t) \\ \text{s.t. } & (\kappa_{i,k}(t) + \eta_{i,k}(t))^\top \mathbb{1} = 1 \\ & \|\kappa_{i,k}\|^2 \leq \psi_{i,k}. \end{aligned} \tag{4.13}$$

The solution of this problem cannot be computed in a closed form. Because of the convexity of the new constraint, the problem is convex. Strong duality holds, and the KKT conditions are necessary and sufficient for optimality; therefore, the primal and dual $(\kappa_{i,k}^*, \eta_{i,k}^*)$ and $(\upsilon_{i,k}^*, \nu_{i,k}^*)$ have to satisfy:

$$(\kappa_{i,k}^*)^\top \kappa_{i,k}^* - \psi_{i,k} \leq 0, \qquad (\kappa_{i,k}^* + \eta_{i,k}^*)^\top \mathbb{1} - 1 = 0,$$

$$\upsilon_{i,k}^* \geq 0, \qquad \upsilon_{i,k}^*((\kappa_{i,k}^*)^\top \kappa_{i,k}^* - \psi_{i,k}) = 0,$$

$$2(\Theta_{i,k} + \upsilon_{i,k}^* I)\kappa_{i,k}^* + \nu_{i,k}^* \mathbb{1} = 0, \quad 2\sigma_k^2(1-\rho_{i,k})\eta_{i,k}^* + \nu_{i,k}^* \mathbb{1} = 0,$$

where last two conditions are obtained from the Lagrangian

$$\begin{aligned} L(\kappa_{i,k}, \eta_{i,k}, \upsilon_{i,k}, \nu_{i,k}) = {} & \kappa_{i,k}^\top \Theta_{i,k}\kappa_{i,k} + (1-\rho_{i,k})\sigma_k^2\eta_i^\top\eta_{i,k} \\ & + \upsilon_{i,k}((\kappa_{i,k})^\top\kappa_{i,k} - \psi_{i,k}) + \nu_{i,k}((\kappa_{i,k} + \eta_{i,k})^\top \mathbb{1} - 1). \end{aligned}$$

By combining these two KKT conditions with the second one, we obtain the optimal values

$$\kappa_{i,k}(t) = \frac{(1-\rho_{i,k})\sigma_k^2(\Theta_{i,k} + \upsilon_{i,k}I)^{-1}\mathbb{1}}{(1-\rho_{i,k})\sigma_k^2\mathbb{1}^\top(\Theta_{i,k} + \upsilon_{i,k}I)^{-1}\mathbb{1} + M_i^k},$$

$$\eta_{i,k}(t) = \frac{\mathbb{1}}{(1-\rho_{i,k})\sigma_k^2 \mathbb{1}^\top (\Theta_{i,k} + \upsilon_{i,k} I)^{-1} \mathbb{1} + M_i^k}.$$

The fourth KKT condition establishes that either $\upsilon_{i,k}^* = 0$ or $(\kappa_{i,k}^*)^\top \kappa_{i,k}^* = \psi_{i,k}$. Comparing the results with the previous ones (4.8), (4.9), it is possible to see that choosing the case $\upsilon_{i,k}^* = 0$, the optimal solutions are the same of the problem (4.7) (where there is no bias constraint), if the constraint $(\kappa_{i,k}^*)^\top \kappa_{i,k}^* \leq \psi_{i,k}$ holds. By observing that, in this case,

$$\begin{aligned}(\kappa_{i,k})^\top \kappa_{i,k} &= \frac{(1-\rho_{i,k})^2 \sigma_k^4 \mathbb{1}^\top (\Theta_{i,k})^{-2} \mathbb{1}}{((1-\rho_{i,k})\sigma_k^2 \mathbb{1}^\top (\Theta_{i,k})^{-1} \mathbb{1} + M_i^k)^2} \\ &\leq \frac{(1-\rho_{i,k})^2 \sigma_k^4 \left\| \Theta_{i,k}^{-1} \right\|^2}{M_i^k}\end{aligned} \tag{4.14}$$

and by choosing

$$\psi_{i,k} = \frac{(1-\rho_{i,k})^2 \sigma^4 \left\| \Theta_{i,k}^{-1} \right\|^2}{M_i^k},$$

the fourth KKT condition is satisfied. We can see that, in this way, it is possible to define an appropriate bound to the bias, by setting an appropriate value of the Pareto parameter $\rho_{i,k}$ and maintaining the results of Equations (4.8) and (4.9).

4.4 The estimation error

The estimates of the nodes converge to a neighborhood of the same value, since it is demonstrated in [86](section B) that the estimation error converges to a neighborhood of the origin. The estimation error can be affected also by bias term when dealing with decentralized scenarios such as sensor networks. Anyway, the size of the cumulative bias can be kept small with respect to the signal to track, following the procedure defined in Section 4.3. It is possible to analyze the entity of the estimation error. This can be useful in order to compute a bound for this quantity. Since, each node can compute at each step the values of mean $B_{i,k}$ (Eq.(4.4)) and variance $V_{i,k}$ (Eq.(4.5)) of the estimation error (in the next section we will see how to practically estimate these values), each sensor can derive a stochastic bound of its own estimation error. The estimation error is, in fact, a random variable. If we do not know the distribution of the random variable, we can use the Chebyshev inequalities in order to define some bounds for the uncertain value of the variable (without any assumption on the distribution):

$$\Pr(\mu_{e_i}^{(k)} - \alpha\sigma_{e_i}^{(k)} \leq e_i^{(k)} \leq \mu_{e_i}^{(k)} + \alpha\sigma_{e_i}^{(k)}) \geq 1 - \frac{1}{\alpha^2}, \tag{4.15}$$

where $\mu_{e_i}^{(k)} = B_{i,k}$ and $\sigma_{e_i}^{(k)} = \sqrt{V_{i,k}}$. It follows that:

- at least the 75% of the values are between $\mu - 2\sigma$ and $\mu + 2\sigma$;
- at least the 88% are between $\mu - 3\sigma$ and $\mu + 3\sigma$;
- at least the 93% are between $\mu - 4\sigma$ and $\mu + 4\sigma$;
- at least the 96% are between $\mu - 5\sigma$ and $\mu + 5\sigma$;
- at least the 99% are between $\mu - 10\sigma$ and $\mu + 10\sigma$.

It is possible to find better results if we assume to know the distribution of $w_i^{(k)}(t)$. As example, let us assume $w_i^{(k)}$ to be normally distributed. Then, it is possible to demonstrate that also $e_i^{(k)}(t)$ has a Gaussian distribution since it is a linear function of Gaussian stochastic variables. In the normal case, we can say therefore that the percentages become:

- $68,3\%$ with $\alpha = 1$;
- $95,5\%$ with $\alpha = 2$;
- $99,0\%$ with $\alpha = 2,58$;
- $99,7\%$ with $\alpha = 3$.

In this way it is possible to define a "α-tube" where we think to find the real value of the estimation error with a certain probability depending on the value of α:

$$\bar{e}_i^{(k)-} \leq e_i^{(k)}(t) \leq \bar{e}_i^{(k)+}(t)$$

with

$$\bar{e}_i^{(k)+}(t) = \mu_{e_i}^{(k)}(t) + \alpha\sigma_{e_i}^{(k)}(t) \tag{4.16}$$

$$\bar{e}_i^{(k)-}(t) = \mu_{e_i}^{(k)}(t) - \alpha\sigma_{e_i}^{(k)}(t) \tag{4.17}$$

being the time-varying upper and lower thresholds. Basing on these considerations, these stochastic bounds can be used as basis knowledge for the definition of an appropriate time-varying bound $\bar{e}_i^{(k)}$ for the distributed estimation error $e_i^{(k)} = \tilde{x}_i^{(k)} - x^{(k)}$, so that:

$$\left|e_i^{(k)}(t)\right| \leq \bar{e}_i^{(k)}(t), \ \forall\, t.$$

Concluding, we can say that it is convenient to use filtered estimates instead of measurements, because it is possible to define less conservative thresholds for fault detection purposes, as we will see in Chapter 6.

4.5 Estimator Structure

Now, we analyze how to implement the distributed estimator. Some of the estimates of the needed quantities are derived in [86]. In our case, we considered a signal that is component–wise quasi–stationary. Since the time-varying linear system in (4.1) is uniformly bounded-input bounded-output stable, then also $x(t)$ is quasi–stationary (see [86] for more details) and hence the mean $\mathbb{E}\varepsilon_{i,k} = \mu_{\varepsilon_{i,k}}(t)$ and the covariance matrix $\Gamma_{i,k}(t)$ can be estimated from the samples as follows:

$$\hat{\mu}_{\varepsilon_{i,k}}(t) = \frac{1}{t}\sum_{\tau=0}^{t}\hat{\varepsilon}_{i,k}(\tau) \tag{4.18}$$

$$\hat{\Gamma}_{i,k}(t) = \frac{1}{t}\sum_{\tau=0}^{t}(\hat{\varepsilon}_{i,k}(\tau) - \hat{\mu}_{\varepsilon_{i,k}}(\tau))(\hat{\varepsilon}_{i,k}(\tau) - \hat{\mu}_{\varepsilon_{i,k}}(\tau))^{\top}, \tag{4.19}$$

where $\hat{\varepsilon}_{i,k}(t)$ is an estimate of the error, that is obtained in [86], taking into account both estimates $\tilde{x}_i^{(k)}(t)$ and measurements $m_i^{(k)}(t)$, by solving a regularized linear least squares problem:

$$\min_{d^{(k)},\hat{\varepsilon}_{i,k}} \left\| \begin{pmatrix} \tilde{\mathbf{x}}_i^{(k)} \\ \mathbf{m}_i^{(k)} \end{pmatrix} - A \begin{pmatrix} d^{(k)} \\ \hat{\varepsilon}_{i,k} \end{pmatrix} \right\|^2 + \nu \left\| D \begin{pmatrix} d^{(k)} \\ \hat{\varepsilon}_{i,k} \end{pmatrix} \right\|^2, \tag{4.20}$$

where the vector $\tilde{\mathbf{x}}_i^{(k)}(t-1) := (\tilde{x}_{i_1}^{(k)}(t-1), \ldots, \tilde{x}_{i_{M_i^k}}^{(k)}(t-1))^T$, with $\{i_1, \ldots, i_{M_i^k}\} \in \mathcal{N}_i^k$, collects estimates, $\mathbf{m}_i^{(k)}(t) := (m_{i_1}^{(k)}(t), \ldots, m_{i_{M_i^k}}^{(k)}(t))^T$ collects neighboring measurements,

$$A = \begin{pmatrix} \mathbb{1} & I \\ \mathbb{1} & 0 \end{pmatrix} \in \mathbb{R}^{2M_i \times M_i^k + 1}, \quad D = \begin{pmatrix} 0 & I \end{pmatrix} \in \mathbb{R}^{M_i^k \times M_i^k + 1}$$

and $\nu > 0$ is a parameter, which can be chosen using the Generalized Cross-Validation method (for more details, see [86], where a sub-optimal result is presented):

$$\nu = \arg\min \frac{\left\| (A^{\top}A + \nu D^{\top}D)^{-1} A^{\top} (\tilde{\mathbf{x}}_i^{(k)\top}, \mathbf{m}_i^{(k)\top}) \right\|}{tr(A^{\top}A + \nu D^{\top}D)^{-1}}.$$

The solution is

$$(d^{(k)}, \hat{\varepsilon}_{i,k}^{\top}) = (\tilde{\mathbf{x}}_i^{(k)\top}, \mathbf{m}_i^{(k)\top}) A (A^{\top}A + \nu D^{\top}D)^{-1}, \tag{4.21}$$

where $\hat{\varepsilon}_{i,k}$ is an estimate of $\varepsilon_{i,k}(t)$ and $d^{(k)}$, estimate of $x^{(k)}$, can be used to estimate $\delta^{(k)}(t) = x^{(k)}(t) - x^{(k)}(t-1)$. Finally, we propose the following solution to estimate $\Lambda_{i,k}(t)$:

$$\hat{\Lambda}_{i,k}(t) = \frac{1}{t}\sum_{\tau=1}^{t}(\hat{\mu}_{\varepsilon_{i,k}}(\tau-1) - \hat{\delta}^{(k)}(\tau)\mathbb{1})(\hat{\mu}_{\varepsilon_{i,k}}(\tau-1) - \hat{\delta}^{(k)}(\tau)\mathbb{1})^{\top}. \tag{4.22}$$

4.5.1 Distributed Estimation Algorithm

In this subsection, the implementation of the proposed algorithm is addressed. Each node has to implement the estimator given by Algorithm 1. Notice that $\rho_{i,k}$ is computed locally by using the Nelder-Mead simplex algorithm [180] in Line 13. Once the parameter has been calculated, the optimal weights $\kappa_{i,k}(t)$ and $\eta_{i,k}(t)$ are computed (Lines 14 - 15) and the local estimate of the signal $\tilde{x}_i^{(k)}(t)$ can be obtained (Line 16). After that, the values of the estimates of $\hat{\Gamma}_{i,k}(t)$, $\hat{\Lambda}_{i,k}(t)$ and $\hat{\mu}_{\varepsilon_{i,k}}(t)$ can be updated using new signal estimates and measurements (Lines 17 - 24).

Algorithm 1 Estimation algorithm for node i

1. $t := 0$
2. $\hat{\mu}_{\varepsilon_{i,k}}(0) := 0$
3. $\hat{\Gamma}_{i,k}(0) := \sigma_k^2 I$
4. $\hat{\Lambda}_i(0) := \sigma_k^2 I$
5. Compute $\rho_{i,k}$
6. $\hat{\Theta}_{i,k}(t) := (1-\rho_{i,k})\hat{\Gamma}_{i,k}(t) + \rho_{i,k}\hat{\Lambda}_{i,k}(t)$
7. $\tilde{x}_i^{(k)}(0) := m_i^{(k)}(0)$
8. **while** forever **do**
9. $\quad M_i^k := \left|\mathcal{N}_i^k\right|$
10. $\quad t := t+1$
11. $\quad$ Collect estimates $\tilde{\mathbf{x}}_i^{(k)}(t-1) := (\tilde{x}_{i_1}^{(k)}(t-1), \ldots, \tilde{x}_{i_{M_i^k}}^{(k)}(t-1))^T$ where $\{i_1, \ldots, i_{M_i^k}\} \in \mathcal{N}_i^k$
12. $\quad$ Collect measurements $\mathbf{m}_i^{(k)}(t) := (m_{i_1}^{(k)}(t), \ldots, m_{i_{M_i^k}}^{(k)}(t))^T$ where $\{i_1, \ldots, i_{M_i^k}\} \in \mathcal{N}_i^k$
13. $\quad$ Compute $\rho_{i,k}$
14. $\quad \kappa_{i,k}(t) := \frac{(1-\rho_{i,k})\sigma_k^2\Theta_{i,k}^{-1}(t)\mathbb{1}}{(1-\rho_{i,k})\sigma_k^2\mathbb{1}^\top\Theta_{i,k}^{-1}(t)\mathbb{1}+M_i^k}$
15. $\quad \eta_{i,k}(t) := \frac{\mathbb{1}}{(1-\rho_{i,k})\sigma_k^2\mathbb{1}^\top\Theta_{i,k}^{-1}(t)\mathbb{1}+M_i^k}$
16. $\quad \tilde{x}_i^{(k)}(t) = \kappa_{i,k}(t)^\top\tilde{\mathbf{x}}_i^{(k)}(t-1) + \eta_{i,k}(t)^\top\mathbf{m}_i^{(k)}(t)$
17. $\quad \hat{\varepsilon}_{i,k} := \frac{\tilde{\mathbf{x}}_i^{(k)}}{1+\nu} - \frac{\nu\mathbb{1}^\top\tilde{\mathbf{x}}_i^{(k)}+(1+\nu)\mathbb{1}^\top\mathbf{m}_i^{(k)}}{M_i^k(1+2\nu)(1+\nu)}\mathbb{1}$
18. $\quad d^{(k)}(t) := \frac{\nu\tilde{\mathbf{x}}_i^{(k)\top}+(1+\nu)\mathbf{m}_i^{(k)\top}}{1+2\nu}\mathbb{1}$
19. $\quad \hat{\delta}^{(k)}(t) := d^{(k)}(t) - d^{(k)}(t-1)$
20. $\quad \hat{\mu}_{\varepsilon_{i,k}}(t) := \frac{t-1}{t}\hat{\mu}_{\varepsilon_{i,k}}(t-1) + \frac{1}{t}\hat{\varepsilon}_{i,k}(t)$
21. $\quad \hat{\Gamma}_{i,k}(t) := \frac{t-1}{t}\hat{\Gamma}_{i,k}(t-1) + \frac{1}{t}(\hat{\varepsilon}_{i,k}(t) - \hat{\mu}_{\varepsilon_{i,k}}(t))(\hat{\varepsilon}_{i,k}(t) - \hat{\mu}_{\varepsilon_{i,k}}(t))^\top$
22. $\quad \hat{\Lambda}_{i,k}(t) := \frac{t-1}{t}\hat{\Lambda}_{i,k}(t-1) + \frac{1}{t}(\hat{\mu}_{\varepsilon_{i,k}}(t-1) - \hat{\delta}^{(k)}(t)\mathbb{1})(\hat{\mu}_{\varepsilon_{i,k}}(t) - \hat{\delta}^{(k)}(t)\mathbb{1})^\top$
23. $\quad \hat{\Theta}_{i,k}(t) := (1-\rho_{i,k})\hat{\Gamma}_{i,k}(t) + \rho_{i,k}\hat{\Lambda}_{i,k}(t)$
24. **end while**

4.5.2 Computational complexity

The computational complexity of the proposed estimator E_p is given mainly by three components: the computational complexity of a matrix inverse, the one of a Nelder-Mead simplex algorithm, and that needed for the estimation of the covariance matrix. The computation of a matrix inverse is required to compute the optimal weights $\kappa_{i,k}(t)$ and $\eta_{i,k}(t)$. It has complexity $O(\left|\mathcal{N}_i^k\right|^2)$. The computation of a Nelder-Mead simplex algorithm is required to compute the optimal Pareto parameter $\rho_{i,k}$. This computation has complexity $O(\mathrm{N_{Iter}}\left|\mathcal{N}_\mathrm{i}^\mathrm{k}\right|^2)$, where $\mathrm{N_{Iter}}$ is the number of iterations. The computation of the covariance matrix is required to calculate the approximate estimates $\hat{\Gamma}_{i,k}(t)$ and $\hat{\Lambda}_{i,k}(t)$: the complexity is $O(\mathrm{Table_{size}}\log(\mathrm{Table_{size}}))$, where the $\mathrm{Table_{size}}$ is the size of a look-up table used to speed up the computation of the quadratically constrained least-square problem [86] given in (4.20).

4.6 Simulation results

In this section, numerical simulations are described to show the effectiveness of the proposed distributed estimator and to compare the performances with estimators available in the literature in the case of a single scalar signal. We consider the following approaches of the state of the art:

E_1: $L_k = 0$ and $H_k = [\eta_{k,i,j}]$ with $\eta_{k,i,j} = 1/M_i^k$ if node i and j communicate, and $\eta_{k,i,j} = 0$ otherwise, resulting in the average of the measurements.

E_2: $L_k = [l_{k,i,j}]$, where $l_{k,i,i} = 1/2M_i^k$, $l_{k,i,j} = 1/M_i^k$ if node i and j communicate, $l_{k,i,j} = 0$ otherwise, whereas $H_k = [\eta_{k,ikj}]$ with $\eta_{k,i,i} = 1/2M_i^k$, and $\eta_{k,i,j} = 0$ elsewhere. This results in the average of the old estimates and node's single measurement.

E_3: $L_k = H_k$ with $l_{k,i,j} = 1/2M_i^k$ if node i and j communicate, and $l_{k,i,j} = 0$ otherwise. This is the average of the old estimates and all local new measurements.

E_4: The estimator proposed in [86], which minimizes only the variance of the estimation error.

E_p: The estimator proposed in this chapter.

A 35-nodes network is obtained, by distributing the nodes randomly over a squared area of size $N_r/3$ and the graph is then obtained by letting two nodes communicate if their relative distance is less than $\sqrt{N_r}$. One example can be seen in Figure 4.1. Figure 4.2 shows the scalar signal that has to be tracked in simulations, that is used in [86] as a benchmark signal to

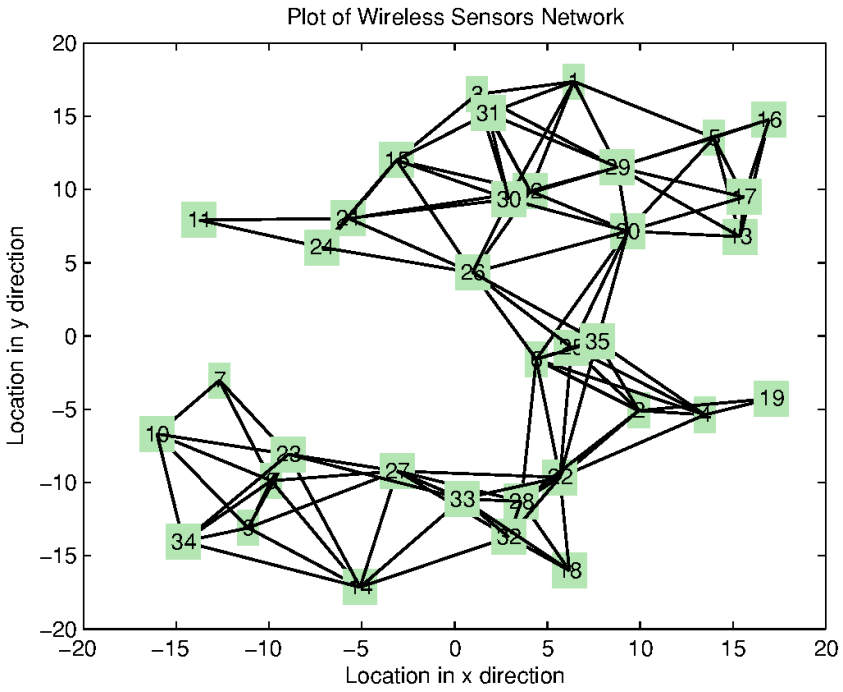

Figure 4.1: An example of the realized sensor networks.

compare estimators performances. The test signals used in simulation are highly non-linear. The noises in the measurements are generated randomly for each node: in the simulation each node has its own measurement noise.

Estimator	MSE
E_1	0.6082
E_2	0.0542
E_3	0.1771
E_4	0.0400
E_p	0.0317

Table 4.1: Simulations results: mean MSE over 50 simulations

In order to compare the performances obtained by the five estimators, we analyze the mean square error of the estimates of each node, averaging the mean square error over all nodes of the network and obtaining what we denote MSE. After that, we averaged the MSE values calculated from 50 different simulations, with different networks and different measurements

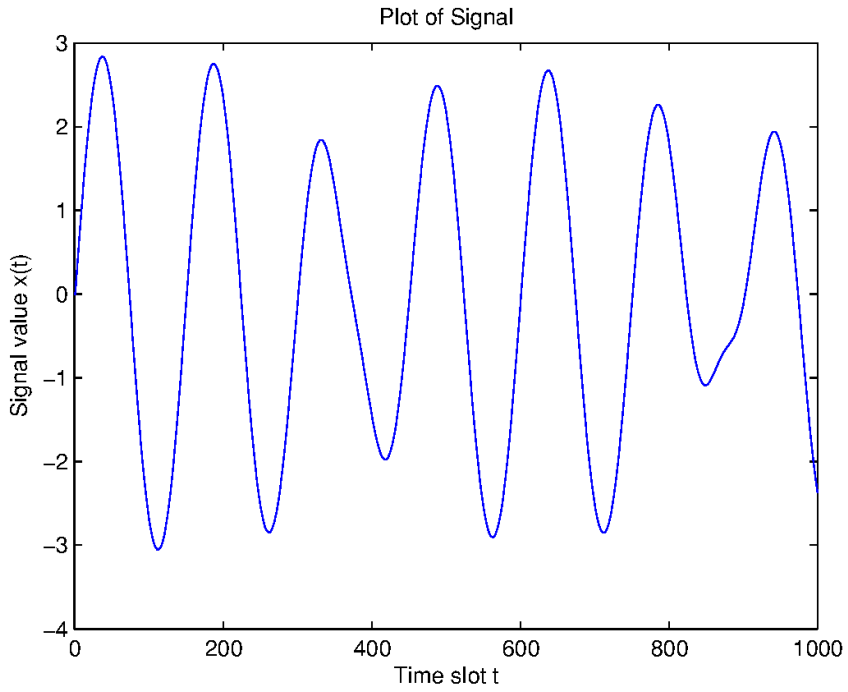

Figure 4.2: The signal to be tracked.

noises. The results are presented in Table 4.1.

Measurements and the resulting estimates for all nodes for each considered estimator are shown in Figure 4.3 for one experiment, but they are similar in all the tested cases, with different networks and different measurements noise. As we can see, all estimators are able to track the signal, but the quality of the estimates is different. The proposed algorithm performs much better than methods E_1, E_2 and E_3 and has results similar to the ones obtained with the minimum variance estimator of [86], presenting a lower averaged value of MSE. Moreover, one drawback of this last method E_4 is the computational cost, while the proposed Pareto approach has lower computational complexity, because E_4 requires in addition the distributed computation of a constraint that permits to guarantee the boundedness of the bias. As regards the computation complexity analysis, in the simulation, the number of iterations of the Nelder-Mead simplex algorithm $\mathrm{N_{Iter}}$ is typically less than 20 and we set $\mathrm{Table_{size}} = 100$. Furthermore, it is important to remark that the computational time required by the Pareto estimator is lower: the proposed approach saves about 20% time to track the signal in simulations run under the same conditions on a Intel i7 processor ($2.80GHz$, 4 Cores, 8 Logical Processors). As example, in the presented simulation case, the time needed for the estimate computation in average

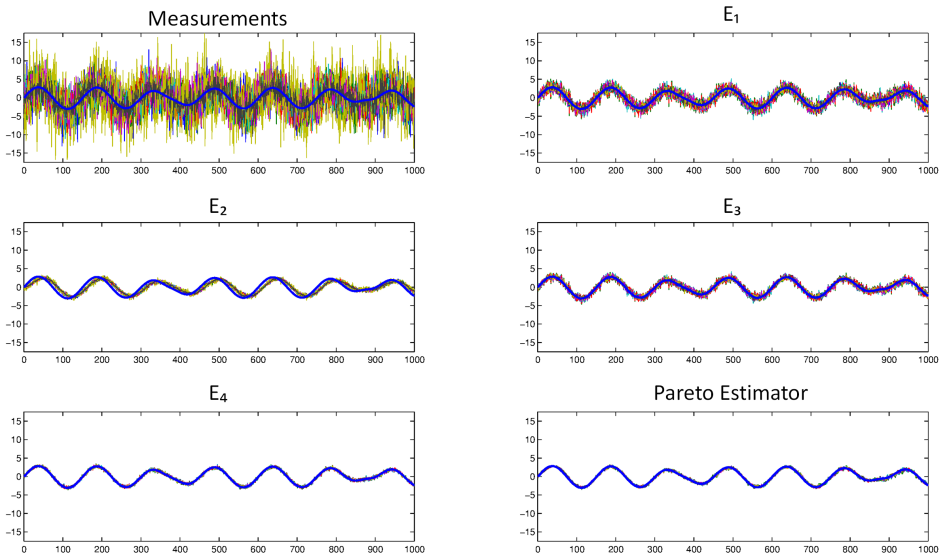

Figure 4.3: In the first graph, the signal to be tracked and the measurements realized by all the $N_r = 35$ nodes. In the following graphs, the estimates obtained by the different five estimators for each node. The tracked signal is represented by the thick blue curve; measurements and estimates have different colors for each node.

for each node is $0.0053s$ for the proposed estimator E_p and $0.0067s$ for estimator E_4. Concluding, also simulation results show the effectiveness of the proposed distributed estimation methodology, guaranteeing a lower computational cost than previous works having similar estimation quality.

4.7 Concluding remarks

In this Chapter the distributed estimation method implemented by each sensor network has been described. It permits to reduce measurements noise and to derive a stochastic bound on the estimation error. This bound and the filtered measurements will be transmitted to the diagnosers with the appropriate time stamp information, allowing to define less conservative detection thresholds. In [181], the proposed methodology has been used for fault detection in sensor networks.

Once the filtering task has been concluded, each sensor network will communicate its estimates and bounds to the diagnoser level. We assume that a distributed protocol is implemented in order to decide which is the sensor that has the transmission task. A not so reliable solution is that a fixed sensor is determined. This may cause some issues: it is not robust to

sensor faults; moreover if the sensor network topology changes, it has to be reconfigured. We propose another solution, consisting in the implementation of what we call a distributed "token protocol": the sensor that has the token communicates its variables to diagnosers. The token can be transmitted to a neighboring sensor if this sensor has lower estimation error bound than the current one. At each step, the sensor with the token receives the bounds of the neighboring nodes and compares these value with its current bound: the token goes to the sensor that has the lowest bound. Of course, other alternatives are possible, depending on the implementation issues and opportunities.

In the following chapters we will analyze the structure and the goals of the diagnosers level. More specifically, in Chapter 5 the re-synchronization method, implemented by the diagnosers to re-organize the filtered measurements received by possibly multi-rate and asynchronous sensor networks, will be presented.

Chapter 5

The diagnosers level

The Local Fault Diagnosers are designed for fault diagnosis purposes in order to monitor the physical system. We assume to have N LFDs, one for each subsystem. Sensor networks communicate filtered measurements $\tilde{x}^{(k)}$, with $k = 1, \ldots, n$, to some diagnosers (we suppose there exists a distributed protocol to determine which sensor of the network will communicate to diagnosers). Therefore, the diagnosers can see part of the physical system, what we call a subsystem or at least a part of it at a certain time instant. In fact, it is possible that some measurements are temporarily not received due to communication network problems or sensor failures.

We consider the decomposition of the system $\mathscr{S}$ into N subsystems $\mathscr{S}_I, I = 1, \ldots, N$. We refer to Chapter 3 for the analysis of the decomposition details. The overlapping of certain states $x^{(s)}$ is allowed, that is, certain states may belong to more than one subsystem: we will refer to them as shared variables. After the decomposition, the I-th subsystem $\mathscr{S}_I$ dynamics can be described as in Eq. (3.3):

$$\mathscr{S}_I : \dot{x}_I = f_I(x_I, u_I) + g_I(x_I, z_I, u_I) + \beta(t - T_0)\phi_I(x_I, z_I, u_I), \qquad (5.1)$$

where $x_I \in \mathbb{R}^{n_I}$ and $u_I \in \mathbb{R}^{p_I}$ are the local state and control input vectors, $z_I \in \mathbb{R}^{q_I}$ is the vector of the interconnection variables, which are state variables of neighboring subsystems that influence $\mathscr{S}_I$. The term $g_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ represents the interconnection function where the local effects of the modeling uncertainty function η have been incorporated, $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \mapsto \mathbb{R}^{n_I}$ represents the local nominal healthy dynamics.

Each diagnoser receives the estimated measurements from a part of the model, a subsystem, and knows the local model representing the dynamics of the measured/received variables. We suppose that each diagnoser knows the local model of the subsystem it is monitoring.

The diagnosers have mainly two tasks: first, they collect some filtered measurements from sensor networks and they re-organize these measurements in order to use them for the second goal: fault detection and isolation.

In this chapter we are considering the synchronization task. In the following we will analyze the proposed model-based re-synchronization procedure.

5.1 The synchronization task

Since sensor networks may not be synchronized and may have different rates, filtered measurements are received at different time instants. Moreover, since filtering and transmission are necessary, estimated measurements are received with a certain delay due to the computation time and the transmission issues. In fact, depending on the network protocol, delays and packet losses are possible. We assume that it is possible to know the time at which the measurement is actually taken. This can be achieved in different ways. The first case considers that the delay due to computation and transmission is known. A more realistic scenario assumes that sensors are able to add a Time Stamp to the transmitted measurement in order to indicate the age of information. The more general scenario considers transmission protocols for which it is possible to define the maximum allowable transmission interval (MATI) [182] and so it is possible to obtain a bound for the transmission delay. In this work we assume to have Time-Stamps. Two different problems are faced: i) the clock synchronization and ii) the re-synchronization procedure. They are addressed in the following.

5.1.1 Clock-synchronization

A first issue emerges when considering that diagnosers and sensors may have different clocks. We propose to adopt the IEEE 1588-2002 standard, officially entitled "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", describing a hierarchical master-slave architecture for clock distribution. Within each sensor network, a synchronization protocol is applied. We propose that each diagnoser is elected as a synchronization master for the sensor networks that communicate with it. Similarly, at the higher level, the diagnosers are synchronized thanks to the same master-slave approach. In this way, all the devices of the monitoring architecture can share the same clock and the use of Time Stamps can be valid. The hierarchical architecture provided by the standard allows to avoid any assumption on the topology of the sensor networks and of the diagnosers.

5.1.2 The re-synchronization procedure

The second issue depends on the fact that different sensor networks may work at different rates. Our proposed solution is that the local diagnosers implement a kind of "virtual sensor", which "synchronizes" the received

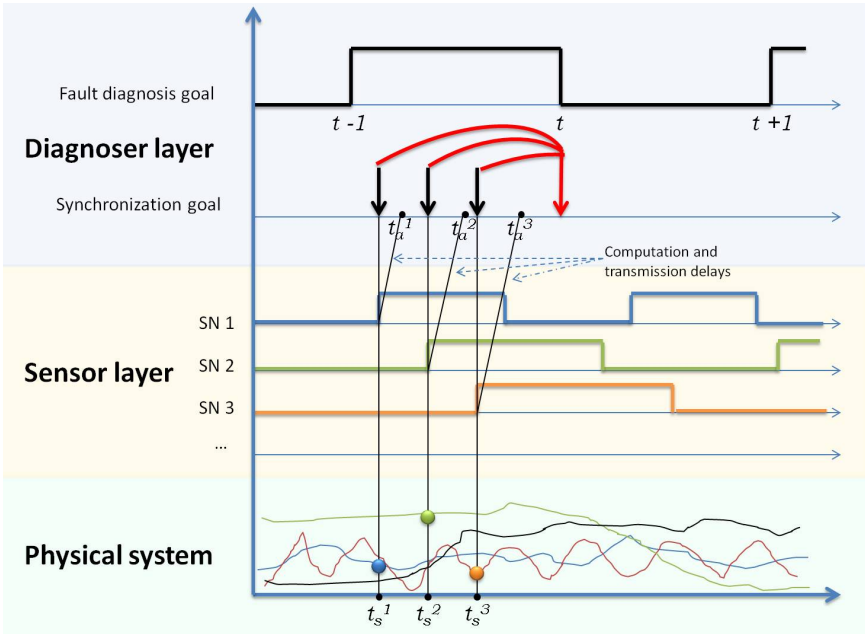

Figure 5.1: Synchronization procedure. We assume there is one diagnoser whose local model depends on three variables, which it receives from three different sensor networks. We plot the clock signals of each layer involved.

measurements. The procedure is shown in Figure 5.1, where an example is illustrated. For the convenience of the reader, we duplicate Figure 2.4 here. Let us then formalize the proposed re-synchronization procedure. Let us consider a state variable belonging to the I-th subsystem. Suppose that a measurement $m_I^{(k)}$ is taken at a certain time instant t_s^k in the continuous-time system. The relative sensor network filters this measurement and computes an estimate $\tilde{x}_I^{(k)}$ and finally, it sends the estimate to the achievable diagnosers. The I-th diagnoser receives the filtered measurement of component k at a certain time t_a^k in the continuous-time reference system, with $t_s^k < t_a^k < t$. Diagnosers work in a discrete-time framework with a sampling time that here we suppose unitary, without loss of generality. In the ideal case, diagnosers should receive all the local state measurements relative to the time instants t, $t+1$, $t+2, \ldots$. The diagnoser computes a projection of all the filtered measurements $\tilde{x}_I^{(k)}(t_s^k)$, with $k = 1, \ldots, n_I$, to the following time instant t, by integrating the local nominal model on the time interval $[t_s^k, t]$.

In healthy conditions, the local nominal model (Eq.(5.1)), in fact, can be rewritten as:

$$x_I^{(k)}(t) = x_I^{(k)}(t_s^k) + \int_{t_s^k}^{t} f_I^{(k)}(x_I, u_I, \tau) + g_I^{(k)}(x_I, z_I, u_I, \tau) d\tau. \tag{5.2}$$

Thus, the diagnoser can estimate therefore the value of the measurement at time t as:

$$\hat{m}_I^{(k)}(t) = \tilde{x}_I^{(k)}(t_s^k) + \int_{t_s^k}^{t} f_I^{(k)}(\hat{m}_I, u_I, \tau) + \hat{g}_I^{(k)}(\hat{m}_I, \hat{m}_{zI}, u_I, \tau)d\tau, \tag{5.3}$$

where $\hat{m}_{zI}$ are the projected measurements of the interconnection variables sent by neighboring diagnosers and $\hat{g}_I$ is the output of an adaptive approximator designed to learn the unknown interconnection function g_I as we will see in the following chapter.

Remark 1: Even if in Eq.(5.3), for analysis purposes, $\hat{g}_I$ represents the output of a continuous-time adaptive approximator, due to implementation issues, a suitable discrete-time approximator will be used, designed as explained in Section 6.2.1.

In this way, it is as if the diagnoser virtually can have all the measurements at the same time t. This comes at the cost of an increasing measurement uncertainty. In fact, the virtual measurement error $\xi_I^{(k)}(t) = \hat{m}_I^{(k)}(t) - x_I^{(k)}(t)$ can be computed as:

$$\begin{aligned} \xi_I^{(k)}(t) &= \tilde{x}_I^{(k)}(t_s^k) - x_I^{(k)}(t_s^k) + \int_{t_s^k}^{t} \Delta_{synch} f_I^{(k)}(\tau) + \Delta_{synch} g_I^{(k)}(\tau)d\tau \\ &= e_I^{(k)}(t_s^k) + \int_{t_s^k}^{t} \Delta_{synch} f_I^{(k)}(\tau) + \Delta_{synch} g_I^{(k)}(\tau)d\tau, \end{aligned} \tag{5.4}$$

where

$$\Delta_{synch} f_I^{(k)}(\tau) \triangleq f_I^{(k)}(\hat{m}_I, u_I, \tau) - f_I^{(k)}(x_I, u_I, \tau),$$
$$\Delta_{synch} g_I^{(k)}(\tau) \triangleq \hat{g}_I^{(k)}(\hat{m}_I, \hat{m}_{zI}, u_I, \tau) - g_I^{(k)}(x_I, z_I, u_I, \tau)$$

and it is possible to see that the virtual measurement uncertainty depends on sensors estimation error $e_I^{(k)}(t_s^k) = \tilde{x}_I^{(k)}(t_s^k) - x_I^{(k)}(t_s^k)$ and diagnoser synchronization error (the integral function). The older the measurements, the bigger the virtual measurement uncertainties.

In this way, it is possible to manage also packet losses in the first level communication network and temporary sensor failures.

At this point, we can collect the projected measurements $\hat{m}_I^{(k)}(t)$ in a vector, which we denote as $y_I(t)$ in the following. The diagnoser labels the projected measurements with a virtual Time Stamp indicating the time instant the variables are referred to. Therefore, it is as if the virtual sensor implemented by the diagnosers takes uncertain local measurements y_I of the state x_I, according to $y_I = x_I + \xi_I$, where ξ_I is the unknown virtual measurement error (Eq. (5.4)). Moreover, it follows that in place of the interconnection variables z, only the vector $v_I = z_I + \varsigma_I$ will be available for diagnosis, where ς_I is composed by the components of ξ_J affecting the

relevant components of y_J, with $J \in \mathscr{I}_I$, the set of the neighbors of $\mathscr{S}_I$.

The virtual measuring errors vectors ξ_I and ς_I are unstructured and unknown, but for each $k = 1, \ldots, n_I$ and $h = 1, \ldots, q_I$, it is possible to compute a bound for their components

$$\left|\xi_I^{(k)}(t)\right| \leq \bar{\xi}_I^{(k)}(t), \qquad \left|\varsigma_I^{(h)}(t)\right| \leq \bar{\varsigma}_I^{(h)}(t),$$

where

$$\bar{\xi}_I^{(k)}(t) = \bar{e}_I^{(k)}(t_s^k) + \int_{t_s^k}^{t} \bar{\Delta}_{synch} f_I^{(k)}(\tau) + \bar{\Delta}_{synch} g_I^{(k)}(\tau) d\tau \tag{5.5}$$

is a positive function, $\bar{e}_I^{(k)}$ is the stochastic bound derived in Section 4.4,

$$\bar{\Delta}_{synch} f_I^{(k)}(\tau) = \max_{x_I \in \mathcal{R}^{x_I}} \left| f_I^{(k)}(\hat{m}_I(\tau), u_I, \tau) - f_I^{(k)}(x_I, u_I, \tau) \right|$$

and $\bar{\Delta}_{synch} g_I^{(k)}(\tau)$ can be computed in an analogous way as in Eq.(6.17), Section 6.2.3, where all the details to bound this term will be explained.

Remark 2: As regards the bounding term $\bar{\Delta}_{synch} f_I$, less conservative thresholds may be obtained if it is possible to assume that there exist some time-varying bounding sets $\mathcal{R}^{x_I}(t) \subset \mathcal{R}^{x_I}$, so that it is possible to say that $x_I(t) \in \mathcal{R}^{x_I}(t)$. Therefore, the bound may be computed as:

$$\bar{\Delta}_{synch} f_I^{(k)}(\tau) = \max_{x_I \in \mathcal{R}^{x_I}(\tau)} \left| f_I^{(k)}(\hat{m}_I(\tau), u_I, \tau) - f_I^{(k)}(x_I, u_I, \tau) \right|.$$

5.2 Concluding remarks

In this short chapter, we have introduced the goals of the diagnosers layer and we have analyzed the synchronization topic. The diagnosers implement a kind of "virtual sensor" in order to re-organize the received filtered measurements, so that, at each step, all the projected variables (virtual measurements) refer to the same time instant. This procedure is needed in order to manage delays and packet losses in the first level communication network between sensors and diagnosers. Moreover, it permits to handle multi-rate and asynchronous systems. In the next Chapter we will go through the main topic of this work in depth: the distributed fault detection and isolation architecture.

Chapter 6

Fault diagnosis

In this Chapter we describe the distributed Fault Diagnosis architecture, implemented by the Local Fault Diagnosers. For diagnosis purposes, each LFD $\mathscr{L}_I$ will communicate with neighboring LFDs $\mathscr{L}_J$, with $J \in \mathcal{V}_I$ (see Definition 3.1.8). It is assumed that the inter-LFD communication will be carried over a packet-switched network, which we call the "second level communication network", subject to packet delays and losses. In order to manage network delays, the data-packets will be Time Stamped (with the "virtual" Time Stamp), so that they contain the information of the time instant the virtual measurements are referred to. Furthermore, we propose to provide each LFD with a buffer to collect the variables sent by neighbors. In the following, we will denote with a ^{b} the most recent value of a variable in the corresponding buffer of a given LFD: for instance v_I^b is the most recent value of the measured interconnection vector v_I contained in the buffer. In this layer we assume to have perfect clock synchronization between the diagnosers. Moreover, thanks to the re-synchronization method proposed in the previous chapter, all the virtual measurements, at each step, will refer to the same time instant.

6.1 Problem formulation

The continuous-time model of the physical system in Eq. 3.1 is discretized in order to obtain a discrete-time model representing the dynamics of the system that has to be monitored. The motivation of the discretization is that in this way it is easier to manage the different sampling rates at which the measurements are taken and the delays occurring in the communication network, and it is possible to design the proposed delay-compensation strategy. We suppose that, for the FDI task, the diagnosers know also a discrete-time model of the monitored subsystem. Let us consider the uncertain non-linear

discrete-time system:

$$\mathscr{S} : x(t+1) = f(x(t), u(t)) + \eta(x(t), u(t), t) + \beta(t - T_0)\phi(x(t), u(t)), \quad (6.1)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$ are the state and the control input of the system, $f : \mathbb{R}^n \times \mathbb{R}^p \mapsto \mathbb{R}^n$ represents the nominal healthy dynamics, η is the uncertainty function and finally ϕ is the unknown fault function, which is null in the case of healthy dynamics. The term $\beta(t - T_0)$ characterizes the time profile of a fault that occurs at some *unknown* discrete-time instant T_0, and $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})$ denotes the nonlinear fault function. The fault time profile $\beta(t - T_0)$ models both abrupt and incipient faults characterized by a decaying exponential time-profile

$$\beta(t - T_0) = \begin{cases} 0 & \text{if} \quad t < T_0 \\ 1 - c^{-(t-T_0)} & \text{if} \quad t \geq T_0 \end{cases}, \quad (6.2)$$

where $c \geq 1$ denotes the unknown fault-evolution rate (the case of an abrupt fault time-profile can be obtained as $c \to \infty$ in (6.2)). As noted for the continuous-time case, the fault time profile given by (6.2) only reflects the developing rate of the fault, while all its other basic features are captured by the function $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})$, which describes the changes in the dynamics due to the fault.

Assumption 6.1.1: The time profile parameter c is unknown but it is lower bounded by a known constant $\bar{c}$, that is $0 < \bar{c} \leq c$.

Assumption 6.1.2: At time $t = 0$ no faults act on the system and the state and control variables, x and u, remain bounded before and after the occurrence of a fault: $(x, u) \in \mathcal{R}$, with $\mathcal{R} = \mathcal{R}^x \times \mathcal{R}^u \subset \mathbb{R}^n \times \mathbb{R}^p$.

Here we consider only the fault detection and isolation problem, not the fault accommodation, therefore Assumption 6.1.2 is just required for well–posedness in order to guarantee that all the signals remain bounded.

We consider the decomposition of the system $\mathscr{S}$ into N subsystems $\mathscr{S}_I, I = 1, \ldots, N$. After the decomposition, the I-th subsystem $\mathscr{S}_I$ dynamics can be described by:

$$\begin{aligned} \mathscr{S}_I : x_I(t+1) = f_I(x_I(t), u_I(t)) + g_I(x_I(t), z_I(t), u_I(t)) \\ + \beta(t - T_0)\phi_I(x_I(t), z_I(t), u_I(t)), \end{aligned} \quad (6.3)$$

where $x_I \in \mathbb{R}^{n_I}$ and $u_I \in \mathbb{R}^{p_I}$ are the local state and control input vectors, $z_I \in \mathbb{R}^{q_I}$ is the vector of the interconnection variables, which are state variables of neighboring subsystems that influence $\mathscr{S}_I$. The term $g_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ represents the interconnection function where the local effects of the modeling uncertainty function η have been incorporated, $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \mapsto \mathbb{R}^{n_I}$ represents the local nominal healthy dynamics. The following assumption is needed:

Assumption 6.1.3: The structural graph and the decomposition are the same before and after the fault event.

In this way we suppose that the possible fault event does not cause a change to the system structure by adding new dependencies between variables belonging to different subsystems. However, it is possible for a fault event to remove some of the interconnections, which can be formally represented by setting some g_I function to zero.

Assumption 6.1.4: The interconnection function g_I is an uncertain nonlinear function, whose k-th component is bounded by some known positive bounded function, i.e., $\left|g_I^{(k)}(x_I, z_I, u_I))\right| \leq \bar{g}_I^{(k)}(x_I, z_I, u_I)$, for all $I = 1, \ldots, N$ and for all $(x, u) \in \mathcal{R}^x \times \mathcal{R}^u$.

Each Local Fault Diagnoser uses for diagnosis purposes the uncertain local filtered virtual measurements y_I of the state x_I, according to $y_I = x_I + \xi_I$, where ξ_I is the unknown virtual measuring error. It follows that in place of the interconnection variables z, only the vector $v_I = z_I + \varsigma_I$ will be available for diagnosis, where ς_I is composed by the components of ξ_J affecting the relevant components of y_J, with $J \in \mathcal{V}_I$, the set of the neighbors of $\mathscr{S}_I$. We assume u_I to be perfectly available.

Assumption 6.1.5: The virtual measuring error vectors ξ_I and ς_I are unstructured and unknown, but for each $k = 1, \ldots, n_I$ and $h = 1, \ldots, q_I$, their components are bounded by some known positive functions:

$$\left|\xi_I^{(k)}(t)\right| \leq \bar{\xi}_I^{(k)}(t), \qquad \left|\varsigma_I^{(h)}(t)\right| \leq \bar{\varsigma}_I^{(h)}(t).$$

In Section 5.1.2, it is explained how to compute these bounds basing on resynchronization error and the sensor networks' estimation error knowledge (Section 4.4).

6.2 Distributed Fault Detection Architecture

At $t = 0$, the DFDI algorithm is initialized turning on each I-th LFD, for which only its FDAE estimator is enabled and monitors the subsystem $\mathscr{S}_I$, providing a *local state estimate* $\hat{x}_{I,0}$ of the local state x_I. The difference between the estimate $\hat{x}_{I,0}$ and the virtual measurements y_I is the *estimation error* $\epsilon_{I,0}(t) \triangleq y_I(t) - \hat{x}_{I,0}(t)$ which plays the role of a residual and is compared component-wise with a suitable *detection threshold* $\bar{\epsilon}_{I,0}(t) \in \mathbb{R}_+^{n_I}$. The condition

$$|\epsilon_{I,0}^{(k)}(t)| \leq \bar{\epsilon}_{I,0}^{(k)}(t)\,, \quad \forall\, k = 1, \ldots, n_I \tag{6.4}$$

is associated with the *fault-free hypothesis*

$$\mathscr{H}_{I,0}: \text{ "The system } \mathscr{S}_I \text{ is healthy"}\,. \tag{6.5}$$

Should condition (6.4) be violated at some time instant t, then the hypothesis $\mathscr{H}_{I,0}$ is falsified and the so-called *local fault detection signature* $S_{I,0}$ is generated, leading to a local fault detection decision. The fault signature is defined as follows.

Definition 6.2.1: The *local detection signature* associated with the subsystem $\mathscr{S}_I$, $I \in \{1, \ldots, N\}$ at time $t > 0$ is the index set

$$S_{I,0}(t) \triangleq \{k \in \{1, \ldots, n_I\} : \exists\, t_1,\, t \geq t_1 > 0 \text{ such that } |\epsilon_{I,0}^{(k)}(t_1)| > \bar{\epsilon}_{I,0}^{(k)}(t_1)\}\,. \quad (6.6)$$

Definition 6.2.2: The *fundamental detection signature* associated with the system $\mathscr{S}$ at time $t > 0$ is the index set

$$S(t) \triangleq \{I \in \{1, \ldots, N\} :\, S_{I,0}(t) \neq \emptyset\}\,. \quad (6.7)$$

At this point, the local fault detection logic for the I–th LFD can be defined by relying on the local detection signature $S_{I,0}(t)$. Specifically, a fault affecting the I–th subsystem is detected by its LFD at the time instant $\bar{t}$ such that $S_{I,0}(\bar{t})$ becomes non-empty. This time instant is called *local fault detection time* $T_{I,d}$:

Definition 6.2.3: The *local fault detection time* $T_{I,d}$ is defined as $T_{I,d} \triangleq \min\{t : S_{I,0}(t) \neq \emptyset\}$.

Finally, the *fault detection time* T_d is simply defined as the earliest among the local detection times.

Definition 6.2.4: The *fault detection time* T_d is defined as $T_d \triangleq \min\{t : S(t) \neq \emptyset\}$.

The fault detection event observed by one (or more) LFD is immediately communicated to the global fault diagnoser $\mathscr{L}$. The GFD computes the fundamental detection signature S and sets T_d as the time at which it becomes non empty. Then, it immediately informs every LFD that a fault has occurred in the system and that the isolation mode should be activated.

Remark 3: The communication between the LFDs and the GFD required to implement the DFDI architecture is event-driven, that is only events such as the detection or isolation of a fault are communicated. As this kind of exchanged information can be limited to simple boolean values, scalability should not be an issue in practical applications.

6.2.1 Fault Detection and Approximation Estimator

For detection purposes, each LFD is equipped with a non-linear adaptive estimator, based on the local discrete-time nominal model.

In the case of non-shared state variables, the local FDAE estimation is designed as follows:

$$\hat{x}_I^{(k)}(t+1) = \lambda(\hat{x}_I^{(k)}(t) - y_I^{(k)}(t)) + f_I^{(k)}(y_I(t), u_I(t)) + \hat{g}_I^{(k)}(y_I(t), v_I^b(t), u_I(t), \hat{\vartheta}_I(t)) \quad (6.8)$$

where $0 < \lambda < 1$ and $\hat{g}_I$ is the output of an adaptive approximator designed to learn the unknown interconnection function g_I, $\hat{\vartheta}_I \in \hat{\Theta}_I$ denotes its adjustable parameters vector and being t_b the virtual Time Stamp of the most recent information received v_I^b. The adaptive approximator starts from the very beginning to learn the uncertain interconnection function in order to facilitate more accurate and faster detection. The following learning law can be obtained using Lyapunov stability methods[1], for every $I \in 1, \ldots, N$:

$$\hat{\vartheta}_I(t+1) = P_{\hat{\Theta}_I}\left[\hat{\vartheta}_I(t) + \gamma_I H_I^\top [\epsilon_I(t+1) - \lambda \epsilon_I(t)\right] \quad (6.9)$$

where $H_I^\top = \partial \hat{g}_I / \partial \hat{\vartheta}_I$ is the gradient matrix and $\gamma_I = \frac{\mu_I}{\varepsilon_I + \|H_I^\top\|_F^2}$, with $P_{\hat{\Theta}_I}$ being a projection operator restricting $\hat{\vartheta}_I$ within $\hat{\Theta}_I$ [183], $\|\cdot\|_F$ denotes the Frobenius norm and $\varepsilon_I > 0$, $0 < \mu_I < 2$ are design constants that guarantee the stability of the learning law [183]. It is worth noting that, to implement (6.8), the I-th LFD needs only to receive from its neighbors the values of the interconnection variables v_I. In the case of variables $x^{(s)} = x_I^{(s_I)} = x_J^{(s_J)}$, shared among more than one LFD, we take advantage of the redundancy obtained by means of the overlap: a deterministic consensus protocol is defined on a generic communication graph $\mathcal{G}_s \triangleq (\mathcal{O}_s, \mathcal{E}_s)$, whose nodes are the LFDs in the overlap set $\mathcal{O}_s$ of $x^{(s)}$, that is the set of the subsystems sharing s. The estimator can be computed as follows:

$$\hat{x}_I^{(s_I)}(t+1) = \lambda(\hat{x}_I^{(s_I)}(t) - y_I^{(s_I)}(t)) + \lambda \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left[\hat{x}_J^{b(s_J)}(t) - \hat{x}_I^{(s_I)}(t)\right] + \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left[f_J^{(s_J)}(y_J, u_J) + \hat{g}_J^{(s_J)}(y_J, v_J^b, u_J, \hat{\vartheta}_J)\right]^b, \quad (6.10)$$

where the terms $W_s^{(I,J)}$ are the components of a stochastic matrix W_s, where the values of each row add up to 1, weighting the terms of the subsystems sharing the variable $\mathbf{x^{(s)}}$ and reflecting the way the various LFDs cooperate

[1] The learning law can be derived from Equation (6.13) that will be presented in Section 6.2.3.

to estimate the shared variable. In Section 6.2.4 it is explained how to define the weight matrix W_s in order to improve detectability. Note that it is possible to extend the formulation of Eq. (6.10) to the case of a non-shared variable component k, since in this case $\mathcal{O}_k = \{I\}$ and so J is simply equivalent to I, and $W_k^{(I,I)} = 1$ by definition. In this way it is possible to give a single general formulation of the state estimator for every state component $k = 1, \ldots, n_I$:

$$\hat{x}_I^{(k)}(t+1) = \lambda(\hat{x}_I^{(k)}(t) - y_I^{(k)}(t)) + \lambda \sum_{J \in \mathcal{O}_k} W_k^{(I,J)} \left[\hat{x}_J^{b(k)}(t) - \hat{x}_I^{(k)}(t)\right] + \sum_{J \in \mathcal{O}_k} W_k^{(I,J)} \left[f_J^{(k)}(y_J, u_J) + \hat{g}_J^{(k)}(y_J, v_J^b, u_J, \hat{\vartheta}_J)\right]^b \quad (6.11)$$

Remark 4: It is important to note that, in (6.10), the I-th LFD does not need the information about the expressions of $f_J^{(s_J)}$ and $\hat{g}_J^{(s_J)}$: it is sufficient that each LFD computes locally the estimate $\hat{x}_J^{(s_J)}$ and the term $f_J^{(s_J)}(y_J, u_J) + \hat{g}_J^{(s_J)}(y_J, v_J^b, u_J, \hat{\vartheta}_J)$ and communicates it to other LFDs by means of the "second level communication network" according to the communication graph $\mathcal{G}_s$.

6.2.2 Delay Compensation Strategy in the second level communication network

In addition to the delays occurring in the first level communication network, which are compensated thanks to the re-synchronization mechanism, also the communication between diagnosers can be affected by delays and packet losses. In order to compute (6.11), each LFD must receive from its neighbors the terms $\hat{x}_J^{(s_J)}$, $f_J^{(s_J)} + \hat{g}_J^{(s_J)}$ and v_I, which are subject to network induced time-varying delays. In order to make the proposed FD scheme robust in this respect, we will employ a simple yet effective delay compensation strategy, first proposed in [184]. As in the approach used in [150], the time stamps of the data packets are considered in order to use only the most recent information received at the destination nodes: when a novel packet arrives, if it has a more recent virtual Time Stamp than the most recent already in the buffer, then it takes its place. In this way each LFD uses only the most recent measurements and information. Let us first clarify what we mean with "up-to-date" information. Let us assume that at a certain time instant t_c, with $t < t_c < t + 1$, a diagnoser has to compute its estimate for the time instant $t + 1$ and therefore it needs information referred to time t. A variable is up-to-date if its virtual Time Stamp indicates time t; otherwise, if the virtual Time Stamp is older, the variable is not up-to-date. Should a delay or a packet loss occur in the "second level communication network", we will proceed as follows:

- if some of the interconnection variables are not up-to-date, then the learning of the interconnection function g_I is paused.

- the summations in (6.11) will be carried on only with the terms received on time.

In order to allow the implementation of this second strategy, we adopt a time varying weighting matrix W_k, able to weight only the up-to-date terms. In Section 6.2.3 we will explain how to choose the weights in order to improve detectability. It is worth noting that with this approach, it is not necessary to know or to estimate the value of the delays. The only information needed is the age of data by means of the virtual Time Stamp. In the following, in Section 6.2.3, we will analyze the behavior of the Local Fault Diagnoser.

6.2.3 The detection threshold

In order to define an appropriate threshold for the detection of faults, we now analyze the dynamics of the FDAE estimation error when the system is healthy. Some considerations are necessary. By assumption, $\sum_{J\in\mathcal{O}_s} W_s^{(I,J)} = 1$. Moreover, due to the way the model decomposition is obtained, the following holds for shared variables, $\forall J \in \mathcal{O}_s$:

$$\begin{aligned}
&f_I^{(s_I)}(x_I, u_I) + g_I^{(s_I)}(x_I, z_I, u_I) \\
&= f_J^{(s_J)}(x_J, u_J) + g_J^{(s_J)}(x_J, z_J, u_J) \\
&= f^{(s)}(x, u) + \eta^{(s)}(x, u, t) \\
&= \sum_{J\in\mathcal{O}_s} W_s^{(I,J)}[f_J^{(s_J)}(x_J, u_J) + g_J^{(s_J)}(x_J, z_J, u_J)].
\end{aligned}$$

Besides, thanks to the fact that only up-to-date information is used in the consensus protocol due to the use of the time-varying consensus matrix (see Sections 6.2.2 and 6.2.4), it is possible to write:

$$\begin{aligned}
\sum_{J\in\mathcal{O}_s} W_s^{(I,J)}[f_J^{(s_J)}(x_J, u_J) + g_J^{(s_J)}(x_J, z_J, u_J)] \\
= \sum_{J\in\mathcal{O}_s} W_s^{(I,J)}[f_J^{(s_J)}(x_J, u_J) + g_J^{(s_J)}(x_J, z_J, u_J)]^b.
\end{aligned}$$

Therefore, basing on these considerations, it is possible to compute the k-th state estimation error component for the general form of Eq. (6.11) as:

$$\begin{aligned}\epsilon_I^{(k)}(t+1) &= y_I^{(k)}(t+1) - \hat{x}_I^{(k)}(t+1) \\ &= x_I^{(k)}(t+1) + \xi_I^{(k)}(t+1) - \hat{x}_I^{(k)}(t+1) \\ &= \sum_{J \in \mathcal{O}_k} W_k^{(I,J)} \left[\lambda \epsilon_J^{(k)} + \Delta f_J^{(k)} + \Delta g_J^{(k)} - \lambda \xi_J^{(k)}\right]^b + \lambda \xi_I^{(k)}(t) \\ &\quad + \xi_I^{(k)}(t+1),\end{aligned} \tag{6.12}$$

where $\Delta f_J^{(k)} \triangleq f_J^{(k)}(x_J, u_J) - f_J^{(k)}(y_J, u_J)$ and $\Delta g_J^{(k)} \triangleq g_J^{(k)}(x_J, z_J, u_J) - \hat{g}_J^{(k)}(y_J, v_J^b, u_J, \hat{\vartheta}_J)$.

Now we introduce a general formulation in vectorial form of the state error equation for analysis purposes. Specifically we define for every k-th state component the extended estimation error vector $\epsilon_{k,E}$, which is a column vector collecting the estimation error vectors of the N sub-systems sharing the k-th state component: $\epsilon_{k,E} \triangleq \text{col}\left(\epsilon_J^{(k)} : J \in \mathcal{O}_k\right)$. If k is non-shared, it simply collects one single element. The dynamics of $\epsilon_{k,E}$ can be described as:

$$\epsilon_{k,E}(t+1) = W_k \left[\lambda \epsilon_{k,E} + \Delta f_{k,E} + \Delta g_{k,E} - \lambda \xi_{k,E}\right]^b + \lambda \xi_{k,E}(t) + \xi_{k,E}(t+1), \tag{6.13}$$

where $\Delta f_{k,E}$ is a column vector, collecting the values $\Delta f_J^{(k_J)}$, for each $J \in \mathcal{O}_k$; $\Delta g_{k,E}(t)$ and $\xi_{k,E}$ are defined in an analogous way as $\Delta f_{k,E}(t)$.

In the following, we propose a threshold for the k-th component estimation error that guarantees no false-positive alarms. In the general form, by taking the absolute value component-wise, we can observe that:

$$\begin{aligned}|\epsilon_{k,E}(t+1)| \leq W_k[\lambda \left|\epsilon_{k,E}\right| + \left|\Delta f_{k,E}\right| + \left|\Delta g_{k,E}\right| + \left|\lambda \xi_{k,E}\right|]^b + \left|\lambda \xi_{k,E}(t)\right| \\ + \left|\xi_{k,E}(t+1)\right|\end{aligned}$$

Using the Comparison Lemma, the estimation error can be bounded by the threshold $\bar{\epsilon}_{k,E}$, that is defined as the solution of the following equation and that can be computed in a distributed way:

$$\begin{aligned}\bar{\epsilon}_{k,E}(t+1) = W_k \left[\lambda \bar{\epsilon}_{k,E} + \bar{\Delta} f_{k,E} + \bar{\Delta} g_{k,E} + \lambda \bar{\xi}_{k,E}\right]^b + \lambda \bar{\xi}_{k,E}(t) \\ + \bar{\xi}_{k,E}(t+1),\end{aligned} \tag{6.14}$$

where

$$\bar{\Delta} f_{k,E}(t) = \max_{\left|\xi^{(k)}\right| \leq \bar{\xi}^{(k)}} \left\{\left|\Delta f_{k,E}(t)\right|\right\}$$

and the equation can be initialized with

$$\bar{\epsilon}_{k,E}(0) = \mathrm{col}(\bar{\xi}_J^{(k)}(0), J \in \mathcal{O}_k).$$

Concerning $\bar{\Delta} g_{k,E}$, some considerations are necessary. The interconnection function error can be described as the sum of four different terms:

$$\Delta g_I = H_I \tilde{\vartheta}_I + \nu_I + \Delta \hat{g}_I + \Delta g_I^{\tau}. \tag{6.15}$$

The first term considers the error made because of the parameters estimation. This may be formalized by introducing an *optimal weight vector* [21] $\hat{\vartheta}_I^*$:

$$\hat{\vartheta}_I^* \triangleq \arg \min_{\hat{\vartheta}_I \in \Theta_I} \sup_{x_I, z_I, u_I} \left\| g_I(x_I, z_I, u_I) - \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I \right\|, \tag{6.16}$$

with x_I, z_I, u_I taking values in their respective domains, and by defining the parameter estimation error $\tilde{\vartheta}_I \triangleq \hat{\vartheta}_I^* - \hat{\vartheta}_I$. The second term is the *Minimum Functional Approximation Error* (MFAE) ν_I, which describes the least possible approximation error that can be obtained at time t if $\hat{\vartheta}_I$ were optimally chosen: $\nu_I(t) \triangleq g_I(x_I, z_I, u_I) - \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I^*)$. Then, a term that represents the error caused by the use of the uncertain measurements instead of the actual values of the state variables is defined: $\Delta \hat{g}_I \triangleq \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I) - \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_I)$. Finally, there is a term that takes into account the contribution to the estimation error due to the use of delayed measurements:

$$\Delta g_I^{\tau} \triangleq \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_I) - \hat{g}_I(y_I, v_I^b, u_I, \hat{\vartheta}_I),$$

where v_I is the current measured variable and v_I^b is the value in the buffer, which is old in the case of delays. This term is null when up-to-date measurements are used, that is $v_I^b = v_I$.

Now, the above terms are bounded as follows:

$$\begin{aligned} \bar{\Delta} g_I(t) \triangleq \|H_I\| \, \kappa_I(\hat{\vartheta}_I) + \bar{\nu}_I(t) + \max_{|\xi_I| \le \bar{\xi}_I(t)} \max_{|\varsigma_I| \le \bar{\varsigma}_I(t)} |\Delta \hat{g}_I(t)| \\ + \max_{v_I \in \mathcal{R}^v} \left| \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_I) - \hat{g}_I(y_I, v_I^b, u_I, \hat{\vartheta}_I) \right|, \end{aligned} \tag{6.17}$$

with the function κ_I being such that $\kappa_I(\hat{\vartheta}_I) \ge \left\| \tilde{\vartheta}_I \right\|$ and $\mathcal{R}^{v_I} \subset \mathbb{R}^{q_I}$, where this last term represents a local domain of the interconnection variable and is communicated by the neighboring LFDs at $t = 0$. It is important to remark that $\mathcal{R}^{v_I}$ coincides with the domain $\mathcal{R}^{x_J}$ for subsystem J, which is defined in Assumption 6.1.2.

Remark 5: The above bound of the term Δg_I^{τ} is conservative, but it is presented since it is general and requires a small amount of data to be

communicated. Other solutions are possible: as an example, let us consider the case where the dynamics of $v_I(t)$ can be assumed to have bounded first derivative, so that $|v_I(t+1) - v_I(t)| \leq K\,|(t+1) - t|$, with K being an a-priori known constant that is valid for a certain interval of time. The neighboring LFDs communicate the current value of K and its validity time to the I-th LFD, which computes the bound of Δg_I^τ as

$$\max_{v_I \in \mathcal{R}_K^v(t)} \left| \hat{g}_I(y_I, v_I(t), u_I, \hat{\vartheta}_I) - \hat{g}_I(y_I, v_I^b, u_I, \hat{\vartheta}_I) \right|,$$

where $\mathcal{R}_K^v(t) = [v_I^b(t) - K(t - t_b), v_I^b(t) + K(t - t_b)]$ is the current local domain, being t_b the virtual Time Stamp of the most recent information received in t: $v_I^b(t)$. When the current value of K is not valid, if a new value is not received, the LFD bounds the term considering $v_I \in \mathcal{R}^v$.

Remark 6: The terms $\bar{\xi}_I(t)$ and $\bar{\varsigma}_I(t)$ are computed by the diagnosers at each step after the re-synchronization task (see Eq. (5.5)) and are available in order to calculate the fault detection threshold.

The extended upper bound $\bar{\Delta} g_E(t)$ simply collects the upper bounds of the subsystems sharing the variable. The threshold in Eq. (6.14) guarantees that no false-positive alarms will be issued until T_0 because of the uncertainties[2]. In rough terms, this comes at the cost of the impossibility of detecting faults "hidden by the uncertainties in the system dynamics". This is formalized in Section 6.3, in which a sufficient condition for distributed detectability will be derived.

6.2.4 The novel consensus approach

It is worth noting that, since the threshold defined in (6.14), is a conservative threshold, it is important that it is as small as possible. The use of old values of the interconnection variables simply implies the addition of the positive term Δg_I^τ in the computation of the threshold, which increases the value of the threshold, deteriorating detectability skills. Therefore, in the case of shared variables, we propose the consensus-weighting matrix W_k to be time varying in order to minimize the adaptive threshold. In the consensus protocol, it is preferable to weight more the subsystem which has got the lowest threshold component, that is the subsystem that has lower uncertainty in its measurements and in the local model and that has the fewest delays and packet losses:

$$W_k^{I,J} = \begin{cases} 1 \text{ if } J = arg \min\limits_{J \in \mathcal{O}_k^b} (\lambda \bar{\epsilon}_J^{(k)} + \bar{\Delta} f_J^{(k)} + \bar{\Delta} g_J^{(k)} + \lambda \bar{\xi}_J^{(k)})^b \\ 0 \text{ otherwise,} \end{cases} \tag{6.18}$$

[2]This is true if the virtual measuring error bound assumption 6.1.5 holds

where $\mathcal{O}_k^b$ is the set of subsystems sharing k for which the I-th LFD has up-to-date information. This means that at each step each LFD uses only up-to-date information, received from only one LFD sharing the considered variable and this choice can change at each step. It is possible that neighboring LFDs sharing the same variable component k use different information for their threshold, since the threshold term $\lambda\bar{\epsilon}_J^{(k)} + \bar{\Delta}f_J^{(k)} + \bar{\Delta}g_J^{(k)} + \lambda\bar{\xi}_J^{(k)}$ depends on the reliability of the communication links, in conjunction with the confidence that each LFD has in its own measurements and estimates. In this way, moreover, we can manage time delays and packet losses: in fact, if the FDAE does not receive some consensus terms from some neighboring LFDs, it simply considers and weights only the up-to-date values. It is worth noting that this approach can be used in any case, with or without delays, and in Section 6.3 we will demonstrate that it improves detectability. In order to guarantee the convergence of the estimator, we demonstrate in the following proposition that the system described by Equation (6.13) is stable.

Proposition 6.2.1: Equation (6.13), with the consensus matrix defined in Equation (6.18), represents the dynamics of a exponentially stable discrete-time system.

Proof: Since W_k is a stochastic matrix, its norm is always equal to 1. Therefor, since $0 < \lambda < 1$, $\|\lambda W_k(t)\| \leq \gamma < 1$, with $0 < \gamma < 1$.

$$\begin{aligned} &\|\epsilon_{k,E}(t+1)\| = \|\lambda W_k(t)\epsilon_{k,E}(t)\| \\ &\leq \|\lambda W_k(t)\| \, \|\lambda W_k(t-1)\| \ldots \|\lambda W_k(0)\| \, \|\epsilon_{k,E}(0)\| \\ &\leq \gamma^t \, \|\epsilon_{k,E}(0)\| \end{aligned} \tag{6.19}$$

For $t \to \infty$, the series converges to zero. Moreover, in [185] it is proved that, given a system $x(t+1) = A(t)x(t)$, with $A(t) \in conv(A_1, \ldots, A_N)$, it is exponentially stable iff $\exists$ a sufficiently large integer k such that

$$\|A_{i_1} \cdot A_{i_2} \ \ldots \cdot A_{i_k}\| \leq \gamma < 1, \quad \forall (i_1, \ldots, i_k) \in \{1, \ldots, N\}^k \,,$$

where $conv(A_1, \ldots, A_N)$ is the convex matrix polyhedron of the set of constant matrices $\{A_1, \ldots, A_N\}$ and $\|\cdot\|$ is any vector induced matrix norm. In our case, therefore, we have to analyze matrix $W_k(t)$. Since each row of $W_k(t)$ has all null elements except one equal to 1, the product $W_k(t) \cdot W_k(t-1) \ \ldots \cdot W_k(0)$ is a stochastic matrix once again. So, being $0 < \lambda < 1$, then $\left\|\lambda^t (W_k(t) \cdot W_k(t-1) \ \ldots \cdot W_k(0))\right\| < 1$ and the hypothesis is satisfied. ■

The state estimation error solution can then be written as:

$$\begin{aligned} \epsilon_{k,E}(t) = \sum_{h=0}^{t-1} (\lambda W_k(h))^{t-1-h} \Big\{ W_k(h) \left[\Delta f_{k,E}(h) + \Delta g_{k,E}(h) - \lambda \xi_{k,E}(h)\right]^b \\ + \lambda \xi_{k,E}(h) + \xi_{k,E}(h+1) \Big\} + \prod_{h=0}^{t-1} (\lambda W_k(h)) \epsilon_{k,E}(0) \end{aligned} \tag{6.20}$$

It is important to note that, while previous results exist in the literature where the convergence of consensus algorithms with unreliable communication is proved by imposing conditions on the graph structure (see among many others the notable [186]), here thanks to the proposed choice of time varying weights, such assumptions are not needed.

6.2.5 The algorithm

Algorithm 2 Fault detection algorithm for the I-th LFD

$t = 1$
Learning = ON
Initialize the estimate $\hat{x}_I(1) = y_I(1)$
Initialize the threshold $\bar{\epsilon}_I(1) = \bar{\xi}_I(1)$
while A fault is not detected **do**
  Measurements $y_I(t)$ are acquired
  $\epsilon_I(t) = y_I(t) - \hat{x}_I(t)$
  Compare $|\epsilon_I(t)|$ with $\bar{\epsilon}_I(t)$
  if $|\epsilon_I(t)| > \bar{\epsilon}_I(t)$ **then**
    A fault is detected
    Learning = OFF
  end if
  Information from neighbors is acquired
  Update consensus weights (Eq. (6.18))
  if Some components k of v_I are not received **then**
    Learning = OFF
  else
    Learning = ON
    $v_I^{b(k)}(t) = v_I^{(k)}(t)$
  end if
  if Learning = ON **then**
    Update $\hat{\vartheta}_I$ (Eq. (6.9))
  else
    $\hat{\vartheta}_I(t) = \hat{\vartheta}_I(t-1)$
  end if
  Compute the novel estimate $\hat{x}_I(t+1)$ (Eq. (6.11))
  Compute the novel threshold $\bar{\epsilon}_I(t+1)$ (Eq. (6.14))
  $t = t + 1$
end while

Now we have all the elements to describe the proposed fault detection scheme, able to cope with delays and packet dropouts in the communication network between the local fault diagnosers. The implementation is explained

in Algorithm 2, that summarizes what is described in Subsections 6.2.2 and 6.2.4.

6.3 Detectability Conditions

We now consider the behavior of the fault detection scheme in the case of a faulty system. We assume that at an unknown time $t = T_0$ a fault ϕ occurs. Let's consider the general case of a variable shared among more than one subsystem, where $\phi_{k,E} = \phi^{(k)} \cdot (1, \ldots, 1)^\top$ denotes the extended fault function vector collecting the fault functions of the subsystems sharing the k-th variable. After the occurrence of the fault, for $t > T_0$, the state estimation error dynamics are

$$\epsilon_{k,E}(t+1) = W_k \left[\lambda\epsilon_{k,E}(t) + \Delta f_{k,E}(t) + \Delta g_{k,E}(t) - \lambda\xi_{k,E}(t)\right]^b + \lambda\xi_{k,E}(t) + \xi_{k,E}(t+1) + \phi_{k,E}(t) \quad (6.21)$$

In the following, we derive a sufficient condition for distributed fault detectability.

Theorem 6.3.1 (Fault Detectability): If there exists a time instant $t_1 > T_0$ such that the fault ϕ satisfies the inequality

$$\left|\sum_{h=T_0}^{t_1-1} \lambda^{t_1-1-h}\phi^{(k)}(h)\right| > 2\bar{\epsilon}_I^{(k)}(t_1) \quad (6.22)$$

for at least one component $k \in n_I$, then the fault will be detected at time t_1, that is $\left|\epsilon_I^{(k)}(t_1)\right| > \bar{\epsilon}_I^{(k)}(t_1)$.

Proof: At time instant $t_1 > T_0$, the estimation error can be described as:

$$\begin{aligned}\epsilon_{k,E}(t_1) = \sum_{h=0}^{t_1-1} (\lambda W_k(h))^{t_1-1-h} \Big[&W_k(h)\Delta f_{k,E}^b(h) + W_k(h)\Delta g_{k,E}^b(h) \\ &-\lambda W_k(h)\xi_{k,E}^b(h) + \lambda\xi_{k,E}(h) \\ &+\xi_{k,E}(h+1) + \phi_{k,E}(h)\Big] + \prod_{h=0}^{t_1-1}(\lambda W_k(h))\epsilon_{k,E}(0)\end{aligned}$$

that can be rewritten as:

$$\epsilon_{k,E}(t_1) = \sum_{h=0}^{t_1-1} (\lambda W_k(h))^{t_1-1-h} \Big[W_k(h)\Delta f^b_{k,E}(h) + W_k(h)\Delta g^b_{k,E}(h) \\ -\lambda W_k(h)\xi^b_{k,E}(h) + \lambda\xi_{k,E}(h) + \xi_{k,E}(h+1) \Big] \\ + \prod_{h=0}^{t_1-1} (\lambda W_k(h))\epsilon_{k,E}(0) + \sum_{h=T_0}^{t_1-1} [\lambda^{t_1-1-h}\phi_{k,E}(h)]$$

Using the triangle inequality we obtain:

$$|\epsilon_{k,E}(t_1)| \geq -\left| \sum_{h=0}^{t_1-1} (\lambda W_k(h))^{t_1-1-h} \Big[W_k(h)\Delta f^b_{k,E}(h) + W_k(h)\Delta g^b_{k,E}(h) \right. \\ \left. -\lambda W_k(h)\xi^b_{k,E}(h) + \lambda\xi_{k,E}(h) + \xi_{k,E}(h+1) \Big] + \prod_{h=0}^{t_1-1} (\lambda W_k(h))\epsilon_{k,E}(0) \right| \\ + \left| \sum_{h=T_0}^{t_1-1} [\lambda^{t_1-1-h}\phi_{k,E}(h)] \right|$$

Recalling how the threshold was defined in Equation (6.14), the following inequality is implied:

$$|\epsilon_{k,E}(t_1)| \geq -\bar{\epsilon}_{k,E}(t_1) + \left| \sum_{h=T_0}^{t_1-1} [\lambda^{t_1-1-h}\phi_{k,E}(h)] \right|.$$

Since $\phi_{k,E}$ is a vector whose components are all equal to $\phi^{(k)} = \phi_I^{(k_I)} = \phi_J^{(k_J)}$, it is easy to see that the fault detection condition $\left|\epsilon_I^{(k)}(t_1)\right| > \bar{\epsilon}_I^{(k)}(t_1)$ is implied by the hypothesis. ■

This theorem provides a sufficient condition for the implicit characterization of a class of faults that can be detected by the proposed fault detection scheme. Based on this result, in Eq. 6.22 it is easy to see that the lower the threshold is, the sooner the fault will be detected. Therefore the use of the proposed time-varying consensus weighting matrix, able to minimize threshold components in the case of shared variables, improves detectability. It is worth noting that this is true in general, also in the case without delays [184].

6.4 Distributed Fault isolation

In this section, we address the distributed fault isolation problem. We assume that the fault function $\boldsymbol{\phi}$ may either be unknown or belong to a known

global fault set $\mathcal{F}$:

$$\mathcal{F} \triangleq \{\boldsymbol{\phi}_{\mathbf{1}}(\mathbf{x},\mathbf{u}),\ldots,\boldsymbol{\phi}_{\mathbf{N}_{\mathcal{F}}}(\mathbf{x},\mathbf{u})\}\,.$$

In general, not all the subsystems are affected by a given fault function $\boldsymbol{\phi}_{\mathbf{l}}$, but only those in the corresponding *fault influence set* $\mathcal{U}_{\mathbf{l}}$. For each l–th fault, $\mathcal{U}_l$ contains the indexes of all the subsystems $\mathscr{S}_I$ that, after the decomposition $\mathcal{D}$, are assigned to at least one global state component $\mathbf{x}^{(s)}$ for which the fault function $\boldsymbol{\phi}_l$ is non–zero for at least one time instant. This is formalized in the following definition:

Definition 6.4.1: The *fault influence set* $\mathcal{U}_{\mathbf{l}}$ for the l–th fault function $\boldsymbol{\phi}_l$ is the index set

$$\mathcal{U}_{\mathbf{l}} \triangleq \{I:\ \exists t,\ \exists \mathbf{s},\ \mathbf{s}\in\mathcal{I}_I,\ \boldsymbol{\phi}_{\mathbf{l}}^{(\mathbf{s})}(\mathbf{x}(t),\mathbf{u}(t))\neq 0\}\,. \tag{6.23}$$

For each subsystem $\mathscr{S}_I$, a *local fault set* $\mathcal{F}_I$ (defined below) can be built with the local fault functions obtained by all the global faults $\boldsymbol{\phi}_{\mathbf{l}}$ such that $I\in\mathcal{U}_{\mathbf{l}}$:

$$\mathcal{F}_I \triangleq \{\phi_{I,1}(x_I,z_I,u_I),\ldots,\phi_{I,N_{\mathcal{F}_I}}(x_I,z_I,u_I)\}\,.$$

It is worth noting that the local fault functions depend only on the local variables x_I, z_I and u_I. The concept of the fault influence sets implies the subdivision of the faults into two categories, depending upon their topology: *local faults*, whose influence set is a singleton, and *distributed faults*, whose influence set includes more than one subsystem (see [29] for a detailed description). As in [187, 29], the generic I–th LFD knows only the local fault set $\mathcal{F}_I$ and has no information about the fault influence sets of the global faults corresponding to the local fault functions belonging to $\mathcal{F}_I$. Consequently, the I–th LFD may only be able to detect and isolate the *local* part of a fault that influences the subsystem $\mathscr{S}_I$, but it does not have enough information to understand whether the isolated local part corresponds to a local fault, or it just describes the local influence of a larger distributed fault. This ambiguity is overcome by the third layer of the DFDI architecture (see Fig. 2.5 and Section 6.4.3), consisting of the global fault diagnoser $\mathscr{L}$, which is assumed to know both the global fault set $\mathcal{F}$ and the fault influence sets of all the global fault functions. By using this knowledge and the local fault decisions d_I^{FD} obtained from all the lower-level LFDs, the GFD may be able to take a correct global fault decision d^{FD}: a successful global isolation of a fault by the GFD requires that all of the fault local parts have been *locally isolated* by the LFDs in its influence set.

6.4.1 Local fault isolation logic

After fault detection at time T_d, every LFD is told by the GFD to stop its FDAE and switch to isolation mode: the interconnection approximator stops updating its parameters vector, in order to stop learning also the influence of the fault function: $\hat{\vartheta}_{I,0}(t) = \hat{\vartheta}_{I,0}(T_d)\,, \forall\, t \geq T_d$. The isolation task is carried on by relying on a bank of $N_{\mathcal{F}_I}$ Fault Isolation Estimators (FIEs), with $I = 1, \ldots, N$, in order to implement a GOS scheme such as the one described in [22]. This scheme relies on the generic l–th FIE of the I–th LFD being matched to the corresponding fault function $\phi_{I,l}$, belonging to the local fault set $\mathcal{F}_I$. Each fault function in $\mathcal{F}_I$ is of the form

$$\phi_{I,l}(x_I, z_I, u_I) = [(\vartheta_{I,l,1})^\top H_{I,l,1}(x_I, z_I, u_I), \ldots, (\vartheta_{I,l,n_I})^\top H_{I,l,n_I}(x_I, z_I, u_I)]^\top, \tag{6.24}$$

where, for $k \in \{1, \ldots, n_I\}$, $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, the *known* functions $H_{I,l,k} : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{q_{I,l,k}}$ provide the functional structure of the fault and the *unknown* parameter vectors $\vartheta_{I,l,k} \in \Theta_{I,l,k} \subset \mathbb{R}^{q_{I,l,k}}$ provide its "magnitude". The parameter domains $\Theta_{I,l,k}$ are again assumed to be origin–centered hyper–spheres with radius $M_{\Theta_{I,l,k}}$.

The generic l–th FIE estimator, with $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, monitors its subsystem $\mathscr{S}_I$, computing a *local state estimate* $\hat{x}_{I,l}$ of the local state x_I, analogously to the FDAE. The difference between the estimate $\hat{x}_{I,l}$ and the measurements y_I produces the *isolation residual* $\epsilon_{I,l} \triangleq y_I - \hat{x}_{I,l}$ which, again, is compared, component by component, to a suitable *isolation threshold* $\bar{\epsilon}_{I,l} \in \mathbb{R}^{n_I}_+$. The condition

$$|\epsilon^{(k)}_{I,l}(t)| \leq \bar{\epsilon}^{(k)}_{I,l}(t)\,,\ k = 1,\ \ldots,\ n_I \tag{6.25}$$

is associated with the l–th fault hypothesis

$$\mathscr{H}_{I,l}:\ \text{"The subsystem } \mathscr{S}_I \text{ is affected by the } l\text{–th fault"}\,, \tag{6.26}$$

where $l = 1, \ldots, N_{\mathcal{F}_I}$. If the condition (6.25) is violated at some time instant t, then the hypothesis $\mathscr{H}_{I,l}$ is falsified and a so–called *local fault isolation signature* $S_{I,l}$ is generated.

Definition 6.4.2: The l–th *local isolation signature* related to the subsystem $\mathscr{S}_I$, $I \in \{1, \ldots, N\}$, $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$ at time $t > 0$ is the index set

$$S_{I,l}(t) \triangleq \{k \in \{1, \ldots, n_I\} : \exists\, t_1,\, t \geq t_1 > 0 \text{ such that } |\epsilon^{(k)}_{I,l}(t_1)| > \bar{\epsilon}^{(k)}_{I,l}(t_1)\}\,. \tag{6.27}$$

As soon as the hypothesis $\mathscr{H}_{I,l}$ is falsified and the corresponding isolation

signature $S_{I,l}(t)$ becomes non-empty, the specific FIE stops its operation and the fault $\phi_{I,l}(t)$ is excluded as a possible cause of the non-empty detection signature. This time instant is called the *exclusion time* $T_{e,I,l}$.

Definition 6.4.3: The *l–th fault exclusion time* $T_{e,I,l}$ is defined as $T_{e,I,l} \triangleq \min\{t : S_{I,l}(t) \neq \emptyset\}$.

Ideally, the goal of the isolation logic is to exclude every but one fault, which may be said to be *isolated*. This is expressed formally in the following further definition.

Definition 6.4.4: A fault $\phi_{I,q} \in \mathcal{F}_I$ is *locally isolated* at time t iff $\forall l$, $l \in \{1, \ldots, N_{\mathcal{F}_I}\} \setminus \{q\}$, $S_{I,l}(t) \neq \emptyset$ and $S_{I,q}(t) = \emptyset$. Furthermore $T_{locisol,I,q} \triangleq \max\{T_{e,I,l},\, l \in \{1, \ldots, N_{\mathcal{F}_I}\} \setminus \{q\}\}$ is the *local fault isolation time.*

Remark 7: It is worth noting that, if a fault has been locally isolated, we can conclude that it actually occurred if we assume that only faults belonging to the set $\mathcal{F}_I$ may occur. Otherwise, we can only say that it cannot be excluded that it occurred.

6.4.2 Local fault isolation and Fault Isolation Estimators

Now the FIEs are described in detail. After the fault $\phi(t)$ has occurred, the state equation of the s_I–th component of the I–th subsystem becomes

$$x_I^{(s_I)}(t+1) = f_I^{(s_I)}(x_I(t), u_I(t)) + g_I^{(s_I)}(x_I(t), z_I(t), u_I(t)) + \beta(t - T_0)\phi^{(s)}(x(t), u(t))\,.$$

The l–th FIE estimator dynamic equation for a shared variable is defined as

$$\begin{aligned}\hat{x}_{I,l}^{(s_I)}(t+1) = \lambda\{\hat{x}_{I,l}^{(s_I)}(t) - y_I^{(s_I)}(t) + \sum_{J \in \mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[\hat{x}_{J,l}^{(s_J)b}(t) - \hat{x}_{I,l}^{(s_I)}(t)]\} \\ + \sum_{J \in \mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[f_J^{(s_J)}(y_J(t), u_J(t)) + \hat{g}_J^{(s_J)}(y_J(t), v_J^b(t), u_J(t), \hat{\vartheta}_J(T_d)) \\ + \hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J^b(t), u_J(t), \hat{\vartheta}_{J,l})]^b\,, \quad (6.28)\end{aligned}$$

where $\hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J^b(t), u_J(t), \hat{\vartheta}_{J,l}) \triangleq (\hat{\vartheta}_{J,l,s_J})^\top H_{J,l,s_J}(y_J(t), v_J(t), u_J(t))$ is the s_J–th component of a linearly-parameterized function that matches the structure of the l–th fault function $\phi_{J,l}$, and the vector $\hat{\vartheta}_{J,l} \triangleq \mathrm{col}(\hat{\vartheta}_{J,l,k}, k \in \{1, \ldots, n_I\})$ has been introduced.

Analogously to the FDAE case, the parameters vectors are updated according to the learning law:

$$\hat{\vartheta}_{J,l,k}(t+1) = \mathcal{P}_{\hat{\Theta}_{J,l,k}}(\hat{\vartheta}_{J,l,k}(t) + \gamma_{J,l,k}(t) H_{J,l,k}^\top(t) r_{J,l,k}(t+1))\,,$$

where

$$r_{J,l,k}(t+1) = \epsilon_{J,l,k}(t+1) - \lambda\epsilon_{J,l,k}(t),$$

and $\mathcal{P}_{\hat{\Theta}_{J,l,k}}$ is the projection operator [183]

$$\mathcal{P}_{\hat{\Theta}_{J,l,k}}(\hat{\vartheta}_{J,l,k}) \triangleq \begin{cases} \hat{\vartheta}_{J,l,k} & \text{if } |\hat{\vartheta}_{J,l,k}| \leq M_{\hat{\Theta}_{J,l,k}} \\ \frac{M_{\hat{\Theta}_{J,l,k}}}{|\hat{\vartheta}_{J,l,k}|}\hat{\vartheta}_{J,l,k} & \text{if } |\hat{\vartheta}_{J,l,k}| > M_{\hat{\Theta}_{J,l,k}} \end{cases},$$

The learning rate $\gamma_{J,l,k}(t)$ is computed at each step as

$$\gamma_{J,l,k}(t) \triangleq \frac{\mu_{J,l,k}}{\varepsilon_{J,l,k} + \|H_{J,l,k}^{\top}(t)\|^2}, \;\; \varepsilon_{J,l,k} > 0, \;\; 0 < \mu_{J,l,k} < 2\,.$$

Remembering that, thanks to the introduction of the time-varying consensus matrix (Section 6.2.4), it is possible to write

$$\sum_{J\in\mathcal{O}_s} W_s^{(I,J)}[f_J^{(s_J)}(x_J,u_J) + g_J^{(s_J)}(x_J,z_J,u_J)] \\ = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)}[f_J^{(s_J)}(x_J,u_J) + g_J^{(s_J)}(x_J,z_J,u_J)]^b$$

since only up-to-date information is used, the corresponding estimation error dynamic equation can be computed as follows

$$\epsilon_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[\lambda\epsilon_{J,l}^{(s_J)}(t) + \chi_J^{(s_J)}(t) \\ + (1-c^{-(t-T_0)})\phi^{(s)}(t) - \hat{\phi}_{J,l}^{(s_J)}(t)]^b + \lambda\xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,.$$

where

$$\chi_J^{(s_J)}(t) = \Delta f_J^{(s_J)}(t) + \Delta g_J^{(s_J)}(t) - \lambda\xi_J^{(s_J)}(t).$$

Let us then consider a matched fault, that is,

$$\phi^{(s)}(t) = \phi_{J,l}^{(s_J)}(x_J(t), z_J(t), u_J(t), \vartheta_{J,l}),$$

$\forall\, J \in \mathcal{O}_{\boldsymbol{s}}$. The error equation can be written as

$$\epsilon_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[\lambda\epsilon_{J,l}^{(s_J)}(t) + \chi_J^{(s_J)}(t) - H_{J,l,s_J}(t)^{\top}\hat{\vartheta}_{J,l,s_J} \\ +(1-c^{-(t-T_0)})(H_{J,l,s_J}(t)^{\top}\vartheta_{J,l,s_J} + \Delta H_{J,l,s_J}^{\top}\vartheta_{J,l,s_J})]^b + \lambda\xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,,$$

where $\Delta H_{J,l,s_J}^{\top}(t) \triangleq H_{J,l,s_J}(x_J(t), z_J(t), u_J(t)) - H_{J,l,s_J}(y_J(t), v_J^b(t), u_J(t))$ is defined.

By introducing the parameter estimation errors $\tilde{\vartheta}_{J,l,s_J} \triangleq \vartheta_{J,l,s_J} - \hat{\vartheta}_{J,l,s_J}$,

the FIE estimation error equation for a matched fault becomes

$$\epsilon_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[\lambda\epsilon_{J,l}^{(s_J)}(t)+\chi_J^{(s_J)}(t)+(1-c^{-(t-T_0)})H_{J,l,s_J}(t)^\top\tilde{\vartheta}_{J,l,s_J} + (1-c^{-(t-T_0)})\Delta H_{J,l,s_J}(t)^\top\vartheta_{J,l,s_J} - c^{-(t-T_0)}H_{J,l,s_J}(t)^\top\hat{\vartheta}_{J,l,s_J}]^b + \lambda\xi_I^{(s_I)}(t)+\xi_I^{(s_I)}(t+1)\,,$$

so that its absolute value can be bounded by a threshold that is solution of the following equation

$$\bar{\epsilon}_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[\lambda\bar{\epsilon}_{J,l}^{(s_J)}(t)+\bar{\chi}_J^{(s_J)}(t)+\|H_{J,l,s_J}(t)\|\kappa_{J,l,s_J}(\hat{\vartheta}_{J,l,s_J}) + \bar{\Delta}H_{J,l,s_J}(t)\bar{\vartheta}_{J,l,s_J} - \bar{c}^{-(t-T_d)}\|H_{J,l,s_J}(t)\|\,\|\hat{\vartheta}_{J,l,s_J}\|]^b + \lambda\bar{\xi}_I^{(s_I)}(t)+\bar{\xi}_I^{(s_I)}(t+1)\,,$$

where

$$\bar{\chi}_J^{(s_J)}(t) = \bar{\Delta}f_J^{(s_J)}(t)+\bar{\Delta}g_J^{(s_J)}(t)+\lambda\bar{\xi}_J^{(s_J)}(t).$$

As regards

$$\bar{\Delta}H_{J,l,s_J}^\top(t) = \max_{\xi_J}\max_{\varsigma_J,v_J\in\mathcal{R}^v}\left|H_{J,l,s_J}(x_J(t),z_J(t),u_J(t))-H_{J,l,s_J}(y_J(t),v_J^b(t),u_J(t))\right|,$$

some considerations can be done analogously as for Eq.(6.17) and $\bar{\vartheta}_{J,l,s_J}$ can be computed remembering that the parameter domains $\Theta_{I,l,k}$ are assumed to be origin–centered hyper–spheres with radius $M_{\Theta_{I,l,k}}$. As in Subsection 6.2.3, the error and threshold solutions can be conveniently expressed in vector form $\epsilon_{s,l}(t) \triangleq \mathrm{col}(\epsilon_{I,l}^{(s_I)}, I\in\mathcal{O}_{\boldsymbol{s}})$, $\bar{\epsilon}_{s,l}(t) \triangleq \mathrm{col}(\bar{\epsilon}_{I,l}^{(s_I)}, I\in\mathcal{O}_{\boldsymbol{s}})$, so that it holds

$$\epsilon_{s,l}(t+1) = W_{\boldsymbol{s}}[\lambda\epsilon_{s,l}(t)+\chi_{\boldsymbol{s}}(t)+\Delta\phi_s(t)]^b+\lambda\xi_{\boldsymbol{s}}(t)+\xi_{\boldsymbol{s}}(t+1)\,,$$

where

$$\Delta\phi_s(t) = \mathrm{col}((1-c^{-(t-T_0)})H_{I,l,s_I}(t)^\top\tilde{\vartheta}_{I,l,s_I}+(1-c^{-(t-T_0)})\Delta H_{I,l,s_I}(t)^\top\vartheta_{I,l,s_I} - c^{-(t-T_0)}H_{I,l,s_I}(t)^\top\hat{\vartheta}_{I,l,s_I}, I\in\mathcal{O}_{\boldsymbol{s}}). \quad (6.29)$$

Then, it becomes:

$$\epsilon_{s,l}(t) = \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h} W_{\boldsymbol{s}}[\chi_{\boldsymbol{s}}(h) + \Delta\phi_s(t)]^b + \sum_{h=T_d}^{t-1} [(\lambda W_{\boldsymbol{s}})^{t-1-h}(\lambda\xi_{\boldsymbol{s}}(h) + \xi_{\boldsymbol{s}}(h+1))] + (\lambda W_{\boldsymbol{s}})^{t-T_d}\epsilon_{s,l}(T_d)\,.$$

Componentwise, the estimation error is given by

$$\epsilon_{I,l}^{(s_I)}(t) = w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h}[\chi_{\boldsymbol{s}}(h) + \Delta\phi_s(h)]^b + \lambda w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-2} [(\lambda W_{\boldsymbol{s}})^{t-2-h}(\lambda\xi_{\boldsymbol{s}}(h) + \xi_{\boldsymbol{s}}(h+1))] + \lambda\xi_I^{(s_I)}(t-1) + \xi_I^{(s_I)}(t) + \lambda w_{\boldsymbol{s},I}(\lambda W_{\boldsymbol{s}})^{t-1-T_d}\epsilon_{s,l}(T_d)\,,$$

and, analogously, the threshold solution is given by

$$\bar{\epsilon}_{I,l}^{(s_I)}(t) = w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h}[\bar{\chi}_{\boldsymbol{s}}(t) + \bar{\Delta}\phi_s(t)]^b + \lambda w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-2} [(\lambda W_{\boldsymbol{s}})^{t-2-h}(\lambda\bar{\xi}_{\boldsymbol{s}}(t) + \bar{\xi}_{\boldsymbol{s}}(t+1))] + \lambda\bar{\xi}_I^{(s_I)}(t-1) + \bar{\xi}_I^{(s_I)}(t) + \lambda w_{\boldsymbol{s},I}(\lambda W_{\boldsymbol{s}})^{t-1-T_d}\bar{\epsilon}_{s,l}(T_d)$$

where

$$\bar{\Delta}\phi_s(t) = \mathrm{col}(\|H_{I,l,s_I}(t)\|\kappa_{I,l,s_I}(\hat{\vartheta}_{I,l,s_I}) + \bar{\Delta}H_{I,l,s_I}(t)\bar{\vartheta}_{I,l,s_I} - \bar{c}^{-(t-T_d)}\|H_{I,l,s_I}(t)\|\,\|\hat{\vartheta}_{I,l,s_I}, I \in \mathcal{O}_{\boldsymbol{s}}\|).$$

This threshold guarantees by definition that no matched fault will be excluded because of uncertainties or the effect of the parameter estimation error $\tilde{\vartheta}_{I,l,s_I}$. In the case of a non–matched fault (that is, $\phi_I^{(s_I)}(x_I(t), z_I(t), u_I(t)) = \phi_{I,q}^{(s_I)}(x_I(t), z_I(t), u_I(t), \vartheta_{I,q})$ for some $I \in \mathcal{O}_{\boldsymbol{s}}$ and with $q \neq l$), the dynamics of the s_I–component of the estimation error of the l–th FIE of the I–th LFD

can be written as

$$\begin{aligned}\epsilon_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_{\boldsymbol{s}}} W_{\boldsymbol{s}}^{(I,J)}[\lambda\epsilon_{J,l}^{(s_J)}(t) + \chi_J^{(s_J)}(t) \\ + (1-c^{-(t-T_0)})\phi_{I,q}^{(s_I)}(x_I(t), z_I(t), u_I(t), \vartheta_{I,q}) \\ - \hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J^b(t), u_J(t), \hat{\vartheta}_{J,l})]^b + \lambda\xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,.\end{aligned}$$

As shown before, a convenient way to study the behavior of the estimation error of the LFDs sharing the variable $\boldsymbol{x}^{(\boldsymbol{s})}$ is to consider the vector $\epsilon_{s,l}$, given by the dynamic equation

$$\epsilon_{s,l}(t+1) = W_{\boldsymbol{s}}\left[\lambda\epsilon_{s,l}(t) + \chi_{\boldsymbol{s}}(t) + \Delta_{\boldsymbol{s},l}\phi_{I,q}(t)\right]^b + \lambda\xi_{\boldsymbol{s}}(t) + \xi_{\boldsymbol{s}}(t+1)\,,$$

where the following *mismatch vector* was introduced

$$\Delta_{\boldsymbol{s},l}\phi_{I,q}(t) \triangleq \mathrm{col}((1-c^{-(t-T_0)})\phi_{I,q}^{(s_I)}(t) - \hat{\phi}_{s,l}(t),\, I\in\mathcal{O}_{\boldsymbol{s}})$$

and I is any index in the overlap set $\mathcal{O}_{\boldsymbol{s}}$. The solution can then be written as

$$\begin{aligned}\epsilon_{s,l}(t) = \sum_{h=T_d}^{t-1}(\lambda W_{\boldsymbol{s}})^{t-1-h}W_{\boldsymbol{s}}\left[\chi_{\boldsymbol{s}}(h) + \Delta_{\boldsymbol{s},l}\phi_{I,q}(h)\right]^b \\ + \sum_{h=T_d}^{t-1}\left[(\lambda W_{\boldsymbol{s}})^{t-1-h}(\lambda\xi_{\boldsymbol{s}}(h) + \xi_{\boldsymbol{s}}(h+1))\right] + (\lambda W_{\boldsymbol{s}})^{t-T_d}\epsilon_{s,l}(T_d)\,,\end{aligned}$$

and componentwise is described by

$$\begin{aligned}\epsilon_{I,l}^{(s_I)}(t) = w_{\boldsymbol{s},I}\sum_{h=T_d}^{t-1}(\lambda W_{\boldsymbol{s}})^{t-1-h}[\chi_{\boldsymbol{s}}(h) + \Delta_{\boldsymbol{s},l}\phi_{I,q}(h)] \\ + \lambda w_{\boldsymbol{s},I}\sum_{h=T_d}^{t-2}\left[(\lambda W_{\boldsymbol{s}})^{t-2-h}(\lambda\xi_{\boldsymbol{s}}(h) + \xi_{\boldsymbol{s}}(h+1))\right] \\ + \lambda\xi_I^{(s_I)}(t-1) + \xi_I^{(s_I)}(t) + \lambda w_{\boldsymbol{s},I}(\lambda W_{\boldsymbol{s}})^{t-1-T_d}\epsilon_{s,l}(T_d)\,.\end{aligned}$$

Now, it is possible to prove a sufficient condition for *fault isolability*, providing a characterization in a non-closed form of a class of faults that can be isolated by the proposed scheme.

Theorem 6.4.1 (Fault Isolability): Given a fault $\phi_{I,q} \in \mathcal{F}_I$, if for each $l \in \{1,\ldots,N_{\mathcal{F}_I}\}\setminus q$ there exists some time instant $T_l > T_d$ and some

$s_I \in \{1, \ldots, n_I\}$ such that the following inequality holds

$$
\begin{aligned}
|w_{\boldsymbol{s},I} \sum_{h=T_d}^{T_l-1} (\lambda W_{\boldsymbol{s}})^{T_l-1-h} \Delta_{\boldsymbol{s},l} \phi^b_{I,q}(h)| > \\
w_{\boldsymbol{s},I} \sum_{h=T_d}^{T_l-1} (\lambda W_{\boldsymbol{s}})^{T_l-1-h} [2\bar{\chi}_{\boldsymbol{s}}(h) + \bar{\Delta}\phi_s(h)]^b \\
+ 2\left\{\lambda w_{\boldsymbol{s},I} \sum_{h=T_d}^{T_l-2} [(\lambda W_{\boldsymbol{s}})^{T_l-2-h} (\lambda \bar{\xi}_{\boldsymbol{s}}(T_l) + \bar{\xi}_{\boldsymbol{s}}(T_l+1))]\right. \\
\left. + \lambda \bar{\xi}^{(s_I)}_I(T_l-1) + \bar{\xi}^{(s_I)}_I(T_l) + \lambda w_{\boldsymbol{s},I} (\lambda W_{\boldsymbol{s}})^{T_l-1-T_d} \bar{\epsilon}_{s,l}(T_d)\right\},
\end{aligned}
$$

then, the q–th fault will be isolated. Furthermore, the local isolation time is upper-bounded by $\max_{l \in \{1,\ldots,N_{\mathcal{F}_I}\} \setminus q} (T_l)$.

Proof: By using the triangle inequality, the absolute value of the s_I–th component of the l–th FIE of the I–th LFD estimation error is bounded for $t > T_d$ by

$$
\begin{aligned}
|\epsilon^{(s_I)}_{I,l}(t)| \geq |w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h} \Delta_{\boldsymbol{s},l} \phi^b_{I,q}(h)| - |w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h} \chi^b_{\boldsymbol{s}}(h)| \\
- |\lambda w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-2} [(\lambda W_{\boldsymbol{s}})^{t-2-h} (\lambda \xi_{\boldsymbol{s}}(h) + \xi_{\boldsymbol{s}}(h+1))]| \\
- |\lambda \xi^{(I)}_{\boldsymbol{s}}(t-1)| - |\xi^{(I)}_{\boldsymbol{s}}(t)| - |\lambda w_{\boldsymbol{s},I} (\lambda W_{\boldsymbol{s}})^{t-1-T_d} \epsilon_{s,l}(T_d)|.
\end{aligned}
$$

Because of the way its threshold has been defined

$$
\begin{aligned}
|\epsilon^{(s_I)}_{I,l}(t)| \geq |w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h} \Delta_{\boldsymbol{s},l} \phi^b_{I,q}(h)| - w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-1} (\lambda W_{\boldsymbol{s}})^{t-1-h} \bar{\chi}^b_{\boldsymbol{s}}(h) \\
- \lambda w_{\boldsymbol{s},I} \sum_{h=T_d}^{t-2} [(\lambda W_{\boldsymbol{s}})^{t-2-h} (\lambda \bar{\xi}_{\boldsymbol{s}}(h) + \bar{\xi}_{\boldsymbol{s}}(h+1))] - \lambda \bar{\xi}^{(I)}_{\boldsymbol{s}}(t-1) \\
- \bar{\xi}^{(I)}_{\boldsymbol{s}}(t) - \lambda w_{\boldsymbol{s},I} (\lambda W_{\boldsymbol{s}})^{t-1-T_d} \bar{\epsilon}_{s,l}(T_d).
\end{aligned}
$$

In order for the l–th fault to be excluded, the inequality $|\epsilon^{(s_I)}_{I,l}(t)| > \bar{\epsilon}^{(s_I)}_{I,l}(t)$

must be satisfied. This translates to the following further inequality

$$|w_{\boldsymbol{s},I}\sum_{h=T_d}^{t-1}(\lambda W_{\boldsymbol{s}})^{t-1-h}\Delta_{\boldsymbol{s},l}\phi^b_{I,q}(h)| \geq \bar{\epsilon}_{I,l}(t) + w_{\boldsymbol{s},I}\sum_{h=T_d}^{t-1}(\lambda W_{\boldsymbol{s}})^{t-1-h}\bar{\chi}^b_{\boldsymbol{s}}(h) + \lambda w_{\boldsymbol{s},I}\sum_{h=T_d}^{t-2}[(\lambda W_{\boldsymbol{s}})^{t-2-h}(\lambda\bar{\xi}_{\boldsymbol{s}}(h)+\bar{\xi}_{\boldsymbol{s}}(h+1))] + \lambda\bar{\xi}^{(I)}_{\boldsymbol{s}}(t-1)| + \bar{\xi}^{(I)}_{\boldsymbol{s}}(t) + \lambda w_{\boldsymbol{s},I}(\lambda W_{\boldsymbol{s}})^{t-1-T_d}\bar{\epsilon}_{s,l}(T_d)\,,$$

which is implied by the inequality in the hypothesis of the present theorem. Should the inequality hold for every fault function of $\mathcal{F}_I$ but the q–th, then this fault will be isolated in the sense of Definition 6.4.4. ■

6.4.3 Global fault isolation logic

The global isolation logic is analogous to the one presented in [187] and [29] concerning the discrete-time context. As previously pointed out, in the present DFDI setting a distinction between local and distributed faults is made. If a fault is local, then it is sufficient that the corresponding LFD excludes every but that fault to declare it isolated. However, for distributed faults the isolation needs that all the LFDs, in the influence set of that fault (Def. 6.4.1), exclude all other faults.

Hence, the following definition is introduced.

Definition 6.4.5: A fault $\boldsymbol{\phi_1} \in \mathcal{F}$ is *globally isolated* if for each J-th LFD in the fault influence set $U_{\mathbf{1}}$, the corresponding local functions ϕ_{J,l_J} have been isolated, with $J \in \mathcal{U}_{\mathbf{1}}$. Furthermore $T_{isol,\mathbf{1}} \triangleq \max\{T_{locisol,J,l_J},\, J \in \mathcal{U}_{\mathbf{1}}\}$ is the *global fault isolation time.*

In practice, the global isolation task is carried on by the GFD, by using the fault influence sets of all the global faults in $\mathcal{F}$, and the LFDs local fault decisions. It must be noted that a locally isolated fault may still be excluded at a later time by its LFD, so that any global or local isolation decision should never be considered definitive.

6.5 Concluding remarks

In this Chapter the Distributed Fault Detection and Isolation Architecture has been analyzed. The issue of delays and packet dropouts in the communication network between diagnosers have been addressed, by proposing a delay compensation strategy. Fault detectability and isolability conditions have been derived and the convergence of the estimator with the time-varying consensus matrix has been proved. In the following chapter, some simulation results will be presented in order to show the effectiveness of the proposed monitoring architecture.

Chapter 7

Simulation results

In this chapter, we present some simulation results in order to prove the effectiveness of the proposed methods. First of all, we consider the delay compensation strategy implemented by the local diagnosers and we will show that the time-varying consensus matrix improves the detectability in all scenarios, with and without delays, comparing the performances to the cases with different consensus matrices. In a second section, we present the simulation results obtained by modeling the monitoring architecture proposed in this work in order to show its effectiveness. In particular, we analyze the effects of the presence of the sensor network layer, by introducing non synchronized measurements and communication delays and by implementing the re-synchronization scheme. Moreover, we consider the advantages of the distributed estimation method.

7.1 First simulation example: the time-varying consensus matrix

In this Section, we illustrate the effectiveness of the proposed DFDI architecture, presented in Section 6, by means of some simulation experiments. More specifically, we analyze the introduction of the time-varying consensus matrix and the delay compensation strategy. We consider therefore only the distributed detection problem, assuming that all the measurements are synchronized. We consider an eleven-tank system, an extension of the well-known benchmark three-tank system ([22], [188], [189]). The monolithic system (see the first level of Fig. 8.1, where the square labels refer to the pipes number) is decomposed into three overlapping subsystems: the decomposition is $\mathcal{D} = \{\mathscr{S}_1, \mathscr{S}_2, \mathscr{S}_3\}$, with index sets $\mathcal{I}_1 = [1\ 2\ 3\ 4\ 5]^\top$, $\mathcal{I}_2 = [4\ 5\ 6\ 7]^\top$ and $\mathcal{I}_3 = [5\ 8\ 9\ 10\ 11]^\top$. The variables $x^{(4)}$ and $x^{(5)}$, which correspond to tanks number 4 and 5, are shared, and so the related overlap index sets are $\mathcal{O}_4 = \{1, 2\}$ and $\mathcal{O}_5 = \{2, 3\}$. There are three pumps, connected to the first, seventh and eleventh tank with the input flows: $u_1 = 1.25 + 0.25 \cdot \sin(0.05 \cdot t)$,

$u_2 = 1.9 - 1 \cdot \sin(0.005 \cdot t)$ and $u_3 = 1.3 + 0.6 \cdot \cos(0.03 \cdot t)$. The tank sections are nominally equal to $A = [\,1\ 0.5\ 1\ 1\ 2\ 2\ 1\ 1\ 0.5\ 0.5\ 0.5\,]\ \mathrm{m}^2$ and the interconnecting pipe cross-sections to

$$A_p = [\,0.2\ 0.22\ 0.38\ 0.2\ 0.16\ 0.18\ 0.24\ 0.2\ 0.18\ 0.14\ 0.42\ 0.2\,]\ \mathrm{m}^2.$$

Moreover, each tank is connected to a drain pipe, whose nominal cross-section are $A_d = [\,0.025\ 0.0125\ 0.0225\ 0.0275\ 0.075\ 0.0375\ 0.025\ 0.03\ 0.01\ 0.0125\ 0.015\,]\ \mathrm{m}^2$. The pipes outflow coefficients are all equal to 1. The uncertainty affecting tanks and pipes cross-sections values are lower than 5% and 8% of the nominal values respectively. The upper bound on uncertainty in outflow coefficients is 10%. Finally, the measurement errors ξ_I of tank levels y_I are upper bounded by $\bar{\xi}_1 = [\,0.05\ 0.05\ 0.05\ 0.05\ 0.05\,]$ m, $\bar{\xi}_2 = [\,0.06\ 0.06\ 0.06\ 0.06\,]$ m, and $\bar{\xi}_3 = [\,0.04\ 0.04\ 0.04\ 0.04\ 0.04\,]$ m. In order to learn interconnection functions, that in this case consists in the flows through pipes crossing a subsystem boundary, each LFD is provided with adaptive approximators $\hat{g}_I$, implemented by RBF neural networks. The parameter domains Θ_I are considered as hyperspheres with radii $\begin{bmatrix} 2 & 3 & 2 \end{bmatrix} \cdot T_s$, where $T_s = 0.2\,\mathrm{s}$ is the sampling period. The learning parameters are set to $\mu_{1,0} = 10^{-4}$, $\varepsilon_{1,0} = 10^{-3}$, $\mu_{2,0} = 0.5 \cdot 10^{-4}$, $\varepsilon_{2,0} = 10^{-3}$, $\mu_{3,0} = 0.5 \cdot 10^{-4}$, $\varepsilon_{3,0} = 10^{-3}$. We set the filter constant $\lambda = 0.9$. The total uncertainties $\chi_I(t) \triangleq \Delta f_I(t) + \Delta g_I(t) - \lambda \xi_I(t)$ were bounded by $\bar{\chi}_1 = \begin{bmatrix} 0.36 & 0.42 & 0.42 & 0.6 & 0.6 \end{bmatrix} \cdot T_s$, $\bar{\chi}_2 = \begin{bmatrix} 0.36 & 0.48 & 0.42 & 0.3 \end{bmatrix} \cdot T_s$, $\bar{\chi}_3 = \begin{bmatrix} 0.6 & 0.6 & 0.42 & 0.72 & 0.54 \end{bmatrix} \cdot T_s$.A fault is introduced in the variable $x^{(4)}$, monitored by both LFD1 and LFD2, by simulating a leakage in tank 4 at time $t = 10$s: this is modeled as a circular hole of unknown radius $0 \leq \rho^{(i)} \leq A^{(i)}$ in the tank bottom, so that the outflow due to the leak is $q_f^{(i)} = \pi(\rho^{(i)})^2 \sqrt{2g x^{(i)}(t)}$, $i = 4$. We consider three different simulation scenarios. The simulation results are summarized in Tables 7.1-7.3. In the first scenario, the system is not affected by delays; in the second a constant communication delay equal to twice the sampling time, is introduced in the link between LFD1 and LFD2; in the third scenario, the delay is a Heavyside step function centered in $t = 5$s. Three different approaches were tested: in the first case the original architecture [29] with a Metropolis matrix as a consensus weighting matrix is used. In the second case, a modified weighting matrix is adopted, weighting more the subsystem that has lower total uncertainty bound $\bar{\chi}_I$, while in the third case, the methodology proposed in this work is implemented: the novel time-varying consensus approach is introduced, weighting more the subsystem that has lower uncertainty and shorter delays, and the delay compensation strategy is applied. The second case is introduced in order to show that the lower performances of the original FD architecture do not depend only on a not optimal choice of the constant weighting matrix. We can see in Tables 7.1-7.3 that in all cases the fault is detected by both the LFDs.

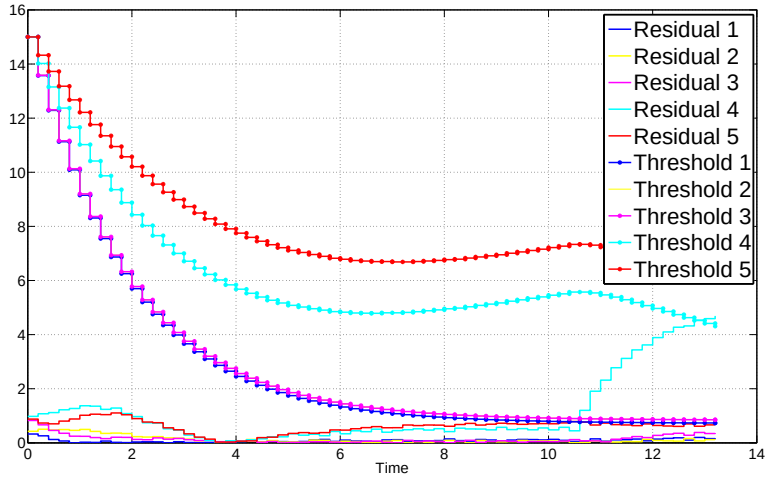

Figure 7.1: Case 1: original approach; Scenario 1: no delays.

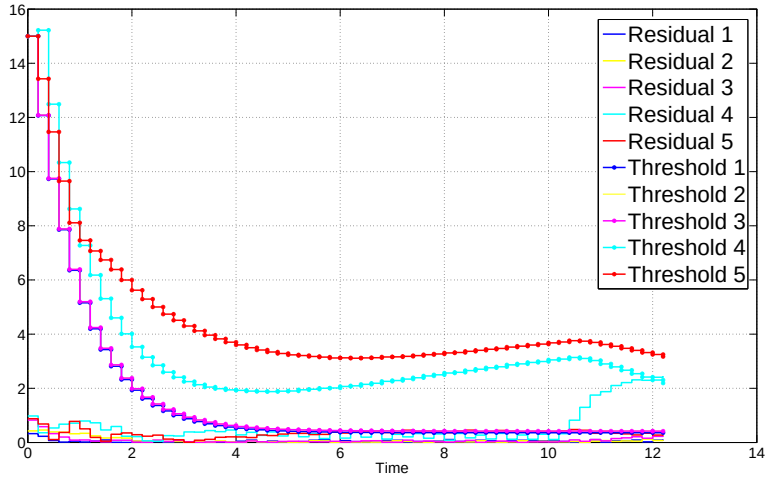

Figure 7.2: Case 3: proposed approach; Scenario 1: no delays.

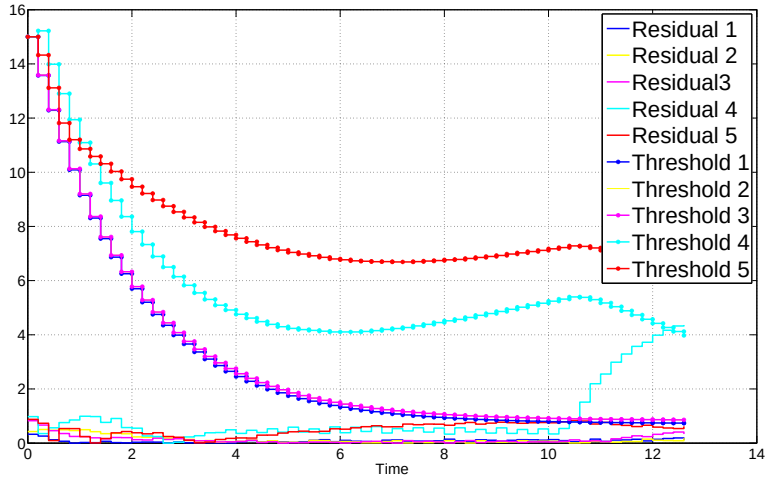

Figure 7.3: Case 3: proposed approach; Scenario 3: step delay.

Figure 7.4: Residuals and thresholds. LFD 1

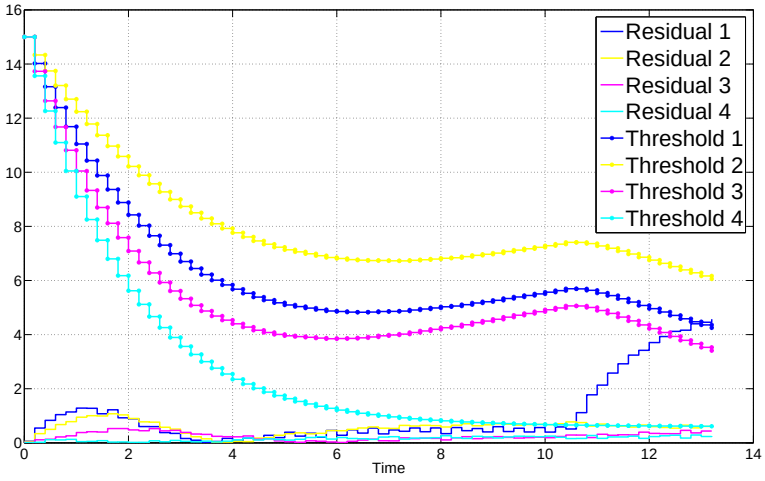

Figure 7.5: Case 1: original approach; Scenario 1: no delays.

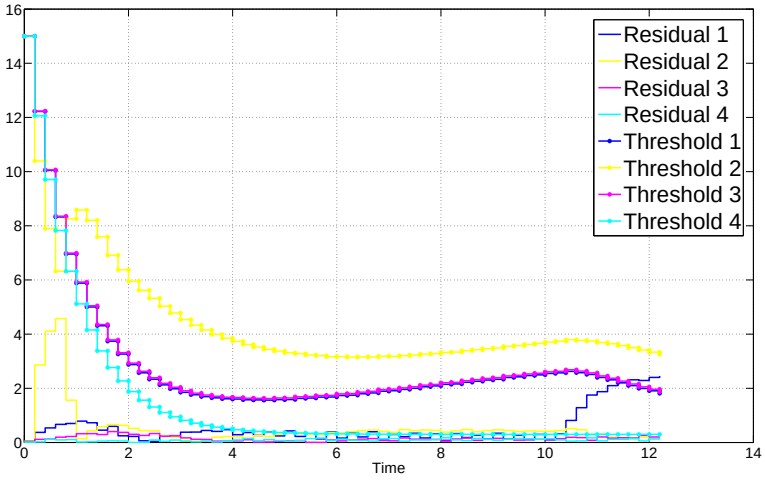

Figure 7.6: Case 3: proposed approach; Scenario 1: no delays.

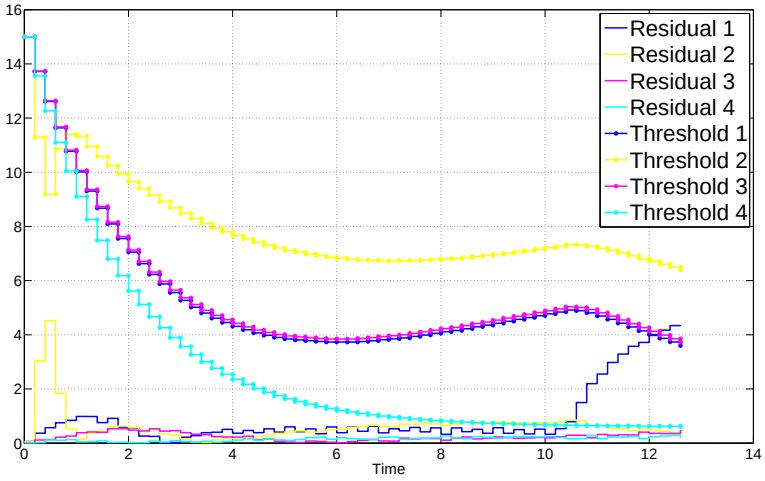

Figure 7.7: Case 3: proposed approach; Scenario 3: step delay.

Figure 7.8: Residuals and thresholds. LFD 2

The introduction of communication delays increases the detection time T_d in all cases. On the other hand, the use of the proposed consensus approach improves the detectability performance, reducing the detection time in all the scenarios, also when no delays are occurring. This demonstrates the effectiveness of the proposed methodology. Figures 7.4 and 7.8 show

Case	W_4 Matrix	LFD1 T_d [s]	LFD2 T_d [s]	T_d [s]
1	$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$	12.8	13	12.8
2	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	12.2	12	12
3	Time-varying	12	11.6	11.6

Table 7.1: Simulation results. No delays. Fault time: $t = 10$s.

Case	W_4 Matrix	LFD1 T_d [s]	LFD2 T_d [s]	T_d [s]
1	$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$	13	13.2	13
2	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	12.8	12.6	12.6
3	Time-varying	12.4	12.2	12.2

Table 7.2: Simulation results. Fixed delay.

residuals and thresholds signals for some of the considered simulation scenarios. We can observe that the novel approach with time-varying consensus matrices presents lower thresholds (Fig.7.2 and 7.6), considering the same simulation scenario. The introduction of delays increases the level of the

Case	W_4 Matrix	LFD1 T_d [s]	LFD2 T_d [s]	T_d [s]
1	$\begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$	13.4	13	13
2	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	12.8	12.6	12.6
3	Time-varying	12.4	12.2	12.2

Table 7.3: Simulation results. Step delay.

thresholds (Fig.7.3 and 7.7). However, the time-varying consensus matrix allows to limit the increase: in fact, in the case of shared variables, the threshold is lower than the scenario without delays using the original approach (see Fig.7.1 and 7.5). This results in a reduction of the detection time using the novel delay-compensation approach.

7.2 The second example: the re - synchronization mechanism

In this section, we present the simulation results obtained by modeling the monitoring architecture proposed in the present work in order to show the effectiveness of the re-synchronization scheme. In particular, we deeply analyze the contributions of the sensor networks layer and the effects of communication delays. For the sake of simplicity, in this section, we consider

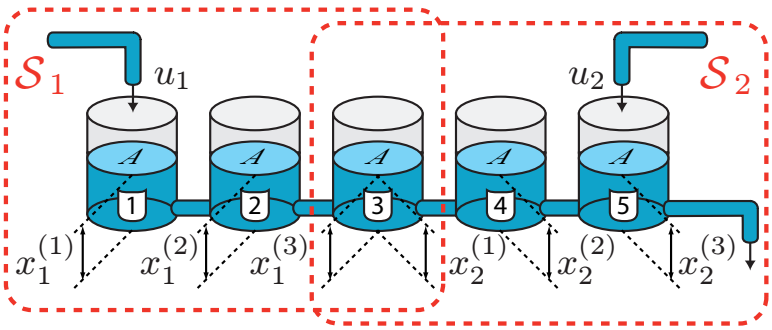

Figure 7.9: Structure of the five-tanks system.

a five-tank string system, monitored by two LFDs (see Fig. 7.9 [27]). The two LFDs monitor three tanks each and share the third tank. The local nominal functions f_1 and f_2 describe the flows through the pipes linking tanks assigned to the same LFD, while the interconnection terms g_1 and g_2 are due to the flow between tanks 3 and 4 and between tanks 2 and 3, respectively. The monolithic system (see Fig. 7.9) is decomposed into two overlapping subsystems, according to the decomposition $\mathcal{D} = \{\mathscr{S}_1, \mathscr{S}_2\}$, with index sets $\mathcal{I}_1 = [1\ 2\ 3]^\top$ and $\mathcal{I}_2 = [3\ 4\ 5]^\top$. The third tank is shared, and therefore the corresponding overlap index set is $\mathcal{O}_3 = \{1, 2\}$. The tank levels are denoted by $x_I^{(i)}$, with $I = \{1, 2\}$ and $i = \{1, 2, 3\}$, and are limited between 0 and 10 m. Two pumps are present, feeding the first and the fifth tank with the following flows: $u_1 = 1.25 + 0.25 \cdot \sin(0.25 \cdot t)$ and $u_2 = 1.75 - 1 \cdot \sin(0.05 \cdot t + 0.4)$. The nominal tank sections are set according to the following vector $A = [\,1\ 1\ 1\ 1\ 1\,]\ \mathrm{m}^2$, while the interconnecting pipe cross-sections are nominally equal to $A_p = [\,0.1\ 0.1\ 0.1\ 0.1\ 0.1\,]\ \mathrm{m}^2$. Furthermore, to each tank are connected drain pipes whose nominal cross-section are $A_d = [\,0.05\ 0.05\ 0.05\ 0.05\ 0.05\,]\ \mathrm{m}^2$. All the pipes outflow coefficients are unitary. By using balance equations and Torricelli's rule, we obtain the state equations (for details about the dynamical equations of a multi-tank system the reader is referred for example to [22]). When building the local models f_I of each LFD, the actual cross-sections used are affected by random uncertainties no larger than 5% and 6% of the nominal values, respectively for the tanks and for the pipes. The outflow coefficients are affected by uncertainties no larger than 10%. Furthermore the tank levels measurements m_I are affected by measuring uncertainties w_I whose components are upper bounded by $\bar{w}_1 = [\,0.2\ 0.25\ 0.3\,]$ m and $\bar{w}_2 = [\,0.3\ 0.15\ 0.2\ ]$ m. The virtual measurement errors are computed on-line basing on the re-synchronization process. In order to learn the interconnection functions of each subsystem, which consist on the flows through pipes crossing a subsystem boundary, each LFD is provided with adaptive approximators $\hat{g}_I$, implemented by RBF neural networks having 3 and 2 neurons respectively along the range of each input dimension. Since the interconnection variables are $z_1 = x_2^{(2)}$ and $z_2 = x_1^{(2)}$, the interconnection functions $g_1(x_1, z_1, u_1)$ and $g_2(x_2, z_2, u_2)$ should be 5-inputs, 3-outputs functions. On the other hand, because of the topology of the specific system, both g_1 and g_2 have only one non-zero output component and depend only on $(x_2^{(2)}, x_1^{(3)})$ and $(x_1^{(2)}, x_2^{(1)})$ respectively. Therefore, the adaptive approximators $\hat{g}_1$ and $\hat{g}_2$ were realized with two 2-inputs, 1-output radial basis neural networks. The network to learn $\hat{g}_1$ is implemented with 9 basis functions, while the network $\hat{g}_2$ is made of 4 basis functions only. After suitable offline simulations, the parameter domains Θ_I were chosen to be hyperspheres with radii equal to $\begin{bmatrix} 3.5 & 3.5 \end{bmatrix} \cdot T_s$, with $T_s = 0.1\,\mathrm{s}$ being the sampling period. The learning

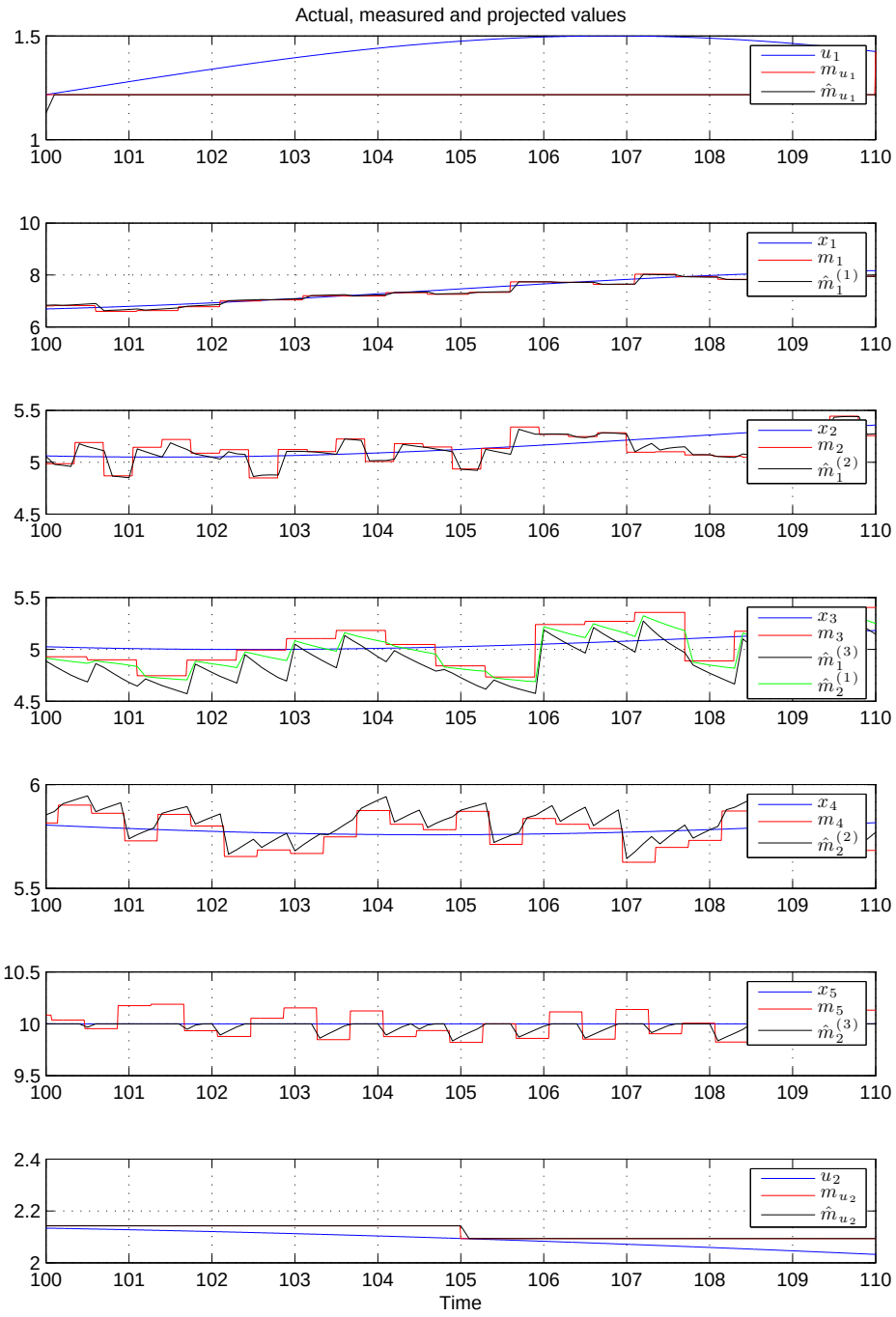

Figure 7.10: The measured and the projected signals.

rate auxiliary coefficients for the interconnection adaptive approximators were set to $\mu_{1,0} = 0.1$, $\varepsilon_{1,0} = 10^{-3}$, $\mu_{2,0} = 0.1$, $\varepsilon_{2,0} = 10^{-3}$, while the filter constants were all set to $\lambda = 0.85$. The different sensor networks, each one measuring a single variable, have different sampling rates. The measurement sampling periods are $[\,10\,15\,0.5\,0.35\,0.6\,0.4\,0.4\,]$, while the offsets with respect to the diagnosers clock are $[\,0\,0\,0.25\,0.3\,0.15\,0.07\,]$. The measurements signals are shown in Figure 7.10, where the real signals, the sampled measurements and the projected signals are illustrated. It is worth noting that the considered case is even more challenging than the one described in the previous chapters, since also the input signals are subject to measurement noise and sampling issues. The communication delays between diagnosers are random and time-varying. In Figure 7.11, the effects of the used delay are shown in the case of two sinusoidal signals as example. In the first figure the received time stamp is plotted, while the second figure shows how the sinusoidal signals are seen by the receiving diagnosers.

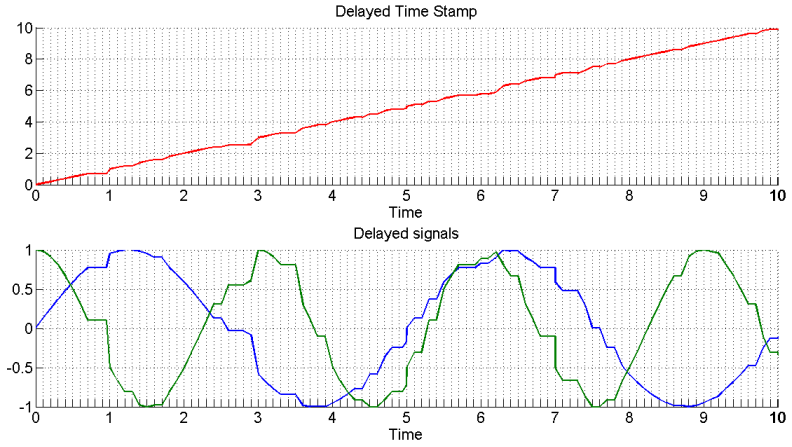

Figure 7.11: The effect of the time-varying communication delays on transmitted signals and time stamps.

The considered fault function represents a leakage (a circular hole of radius 0.15 m) in the third tank occurring at time $t = 200$ s. The simulation results are shown in Figures 7.12-7.15. In Figure 7.12 and 7.13 the detection residuals and the time-varying thresholds are represented. It is possible to see that both the first and the second local fault diagnosers are able to detect the fault occurring on the third tank. In Figure 7.14 and 7.15, it is possible to see that the fault is detected at time $T_d = 200.8$ s by both diagnosers.

For the sake of completeness, we compared the obtained results to the case in which all the measurements are synchronized and no communication

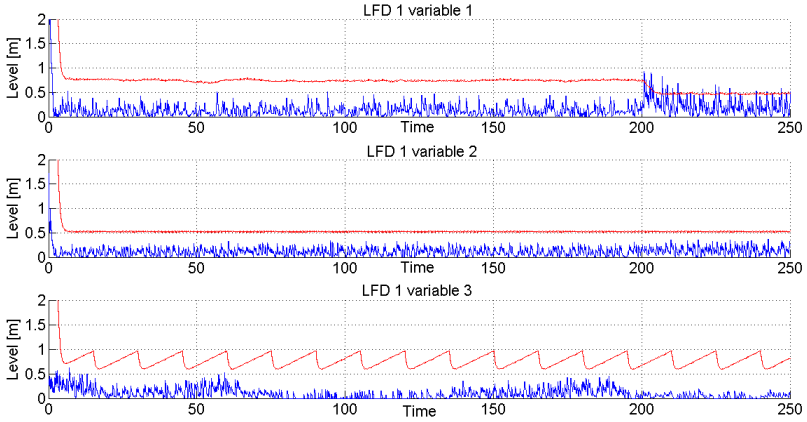

Figure 7.12: Detection residuals and thresholds: LFD 1

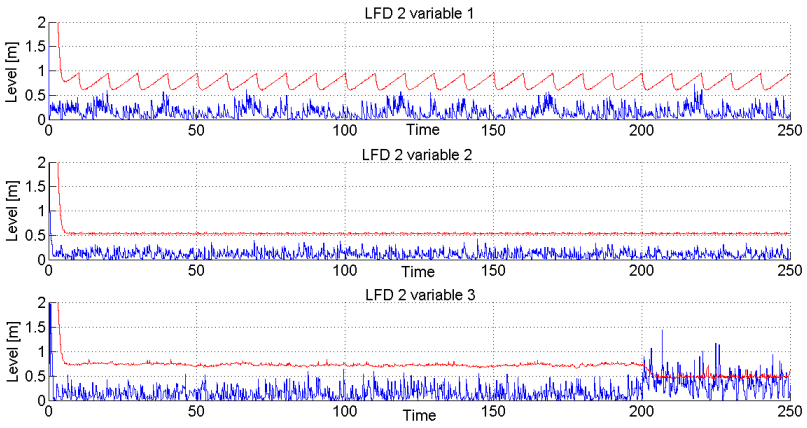

Figure 7.13: Detection residuals and thresholds: LFD 2

delays are present, which is an ideal case. The model and fault parameters are the same used in the case with multi-rate measurements and delayed communication. As it is possible to see in Figure 7.16 and 7.17, in this ideal scenario, the first local fault diagnoser can detect the fault at time $t = 200.6$, while the detection time of the second LFD is $t = 200.8$. In this way, simulation results show that the introduction of the re-synchronization scheme and of the delay compensation strategy allows to obtain fault detection even when the measurements are non synchronized and the communication network is not reliable. Moreover, the detection time has a small delay and is comparable to the ideal case without delays.

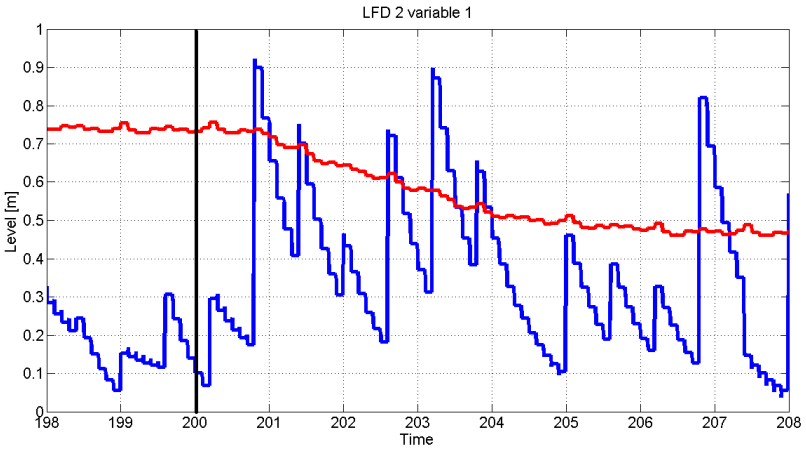

Figure 7.14: Detection residual and threshold: LFD 1 Tank 3

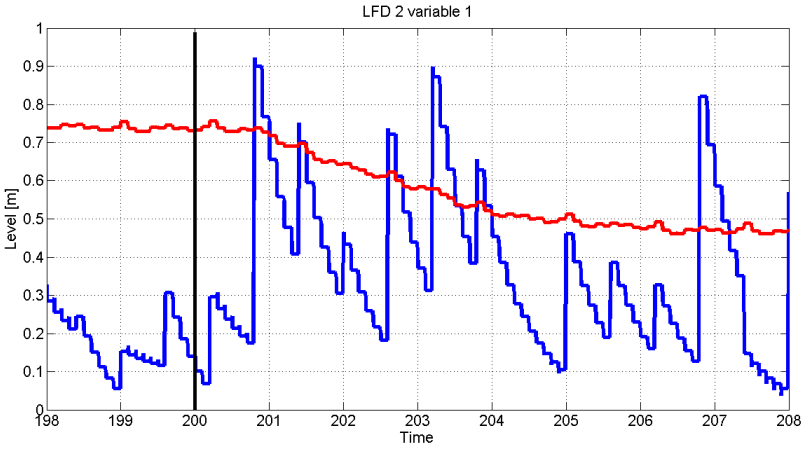

Figure 7.15: Detection residual and threshold: LFD 2 Tank 3

After that, the distributed estimation effectiveness is investigated. We suppose that each state variable is measured by one sensor network. The sensor networks are created as explained in Section 4.6, by distributing 20 sensor nodes randomly in a defined area and letting them communicate with a certain radio range. Each sensor has a different random measurement noise, having zero mean and standard deviation equal to 0.1. The bound on the estimation error is computed by each sensor as explained in Section 4.4. The sensors are affected by a Gaussian zero-mean noise, each one with different standard deviation, which is a random number with standard deviation equal to 1. In Figure 7.18 and 7.19, it is possible to see that, thanks to the Pareto distributed estimation method implemented by the sensor net-

works layer, the detection is possible even with larger measurement noise, as illustrated in Figures 7.20, where the effectiveness of the filtering task is shown.

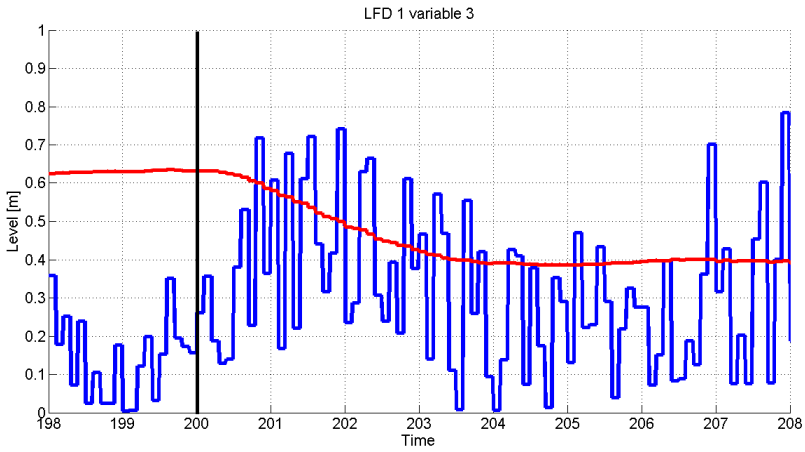

Figure 7.16: Detection residual and threshold: LFD 1 Tank 3 Ideal case

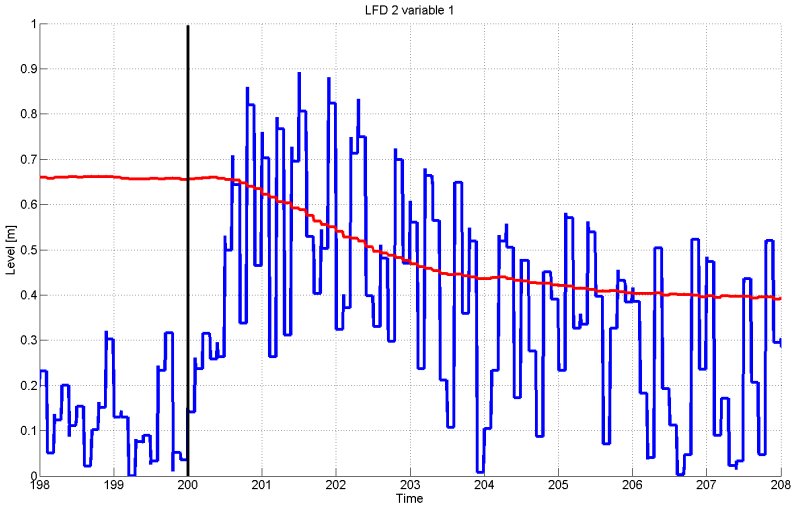

Figure 7.17: Detection residual and threshold: LFD 2 Tank 3 Ideal case

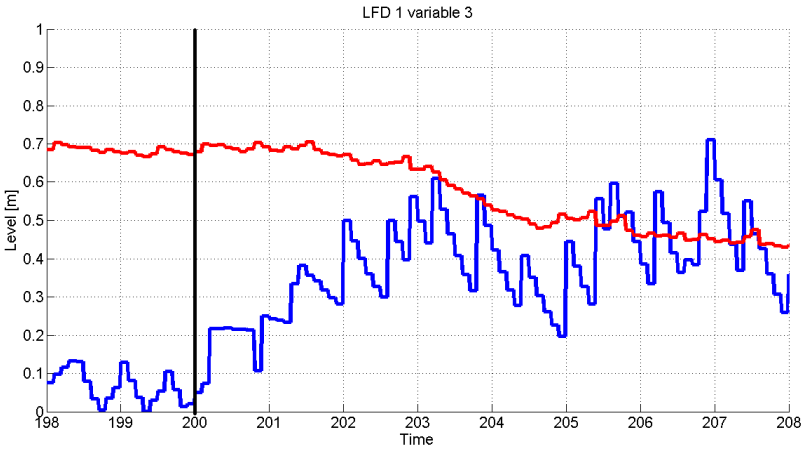

Figure 7.18: Detection residual and threshold: LFD 1 with distributed estimation

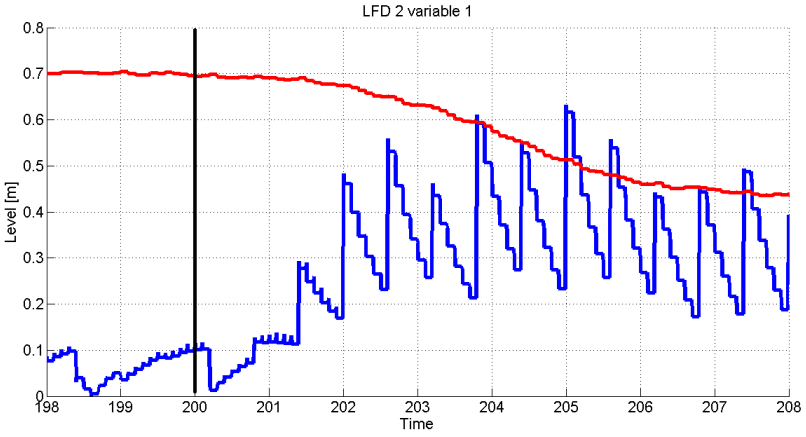

Figure 7.19: Detection residual and threshold: LFD 2 with distributed estimation

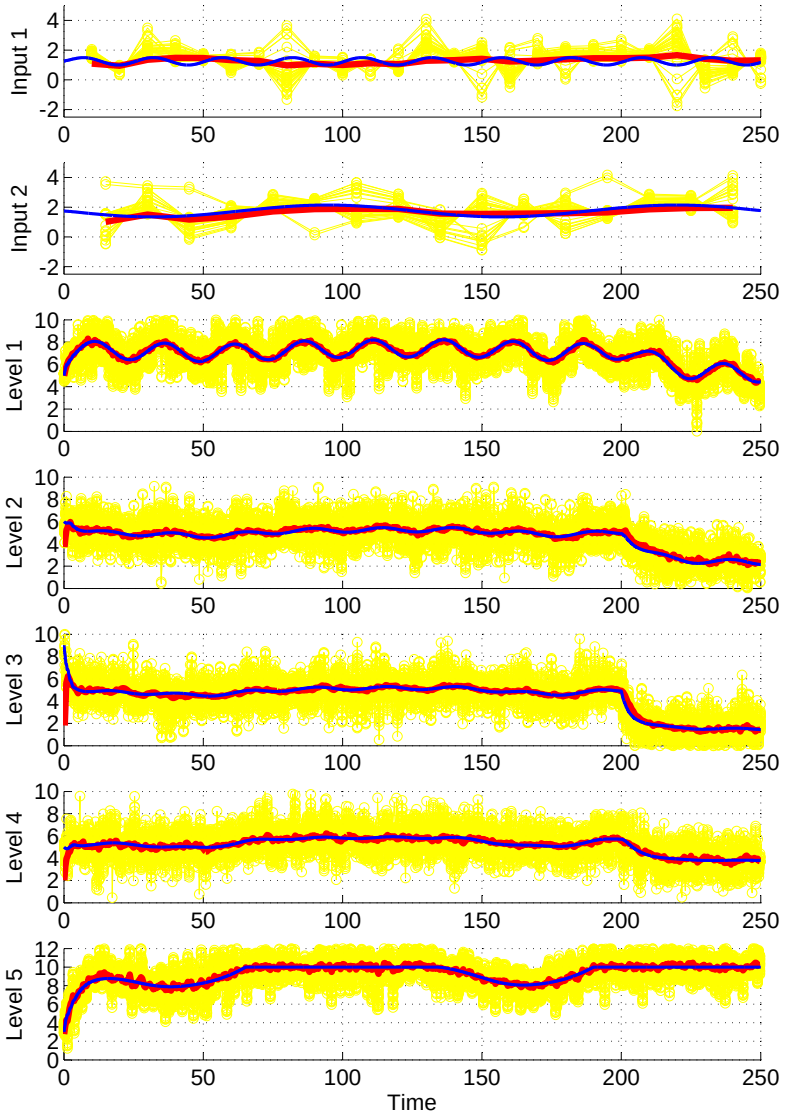

Figure 7.20: Measurements taken by each sensor in the sensor network (yellow), real signal (blue) and transmitted filtered estimates (red).

7.3 Concluding remarks

In this Chapter, some simulation results have been presented in order to show the effectiveness of the proposed monitoring architecture.

In this first Part, some of the issues emerging when dealing with real large-scale networked systems have been addressed, by considering not-synchronized multi-rate systems and not reliable communication networks affected by unknown time-varying delays and packet dropouts. A comprehensive monitoring architecture has been proposed, analyzing all the levels composing complex systems: the physical environment, the sensor layer and the control and monitoring level. The relationships and influences between these layers are considered, motivated by the need for integration emerged by recent scientific research in the fields of Cyber-Physical Systems, Systems of Systems, and in general, of networked and distributed systems.

In Part II, for the sake of completeness, different Fault Detection and Isolation architectures are presented, considering the continuous-time framework and the case in which the state is only partially measurable.

Part II

Other DFDI frameworks

In this part, for the sake of completeness, we present some monitoring architectures designed for different frameworks. More specifically, we extend the discrete-time DFDI scheme presented in Chapter 6 to the continuous-time context and we analyze the case of not completely measurable state. Here, we consider only the part designed for monitoring purposes: a simplified version of the fault diagnosis scheme presented for the discrete-time case, without delay compensation strategy, is designed specifically for different scenarios. The detection and isolation logics are the same, but these architectures differ from the discrete-time case for different problem formulation, different estimators form and different convergence properties. As a future work, these architectures could be extended by considering the issues dealt with in Part I.

Chapter 8

The Continuous-time case

In this chapter we extend the distributed fault detection and isolation architecture presented in Chapter 6 for the continuous-time context [190].

8.1 Problem formulation

Let us consider a nonlinear dynamic system $\mathscr{S}$, described by the following continuous-time model

$$\mathscr{S} \ : \ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\eta}(\mathbf{x}, \mathbf{u}, t) + \beta(t - T_0)\boldsymbol{\phi}(\mathbf{x}, \mathbf{u}). \tag{8.1}$$

Similarly as before, $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ and $\mathbf{u} \in \mathbb{R}^{\mathbf{m}}$ denote the state and input vectors, respectively, and $\mathbf{f} : \mathbb{R}^{\mathbf{n}} \times \mathbb{R}^{\mathbf{m}} \mapsto \mathbb{R}^{\mathbf{n}}$ represents the *nominal healthy dynamics*, $\boldsymbol{\eta} : \mathbb{R}^{\mathbf{n}} \times \mathbb{R}^{\mathbf{m}} \times \mathbb{R}_+ \mapsto \mathbb{R}^{\mathbf{n}}$ the uncertainty function, including external disturbances and modeling errors and $\boldsymbol{\phi}(\mathbf{x}, \mathbf{u})$ denotes the non-linear fault function. The fault time profile is the same as in Eq. (3.2).

The following general decomposition is employed:

$$\mathscr{S}_I : \ \dot{x_I} = f_I(x_I, u_I) + g_I(x_I, z_I, u_I) + \beta(t - T_0)\phi_I(x_I, z_I, u_I)\,, \tag{8.2}$$

where $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ is the *local nominal* function and $g_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ the *interconnection function*. Moreover, $u_I \in \mathbb{R}^{m_I}$, $(m_I \leq \mathbf{m})$ is the *local input*, $z_I \in \mathbb{R}^{p_I}$, $(p_I \leq \mathbf{n} - n_I)$ is the vector of *interconnection variables*, and $\phi_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ is the *local fault function*.

The following assumptions are now introduced. As in the discrete-time case, also here we need the well-posedness assumption:

Assumption 8.1.1: For each $\mathscr{S}_I, I = 1, \ldots, N$, the state variables $x_I(t)$ and control variables $u_I(t)$ remain bounded before and after the occurrence of a fault, i.e., there exist some stability regions $\mathcal{R}_I = \mathcal{R}_I^x \times \mathcal{R}_I^u \subset \mathbb{R}^{n_I} \times \mathbb{R}^{m_I}$, such that $(x_I(t), u_I(t)) \in \mathcal{R}_I^x \times \mathcal{R}_I^u$, $\forall\, I = 1, \ldots, N, \forall\, t \geq 0$.

Owing to Assumption 8.1.1, for each subsystem $\mathscr{S}_I$, $I = 1, \ldots, N$, it is

possible to define some stability regions $\mathcal{R}_I^z$ for the interconnecting variables z_I.

Assumption 8.1.2: The time profile parameter α is unknown but it is lower bounded by a known constant $\bar{\alpha}$.

Assumption 8.1.3: The interconnection function g_I is an unstructured and uncertain nonlinear function, whose k–th component is bounded by some known function, i.e.,

$$|g_I^{(k)}(x_I, z_I, u_I)| \leq \bar{g}_I^{(k)}(x_I, z_I, u_I)\,, \quad \forall\, x_I \in \mathcal{R}_I^x, \forall\, z_I \in \mathcal{R}_I^z, \forall\, u_I \in \mathcal{R}_I^u, \tag{8.3}$$

where the bounding function $\bar{g}_I^{(k)} \geq 0$ is known and bounded for all $I = 1, \ldots, N$.

8.2 Distributed Fault Detection and Isolation Architecture

As in Chapter 6, the proposed DFDI architecture consists of N agents $\mathscr{L}_I$, each one monitoring a single subsystem $\mathscr{S}_I$, $I \in \{1, \ldots, N\}$ and providing a local fault decision d_I^{FD}, regarding the status of the subsystems $\mathscr{S}_I$. The Global Fault Diagnoser coordinates the LFDs and formulates the *global fault decision* d^{FD} about the health of the global system $\mathscr{S}$. Fig. 8.1 illustrates the architecture considered in this chapter, which is a particular case of the monitoring architecture presented in Part I.

The DFDI architecture will be based on the framework portrayed in Chapter 2, Section 2.3.2, where each LFD is equipped with $N_{\mathcal{F}_I}+1$ nonlinear adaptive estimators of the local state x_I, with $I \in \{1, \ldots, N\}$. The first estimator, the FDAE, is used for fault detection, while the remaining $N_{\mathcal{F}_I}$ estimators, the FIEs, are used to determine which of the possible $N_{\mathcal{F}_I}$ fault in the set $\mathcal{F}_I$ has occurred.

The measurement equation of each LFD is assumed to take on the form

$$y_I \triangleq x_I + \xi_I\,,$$

where ξ_I is an unknown function characterizing the measurement error on x_I. As each LFD must communicate with the neighboring LFDs in $\mathcal{V}_I$ in order to fill the interconnection vector z_I, it follows that, instead of receiving the actual interconnection vector z_I, each LFD receives from its neighbors the vector

$$v_I \triangleq z_I + \zeta_I,$$

where ζ_I is made of the components of ξ_J affecting the components of the measurements y_J, $J \in \mathcal{V}_I$.

The following further assumption is needed.

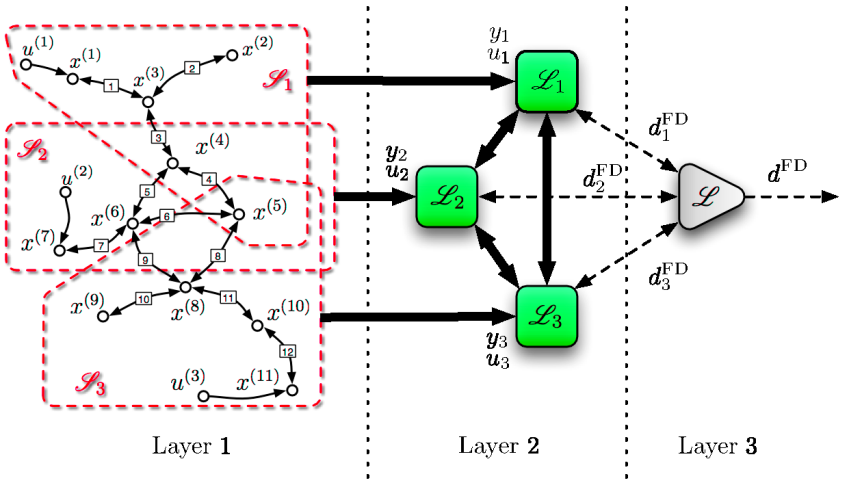

Figure 8.1: A scheme of the proposed DFDI architecture. In this example three subsystems, $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$, each one physically interacting with the other two, are represented in the first layer. In the middle layer each local fault diagnoser $\mathscr{L}_I$ is rendered as a square, with thick black arrows depicting information flows. The arrows from the corresponding subsystem symbolize the direct measurements of local variables by the LFD, while the arrows between the diagnosers account for information exchange between them. The global diagnoser $\mathscr{L}$ in the third layer communicates with each of the lower level LFDs in order to formulate a global fault decision. These information exchanges are rendered with dashed arrows because they are sporadic and event-driven.

Assumption 8.2.1: The measuring uncertainties represented by the vectors ξ_I and ζ_I are unstructured and unknown, but, for each $h = 1, \ldots, n_I$ and for each $k = 1, \ldots, p_I$, the components of ξ_I and of ζ_I are bounded, respectively, as

$$|\xi_I^{(h)}| \leq \bar{\xi}_I^{(h)}, \quad |\zeta_I^{(k)}| \leq \bar{\zeta}_I^{(k)}, \quad \forall\, t \geq 0\,, \tag{8.4}$$

where $\bar{\xi}_I^{(h)}$ and $\bar{\zeta}_I^{(k)}$ are known positive scalars. Hence, it is possible to define *a priori* two compact regions of interest $\mathcal{R}_I^{\xi}$ and $\mathcal{R}_I^{\zeta}$ such that $\xi_I \in \mathcal{R}_I^{\xi}$ and $\zeta_I \in \mathcal{R}_I^{\zeta}$.

In order to take advantage of the multiple measurements (affected by distinct uncertainties) made by the LFDs in the overlapping set of a shared variable, in the following a cooperation mechanism between LFDs is designed to improve the diagnosis performances.

8.3 Distributed Fault Detection

At $t = 0$, the DFDI algorithm is initialized turning on each I-th LFD, for which only its FDAE estimator is enabled and monitors the subsystem $\mathscr{S}_I$, providing a *local state estimate* $\hat{x}_{I,0}$ of the local state x_I. The difference between the estimate $\hat{x}_{I,0}$ and the measurements y_I is the *estimation error* $\epsilon_{I,0}(t) \triangleq y_I(t) - \hat{x}_{I,0}(t)$ used as a residual and compared component-wise with a suitable *detection threshold* $\bar{\epsilon}_{I,0}(t) \in \mathbb{R}_+^{n_I}$.

8.3.1 Local Fault Detection and Approximation Estimator

In this subsection, we design the local fault detection and approximation estimator. Extending to the continuous-time context the approach described in the previous chapter, the local FDAE is a nonlinear adaptive estimator based on the subsystem model (8.2).

We start by considering the simpler case of a non-shared state variable. The estimate of the k-th component $\hat{x}_{I,0}^{(k)}$ is computed as

$$\dot{\hat{x}}_{I,0}^{(k)} = -\lambda_I^{(k)}(\hat{x}_{I,0}^{(k)} - y_I^{(k)}) + f_I^{(k)}(y_I, u_I) + \hat{g}_I^{(k)}(y_I, v_I, u_I, \hat{\vartheta}_{I,0})\,, \tag{8.5}$$

where $-\lambda_I^{(k)} < 0$ is the $k-$estimator pole and $\Lambda \triangleq \text{diag}\,(\lambda_I^1, ..., \lambda_I^{n_I})$, the term $\hat{g}_I^{(k)}$ denotes the k–th output of a linear-in-the-parameters adaptive approximator designed to learn the unknown non-linear interconnection function g_I, and $\hat{\vartheta}_{I,0} \in \hat{\Theta}_{I,0}$ denotes its adjustable parameters vector, with $\hat{\Theta}_{I,0} \subset \mathbb{R}^{q_{I,0}}$ being a given compact set[1].

[1]For the sake of simplicity we assume $\hat{\Theta}_{I,0}$ to be a origin–centered hypersphere with radius $M_{\hat{\Theta}_{I,0}}$ (see [22] for some remarks on this geometrical simplification).

In order to take advantage of the redundancy introduced by the overlap, we adopt the following deterministic consensus scheme between the LFDs in $\mathcal{O}_{\mathbf{s}}$ so that their FDAEs cooperate towards the estimation of the shared state variable $\mathbf{x}^{(\mathbf{s})}$. In this way, the FDAE dynamic equation for the generic I–th LFD, $I \in \mathcal{O}_{\mathbf{s}}$, becomes:

$$\begin{aligned}\dot{\hat{x}}_{I,0}^{(s_I)} = &-\lambda_I^{(s_I)}(\hat{x}_{I,0}^{(s_I)}(t)-y_I^{(s_I)}(t))+W_{\mathbf{s}}^{(I,I)}[f_I^{(s_I)}(y_I,u_I)+\hat{g}_I^{(s_I)}(y_I,v_I,u_I,\hat{\vartheta}_{I,0})] \\ &- \lambda_I^{(s_I)} \sum_{J\in\mathcal{O}_{\mathbf{s}}\setminus\{I\}} W_{\mathbf{s}}^{(I,J)} \left[\hat{x}_{I,0}^{(s_I)}(t) - \hat{x}_{J,0}^{(s_J)}(t)\right] \\ &+ \sum_{J\in\mathcal{O}_{\mathbf{s}}\setminus\{I\}} W_{\mathbf{s}}^{(I,J)}[f_J^{(s_J)}(y_J,u_J) + \hat{g}_J^{(s_J)}(y_J,v_J,u_J,\hat{\vartheta}_{J,0})]\,. \quad (8.6)\end{aligned}$$

It is important to remark that, in order to implement (8.6), the LFD $\mathscr{L}_I$ does not need the information about the functional form of $f_J^{(s_J)}$ and of $\hat{g}_J^{(s_J)}$; instead, it is sufficient that $\mathscr{L}_J$, $J \in \mathcal{O}_{\mathbf{s}}$, computes locally the term $f_J^{(s_J)} + \hat{g}_J^{(s_J)}$ and communicates it, with its actual state estimate $\hat{x}_{J,0}^{(s_J)}$, to other LFDs according to the communication graph $\mathcal{G}_s$.

The term $W_{\mathbf{s}} = [W_{\mathbf{s}}^{(I,J)}]$ is a weighted adjacency matrix reflecting the way the various LFDs cooperate assuming to have a generic communication graph $\mathcal{G}_{\mathbf{s}} \triangleq (\mathcal{O}_{\mathbf{s}}, \mathcal{E}_{\mathbf{s}})$. In this chapter, we consider a doubly-stochastic matrix, as example the *Metropolis* adjacency matrices [191, 192] $W_{\mathbf{s}} \in \mathbb{R}^{N_{\mathbf{s}}\times N_{\mathbf{s}}}$, with $N_{\mathbf{s}}$ being the number of subsystems sharing the variable $\mathbf{s}$, defined as

$$W_{\mathbf{s}}^{(I,J)} \triangleq \begin{cases} 0 & , \text{ if } (I,J) \notin \mathcal{E}_{\mathbf{s}} \\ \frac{1}{1+\max\{d_{\mathbf{s}}^{(I)}, d_{\mathbf{s}}^{(J)}\}} & , \text{ if } (I,J) \in \mathcal{E}_{\mathbf{s}},\ I \neq J \\ 1 - \sum_{K\neq I} W_{\mathbf{s}}^{(I,K)} & , \text{ if } I = J \end{cases}, \quad (8.7)$$

where $d_{\mathbf{s}}^{(I)}$ is the degree of the I–th node in the communication graph $\mathcal{G}_{\mathbf{s}}$. Other solutions are possible.

Now, we describe the design of the adaptive approximator. In order for $\hat{g}_I$ to learn the interconnection function g_I, the parameter vector $\hat{\vartheta}_{I,0}$ is updated according to the following updating law:

$$\dot{\hat{\vartheta}}_{I,0} = \mathcal{P}_{\hat{\Theta}_{I,0}}(\Gamma_{I,0} H_{I,0}^{\top} W_{I,I} \epsilon_{I,0})\,, \quad (8.8)$$

where $H_{I,0} \triangleq \partial \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_{I,0})/\partial \hat{\vartheta}_{I,0} \in \mathbb{R}^{n_I \times q_{I,0}}$ denotes the gradient matrix of the on–line approximator with respect to its adjustable parameters; $\Gamma_{I,0} \in \mathbb{R}^{q_{I,0}\times q_{I,0}}$ is a symmetric and positive definite learning rate matrix; $W_{I,I} \in \mathbb{R}^{n_I \times n_I}$ is a matrix that takes the consensus weights into account: $W_{I,I} = \mathrm{diag}\,(W_1^{(I,I)}, W_2^{(I,I)}, ..., W_{n_I}^{(I,I)})$, where $W_k^{(I,I)} = 1$ if the $k-$th state component is not shared. The initial weight vector $\hat{\vartheta}_{I,0}$ is chosen such that

$\hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_{I,0}(0)) = 0$, which corresponds to the case in which the dynamics of the estimator are described only in terms of the local nominal dynamics.

$\mathcal{P}_{\hat{\Theta}_{I,0}}$ is a projection operator (see [183]) restricting $\hat{\vartheta}_{I,0}$ within $\hat{\Theta}_{I,0}$ according to:

$$\mathcal{P}_{\hat{\Theta}_{I,0}}(\hat{\vartheta}_{I,0}) \triangleq \Gamma_{I,0} H_{I,0}^{\top} W_{I,I} \epsilon_{I,0} - \iota \Gamma_{I,0} \frac{\hat{\vartheta}_{I,0} \hat{\vartheta}_{I,0}^{T}}{\hat{\vartheta}_{I,0}^{T} \Gamma_{I,0} \hat{\vartheta}_{I,0}} \Gamma_{I,0} H_{I,0}^{\top} W_{I,I} \epsilon_{I,0}, \quad (8.9)$$

where ι is the indicator function

$$\iota \triangleq \begin{cases} 1, & \text{if} \quad \|\hat{\vartheta}_{I,0}\| = M_{\hat{\Theta}_{I,0}} \text{ and } \hat{\vartheta}_{I,0}^{\top} \Gamma_{I,0} H_{I,0}^{\top} W_{I,I} \epsilon_{I,0} > 0, \\ 0, & \text{otherwise .} \end{cases}$$

Bearing (8.7) in mind, after some algebra, (8.6) can be rewritten in more compact form as

$$\dot{\hat{x}}_{I,0}^{(s_I)} = -\lambda_I^{(s_I)} \{ \hat{x}_{I,0}^{(s_I)} - y_I^{(s_I)} + \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)} [\, \hat{x}_{I,0}^{(s_I)} - \hat{x}_{J,0}^{(s_J)} \,] \} + \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)} [f_J^{(s_J)}(y_J, u_J) + \hat{g}_J^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,0})] \,. \quad (8.10)$$

Therefore, before a fault occurs (i.e., for $t < T_0$), the dynamics of the LFD estimation error component $\epsilon_{I,0}^{(s_I)}$ can be written as

$$\dot{\epsilon}_{I,0}^{(s_I)} = \lambda_I^{(s_I)} \{ -\epsilon_{I,0}^{(s_I)} + \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)} [\, \epsilon_{J,0}^{(s_J)} - \epsilon_{I,0}^{(s_I)} + \xi_I^{(s_I)} - \xi_J^{(s_J)} \,] \} + \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)} [\, f_J^{(s_J)}(x_J, u_J) - f_J^{(s_J)}(y_J, u_J) + g_J^{(s_J)}(x_J, z_J, u_J) - \hat{g}_J^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,0}) \,] + \dot{\xi}_I^{(s_I)} \,.$$

Since $\sum_{I \neq J} W_{\mathbf{s}}^{(I,J)} = 1 - W_{\mathbf{s}}^{(I,I)}$ by assumption, the estimation error dynamics can be reformulated as

$$\dot{\epsilon}_{I,0}^{(s_I)} = \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)} \{ \lambda_I^{(s_I)} [\epsilon_{J,0}^{(s_J)} - 2\epsilon_{I,0}^{(s_I)} - \xi_J^{(s_J)}] + \Delta f_J^{(s_J)} + \Delta g_J^{(s_J)} \} + \lambda_I^{(s_I)} \xi_I^{(s_I)} + \dot{\xi}_I^{(s_I)} \,, \quad (8.11)$$

where the following scalar quantities are defined: $\Delta f_J^{(s_J)} \triangleq f_J^{(s_J)}(x_J, u_J) - f_J^{(s_J)}(y_J, u_J)$, $\Delta g_J^{(s_J)} \triangleq g_J^{(s_J)}(x_J, z_J, u_J) - \hat{g}_J^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{I,0})$.

The second term can be computed, following some considerations, similarly as the discrete-time case, but here we do not consider the contribution

derived from the use of possibly not up-to-date information, since in this chapter we assume to have a reliable communication network between diagnosers, without delays or packet dropouts. Although the aim of the adaptive approximator $\hat{g}_I$ is to learn the uncertain function g_I, generally it cannot be expected to match the actual term g_I even if the weights of the adaptive approximator could be optimally selected. This may be formalized by introducing the optimal weight vector $\hat{\vartheta}^*_{I,0}$:

$$\hat{\vartheta}^*_{I,0} \triangleq \arg \min_{\hat{\vartheta}_{I,0} \in \Theta_{I,0}} \sup_{x_I, z_I, u_I} \|g_I(x_I, z_I, u_I) - \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_{I,0})\| ,$$

with x_I, z_I, u_I taking values in their respective domains, and defining the Minimum Functional Approximation Error (MFAE) ν_I, which describes the least possible approximation error that can be achieved at time t if $\hat{\vartheta}_{I,0} = \hat{\vartheta}^*_{I,0}$:

$$\nu_I(t) \triangleq g_I(x_I(t), z_I(t), u_I(t)) - \hat{g}_I(x_I(t), z_I(t), u_I(t), \hat{\vartheta}^*_{I,0}) .$$

Finally we introduce the parameter estimation error $\tilde{\vartheta}_{I,0} \triangleq \hat{\vartheta}^*_{I,0} - \hat{\vartheta}_{I,0}$ and the function

$$\Delta\hat{g}_I \triangleq \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_{I,0}) - \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_{I,0}) .$$

In this way, Δg_I can be written as $\Delta g_I = H_{I,0}\tilde{\vartheta}_{I,0} + \nu_I + \Delta\hat{g}_I$.

By using (8.11), the dynamics of the LFD estimation error component $\epsilon^{(s_I)}_{I,0}$ before the occurrence of a fault ($t < T_0$) can be written as

$$\dot{\epsilon}^{(s_I)}_{I,0} = \sum_{J \in \mathcal{O}_\mathbf{s}} W^{(I,J)}_\mathbf{s} \left[\lambda^{(s_I)}_I (\epsilon^{(s_J)}_{J,0} - 2\epsilon^{(s_I)}_{I,0}) + \chi^{(s_J)}_J\right] + \lambda^{(s_I)}_I \xi^{(s_I)}_I + \dot{\xi}^{(s_I)}_I , \quad (8.12)$$

where we introduced the following *total uncertainty* term

$$\chi^{(s_I)}_I \triangleq \Delta f^{(s_I)}_I + \Delta g^{(s_I)}_I - \lambda^{(s_I)}_I \xi^{(s_I)}_I .$$

In order to analyze the behavior of $\epsilon^{(s_I)}_{I,0}$ and define the threshold $\bar{\epsilon}^{(s_I)}_{I,0}(t)$, it is convenient to introduce the following vectors related to the detection estimator of all the LFDs sharing the variable $\mathbf{x}^{(\mathbf{s})}$: $\epsilon_{\mathbf{s},0} \triangleq \mathrm{col}(\epsilon^{(s_I)}_{I,0}, I \in \mathcal{O}_\mathbf{s})$, $\chi_\mathbf{s} \triangleq \mathrm{col}(\chi^{(s_I)}_I, I \in \mathcal{O}_\mathbf{s})$, $\Lambda_\mathbf{s} \triangleq \mathrm{diag}\,(\lambda^{(s_I)}_I, I \in \mathcal{O}_\mathbf{s})$, and $\xi_\mathbf{s} \triangleq \mathrm{col}(\xi^{(s_I)}_I, I \in \mathcal{O}_\mathbf{s})$. The FDAE estimation error dynamics of all the LFDs in $\mathcal{O}_\mathbf{s}$ can then be written in a more useful and compact form for analysis purposes:

$$\dot{\epsilon}_{\mathbf{s},0} = -A_\mathbf{s}\epsilon_{\mathbf{s},0} + W_\mathbf{s}\chi_\mathbf{s} + \Lambda_\mathbf{s}\xi_\mathbf{s} + \dot{\xi}_\mathbf{s} , \quad (8.13)$$

where the matrix $A_{\mathbf{s}}$ is defined as:

$$a_{\mathbf{s}}^{(I,J)} \triangleq \begin{cases} -\lambda_I^{(s_I)} W_{\mathbf{s}}^{(I,J)} & , \text{ if } I \neq J \\ \lambda_I^{(s_I)} \left(W_{\mathbf{s}}^{(I,I)} + 2 \sum_{J \in \mathcal{O}_{\mathbf{s}} \setminus \{I\}} W_{\mathbf{s}}^{(I,J)} \right) & , \text{ if } I = J \end{cases} . \tag{8.14}$$

Since $\sum_{J \in \mathcal{O}_{\mathbf{s}} \setminus \{I\}} W_{\mathbf{s}}^{(I,J)} = 1 - W_{\mathbf{s}}^{(I,I)}$, the matrix can be expressed as

$$-A_{\mathbf{s}} = \lambda (W_{\mathbf{s}} - 2\mathbb{I}), \tag{8.15}$$

in the case $\lambda = \lambda_J^{(s_J)}$ for every $J \in \mathcal{O}_{\mathbf{s}}$. Using Gerschgorin circles [193], it is possible to demonstrate that all the eigenvalues of $W_{\mathbf{s}} - 2\mathbb{I}$ are trapped in the collection of circles centered at $W_{\mathbf{s}}^{(I,I)} - 2$ with radii $\sum_{J \in \mathcal{O}_{\mathbf{s}} \setminus \{I\}} W_{\mathbf{s}}^{(I,J)} = 1 - W_{\mathbf{s}}^{(I,I)}$. As a consequence, since $0 < W_{\mathbf{s}}^{(I,I)} < 1$ and $\lambda > 0$ by assumption, all the eigenvalues of $-A_{\mathbf{s}}$ have negative real part: it follows that (8.13) represents the dynamics of a stable LTI continuous–time system. The solution to (8.13) is given by

$$\epsilon_{\mathbf{s},0}(t) = \int_0^t e^{-A_{\mathbf{s}}(t-\tau)} [W_{\mathbf{s}} \chi_{\mathbf{s}}(\tau) + \Lambda_{\mathbf{s}} \xi_{\mathbf{s}}(\tau) + \dot{\xi}_{\mathbf{s}}(\tau)] d\tau + e^{-A_{\mathbf{s}} t} \epsilon_{\mathbf{s},0}(0), \tag{8.16}$$

and, using integration by parts,

$$\epsilon_{\mathbf{s},0}(t) = \int_0^t e^{-A_{\mathbf{s}}(t-\tau)} [W_{\mathbf{s}} \chi_{\mathbf{s}}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}}) \xi_{\mathbf{s}}(\tau)] d\tau + e^{-A_{\mathbf{s}} t} (\epsilon_{\mathbf{s},0}(0) - \xi_{\mathbf{s}}(0)) + \xi_{\mathbf{s}}(t), \tag{8.17}$$

so that, component-wise, we obtain

$$\epsilon_{I,0}^{(s_I)}(t) \equiv \epsilon_{\mathbf{s},0}^{(I)}(t) = a_{\mathbf{s},I}^{\top}(t) \Big\{ \int_0^t e^{A_{\mathbf{s}} \tau} [W_{\mathbf{s}} \chi_{\mathbf{s}}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}}) \xi_{\mathbf{s}}(\tau)] d\tau + \epsilon_{\mathbf{s},0}(0) - \xi_{\mathbf{s}}(0) \Big\} + \xi_I^{(s_I)}(t) , \tag{8.18}$$

where $a_{\mathbf{s},I}^{\top}(t)$ is a vector containing the I–th row of matrix $e^{-A_{\mathbf{s}} t}$. Now, we are able to define a threshold on the estimation error that guarantees no false–positive fault detections for $t < T_0$. In the following, a formulation of this threshold is given. The absolute value of the estimation error for $t < T_0$

can be upper-bounded by using the triangular inequality as follows:

$$\begin{aligned}\left|\epsilon_{I,0}^{(s_I)}(t)\right| &\leq \left|a_{\mathbf{s},I}^{\top}(t)\right| \Big\{ \int_0^t \left| e^{A_{\mathbf{s}}\tau}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}})\xi_{\mathbf{s}}(\tau)] \right| d\tau \\ &\qquad + |\epsilon_{\mathbf{s},0}(0)| + |\xi_{\mathbf{s}}(0)| \Big\} + \left|\xi_I^{(s_I)}(t)\right| \\ &\leq \left|a_{\mathbf{s},I}^{\top}(t)\right| \Big\{ \int_0^t \left\| e^{A_{\mathbf{s}}\tau} \right\| [W_{\mathbf{s}}\bar{\chi}_{\mathbf{s}}(\tau) + \|(\Lambda_{\mathbf{s}} - A_{\mathbf{s}})\| \, \bar{\xi}_{\mathbf{s}}(\tau)] d\tau \\ &\qquad + \bar{\epsilon}_{\mathbf{s},0}(0) + \bar{\xi}_{\mathbf{s}}(0) \Big\} + \bar{\xi}_I^{(s_I)}(t) = \bar{\epsilon}_{I,0}^{(s_I)}(t), \quad (8.19)\end{aligned}$$

with initial conditions $\bar{\epsilon}_{\mathbf{s}}(0) \triangleq \bar{\xi}_I^{(s_I)}(0)$ and where $|\cdot|$ denotes the element by element absolute value, $\|\cdot\|$ the matrix norm; the upper bound on the total uncertainty term is defined as[2]

$$\begin{aligned}\bar{\chi}_J^{(s_J)}(t) \triangleq \max_{\xi_J} |\Delta f_J^{(s_J)}(t)| + \|H_{J,0}\| \kappa_{J,0}(\hat{\vartheta}_{J,0}) + \bar{\nu}_J(t) + \max_{\xi_J} \max_{\zeta_J} |\Delta \hat{g}_J(t) \\ + \lambda \bar{\xi}_J^{(s_J)}(t)| \,,\end{aligned}$$

with the function $\kappa_{J,0}$ being such that[3] $\kappa_{J,0}(\hat{\vartheta}_{J,0}) \geq \|\tilde{\vartheta}_{J,0}\|$. It is worth noting that the adaptive threshold defined in (8.19) can be easily implemented by any LFD in $\mathcal{O}_{\mathbf{s}}$ by means of a linear first-order filter driven by a suitable input (see [22]).

Finally, the estimator equation (8.10) and the error dynamics (8.12) for a non–shared component $x_{I,0}^{(j)}$ are:

$$\hat{x}_{I,0}^{(j)}(t+1) = -\lambda_I^{(j)} \left[\hat{x}_{I,0}^{(j)}(t) - y_I^{(j)}(t) \right] + f_I^{(j)}(y_I(t), u_I(t)) + \hat{g}_I^{(j)}(t) \,,$$

$$\dot{\epsilon}_{I,0}^{(j)}(t) + \lambda_I^{(j)} \epsilon_{I,0}^{(j)}(t) = \chi_I^{(j)}(t) + \lambda_I^{(j)} \xi_I^{(j)}(t) + \dot{\xi}_I^{(j)}(t) \,,$$

and, accordingly, the threshold function can be defined as

$$\bar{\epsilon}_{I,0}^{(j)}(t) \triangleq \int_0^t e^{-\lambda_I^{(j)}(t-\tau)} \bar{\chi}_I^{(j)}(\tau) d\tau + e^{-\lambda_I^{(j)} t} [\bar{\epsilon}_I^{(j)}(0) + \bar{\xi}_I^{(j)}(0)] + \bar{\xi}_I^{(j)}(t)),$$

with $\bar{\epsilon}_{I,0}^{(j)}(0) \triangleq \bar{\xi}_I^{(j)}(0)$.

8.3.2 Faulty behavior and Fault Detectability

We now analyze the behavior of the fault detection methodology in the presence of a fault, and its detectability capabilities. Assuming that at time

[2]The notation $\max_{\xi_J}$ is short-hand for $\max_{\xi_J \in \mathcal{R}^{\xi_J}}$.

[3]As $\Theta_{J,0}$ is compact, the function $\kappa_{J,0}$ can always be defined.

$t = T_0$ a fault $\boldsymbol{\phi}$ occurs, let

$$\phi_{\mathbf{s}}(\mathbf{x}, \mathbf{u}) = \mathrm{col}\,(\boldsymbol{\phi}^{(\mathbf{s})}(\mathbf{x}, \mathbf{u}),\, \mathbf{s} = \mathbf{1}, \ldots, \mathbf{n}) \tag{8.20}$$

where $\boldsymbol{\phi}^{(\mathbf{s})}$ denotes the component of the fault function affecting the s-th state equation of the monolithic system (see (8.1)). For $t \geq T_0$, the estimation error dynamics for a shared state variable $\mathbf{x}^{(\mathbf{s})}$ given by (8.13) becomes

$$\dot{\epsilon}_{\mathbf{s},0}(t) = -A_{\mathbf{s}}\epsilon_{\mathbf{s},0}(t) + W_{\mathbf{s}}\chi_{\mathbf{s}}(t) + (1 - e^{-\alpha(t-T_0)})\phi_{\mathbf{s}}(t) + \lambda_{\mathbf{s}}\xi_{\mathbf{s}}(t) + \dot{\xi}_{\mathbf{s}}(t)\,, \tag{8.21}$$

where $\phi_{\mathbf{s}}(t) \in \mathbb{R}^{N_{\mathbf{s}}}$ is a vector whose components are all equal to $\boldsymbol{\phi}^{(\mathbf{s})}$. The following theorem gives a sufficient condition for a fault to be detected by the I-th LFD, thus defining, in a non-closed form, a set of faults that can be detected by the proposed scheme with the previously-introduced assumptions.

Theorem 8.3.1: Given a subsystem $\mathscr{S}_I$, if there exists a time instant $t_1 > T_0$ such that the fault ϕ_I satisfies the inequality

$$\left| a_{\mathbf{s},I}^{\top}(t_1) \int_{T_0}^{t_1} e^{A_{\mathbf{s}}\Lambda_{\mathbf{s}}\tau}(1 - e^{-\alpha(\tau - T_0)})\phi_s(\tau)d\tau \right| > 2\bar{\epsilon}_{I,0}^{(s_I)}(t_1)\,, \tag{8.22}$$

for at least one component $s_I \in \{1, \ldots, n_I\}$, then the fault will be detected at time t_1, that is $|\epsilon_{I,0}^{(s_I)}(t_1)| > \bar{\epsilon}_{I,0}^{(s_I)}(t_1)$.

Proof: At time instant $t_1 > T_0$, by using (8.16) and (8.20), the estimation error vector $\epsilon_{\mathbf{s},0}$ can be written as

$$\epsilon_{\mathbf{s},0}(t_1) = \int_0^{t_1} e^{-A_{\mathbf{s}}(t_1 - \tau)}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + \Lambda_{\mathbf{s}}\xi_{\mathbf{s}}(\tau) + \dot{\xi}_{\mathbf{s}}(\tau) + (1 - e^{-\alpha(\tau - T_0)})\phi_s(\tau)]d\tau + e^{-A_{\mathbf{s}}t_1}\epsilon_{\mathbf{s},0}(0). \tag{8.23}$$

Then, we apply the same expansion as in equation (8.18): the solution for the estimation error for the s_I–th component of the I–th subsystem can be written as[4]

$$\epsilon_{I,0}^{(s_I)}(t_1) = a_{\mathbf{s},I}^{\top}(t_1) \int_0^{t_1} e^{A_{\mathbf{s}}\tau}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + \Lambda_{\mathbf{s}}\xi_{\mathbf{s}}(\tau) + \dot{\xi}_{\mathbf{s}}(\tau)]d\tau + a_{\mathbf{s},I}^{\top}(t_1) \int_{T_0}^{t_1} e^{A_{\mathbf{s}}\tau}(1 - e^{-\alpha(\tau - T_0)})\phi_s(\tau)d\tau + a_{\mathbf{s},I}^{\top}(t_1)\epsilon_{\mathbf{s},0}(0)\,.$$

[4]Remembering that $W_{\mathbf{s}}$ is doubly stochastic and all the components of $\phi_{\mathbf{s}}$ are equal to $\boldsymbol{\phi}^{(\mathbf{s})}$.

Using the triangle inequality, we obtain

$$\left|\epsilon_{I,0}^{(s_I)}(t_1)\right| \geq -\left|a_{\mathbf{s},I}^{\top}(t_1)\int_0^{t_1} e^{-W_{\mathbf{s}}\Lambda_{\mathbf{s}}\tau}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) - \Lambda_{\mathbf{s}}\xi_{\mathbf{s}}(\tau) + \dot{\xi}_{\mathbf{s}}(\tau)]d\tau\right| - \left|a_{\mathbf{s},I}^{\top}(t_1)\epsilon_{\mathbf{s},0}(0)\right| + \left|a_{\mathbf{s},I}^{\top}(t_1)\int_{T_0}^{t_1} e^{A_{\mathbf{s}}\tau}(1-e^{-\alpha(\tau-T_0)})\phi_s(\tau)d\tau\right|$$

By recalling how the threshold $\bar{\epsilon}_{I,0}^{(s_I)}$ was defined in Subsection 8.3.1, it is easy to see that the last inequality is implied by

$$\left|\epsilon_{I,0}^{(s_I)}(t_1)\right| \geq -\bar{\epsilon}_{I,0}^{(s_I)}(t_1) + \left|a_{\mathbf{s},I}^{\top}(t_1)\int_{T_0}^{t_1} e^{A_{\mathbf{s}}\tau}(1-e^{-\alpha(\tau-T_0)})\phi_s(\tau)d\tau\right| . \quad (8.24)$$

so that the fault detection condition $|\epsilon_{I,0}^{(s_I)}(t_1)| \geq \bar{\epsilon}_{I,0}^{(s_I)}(t_1)$ is implied by the theorem hypothesis. ■

Remark 8: Theorem 8.3.1 provides a sufficient condition for fault detectability and can be easily specialized to the case of non–shared variables. Qualitatively speaking, the inequality on the left-hand side of (8.22) characterizes the relative magnitude of the effect of the fault versus the upper bound on the unknown functions quantified by the right-hand side of (8.22). In [8], possibly less conservative results are given for the case of equally weighted consensus matrices.

8.4 Distributed Fault Isolation

In this section, we address the distributed fault isolation problem by extending to the continuous-time context the approach described in Section 6.4. More specifically, we assume that the fault function $\boldsymbol{\phi}$ may either be unknown or belong to a known global fault set $\mathcal{F}$:

$$\mathcal{F} \triangleq \{\boldsymbol{\phi}_{\mathbf{1}}(\mathbf{x},\mathbf{u}),\ldots,\boldsymbol{\phi}_{\mathbf{N}_{\mathcal{F}}}(\mathbf{x},\mathbf{u})\}.$$

For each subsystem $\mathscr{S}_I$, a local fault set $\mathcal{F}_I$ can be built with the local fault functions obtained by all the global faults $\boldsymbol{\phi}_{\mathbf{l}}$ such that $I \in \mathcal{U}_{\mathbf{l}}$, where $\mathcal{U}_{\mathbf{l}}$ is the fault influence set (definition 6.4.1):

$$\mathcal{F}_I \triangleq \{\phi_{I,1}(x_I,z_I,u_I),\ldots,\phi_{I,N_{\mathcal{F}_I}}(x_I,z_I,u_I)\}.$$

As in the discrete-time case, the generic I–th LFD is only able to exploit the knowledge of the local fault set $\mathcal{F}_I$ and no information about the fault influence sets of the global faults corresponding to the local fault functions belonging to $\mathcal{F}_I$ turns out to be available to the I–th LFD.

8.4.1 Local fault isolation logic

After fault detection at time T_d, every LFD is told by the GFD to stop its FDAE and switch to isolation mode. At the same time the interconnection approximator stops updating its parameters vector, in order to prevent learning also the contribution of the fault function: $\hat{\vartheta}_{I,0}(t) = \hat{\vartheta}_{I,0}(T_d)\,, \forall\, t \geq T_d$. The isolation task is carried on by relying on a bank of $N_{\mathcal{F}_I}$, $I = 1, \ldots, N$, FIEs. This scheme relies on the generic l–th FIE of the I–th LFD being matched to the corresponding fault function $\phi_{I,l}$, belonging to the local fault set $\mathcal{F}_I$. Each fault function in $\mathcal{F}_I$ is of the form

$$\phi_{I,l}(x_I, z_I, u_I) = [(\vartheta_{I,l,1})^\top H_{I,l,1}(x_I, z_I, u_I),\, \ldots,\, (\vartheta_{I,l,n_I})^\top H_{I,l,n_I}(x_I, z_I, u_I)]^\top, \tag{8.25}$$

where, for $k \in \{1, \ldots, n_I\}$, $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, the *known* functions $H_{I,l,k} : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{q_{I,l,k}}$ provide the functional structure of the fault and the *unknown* parameter vectors $\vartheta_{I,l,k} \in \Theta_{I,l,k} \subset \mathbb{R}^{q_{I,l,k}}$ provide its "magnitude". The parameter domains $\Theta_{I,l,k}$ are again assumed to be origin–centered hyper–spheres with radius $M_{\Theta_{I,l,k}}$ (see footnote 1, Subsection 8.3.1). The generic l–th FIE estimator, with $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, monitors its subsystem $\mathscr{S}_I$, computing a *local state estimate* $\hat{x}_{I,l}$ of the local state x_I, analogously to the FDAE. The difference between the estimate $\hat{x}_{I,l}$ and the measurements y_I produces the *isolation residual* $\epsilon_{I,l} \triangleq y_I - \hat{x}_{I,l}$ which, again, is compared, component by component, to the suitable *isolation threshold* $\bar{\epsilon}_{I,l} \in \mathbb{R}^{n_I}_+$. Ideally, the goal of the isolation logic is to exclude every but one fault.

In the following, the FIEs are described in some detail. After the fault $\boldsymbol{\phi}(t)$ has occurred at time $t = T_0$, the state equation of the s_I–th component of the I–th subsystem becomes

$$\dot{x}_I^{(s_I)} = f_I^{(s_I)}(x_I, u_I) + g_I^{(s_I)}(x_I, z_I, u_I) + (1 - e^{-\alpha(t-T_0)})\phi^{(s)}(x, u)\,.$$

The l–th FIE estimator dynamic equation for the most general case of a distributed fault with a shared variable is defined as

$$\begin{aligned}\dot{\hat{x}}_{I,l}^{(s_I)} = -\lambda_I^{(s_I)}\{\hat{x}_{I,l}^{(s_I)} - y_I^{(s_I)} + \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[\hat{x}_{I,l}^{(s_I)} - \hat{x}_{J,l}^{(s_J)}]\} \\ + \sum_{J \in \mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[f_J^{(s_J)}(y_J, u_J) + \hat{g}_J^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,0}(T_d)) \\ + \hat{\phi}_{J,l}^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,l})]\,,\end{aligned} \tag{8.26}$$

where $\hat{\phi}_{J,l}^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,l}) \triangleq (\hat{\vartheta}_{J,l,s_J})^\top H_{J,l,s_J}(y_J, v_J, u_J)$ is the s_J–th component of a linearly-parameterized function that matches the structure of the l–th fault function $\phi_{J,l}$, and the vector $\hat{\vartheta}_{J,l} \triangleq \mathrm{col}(\hat{\vartheta}_{J,l,k}, k \in \{1, \ldots, n_I\})$ has been introduced.

Analogously to the FDAE case, the parameters vectors are updated ac-

cording to the learning law:

$$\dot{\hat{\vartheta}}_{J,l,k} = \mathcal{P}_{\hat{\Theta}_{J,l,k}}(\Gamma_{J,l,k} H_{J,l,k}^{\top} W_{J,J} \epsilon_{J,l,k}) \,,$$

where $\mathcal{P}_{\hat{\Theta}_{J,l,k}}$ is again a suitable projection operator [183], $\Gamma_{J,l,k}(t)$ is the learning rate and $W_{J,J}$ the consensus weights correction matrix. The corresponding estimation error dynamic equation can be written as:

$$\begin{aligned}\dot{\epsilon}_{I,l}^{(s_I)}(t) = -\lambda_I^{(s_I)} \{\epsilon_{I,l}^{(s_I)}(t) + \sum_{J\in\mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[-\epsilon_{J,l}^{(s_J)}(t) + \epsilon_{I,l}^{(s_I)}(t) \\ - \xi_I^{(s_I)}(t) + \xi_J^{(s_J)}(t)]\} + \sum_{J\in\mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[\Delta f_J^{(s_J)}(t) + \Delta g_J^{(s_J)}(t) \\ + (1 - e^{-\alpha(t-T_0)})\phi^{(s)}(t) - \hat{\phi}_{J,l}^{(s_J)}(t)] + \dot{\xi}_I^{(s_I)}(t) \,,\end{aligned}$$

which implies

$$\begin{aligned}\dot{\epsilon}_{I,l}^{(s_I)}(t) = \sum_{J\in\mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[\lambda_I^{(s_I)}(\epsilon_{J,l}^{(s_J)}(t) - 2\epsilon_{I,l}^{(s_I)}(t)) + \chi_J^{(s_J)}(t) \\ + (1 - e^{-\alpha(t-T_0)})\phi^{(s)}(t) - \hat{\phi}_{J,l}^{(s_J)}(t)] + \lambda_I^{(s_I)}\xi_I^{(s_I)}(t) + \dot{\xi}_I^{(s_I)}(t) \,.\end{aligned}$$

When considering a matched fault (that is, $\phi^{(s)} = \phi_{J,l}^{(s_J)}$, $\forall\, J \in \mathcal{O}_{\mathbf{s}}$), the error equation can be written as

$$\begin{aligned}\dot{\epsilon}_{I,l}^{(s_I)}(t) = \sum_{J\in\mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[\lambda_I^{(s_I)}(\epsilon_{J,l}^{(s_J)}(t) - 2\epsilon_{I,l}^{(s_I)}(t)) + \chi_J^{(s_J)}(t) \\ + (1 - e^{-\alpha(t-T_0)})(H_{J,l,s_J}(t)^{\top}\vartheta_{J,l,s_J} + \Delta H_{J,l,s_J}^{\top}\vartheta_{J,l,s_J}) - H_{J,l,s_J}(t)^{\top}\hat{\vartheta}_{J,l,s_J}] \\ + \lambda_I^{(s_I)}\xi_I^{(s_I)}(t) + \dot{\xi}_I^{(s_I)}(t) \,,\end{aligned}$$

where $\Delta H_{J,l,s_J}^{\top} \triangleq H_{J,l,s_J}(x_J, z_J, u_J) - H_{J,l,s_J}(y_J, v_J, u_J)$. Let us introduce the parameter estimation errors $\tilde{\vartheta}_{J,l,s_J} \triangleq \vartheta_{J,l,s_J} - \hat{\vartheta}_{J,l,s_J}$; then, the FIE estimation error equation for a matched fault becomes

$$\begin{aligned}\dot{\epsilon}_{I,l}^{(s_I)}(t) = \sum_{J\in\mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[\lambda_I^{(s_I)}(\epsilon_{J,l}^{(s_J)}(t) - 2\epsilon_{I,l}^{(s_I)}(t)) + \chi_J^{(s_J)}(t) \\ + (1 - e^{-\alpha(t-T_0)})H_{J,l,s_J}(t)^{\top}\tilde{\vartheta}_{J,l,s_J} \\ + (1 - e^{-\alpha(t-T_0)})\Delta H_{J,l,s_J}(t)^{\top}\vartheta_{J,l,s_J} \\ - e^{-\alpha(t-T_0)}H_{J,l,s_J}(t)^{\top}\hat{\vartheta}_{J,l,s_J}] + \lambda_I^{(s_I)}\xi_I^{(s_I)}(t) + \dot{\xi}_I^{(s_I)}(t) \,.\end{aligned}$$

As in Subsection 8.3.1, it can be conveniently expressed in vector form

$$\epsilon_{s,l} \triangleq \mathrm{col}(\epsilon_{I,l}^{(s_I)}, I \in \mathcal{O}_{\mathbf{s}}), \qquad \bar{\epsilon}_{s,l} \triangleq \mathrm{col}(\bar{\epsilon}_{I,l}^{(s_I)}, I \in \mathcal{O}_{\mathbf{s}}) \,,$$

and so

$$\dot{\epsilon}_{s,l}(t) = -A_{\mathbf{s}}\epsilon_{s,l}(t) + W_{\mathbf{s}}[\chi_{\mathbf{s}}(t) + \mathrm{col}((1-e^{-\alpha(t-T_0)})H_{I,l,s_I}(t)^\top\tilde{\vartheta}_{I,l,s_I} \\ +(1-e^{-\alpha(t-T_0)})\Delta H_{I,l,s_I}(t)^\top\vartheta_{I,l,s_I} - e^{-\alpha(t-T_0)}H_{I,l,s_I}(t)^\top\hat{\vartheta}_{I,l,s_I})] + \Lambda_{\mathbf{s}}\xi_{\mathbf{s}}(t) + \dot{\xi}_{\mathbf{s}}(t)\,.$$

The solution, using integration by parts, is:

$$\epsilon_{\mathbf{s},l}(t) = \int_{T_d}^{t} e^{-A_{\mathbf{s}}(t-\tau)}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + W_{\mathbf{s}}\Delta\phi_{\mathbf{s}}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}})\xi_{\mathbf{s}}(\tau)]d\tau \\ + e^{-A_{\mathbf{s}}(t-T_d)}(\epsilon_{\mathbf{s},l}(T_d) - \xi_{\mathbf{s}}(T_d)) + \xi_{\mathbf{s}}(t), \quad (8.27)$$

where

$$\Delta\phi_{\mathbf{s}}(t) \triangleq \mathrm{col}[(1-e^{-\alpha(t-T_0)})H_{I,l,s_I}(t)^\top\tilde{\vartheta}_{I,l,s_I} \\ + (1-e^{-\alpha(t-T_0)})\Delta H_{I,l,s_I}(t)^\top\vartheta_{I,l,s_I} - e^{-\alpha(t-T_0)}H_{I,l,s_I}(t)^\top\hat{\vartheta}_{I,l,s_I},\, I \in \mathcal{O}_{\mathbf{s}}]\,.$$

Componentwise, the estimation error is given by

$$\epsilon_{I,l}^{(s_I)}(t) = a_{\mathbf{s},I}(t)\Big\{\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + W_{\mathbf{s}}\Delta\phi_{\mathbf{s}}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}})\xi_{\mathbf{s}}(\tau)]d\tau \\ + e^{A_{\mathbf{s}}T_d}(\epsilon_{\mathbf{s},l}(T_d) - \xi_{\mathbf{s}}(T_d))\Big\} + \xi_I^{(s_I)}(t), \quad (8.28)$$

and the threshold can be defined as:

$$\bar{\epsilon}_{I,l}^{(s_I)}(t) = |a_{\mathbf{s},I}(t)|\,\Big\{\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\|[W_{\mathbf{s}}\bar{\chi}_{\mathbf{s}}(\tau) + W_{\mathbf{s}}\bar{\Delta}\phi_{\mathbf{s}}(\tau) + \|\Lambda_{\mathbf{s}} - A_{\mathbf{s}}\|\,\bar{\xi}_{\mathbf{s}}(\tau)]d\tau \\ + \left\|e^{A_{\mathbf{s}}T_d}\right\|(\bar{\epsilon}_{\mathbf{s},l}(T_d) + \bar{\xi}_{\mathbf{s}}(T_d))\Big\} + \bar{\xi}_I^{(s_I)}(t), \quad (8.29)$$

where

$$\bar{\Delta}\phi_{\mathbf{s}}(t) = \mathrm{col}[\|H_{I,l,s_I}(t)\|\kappa_{I,l,s_I}(\hat{\vartheta}_{I,l,s_I}) + \bar{\Delta}H_{I,l,s_I}(t)\bar{\vartheta}_{I,l,s_I} \\ - e^{-\bar{\alpha}(t-T_d)}\|H_{I,l,s_I}(t)\|\,\|\hat{\vartheta}_{I,l,s_I}\|,\, I \in \mathcal{O}_{\mathbf{s}}]\,.$$

This threshold guarantees by definition that no matched fault is excluded because of uncertainties or the effect of the parameter estimation error $\tilde{\vartheta}_{I,l,s_I}$.

When considering the case of a non–matched fault, that is,

$$\phi_I^{(s_I)}(x_I, z_I, u_I) = \phi_{I,\gamma}^{(s_I)}(x_I, z_I, u_I, \vartheta_{I,\gamma})$$

for some $I \in \mathcal{O}_{\mathbf{s}}$ and with $\gamma \neq l$, the dynamics of the s_I–component of the

estimation error of the l–th FIE of the I–th LFD can be written as

$$\dot{\epsilon}_{I,l}^{(s_I)}(t) = \sum_{J\in\mathcal{O}_{\mathbf{s}}} W_{\mathbf{s}}^{(I,J)}[\lambda_I^{(s_I)}(\epsilon_{J,l}^{(s_J)} - 2\epsilon_{I,l}^{(s_I)})(t) + \chi_J^{(s_J)}(t) + (1 - e^{-\alpha(t-T_0)})\phi_{I,\gamma}^{(s_I)}(x_I(t), z_I(t), u_I(t), \vartheta_{I,\gamma}) - \hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,l})] + \lambda\xi_I^{(s_I)}(t) + \dot{\xi}_I^{(s_I)}(t)\,.$$

As shown before, a convenient way to study the behavior of the estimation error of the LFDs sharing the variable $\mathbf{x^{(s)}}$ is to consider the vector $\epsilon_{s,l}$, given by the dynamic equation

$$\dot{\epsilon}_{s,l}(t) = -A_{\mathbf{s}}\epsilon_{s,l}(t) + W_{\mathbf{s}}\left[\chi_{\mathbf{s}}(t) + \Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(t)\right] + \lambda\xi_{\mathbf{s}}(t) + \dot{\xi}_{\mathbf{s}}(t)\,,$$

where the following *mismatch vector* was introduced

$$\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(t) \triangleq \mathrm{col}((1 - e^{-\alpha(t-T_0)})\phi_{I,\gamma}^{(s_I)}(t),\, I \in \mathcal{O}_{\mathbf{s}}) - \hat{\phi}_{s,l}(t)\,.$$

The solution is given by:

$$\epsilon_{\mathbf{s},l}(t) = \int_{T_d}^{t} e^{-A_{\mathbf{s}}(t-\tau)}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + W_{\mathbf{s}}\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}})\xi_{\mathbf{s}}(\tau)]d\tau + e^{-A_{\mathbf{s}}(t-T_d)}(\epsilon_{\mathbf{s},l}(T_d) - \xi_{\mathbf{s}}(T_d)) + \xi_{\mathbf{s}}(t)\,,$$

and componentwise

$$\epsilon_{I,l}^{(s_I)}(t) = a_{\mathbf{s},I}^{\top}(t)\left\{\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}[W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau) + W_{\mathbf{s}}\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(\tau) + (\Lambda_{\mathbf{s}} - A_{\mathbf{s}})\xi_{\mathbf{s}}(\tau)]d\tau + e^{A_{\mathbf{s}}T_d}(\epsilon_{\mathbf{s},l}(T_d) - \xi_{\mathbf{s}}(T_d))\right\} + \xi_I^{(s_I)}(t)\,.$$

Then, the following sufficient condition for *local fault isolability* can be easily proved.

Theorem 8.4.1 (Local Fault Isolability): Given a fault $\phi_{I,\gamma} \in \mathcal{F}_I$, if, for each $l \in \{1, \ldots, N_{\mathcal{F}_I}\} \setminus \{\gamma\}$, there exists some time instant $t = T_l > T_d$ and some $s_I \in \{1, \ldots, n_I\}$ such that the following inequality holds

$$|a_{\mathbf{s},I}^{\top}(t)\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}W_{\mathbf{s}}\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(h)| > \bar{\epsilon}_{I,l}^{(s_I)}(t) + \left|a_{\mathbf{s},I}^{\top}(t)\right|\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\| W_{\mathbf{s}}\bar{\chi}_{\mathbf{s}}(\tau)d\tau + \left|a_{\mathbf{s},I}^{\top}(t)\right|\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\|\|\Lambda_{\mathbf{s}} - A_{\mathbf{s}}\|\,\bar{\xi}_{\mathbf{s}}(\tau)d\tau + \left|a_{\mathbf{s},I}^{\top}(t)\right|\left\|e^{A_{\mathbf{s}}T_d}\right\|(\bar{\epsilon}_{\mathbf{s},l}(T_d) + \bar{\xi}_{\mathbf{s}}(T_d)) + \bar{\xi}_I^{(s_I)}(t),$$

then, the γ–th fault is isolated. Furthermore, the local isolation time is

upper-bounded by $\max_{l\in\{1,\dots,N_{\mathcal{F}_I}\}\setminus\{\gamma\}} T_l$.

Proof: By using the triangle inequality, the absolute value of the s_I–th component of the l–th FIE of the I–th LFD estimation error is lower-bounded for $t > T_d$ by

$$\begin{aligned}\left|\epsilon_{I,l}^{(s_I)}(t)\right| \geq &-\left|a_{\mathbf{s},I}^{\top}(t)\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}W_{\mathbf{s}}\chi_{\mathbf{s}}(\tau)d\tau\right| \\ &+\left|a_{\mathbf{s},I}^{\top}(t)\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}W_{\mathbf{s}}\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(\tau)d\tau\right| - \left|a_{\mathbf{s},I}^{\top}(t)\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}(\Lambda_{\mathbf{s}}-A_{\mathbf{s}})\xi_{\mathbf{s}}(\tau)d\tau\right| \\ &- \left|a_{\mathbf{s},I}^{\top}(t)e^{A_{\mathbf{s}}T_d}(\epsilon_{\mathbf{s},l}(T_d)-\xi_{\mathbf{s}}(T_d))\right| - \left|\xi_I^{(s_I)}(t)\right|.\end{aligned}$$

Thanks to the known bounds on $\chi_{\mathbf{s}}$ and $\xi_{\mathbf{s}}$ and the fact that the l–th fault cannot already be excluded at time T_d because of the way its threshold has been defined, we have

$$\begin{aligned}\left|\epsilon_{I,l}^{(s_I)}(t)\right| \geq &-\left|a_{\mathbf{s},I}^{\top}(t)\right|\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\| W_{\mathbf{s}}\bar{\chi}_{\mathbf{s}}(\tau)d\tau \\ &+\left|a_{\mathbf{s},I}^{\top}(t)\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}W_{\mathbf{s}}\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(\tau)d\tau\right| - \left|a_{\mathbf{s},I}^{\top}(t)\right|\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\|\|\Lambda_{\mathbf{s}}-A_{\mathbf{s}}\|\,\bar{\xi}_{\mathbf{s}}(\tau)d\tau \\ &- \left|a_{\mathbf{s},I}^{\top}(t)\right|\left\|e^{A_{\mathbf{s}}T_d}\right\|(\bar{\epsilon}_{\mathbf{s},l}(T_d)+\bar{\xi}_{\mathbf{s}}(T_d)) - \bar{\xi}_I^{(s_I)}(t).\end{aligned}$$

In order for the l–th fault to be excluded, the inequality $|\epsilon_{I,l}^{(s_I)}(t)| > \bar{\epsilon}_{I,l}^{(s_I)}(t)$ must be satisfied. This translates to the following further inequality

$$\begin{aligned}\left|a_{\mathbf{s},I}^{\top}(t)\int_{T_d}^{t} e^{A_{\mathbf{s}}\tau}W_{\mathbf{s}}\Delta_{\mathbf{s},l}\phi_{\mathbf{s},\gamma}(\tau)d\tau\right| \geq &\,\bar{\epsilon}_{I,l}^{(s_I)}(t)+\left|a_{\mathbf{s},I}^{\top}(t)\right|\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\| W_{\mathbf{s}}\bar{\chi}_{\mathbf{s}}(\tau)d\tau \\ &+\left|a_{\mathbf{s},I}^{\top}(t)\right|\int_{T_d}^{t}\left\|e^{A_{\mathbf{s}}\tau}\right\|\|\Lambda_{\mathbf{s}}-A_{\mathbf{s}}\|\,\bar{\xi}_{\mathbf{s}}(\tau)d\tau + \left|a_{\mathbf{s},I}^{\top}(t)\right|\left\|e^{A_{\mathbf{s}}T_d}\right\|(\bar{\epsilon}_{\mathbf{s},l}(T_d)+\bar{\xi}_{\mathbf{s}}(T_d)) \\ &+\bar{\xi}_I^{(s_I)}(t)\end{aligned}$$

which is implied by the inequality in the hypothesis of the present theorem. Since the inequality holds for every fault function of $\mathcal{F}_I$ but the γ–th, then this fault is locally isolated in the sense of Definition 6.4.4. ■

8.4.2 Global fault isolation logic

The global isolation logic is analogous to the one presented in Section 6.4.3 concerning the discrete-time context. As previously pointed out, in the present DFDI setting a distinction has to be made on the way local and distributed faults are isolated. If a fault is local, then it is sufficient that

the corresponding LFD excludes every but that fault to declare it isolated. However, for distributed faults the isolation needs that all the LFDs, in the influence set of that fault [5], exclude all other faults and therefore the presence of the Global Fault Diagnoser is required.

[5]The fault influence set was introduced in Def. 6.4.1.

Chapter 9

The Input-Output Discrete-time case

In this and in the following chapter, we will consider the case of not completely measurable state. We will face first the detection and isolation problem in the discrete-time context (see [128] for the detection problem) and then in a continuous-time framework.

9.1 Problem formulation

Let us consider a multi-input multi-output uncertain nonlinear system, referred to as *monolithic system*, described by the following discrete-time dynamic equations:

$$\mathscr{S}: \begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{\eta}_x(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \qquad\qquad + \beta(t - T_0)\boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{\eta}_y(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \end{cases} \tag{9.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^p$ are the state, the control input and the measured output vectors respectively, the matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and the vector field $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ represent the nominal healthy dynamics, $\boldsymbol{C} \in \mathbb{R}^{p \times n}$ is the nominal output equation, $\boldsymbol{\eta}_x$ and $\boldsymbol{\eta}_y$ are the uncertainties in the state and in the output equations. The term $\beta(t - T_0)\boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ denotes the changes in the system dynamics due to the occurrence of a fault. $\beta(t - T_0)$, characterizing the time profile of the fault, is defined in Eq. (6.2). The following assumptions are needed.

Assumption 9.1.1: At time $t = 0$ no faults act on the system. Moreover, the state variables $\boldsymbol{x}(t)$ and control variables $\boldsymbol{u}(t)$ remain bounded before and after the occurrence of a fault, i.e., there exist some bounded regions $\mathcal{R} = \mathcal{R}^x \times \mathcal{R}^u \subset \mathbb{R}^n \times \mathbb{R}^m$, such that $(\boldsymbol{x}(t), \boldsymbol{u}(t)) \in \mathcal{R}^x \times \mathcal{R}^u$, $\forall t \geq 0$.

Assumption 9.1.2: The fault–evolution rate parameter b is unknown, but it is lower bounded by a known constant $\bar{b}$, so that $1 < \bar{b} < b$.

Assumption 9.1.3: The modeling and measuring uncertainty terms represented by the vectors $\boldsymbol{\eta}_x$ and $\boldsymbol{\eta}_y$ are unstructured and possibly unknown nonlinear functions of $\boldsymbol{x}$, $\boldsymbol{u}$, and t, but are bounded by some positive, known and bounded functions $\bar{\boldsymbol{\eta}}_x$ and $\bar{\boldsymbol{\eta}}_y$, i.e.,

$$\left|\boldsymbol{\eta}_x^{(h)}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\right| \leq \bar{\boldsymbol{\eta}}_x^{(h)}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$$

and

$$\left|\boldsymbol{\eta}_y^{(k)}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\right| \leq \bar{\boldsymbol{\eta}}_y^{(k)}(\boldsymbol{x}(t), \boldsymbol{u}(t), t),$$

for every h-th and k-th component of the vector, with $h = 1, \ldots, n$, $k = 1, \ldots, p$, for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{R}^x$ and for all t.

Assumption 9.1.3 is required for the analysis but, in practical situations, if some a-priori knowledge on healthy and faulty behavior is available, these assumptions do not cause a significant loss of generality.

As before, we consider the decomposition of the monolithic system $\mathscr{S}$ into N subsystems $\mathscr{S}_I, I = 1, \ldots, N$. As in the case of completely measurable state, the overlapping of certain states $x^{(s)}$ is allowed, but it is worth noting that here, in the Input-Output case, for the sake of generality, we continue considering the decomposition of the states graph, instead of a decomposition made only with respect to the output variables. More specifically, we are concerned with a scenario in which some subsystems may have common state variables, but may differ in their output variables. For example, consider the case of a subsystem where the position of a rigid mechanical body is estimated by measuring its acceleration, while in another subsystem the same position is estimated by measuring its speed: both subsystems share the body position state variable, but they have no common output.

After decomposing the monolithic system (9.1), the I-th subsystem $\mathscr{S}_I$ dynamics can be described by:

$$\begin{aligned} x_I(t+1) =& A_I x_I(t) + f_I(x_I(t), u_I(t)) + g_I(C_I x_I(t), \\ & u_I(t), z_I(t)) + \beta(t - t_0)\phi_I(x_I(t), z_I(t), u_I(t)) \\ y_I(t) =& C_I x_I(t) + \eta_{y,I}(x_I(t), u_I(t), t), \end{aligned} \tag{9.2}$$

where $x_I \in \mathbb{R}^{n_I}$, $u_I \in \mathbb{R}^{m_I}$ and $y_I \in \mathbb{R}^{p_I}$ are the local state, the local control input, and the local measured output vectors respectively, $z_I \in \mathbb{R}^{q_I}$ is the vector of the interconnection variables, which are the neighbor subsystems nodes having a connection with the elements of I. The term $g_I : \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ represents the interconnection function where the effects of the local modeling uncertainty term $\eta_{x,I}$ have been incorporated. The following assumption is needed.

Assumption 9.1.4: The decomposition of the monolithic system (9.1) is such that z_I is made of measured variables only.

In this way, it turns out that all the arguments of the interconnection g_I are known: Assumption 3 is needed in order to allow the learning of the interconnection function. This is a key difference between input–output case and the full–state case. Although this assumption is restrictive, there exist some physical systems that satisfy it: an example may be given by an electric distribution network, where we measure power flows in and out different subsystems. The matrix $A_I \in \mathbb{R}^{n_I \times n_I}$ and the vector field $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ represent the local nominal healthy dynamics, $C_I \in \mathbb{R}^{p_I \times n_I}$ is the nominal local output matrix. $\eta_{y,I}$ is the uncertainty function in the local output equation and includes the measurement error; $\phi_I : \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ is the local fault function. Finally, the following further assumptions are required.

Assumption 9.1.5: The fundamental graph ([96]) is the same before and after the fault event.

This means that we assume that the fault function is such that it does not cause a change to the system structure, but it is however possible for a fault to remove some of the interconnections: this formally corresponds to set some g_I function to zero.

Assumption 9.1.6: (A_I, C_I) is an observable pair.

Assumption 9.1.7: The interconnection function g_I is an unstructured and uncertain nonlinear function, whose k-th component is bounded by some known and bounded function, i.e.,

$$\left| g_I^{(k)}(C_I x_I(t), z_I(t), u_I(t)) \right| \leq \bar{g}_I^{(k)}(C_I x_I(t), z_I(t), u_I(t)),$$

for all $I = 1, \ldots, N$ and for all $(\boldsymbol{x}(t), \boldsymbol{u}(t)) \in \mathcal{R}^x \times \mathcal{R}^u$.

9.2 Distributed Fault Detection Architecture

Similarly as before, the proposed Distributed Fault Detection (FD) architecture is made of two layers: the physical system $\mathscr{S}$, decomposed into N subsystems $\mathscr{S}_I$, and the detection architecture, that is decomposed as well into N entities $\mathscr{L}_I$, the Local Fault Diagnosers (LFD). Each LFD is devoted to monitor exactly one subsystem, by taking local measurements and by communicating only with neighboring LFDs. For detection purposes, each LFD is equipped with a non-linear adaptive estimator, which estimates both the local state x_I and the local output y_I, with $I = 1, ..., N$. The difference between the estimated output $\hat{y}_I$ and the measurements y_I is the output estimation error $\epsilon_{y,I}(t) \triangleq y_I(t) - \hat{y}_I(t)$, which plays the role of a residual and will be compared, component by component, to a suitable threshold

$\bar{\epsilon}_{y,I}(t) \in \mathbb{R}^{p_I}$. The following

$$\left|\epsilon_{y,I}^{(k)}(t)\right| \leq \bar{\epsilon}_{y,I}^{(k)}(t), \forall k = 1, ..., p_I \tag{9.3}$$

is a necessary (but generally not sufficient) condition for the fault-free hypothesis $\mathcal{H}_I$: "The system $\mathscr{S}_I$ is healthy". If the condition is violated at some time instant t, then the hypothesis $\mathcal{H}_I$ is falsified.

Definition 9.2.1: The local fault detection time is defined as $T_{d,I} = \min\left\{t : \exists k, k \in 1, ..., p_I, \left|\epsilon_{y,I}^{(k)}(t)\right| > \bar{\epsilon}_{y,I}^{(k)}(t)\right\}$.

The local FDAE estimation, in the case of non-shared state variables, can be computed as:

$$\begin{cases} \hat{x}_I(t+1) = A_I\hat{x}_I(t) + f_I(\hat{x}_I(t), u_I(t)) \\ \qquad\qquad + \hat{g}_I(y_I(t), u_I(t), v_I(t), \hat{\vartheta}_I) + L_I(y_I(t) - \hat{y}_I(t)) \ , \\ \quad \hat{y}_I(t) = C_I\hat{x}_I(t) \end{cases} \tag{9.4}$$

where where L_I is the local output error gain, $\hat{g}_I$ is the output of an adaptive approximator designed to learn the unknown interconnection function g_I and $\hat{\vartheta}_I \in \hat{\Theta}_I$ denotes its adjustable parameters vector. Due to the uncertain output measurements, it follows that, instead of receiving the actual interconnection vector z_I, each LFD receives from its neighbors the vector $v_I(t) = z_I(t) + \varsigma_I(t)$, where $\varsigma_I(t)$ is made with the components of $\eta_{y,J}$ that affect the relevant components of the neighboring subsystems measurements y_J. In the case of variables $x^{(s)}$ shared among more than one LFDs, we use a deterministic consensus protocol defined on a generic communication graph $\mathcal{G}_s \triangleq (\mathcal{O}_s, \mathcal{E}_s)$, whose nodes are the LFDs in the overlap set $\mathcal{O}_s$ of $x^{(s)}$:

$$\begin{aligned} \hat{x}_I^{(s_I)}(t+1) = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[& A_J^{(s_J)}\hat{x}_J(t) + f_J^{(s_J)}(\hat{x}_J(t), u_J(t)) \\ & + \hat{g}_J^{(s_J)}(y_J(t), u_J(t), v_J(t), \hat{\vartheta}_J) + L_J^{(s_J)}(y_J(t) - \hat{y}_J(t)) \Big] \end{aligned} \tag{9.5}$$

where the terms $W_s^{(I,J)}$ are the components of a doubly stochastic weighted adjacency matrix, as for instance the Metropolis matrix. It is important to note that, in order to implement (9.5), the I-th LFD does not need the information about the expressions of $A_J^{(s_J)}$, $f_J^{(s_J)}$, $\hat{g}_J^{(s_J)}$ and of $L_J^{(s_J)}$; instead, it is sufficient that each LFD computes locally the term $A_J^{(s_J)}\hat{x}_J(t) + f_J^{(s_J)}(t) + \hat{g}_J^{(s_J)}(t) + L_J^{(s_J)}(y_J(t) - \hat{y}_J(t))$ and communicates it to other LFDs according to the communication graph $\mathcal{G}_s$.

9.3 Analysis of the FDAE estimation error

We now analyze the dynamics of the FDAE estimation errors before the occurrence of a fault. In the non-shared case, the i-th state estimation error component is:

$$\begin{aligned}\epsilon_{x,I}^{(i)}(t+1) = & A_I^{(i)} x_I(t) + f_I^{(i)}(x_I(t), u_I(t)) + g_I^{(i)}(C_I x_I(t), u_I(t), z_I(t)) \\ & - A_I^{(i)} \hat{x}_I(t) - f_I^{(i)}(\hat{x}_I(t), u_I(t)) - \hat{g}_I^{(i)}(y_I(t), u_I(t), v_I(t), \hat{\vartheta}_I) \\ & - L_I^{(i)}(y_I(t) - \hat{y}_I(t)) - L_I^{(i)} \eta_{y,I}(t) \\ = & A_{0,I}^{(i)} \epsilon_{x,I}(t) + \Delta f_I^{(i)}(t) + \Delta g_I^{(i)}(t) - L_I^{(i)} \eta_{y,I}(t),\end{aligned} \tag{9.6}$$

where $A_{0,I} \triangleq A_I - L_I C_I$ is a stable matrix (thanks to Assumption 9.1.6),

$$\Delta f_I^{(i)}(t) \triangleq f_I^{(i)}(x_I(t), u_I(t)) - f_I^{(i)}(\hat{x}_I(t), u_I(t))$$

and

$$\Delta g_I^{(i)}(t) \triangleq g_I^{(i)}(C_I x_I(t), u_I(t), z_I(t)) - \hat{g}_I^{(i)}(y_I(t), u_I(t), v_I(t), \hat{\vartheta}_I).$$

We denote with $A^{(i)}$ the i-th row of the matrix A.

In the case of shared variables, the dynamics of the LFD state estimation error component can be written as:

$$\begin{aligned}\epsilon_{x,I}^{(s_I)}(t+1) &= x_I^{(s_I)}(t+1) - \hat{x}_I^{(s_I)}(t+1) \\ &= A_I^{(s_I)} x_I(t) + f_I^{(s_I)}(x_I(t), u_I(t)) + g_I^{(s_I)}(C_I x_I(t), u_I(t), z_I(t)) \\ &- \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[A_J^{(s_J)} \hat{x}_J(t) + f_J^{(s_J)}(\hat{x}_J(t), u_J(t)) + \hat{g}_J^{(s_J)}(y_J(t), u_J(t), v_J(t), \hat{\vartheta}_J) \\ &\qquad + L_J^{(s_J)}(y_J(t) - \hat{y}_J(t)) \Big] .\end{aligned}$$

Since by assumption it holds $\sum_{J \in \mathcal{O}_s} W_s^{(I,J)} = 1$ and due to the way the system decomposition is defined, it is possible to rewrite the state estimation error component as:

$$\begin{aligned}\epsilon_{x,I}^{(s_I)}(t+1) = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[& A_{0,J}^{(s_J)} \epsilon_{x,J}(t) + f_J^{(s_J)}(x_J(t), u_J(t)) \\ & - f_J^{(s_J)}(\hat{x}_J(t), u_J(t)) + g_J^{(s_J)}(C_J x_J(t), u_J(t), z_J(t)) \\ & - \hat{g}_J^{(s_J)}(y_J(t), u_J(t), v_J(t), \hat{\vartheta}_J) - L_J^{(s_J)} \eta_{y,J}(t) \Big] .\end{aligned} \tag{9.7}$$

Summing up, the dynamics of a component of the state estimation error can be written as in (9.6) in the case of non-shared state variables, while,

for shared state variables, we have:

$$\epsilon_{x,I}^{(s_I)}(t+1) = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[A_{0,J}^{(s_J)} \epsilon_{x,J}(t) + \Delta f_J^{(s_J)}(t) + \Delta g_J^{(s_J)}(t) - L_J^{(s_J)} \eta_{y,J}(t) \Big]. \quad (9.8)$$

We now introduce a general formulation of the state error equation for analysis purpose. To this end we define the extended state estimation error vector $\epsilon_{x,E} \in \mathbb{R}^{n_E \times 1}$, with $n_E = \sum_{J=1}^{N} n_J$, that is a column vector collecting the state estimation error vectors of the N sub-systems: $\epsilon_{x,E}(t) \triangleq \text{col}\,(\epsilon_{x,J}(t) : J = 1, ..., N)$. The dynamics of $\epsilon_{x,E}(t)$ are:

$$\epsilon_{x,E}(t+1) = W \left[A_{0,E} \epsilon_{x,E}(t) + \Delta f_E(t) + \Delta g_E(t) - L_E \eta_{yE}(t) \right] \quad (9.9)$$

where W is a $N \times N$ block matrix

$$W \triangleq \begin{bmatrix} W_{1,1} & \dots & W_{1,N} \\ \dots & \dots & \dots \\ W_{N,1} & \dots & W_{N,N} \end{bmatrix},$$

such that each block $W_{I,J}$, with $J = 1, ..., N$ and $I = 1, ..., N$ collects the consensus weights of the subsystem I with regard to the subsystem J. The diagonal blocks $W_{I,I}$ are square diagonal matrices in $\mathbb{R}^{n_I \times n_I}$, whose s_I–th diagonal element, with $s_I = 1, ..., n_I$, is equal to the weight $W_s^{(I,I)}$ defined in Eq. (8.7) if $x_I^{(s_I)}$ is a shared variable, and is equal to 1 otherwise. The matrices $W_{I,J} \in \mathbb{R}^{n_I \times n_J}$, with $J \neq I$, have non-null elements only in positions (s_I, s_J) corresponding to shared variables x_s, and here they take the value of the consensus weight $W_s^{(I,J)}$. This results in W being a symmetrical and doubly–stochastic $n_E \times n_E$ matrix. $A_{0,E}$ is a $N \times N$ diagonal block matrix:

$$A_{0,E} \triangleq \begin{bmatrix} A_{0,1} & 0 & 0 & 0 \\ 0 & A_{0,2} & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & A_{0,N} \end{bmatrix},$$

where the generic block is $A_{0,J} = A_J - L_J C_J \in \mathbb{R}^{n_J \times n_J}$, for $J = 1, .., N$, resulting in $A_{0,E}$ being a sparse $n_E \times n_E$ matrix. $\Delta f_E(t)$ is a $n_E \times 1$ matrix, collecting the values $\Delta f_J^{(s_J)}(t)$, for each $s_J = 1, ..., n_J$ and for every $J = 1, ..., N$. $\Delta g_E(t)$ is defined in an analogous way as $\Delta f_E(t)$. Furthermore, $L_E \triangleq \text{blkdiag}(L_J : J = 1, .., N)$ is a $N \times N$ diagonal block matrix with dimension $n_E \times p_E$, where $p_E \triangleq \sum_{J=1}^{N} p_J$, while $\eta_{y,E}(t)$ is a $p_E \times 1$ column vector collecting the uncertainty terms of the N subsystems: $\eta_{y,E} \triangleq \text{col}\,(\eta_{y,I} : J = 1, ..., N)$. The state estimation error solution can be

written as:

$$\epsilon_{x,E}(t) = \sum_{h=0}^{t-1} (WA_{0,E})^{t-1-h} \left[W\Delta f_E(h) + W\Delta g_E(h) - WL_E\eta_{y,E}(h) \right] + (WA_{0,E})^t \epsilon_{x,E}(0) \quad (9.10)$$

The extended output estimation error is then defined as:

$$\epsilon_{y,E}(t) \triangleq C_E\epsilon_{x,E}(t) + \eta_{y,E}(t) \quad (9.11)$$

where $C_E \triangleq \text{blkdiag}(C_J : J = 1, .., N)$ is a $N \times N$ diagonal block matrix, with dimension $p_E \times n_E$. From (9.9), (9.11) and the definition of C_E, the following learning law for the adjustable parameter vector $\hat{\vartheta}_I$ of the adaptive approximator $\hat{g}_I$, $I \in 1, \ldots, N$ can be derived:

$$\hat{\vartheta}_I(t+1) = P_{\hat{\Theta}_I} \left[\hat{\vartheta}_I(t) + \gamma_I(t) H_I^\top(t) W_{I,I}^\top C_I^\top \epsilon_{y,I}(t+1) \right] \quad (9.12)$$

$$H_I^\top(t) = \partial \hat{g}_I(t) / \partial \hat{\vartheta}_I$$

$$\gamma_I(t) = \frac{\mu_I}{\varepsilon_I + \left\| H_I^\top(t) W_{I,I}^\top C_I^\top \right\|_F^2},$$

where $P_{\hat{\Theta}_I}$ is a projection operator restricting $\hat{\vartheta}_I$ within $\hat{\Theta}_I$ [183], $\| \cdot \|_F$ denotes the Frobenius norm and $\varepsilon_I > 0$, $0 < \mu_I < 2$ are design constants that guarantee the stability of the learning law [194, 195, 196, 183, 197, 198]. The component-wise output estimation error can be written as: $\epsilon_{y,E}^{(k)}(t) = C_E^{(k)}\epsilon_{x,E}(t) + \eta_{y,E}^{(k)}(t)$, for all $k = 1, ..., p_E$. Since each row of C_E, because of the way the matrix was defined, presents non-null values only in correspondence to the state components of a single subsystem, it is possible to write: $\epsilon_{y,I}^{(k)}(t) = C_I^{(k)}\epsilon_{x,I}(t) + \eta_{y,I}^{(k)}(t)$, for all $k = 1, ..., p_I$, and for each subsystem $\mathscr{S}_I$, $I \in 1, \ldots, N$. In the general form, the component-wise output estimation error can be bounded by the following threshold, that can be computed in a distributed way:

$$\begin{aligned} \left| \epsilon_{y,E}^{(k)}(t) \right| &\leq \left| C_E^{(k)} \epsilon_{x,E}(t) \right| + \left| \eta_{y,E}^{(k)}(t) \right| \\ &\leq \Big| C_E^{(k)} \Big\{ \sum_{h=0}^{t-1} (WA_{0,E})^{t-1-h} W \Big[\Delta f_E(h) + \Delta g_E(h) - L_E \eta_{y,E}(h) \Big] \\ &\quad + (WA_{0,E})^t \epsilon_{x,E}(0) \Big\} \Big| + \bar{\eta}_{y,E}^{(k)}(t) \end{aligned}$$

$$
\begin{aligned}
&\leq \left|C_E^{(k)}\right| \Big\{ \sum_{h=0}^{t-1} \left\| (WA_{0,E})^{t-1-h} \right\| \left| W \Big[\Delta f_E(h) + \Delta g_E(h) - L_E \eta_{y,E}(h) \Big] \right| \\
&+ \left\| (WA_{0,E})^{t} \right\| |\epsilon_{x,E}(0)| \Big\} \Big| + \bar{\eta}_{y,E}^{(k)}(t) \\
&\leq \left|C_E^{(k)}\right| \Big\{ \sum_{h=0}^{t-1} \alpha \delta^{t-1-h} W \left[\bar{\Delta} f_E(h) + \bar{\Delta} g_E(h) + |L_E| \, \bar{\eta}_{y,E}(h) \right] \\
&+ \alpha \delta^t \bar{\epsilon}_{x,E}(0) \Big\} + \bar{\eta}_{y,E}^{(k)}(t) \\
&\triangleq \bar{\epsilon}_{y,E}^{(k)}(t)
\end{aligned}
\tag{9.13}
$$

where we denote with $|A|$ the element by element absolute value of the matrix A; α and δ, analogously to [199], are two constants such that $\left\|(WA_{0,E})^t\right\| \leq \alpha\delta^t \leq \|WA_{0,E}\|^t$, $\alpha > 0$, $0 < \delta \leq 1$. Furthermore,

$$
\bar{\Delta} f_E^{(s)}(t) = \max_{x^{(s)} \in R^{x^{(s)}}} \left\{ \left| \Delta f_E^{(s)}(t) \right| \right\},
$$

$$
\bar{\epsilon}_{x,E}^{(s)}(0) = \max_{x^{(s)} \in R^{x^{(s)}}} \left\{ \left| x^{(s)} - \hat{x}^{(s)}(0) \right| \right\},
$$

for every $s = 1, ..., n_E$. As regards $\bar{\Delta} g_E(t)$, some considerations are expressed in the previous chapters: Δg_I can be upper bounded by $\bar{\Delta} g_I(t) \triangleq \|H_{I,0}\| \, \kappa_{I,0}(\hat{\vartheta}_{I,0}) + \bar{\nu}_I(t) + \max_{\eta_{yI}} \max_{\varsigma_I} |\Delta \hat{g}_I(t)|$, where $\kappa_{I,0}$ is such that $\kappa_{I,0}(\hat{\vartheta}_{I,0}) \geq \left\| \tilde{\vartheta}_{I,0} \right\|$. In fact, by defining the parameter estimation error $\tilde{\vartheta}_{,0}I \triangleq \vartheta_{I,0}^{\hat{*}} - \hat{\vartheta}_{I,0}$ and the function

$$
\Delta \hat{g}_I \triangleq \hat{g}_I(C_I x_I, z_I, u_I, \hat{\vartheta}_{I,0}) - \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_{I,0}),
$$

can be written as

$$
\Delta g_I(t) = H_I \tilde{\vartheta}_{I,0} + \nu_I(t) + \Delta \hat{g}_I(t),
$$

where $\nu_I(t) \triangleq g_I(C_I x_I, z_I, u_I) - \hat{g}_I(C_I x_I, z_I, u_I, \hat{\vartheta}_{I,0}^*)$ is the *Minimum Functional Approximation Error*, with

$$
\hat{\vartheta}_I^* \triangleq \arg \min_{\hat{\vartheta}_I \in \Theta_I} \sup_{x_I, z_I, u_I} \left\| g_I(C_I x_I, z_I, u_I) - \hat{g}_I(C_I x_I, z_I, u_I, \hat{\vartheta}_I \right\|
$$

is the optimal weight vector. The extended upper bound $\bar{\Delta} g_E(t)$ simply collects the upper bounds of the N subsystems.

The threshold in Eq. (9.13) guarantees that no false-positive alarms will be issued until T_0 because of the uncertainties. This, of course and in rough terms, comes at the cost of the impossibility of detecting faults "hidden by the uncertainties in the system dynamics". This is formalized in

the following section in which a distributed detectability sufficient condition will be derived.

9.4 Detectability Sufficient Conditions

Let us assume that at time $t = T_0$ a fault $\boldsymbol{\phi}$ occurs in the monolithic system. ϕ_E denotes the extended fault function vector collecting the N subsystems fault functions. After the occurrence of the fault, for $t > T_0$, the state estimation error dynamics becomes

$$\epsilon_{x,E}(t+1) = W\left[A_{0,E}\epsilon_{x,E}(t) + \Delta f_E(t) + \Delta g_E(t) - L_E\eta_{y,E}(t)\right] + (1 - b^{-(t-T_0)})\phi_E(t) \quad (9.14)$$

and the output estimation error equation for the k-th component is:

$$\begin{aligned}\epsilon_{y,E}^{(k)}(t) =& C_E^{(k)}\epsilon_{x,E}(t) + \eta_{y,E}^{(k)}(t)\\ =& C_E^{(k)}\Bigg\{\sum_{h=0}^{t-1}(WA_{0,E})^{t-1-h}\Big[W\Delta f_E(h) + W\Delta g_E(h) - WL_E\eta_{y,E}(h)\\ &+ (1 - b^{-(h-T_0)})\phi_E(h)\Big] + (WA_{0,E})^t\epsilon_{x,E}(0)\Bigg\} + \eta_{y,E}^{(k)}(t).\end{aligned} \quad (9.15)$$

Now, we are able to state and prove a sufficient condition for distributed fault detectability.

Theorem 9.4.1 (Fault Detectability): If there exists a time instant $t_1 > T_0$ such that the fault ϕ_E satisfies the inequality

$$\left|\sum_{h=T_0}^{t_1-1} C_E^{(k)}(WA_{0,E})^{t_1-1-h}(1 - b^{-(h-T_0)})\phi_E(h)\right| > 2\bar{\epsilon}_{y,E}^{(k)}(t_1) \quad (9.16)$$

for at least one component $k \in \{1, ..., p_E\}$, then the fault will be detected at time t_1, that is $\left|\epsilon_{y,E}^{(k)}(t_1)\right| > \bar{\epsilon}_{y,E}^{(k)}(t_1)$.

Proof: At time instant $t_1 > T_0$, the output estimation error can be written as:

$$\begin{aligned}\epsilon_{y,E}^{(k)}(t_1) =& \sum_{h=0}^{t_1-1} C_E^{(k)}(WA_{0,E})^{t_1-1-h}W\left[\Delta f_E(h) + \Delta g_E(h) - L_E\eta_{y,E}(h)\right]\\ &+ C_E^{(k)}(WA_{0,E})^{t_1}\epsilon_{x,E}(0) + \eta_{y,E}^{(k)}(t_1)\\ &+ \sum_{h=T_0}^{t_1-1} C_E^{(k)}(WA_{0,E})^{t_1-1-h}(1 - b^{-(h-T_0)})\phi_E(h).\end{aligned}$$

Using the triangle inequality we obtain:

$$\left|\epsilon_{y,E}^{(k)}(t_1)\right| \geq -\left|\sum_{h=0}^{t_1-1} C_E^{(k)}(WA_{0,E})^{t_1-1-h}W\left[\Delta f_E(h)+\Delta g_E(h)-L_E\eta_{y,E}(h)\right]\right.$$
$$\left.+\,C_E^{(k)}(WA_{0,E})^{t_1}\epsilon_{x,E}(0)+\eta_{y,E}^{(k)}(t_1)\right|$$
$$+\left|\sum_{h=T_0}^{t_1-1} C_E^{(k)}(WA_{0,E})^{t_1-1-h}(1-b^{-(h-T_0)})\phi_E(h)\right|.$$

By recalling how the threshold was defined (Eq. 9.13), it is easy to see that the following inequality is implied:

$$\left|\epsilon_{y,E}^{(k)}(t_1)\right| \geq -\bar{\epsilon}_{y,E}^{(k)}(t_1)+\left|\sum_{h=T_0}^{t_1-1} C_E^{(k)}(WA_{0,E})^{t_1-1-h}(1-b^{-(h-T_0)})\phi_E(h)\right|.$$

In this way the fault detection condition $\left|\epsilon_{y,E}^{(k)}(t_1)\right| > \bar{\epsilon}_{y,E}^{(k)}(t_1)$ is implied by the theorem hypothesis. ■

Theorem 9.4.1 represents a sufficient condition for the off-line characterization, in a non-closed form, of a class of faults that can be detected by the proposed FD methodology.

9.5 Distributed Isolation Architecture

After a fault is detected by any of the N LFDs, the *Global Fault Diagnosers* (GFD) receives the corresponding local fault decision and switches each LFD from fault detection to fault isolation operating mode.

For isolation purposes once again we assume that the fault function $\boldsymbol{\phi}$ may belong to a known global fault set $\mathcal{F}$ or be unknown:

$$\mathcal{F} \triangleq \{\boldsymbol{\phi_1}(\boldsymbol{C}x,\boldsymbol{u}),\ldots,\boldsymbol{\phi_{N_\mathcal{F}}}(\boldsymbol{C}x,\boldsymbol{u})\}.$$

It is possible that not all the subsystems are affected by a given fault function $\boldsymbol{\phi_l}$, but only those contained in the corresponding *fault influence set* $\mathcal{U}_{\boldsymbol{l}}$. As a consequence, a *local fault set* $\mathcal{F}_I$ can be defined for each subsystem $\mathscr{S}_I$, collecting the local fault functions $\phi_{I,l}$ such that $I \in \mathcal{U}_{\boldsymbol{l}}$:

$$\mathcal{F}_I \triangleq \{\phi_{I,1}(C_Ix_I,z_I,u_I),\ldots,\phi_{I,N_{\mathcal{F}_I}}(C_Ix_I,z_I,u_I)\}.$$

It is worth noting that, thanks to Assumption (9.1.5), the local fault functions depend only on the local variables x_I, z_I and u_I. Besides the FDAE, each LFD uses other $N_{\mathcal{F}_I}$ estimators, the FIEs (Fault Isolation Estimators), one for each fault in the local fault set $\mathcal{F}_I$. In the isolation mode, each LFD

activates its set of FIEs in order to locally isolate the fault that is acting on the subsystem I. The GFD, which is assumed to know both the global fault set $\mathcal{F}$ and the fault influence sets of all the global fault functions, will receive the local fault decisions d_I^{FD} and will determine which one of the faults, if any, in the global set $\mathcal{F}$ affects the system $\mathscr{S}$, following the *Generalized Observer Scheme* ([17]) presented in Section 6.4: in this way it will be able to take a correct global fault decision d^{FD}.

For each LFD $\mathscr{L}_I$, with $I = 1, ..., N$, the generic l–th FIE, with $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, monitors the corresponding fault function $\phi_{I,l}$, belonging to the local fault set $\mathcal{F}_I$. We assume that each fault function in $\mathcal{F}_I$ can be expressed as:

$$\phi_{I,l}(C_I x_I(t), z_I(t), u_I(t)) = \Big[(\vartheta_{I,l,1})^\top H_{I,l,1}(C_I x_I(t), z_I(t), u_I(t)), \\ \ldots, (\vartheta_{I,l,n_I})^\top H_{I,l,n_I}(C_I x_I(t), z_I(t), u_I(t))\Big]^\top , \quad (9.17)$$

where $H_{I,l,k} : \mathbb{R}^{p_I} \times \mathbb{R}^{q_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{q_{I,l,k}}$, with $k \in \{1, \ldots, n_I\}$ and $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, are the known functions describing the functional structure of the fault and $\vartheta_{I,l,k} \in \Theta_{I,l,k} \subset \mathbb{R}^{q_{I,l,k}}$ are the unknown parameter vectors providing its "magnitude", where the parameter domains $\Theta_{I,l,k}$ are assumed to be origin–centered hyper–spheres with radius $M_{\Theta_{I,l,k}}$, without much loss of generality.

After $t = T_d$, the generic l–th FIE estimator is activated for isolation purposes and monitors its subsystem, computing a *local state estimate* $\hat{x}_{I,l}$ and a *local output estimate* $\hat{y}_{I,l}$. The difference between the estimate $\hat{y}_{I,l}$ and the measurements y_I is the *estimation error* $\epsilon_{y,I,l} \triangleq y_I - \hat{y}_{I,l}$, used as a residual and compared, component by component, to an appropriate *isolation threshold* $\bar{\epsilon}_{y,I,l} \in \mathbb{R}_+^{p_I}$. After the fault $\phi(t)$ has occurred, the s_I–th component of the I–th local state equation becomes

$$x_I^{(s_I)}(t+1) = A_I^{(s_I)} x_I(t) + f_I^{(s_I)}(x_I(t), u_I(t)) + g_I^{(s_I)}(C_I x_I(t), z_I(t), u_I(t)) \\ + \beta(t - T_0)\phi^{(s)}(Cx(t), u(t)) .$$

The l-th FIE computes a local estimate, that, in the case of non-shared state variables, can be defined as:

$$\begin{cases} \hat{x}_{I,l}(t+1) = A_I \hat{x}_{I,l}(t) + f_I(\hat{x}_{I,l}(t), u_I(t)) + \hat{g}_I(y_I(t), v_I(t), u_I(t), \hat{\vartheta}_{I,0}) \\ \qquad\qquad + L_I(y_I(t) - \hat{y}_{I,l}(t)) + \hat{\phi}_{I,l}(y_I(t), v_I(t), u_I(t), \hat{\vartheta}_{I,l}) \\ \qquad \hat{y}_{I,l}(t) = C_I \hat{x}_{I,l}(t), \end{cases} \quad (9.18)$$

where $L_I \in \mathcal{R}^{n_i \times p_I}$ is the local output error gain, $\hat{g}_I$ is the output of the FDAE approximator designed to match the unknown interconnection function g_I and $\hat{\vartheta}_{I,0}$ represent its parameters, that remain fixed after fault de-

tection,

$$\hat{\phi}_{I,l}^{(s_I)}(y_I(t), v_I(t), u_I(t), \hat{\vartheta}_{I,l}) \triangleq (\hat{\vartheta}_{I,l,s_I})^\top H_{I,l,s_I}(y_I(t), v_I(t), u_I(t))$$

is the s_I–th component of a linearly-parameterized function that learns the structure of the l–th fault function $\phi_{I,l}$, where the vector $\hat{\vartheta}_{I,l} \triangleq \text{col}(\hat{\vartheta}_{I,l,k}, k \in \{1, \ldots, n_I\})$ contains its adjustable parameters. The learning law will be described in the following.

The dynamics of the l–th FIE estimator for the most general case of a distributed fault can be defined as

$$\begin{aligned}\hat{x}_{I,l}^{(s_I)}(t+1) = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[& A_J^{(s_J)} \hat{x}_{J,l}(t) + f_J^{(s_J)}(\hat{x}_{J,l}(t), u_J(t)) \\ & + \hat{g}_J^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,0}) + L_J^{(s_J)}(y_{J,l}(t) - \hat{y}_J(t)) \\ & + \hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,l}) \Big]. \quad (9.19)\end{aligned}$$

Now, we can compute the FIE state estimation error: the i-th component, in the non-shared case, is

$$\begin{aligned}\epsilon_{x,I,l}^{(i)}(t+1) &= x_I^{(i)}(t+1) - \hat{x}_{I,l}^{(i)}(t+1) \\ &= A_I^{(i)} x_I(t) + f_I^{(i)}(x_I(t), u_I(t)) + g_I^{(i)}(C_I x_I(t), z_I(t), u_I(t)) \\ &\quad - A_I^{(i)} \hat{x}_{I,l}(t) - f_I^{(i)}(\hat{x}_{I,l}(t), u_I(t)) - \hat{g}_I^{(i)}(y_I(t), v_I(t), u_I(t), \hat{\vartheta}_{I,0}) \\ &\quad - L_I^{(i)}(y_I(t) - \hat{y}_{I,l}(t)) - L_I^{(i)} \eta_{y,I}(t) - \hat{\phi}_{J,l}^{(i)}(t) + (1 - b^{-(t-T_0)}) \phi^{(i)}(t) \\ &= A_{0,I}^{(i)} \epsilon_{x,I,l}(t) + \Delta f_I^{(i)}(t) + \Delta g_I^{(i)}(t) - L_I^{(i)} \eta_{y,I}(t) \\ &\quad + (1 - b^{-(t-T_0)}) \phi^{(i)}(t) - \hat{\phi}_{J,l}^{(i)}(t). \quad (9.20)\end{aligned}$$

On the other hand, the dynamics of the LFD state estimation error component for shared variables can be described as:

$$\begin{aligned}\epsilon_{x,I,l}^{(s_I)}(t+1) &= A_I^{(s_I)} x_I(t) + f_I^{(s_I)}(x_I(t), u_I(t)) + g_I^{(s_I)}(C_I x_I(t), u_I(t), z_I(t)) \\ &+ (1 - b^{-(t-T_0)}) \phi^{(s)}(t) - \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[A_J^{(s_J)} \hat{x}_{J,l}(t) + f_J^{(s_J)}(\hat{x}_{J,l}(t), u_J(t)) \\ &\quad + \hat{g}_J^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,0}) + L_J^{(s_J)}(y_J(t) - \hat{y}_{J,l}(t)) + \hat{\phi}_{J,l}^{(s_J)}(t) \Big].\end{aligned}$$

The state estimation error component can be rewritten as:

$$\begin{aligned}\epsilon_{x,I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \Big[& A_{0,J}^{(s_J)} \epsilon_{x,J,l}(t) \\ & + f_J^{(s_J)}(x_J(t), u_J(t)) - f_J^{(s_J)}(\hat{x}_{J,l}(t), u_J(t)) \\ & + g_J^{(s_J)}(C_J x_J(t), u_J(t), z_J(t)) - \hat{g}_J^{(s_J)}(y_J(t), u_J(t), v_J(t), \hat{\vartheta}_{J,0}) \\ & - L_J^{(s_J)} \eta_{y,J}(t) + (1 - b^{-(t-T_0)})\phi^{(s)}(t) - \hat{\phi}_{J,l}^{(s_J)}(t) \Big] . \end{aligned} \tag{9.21}$$

When we consider a matched fault $\phi^{(s)}(t) = \phi_{J,l}^{(s_J)}(x_J(t), z_J(t), u_J(t), \vartheta_{J,l})$, $\forall\, J \in \mathcal{O}_{\boldsymbol{s}}$), the error dynamics are:

$$\begin{aligned}\epsilon_{x,I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \Big[& A_{0,J}^{(s_J)} \epsilon_{x,J,l}(t) + \Delta f_J^{(s_J)}(t) + \Delta g_J^{(s_J)}(t) \\ & - L_J^{(s_J)} \eta_{y,J}(t) + \Delta \phi_{J,l}^{(s_J)} \Big] \end{aligned} \tag{9.22}$$

where

$$\begin{aligned}\Delta\phi_{J,l}^{(s_J)} & \triangleq (1 - b^{-(t-T_0)})\phi^{(s)}(t) - \hat{\phi}_{J,l}^{(s_J)}(t) \\ & = (1 - b^{-(t-T_0)})(H_{J,l,s_J}(t)^\top \vartheta_{J,l,s_J} + \Delta H_{J,l,s_J}^\top \vartheta_{J,l,s_J}) - H_{J,l,s_J}(t)^\top \hat{\vartheta}_{J,l,s_J} \end{aligned} \tag{9.23}$$

with $\Delta H_{J,l,s_J}^\top(t) \triangleq H_{J,l,s_J}(x_J(t), z_J(t), u_J(t)) - H_{J,l,s_J}(y_J(t), v_J(t), u_J(t))$. If we introduce the parameter estimation errors $\tilde{\vartheta}_{J,l,s_J} \triangleq \vartheta_{J,l,s_J} - \hat{\vartheta}_{J,l,s_J}$, it can be rewritten as

$$\begin{aligned}\Delta\phi_{J,l}^{(s_J)} = (1 - b^{-(t-T_0)})H_{J,l,s_J}(t)^\top \tilde{\vartheta}_{J,l,s_J} + (1 - b^{-(t-T_0)})\Delta H_{J,l,s_J}(t)^\top \vartheta_{J,l,s_J} \\ - b^{-(t-T_0)} H_{J,l,s_J}(t)^\top \hat{\vartheta}_{J,l,s_J} \end{aligned}$$

By introducing the general formulation for analysis purposes, we can define the FIE extended state estimation error vector, $\epsilon_{x,E,l} \in \mathbb{R}^{n_E \times 1}$, with $n_E = \sum_{J=1}^N n_J$, which is a column vector that collects the FIE state estimation error vectors of the N sub-systems: $\epsilon_{x,E,l}(t) \triangleq \mathrm{col}\,(\epsilon_{x,J,l}(t) : J = 1, ..., N)$. We can then express the dynamics of the estimation error in the case of a matched fault, as:

$$\epsilon_{x,E,l}(t+1) = W\left[A_{0,E}\, \epsilon_{x,E,l}(t) + \Delta f_E(t) + \Delta g_E(t) - L_E \eta_{yE}(t) + \Delta\phi_{E,l}\right]. \tag{9.24}$$

We can now compute the state estimation error solution:

$$\epsilon_{x,E,l}(t) = \sum_{h=T_d}^{t-1} (WA_{0,E})^{t-1-h} \left[W\Delta f_E(h) + W\Delta g_E(h) - WL_E\eta_{y,E}(h) + W\Delta\phi_{E,l}\right] + (WA_{0,E})^t \epsilon_{x,E,l}(0) \quad (9.25)$$

It is then possible to define the extended output estimation error:

$$\epsilon_{y,E,l}(t) \triangleq C_E\epsilon_{x,E,l}(t) + \eta_{y,E}(t) \quad (9.26)$$

where $C_E \triangleq \text{blkdiag}(C_J : J = 1, .., N)$ is a $N \times N$ diagonal block matrix, with dimension $p_E \times n_E$. From (9.24) and (9.26), we developed the following learning law for the parameter vector $\hat{\vartheta}_{I,l}$ of the fault function adaptive approximator $\hat{\phi}_{I,l}$, for every $I = 1, \ldots, N$, $l = 1, ..., N_{\mathcal{F}_I}$:

$$\hat{\vartheta}_{I,l}(t+1) = P_{\hat{\Theta}_{I,l}}\left[\hat{\vartheta}_{I,l}(t) + \gamma_{I,l}(t)H_{I,l}^\top(t)W_{I,I}^\top C_I^\top \epsilon_{y,I,l}(t+1)\right] \quad (9.27)$$

$$H_{I,l}^\top(t) = \partial\hat{\phi}_{I,l}(t)/\partial\hat{\vartheta}_{I,l},$$

$$\gamma_{I,l}(t) = \frac{\mu_{I,l}}{\varepsilon_{I,l} + \left\|H_{I,l}^\top(t)W_{I,I}^\top C_I^\top\right\|_F^2},$$

where $P_{\hat{\Theta}_{I,l}}$ is the projection operator, restricting $\hat{\vartheta}_{I,l}$ within $\hat{\Theta}_{I,l}$, $\|\cdot\|_F$ represents the Frobenius norm and $\varepsilon_{I,l} > 0$, $0 < \mu_{I,l} < 2$ are constants designed to guarantee the stability of the learning law. It differs from the completely measurable state case due to the presence of the output matrix. In the general form, the output estimation error in the case of a matched fault can be written componentwise as: $\epsilon_{y,E,l}^{(k)}(t) = C_E^{(k)}\epsilon_{x,E,l}(t) + \eta_{y,E}^{(k)}(t)$, for all $k = 1, ..., p_E$. It can be bounded by:

$$\begin{aligned}
&\left|\epsilon_{y,E,l}^{(k)}(t)\right| \leq \left|C_E^{(k)}\epsilon_{x,E,l}(t)\right| + \left|\eta_{y,E}^{(k)}(t)\right| \\
&\leq \left|C_E^{(k)}\Big\{\sum_{h=0}^{t-1}(WA_{0,E})^{t-1-h}W\Big[\Delta f_E(h) + \Delta g_E(h) - L_E\eta_{y,E}(h) + \Delta\phi_{E,l}\Big] \right. \\
&\qquad \left. + (WA_{0,E})^t\epsilon_{x,E,l}(0)\Big\}\right| + \bar{\eta}_{y,E}^{(k)}(t) \\
&\leq \left|C_E^{(k)}\right|\Big\{\sum_{h=0}^{t-1}\left\|(WA_{0,E})^{t-1-h}\right\|\left|W\Big[\Delta f_E(h) + \Delta g_E(h) - L_E\eta_{y,E}(h) + \Delta\phi_{E,l}\Big]\right| \\
&\qquad + \left\|(WA_{0,E})^t\right\|\left|\epsilon_{x,E,l}(0)\right|\Big\} + \bar{\eta}_{y,E}^{(k)}(t)
\end{aligned}$$

In this way, it is possible to define the following threshold:

$$\bar{\epsilon}^{(k)}_{y,E,l}(t) \triangleq \left|C^{(k)}_E\right| \Big\{ \sum_{h=0}^{t-1} \alpha\delta^{t-1-h} W \left[\bar{\Delta} f_E(h) + \bar{\Delta} g_E(h) + |L_E|\, \bar{\eta}_{y,E}(h) + \bar{\Delta}\phi_{E,l} \right] + \alpha\delta^t \bar{\epsilon}_{x,E,l}(0) \Big\} + \bar{\eta}^{(k)}_{y,E}(t) \quad (9.28)$$

where α and δ, analogously to Equation (9.13), are two constants such that $\|(WA_{0,E})^t\| \leq \alpha\delta^t \leq \|WA_{0,E}\|^t$, $\alpha > 0$, $0 < \delta \leq 1$. Besides,

$$\bar{\epsilon}^{(s)}_{x,E,l}(0) = \max_{x^{(s)} \in R^{x^{(s)}}} \left\{ \left| x^{(s)} - \hat{x}^{(s)}_l(0) \right| \right\},$$

for every $s = 1, ..., n_E$, and

$$\bar{\Delta}\phi_{E,l} = \mathrm{col}(\|H_{I,l,s_I}(t)\| \kappa_{I,l,s_I}(\hat{\vartheta}_{I,l,s_I}) + \bar{\Delta} H_{I,l,s_I}(t) \bar{\vartheta}_{I,l,s_I} - \bar{b}^{-(t-T_d)} \|H_{I,l,s_I}(t)\|\, \|\hat{\vartheta}_{I,l,s_I}\|,\ s_I = 1, ..., n_I, I = 1, ..., N)\,.$$

where $\kappa_{I,l}(\hat{\vartheta}_{I,l}) \geq \left\|\tilde{\vartheta}_{I,l}\right\|$.

The threshold (9.28) can be computed in a distributed way, since each row of C_E presents non–null values only in positions corresponding to the state components of a single subsystem, and, because of the way it is defined, guarantees that no matched fault will be excluded due to the presence of uncertainties or to the effect of the parameter estimation error $\tilde{\vartheta}_{I,l}$.

9.5.1 Fault isolability analysis

We now consider the case of a non–matched fault $\phi^{(s_I)}_I(x_I(t), z_I(t), u_I(t)) = \phi^{(s_I)}_{I,\gamma}(x_I(t), z_I(t), u_I(t), \vartheta_{I,\gamma})$, with $\gamma \neq l$. In this case, the dynamics of the shared s_I–component of the estimation error for the l–th FIE of the I–th LFD can be written as

$$\epsilon^{(s_I)}_{x,I,l}(t+1) = \sum_{J \in \mathcal{O}_s} W^{(I,J)}_s [A^{(s_J)}_{0,J} \epsilon_{x,J,l}(t) + \Delta f^{(s_J)}_J(t) + \Delta g^{(s_J)}_J(t) - L^{(s_J)}_J \eta_{y,J}(t) + (1 - b^{-(t-T_0)}) \phi^{(s_I)}_{I,\gamma}(x_I(t), z_I(t), u_I(t), \vartheta_{I,\gamma}) - \hat{\phi}^{(s_J)}_{J,l}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,l})].$$

We have shown before that, in order to study the behavior of the estimation error, it is convenient to consider the vector $\epsilon_{x,E,l}$:

$$\epsilon_{x,E,l}(t+1) = W\left[A_{0,E}\, \epsilon_{x,E,l}(t) + \Delta f_E(t) + \Delta g_E(t) - L_E \eta_{yE}(t) + \Delta_l \phi_{E,\gamma}(t)\right],$$

where the *mismatch vector* is introduced

$$\Delta_l \phi_{E,\gamma}(t) \triangleq \mathrm{col}((1 - b^{-(t-T_0)}) \phi^{(s_I)}_{I,\gamma}(t) - \hat{\phi}^{(s_I)}_{I,l}(t),\ s_I = 1, ..., n_I, I = 1, ..., N).$$

The solution can then be written as

$$\epsilon_{x,E,l}(t) = \sum_{h=0}^{t-1}(WA_{0,E})^{t-1-h}W\left[\Delta f_E(h) + \Delta g_E(h) - L_E\eta_{y,E}(h) + \Delta_l\phi_{E,\gamma}(h)\right] + (WA_{0,E})^t\epsilon_{x,E,l}(0).$$

Then, the output residual can be expressed componentwise by the following equation:

$$\begin{aligned}\epsilon^{(k)}_{y,E,l}(t) &= C_E^{(k)}\epsilon_{x,E,l}(t) + \eta^{(k)}_{y,E}(t) \\ &= C_E^{(k)}\Big\{\sum_{h=0}^{t-1}(WA_{0,E})^{t-1-h}\Big[W\Delta f_E(h) + W\Delta g_E(h) - WL_E\eta_{y,E}(h) \\ &\qquad + W\Delta_l\phi_{E,\gamma}(h)\Big] + (WA_{0,E})^t\epsilon_{x,E,l}(0)\Big\} + \eta^{(k)}_{y,E}(t). \quad (9.29)\end{aligned}$$

At this point, a sufficient condition for fault isolability can be proved.

Theorem 9.5.1 (Fault Isolability): Given a fault $\phi_{I,\gamma} \in \mathcal{F}_I$, if for each $l \in \{1,\dots,N_{\mathcal{F}_I}\} \setminus \{\gamma\}$, the following inequality holds

$$\begin{aligned}&\left|\sum_{h=T_d}^{T_l-1} C_E^{(k)}(WA_{0,E})^{T_l-1-h}\Delta_l\phi_{E,\gamma}(h)\right| > \bar{\epsilon}^{(k)}_{y,E,l}(T_l) \\ &\quad + \left|C_E^{(k)}\right|\left\{\sum_{h=0}^{t-1}\alpha\delta^{t-1-h}W[\bar{\Delta}f_E(h) + \bar{\Delta}g_E(h) \right. \\ &\quad \left. + |L_E|\,\bar{\eta}_{y,E}(h)] + \alpha\delta^t\bar{\epsilon}_{x,E,l}(0)\right\} + \bar{\eta}^{(k)}_{y,E}(t)\end{aligned}$$

at some time instant $T_l > T_d$, for some $k \in \{1,\dots,p_I\}$, then the γ–th fault will be isolated. The local isolation time is upper-bounded by $\max_{l\in\{1,\dots,N_{\mathcal{F}_I}\}\setminus\gamma}(T_l)$.

Proof: After fault detection, for $t > T_d$, we can bound the absolute value of the k–th component of the l–th FIE estimation error using the triangle inequality:

$$|\epsilon^{(k)}_{y,E,l}(t)| \geq \left| \sum_{h=T_d}^{T_l-1} C^{(k)}_E (WA_{0,E})^{T_l-1-h} \Delta_l \phi_{E,\gamma}(h) \right|$$
$$- \left| C^{(k)}_E \sum_{h=0}^{t-1} (WA_{0,E})^{t-1-h} \Big[W\Delta f_E(h) + W\Delta g_E(h) - WL_E \eta_{y,E}(h) \Big] \right|$$
$$- \left| C^{(k)}_E (WA_{0,E})^t \epsilon_{x,E,l}(0) \right| - \left| \eta^{(k)}_{y,E}(t) \right|$$

We want the inequality $|\epsilon^{(k)}_{y,E,l}(t)| > \bar{\epsilon}^{(k)}_{y,E,l}(t)$ to be satisfied in order to exclude the l–th fault. This results in the following inequality

$$\left| \sum_{h=T_d}^{T_l-1} C^{(k)}_E (WA_{0,E})^{T_l-1-h} \Delta_l \phi_{E,\gamma}(h) \right| > \bar{\epsilon}^{(k)}_{y,E,l}(T_l)$$
$$+ \left| C^{(k)}_E \sum_{h=0}^{t-1} (WA_{0,E})^{t-1-h} [W\Delta f_E(h) + W\Delta g_E(h) - WL_E \eta_{y,E}(h)] \right|$$
$$+ \left| C^{(k)}_E (WA_{0,E})^t \epsilon_{x,E,l}(0) \right| + \left| \eta^{(k)}_{y,E}(t) \right|$$

which is implied by the inequality in the hypothesis of the theorem. This fault is isolated in the sense of Definition 6.4.4 if the inequality holds for every fault function of $\mathcal{F}_I$ but the γ–th. ■

9.5.2 Global fault isolation logic

The global fault isolation logic is the same designed for the discrete-time case in Section 6.4.3 and for the continuous-time case in Section 8.4.2. As already written, a distinction is made between local and distributed faults. For a local fault, it is sufficient that the corresponding LFD excludes every but that fault for concluding that it is isolated. Instead, in the case of distributed faults, the isolation requires that all the LFDs in the influence set of that fault, exclude all other faults.

Chapter 10

The Input-Output Continuous-time case

In this chapter, we present the continuous-time distributed monitoring architecture designed for the Input-Output case.

10.1 Problem Formulation

We consider a multi-input multi-output uncertain nonlinear continuous-time system:

$$\mathscr{S}: \begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\eta}_x(\boldsymbol{x}, \boldsymbol{u}, t) + \beta(t - T_0)\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{\eta}_y(\boldsymbol{x}, \boldsymbol{u}, t), \end{cases} \tag{10.1}$$

where, analogously as in Chapter 9, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^p$ denote the state, the control input and the measured output vectors respectively, the matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and the vector field $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ describe the nominal healthy dynamics, $\boldsymbol{C} \in \mathbb{R}^{p \times n}$ is the nominal output matrix, $\boldsymbol{\eta}_x$ and $\boldsymbol{\eta}_y$ are the uncertainties in the state and in the output equations. The term $\beta(t - T_0)\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})$ represents the fault function dynamics: $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u})$ denotes the functional structure and $\beta(t - T_0)$ characterizes the time profile of the fault. The following assumptions are needed.

Assumption 10.1.1: The state variables $\boldsymbol{x}$ and control variables $\boldsymbol{u}$ are bounded before and after the occurrence of a fault: $\exists\, \mathcal{R}$, compact region of $\mathbb{R}^n \times \mathbb{R}^m :\ (\boldsymbol{x}(t), \boldsymbol{u}(t)) \in \mathcal{R}$, $\forall t \geq 0$.

Assumption 10.1.2: The fault–evolution rate parameter α is unknown, but lower bounded by a known constant $\bar{\alpha}$.

Assumption 10.1.3: The measurement uncertainty term $\boldsymbol{\eta}_y$ is an unstructured and unknown nonlinear function, bounded by a positive, known

and bounded function $\bar{\boldsymbol{\eta}}_y$:

$$\left|\boldsymbol{\eta}_y^{(k)}(\boldsymbol{x},\boldsymbol{u},t)\right| \leq \bar{\boldsymbol{\eta}}_y^{(k)}(\boldsymbol{x},\boldsymbol{u},t),$$

for all $k = 1,\ldots,p$, $(\boldsymbol{x},\boldsymbol{u}) \in \mathcal{R}$ and for all $t \geq 0$.

As before, for the sake of generality, we consider an overlapping decomposition of the structural graph, instead of a decomposition made only with respect to the measurable variables. After the decomposition, the I-th subsystem $\mathscr{S}_I$ dynamics can be described by:

$$\begin{cases} \dot{x}_I = A_I x_I + f_I(x_I,u_I) + g_I(C_I x_I, u_I, z_I) \\ \qquad\quad + \beta(t-T_0)\phi_I(C_I x_I, z_I, u_I) \\ y_I = C_I x_I + \eta_{y,I}(x_I,u_I,t), \end{cases} \tag{10.2}$$

where $x_I \in \mathbb{R}^{n_I}$, $u_I \in \mathbb{R}^{m_I}$ and $y_I \in \mathbb{R}^{p_I}$ are the local state, the local control input, and the local measured output vectors respectively, and $z_I \in \mathbb{R}^{q_I}$ is the vector of the interconnection variables. The matrix $A_I \in \mathbb{R}^{n_I \times n_I}$ and the vector field $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ represent the local nominal healthy dynamics, $C_I \in \mathbb{R}^{p_I \times n_I}$ is the nominal local output matrix, $g_I : \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ is the interconnection function that incorporates the effects of the corresponding local state modeling uncertainty term $\eta_{x,I}$. The term $\eta_{y,I}$ is the uncertainty function in the local output equation that takes into account the measurement error, while $\phi_I : \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ is the local fault function. The following assumptions are in place.

Assumption 10.1.4: The decomposition of the monolithic system (10.1) is such that z_I is made of measurable variables only.
This assumption is needed in order to allow the learning of the interconnection function and of the fault function.

Assumption 10.1.5: The structural graph and the decomposition are the same before and after the fault event.

Assumption 10.1.6: (A_I, C_I) is an observable pair, $\forall I = 1,\ldots,N$.

Assumption 10.1.7: The interconnection function g_I is an unstructured and uncertain nonlinear function, bounded by a known and bounded function, i.e.,

$$\left|g_I^{(k)}(C_I x_I, z_I, u_I)\right| \leq \bar{g}_I^{(k)}(C_I x_I, z_I, u_I),$$

for all $I = 1,\ldots,N$, $k = 1,...,n_I$ and for all $(\boldsymbol{x},\boldsymbol{u}) \in \mathcal{R}$.

10.2 Distributed Detection Architecture

The distributed Fault Detection Architecture is the same we defined in Chapter 6, consisting of N agents, the LFDs, each one equipped with a non-linear adaptive estimator, the FDAE. The local FDAE estimation, in

the case of non-shared state variables, can be computed as:

$$\begin{aligned}\dot{\hat{x}}_I =& A_I\hat{x}_I + f_I(\hat{x}_I, u_I) + \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_I) \\ &+ L_I(y_I - \hat{y}_I) \\ \hat{y}_I =& C_I\hat{x}_I,\end{aligned} \tag{10.3}$$

where $\hat{g}_I$ is the output of an adaptive approximator designed to learn the unknown interconnection function g_I and $\hat{\vartheta}_I \in \hat{\Theta}_I$ denotes its adjustable parameters vector. The learning law is derived in the following. Due to the uncertain output measurements, each LFD receives from its neighbors the vector $v_I = z_I + \varsigma_I$, where ς_I is made with the components of $\eta_{y,J}$ that affect the relevant components of the neighboring subsystems measurements y_J. In the case of variables $x^{(s)}$ shared among more than one LFD, the estimate of the local state becomes:

$$\begin{aligned}\dot{\hat{x}}_I^{(s_I)} = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \Big[A_J^{(s_J)}\hat{x}_J + f_J^{(s_J)}(\hat{x}_J, u_J) + \hat{g}_J^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_J) \\ + L_J^{(s_J)}(y_J - \hat{y}_J)\Big]\end{aligned} \tag{10.4}$$

where the terms $W_s^{(I,J)}$ are the components of a doubly stochastic weighted adjacency matrix. Many choices are possible, but in all cases the weights $W_s^{(I,J)}$ can be seen as a level of how much the I-th subsystem feels confident about the information received from subsystem J. In the following (see Subsection 10.2.1) we provide a condition for the definition of the consensus matrix in order to guarantee the convergence of the estimator.

We now analyze the dynamics of the FDAE estimation errors before the occurrence of a fault. In the non-shared case, the i-th state estimation error component is:

$$\begin{aligned}\dot{\epsilon}_{x,I}^{(i)} =& A_I^{(i)}x_I + f_I^{(i)}(x_I, u_I) + g_I^{(i)}(C_I x_I, z_I, u_I) \\ &- A_I^{(i)}\hat{x}_I - f_I^{(i)}(\hat{x}_I, u_I) - \hat{g}_I^{(i)}(y_I, v_I, u_I, \hat{\vartheta}_I) - L_I^{(i)}(y_I - \hat{y}_I) \\ =& A_{0,I}^{(i)}\epsilon_{x,I} + \Delta f_I^{(i)} + \Delta g_I^{(i)} - L_I^{(i)}\eta_{y,I},\end{aligned} \tag{10.5}$$

where $A_{0,I} \triangleq A_I - L_I C_I$ is a stable matrix,

$$\Delta f_I^{(i)} \triangleq f_I^{(i)}(x_I, u_I) - f_I^{(i)}(\hat{x}_I, u_I)$$

and

$$\Delta g_I^{(i)} \triangleq g_I^{(i)}(C_I x_I, z_I, u_I) - \hat{g}_I^{(i)}(y_I, v_I, u_I, \hat{\vartheta}_I).$$

We denote with $A^{(i)}$ the i-th row of the matrix A.

In the case of shared variables, the dynamics of the LFD state estimation

error component prior to the occurrence of a fault can be written as:

$$\dot{\epsilon}_{x,I}^{(s_I)} = \dot{x}_I^{(s_I)} - \dot{\hat{x}}_I^{(s_I)} = A_I^{(s_I)} x_I + f_I^{(s_I)}(x_I, u_I) + g_I^{(s_I)}(C_I x_I, u_I, z_I) - \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[A_J^{(s_J)} \hat{x}_J + f_J^{(s_J)}(\hat{x}_J, u_J) + \hat{g}_J^{(s_J)}(y_J, u_J, v_J, \hat{\vartheta}_J) + L_J^{(s_J)}(y_J - \hat{y}_J) \Big].$$

By assumption it holds $\sum_{J \in \mathcal{O}_s} W_s^{(I,J)} = 1$ and, thanks to the way the model decomposition was obtained, the state estimation error component can be rewritten as:

$$\dot{\epsilon}_{x,I}^{(s_I)} = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left[A_{0,J}^{(s_J)} \epsilon_{x,J} + \Delta f_J^{(s_J)} + \Delta g_J^{(s_J)} - L_J^{(s_J)} \eta_{y,J} \right].$$

By introducing the general formulation for analysis purpose, we define the following extended vectors: the extended state estimation error vector $\epsilon_{x,E} \triangleq \text{col}\,(\epsilon_{x,J} : J = 1, ..., N) \in \mathbb{R}^{n_E \times 1}$, with $n_E = \sum_{J=1}^{N} n_J$, collecting the state estimation error vectors of the N sub-systems; $\Delta f_E(t)$, which is a $n_E \times 1$ matrix, collecting the values $\Delta f_J^{(s_J)}(t)$, for each $s_J = 1, ..., n_J$ and for every $J = 1, ..., N$; $\Delta g_E(t)$, defined in an analogous way as $\Delta f_E(t)$. And then we define the extended matrices:

$$W \triangleq \begin{bmatrix} W_{1,1} & \dots & W_{1,N} \\ \dots & \dots & \dots \\ W_{N,1} & \dots & W_{N,N} \end{bmatrix},$$

which is a $N \times N$ block matrix such that each block $W_{I,J}$, with $J = 1, ..., N$ and $I = 1, ..., N$, collects the consensus weights of the subsystem I with regard to the subsystem J[1], and

$$A_{0,E} \triangleq \begin{bmatrix} A_{0,1} & 0 & 0 & 0 \\ 0 & A_{0,2} & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & A_{0,N} \end{bmatrix},$$

which is a $N \times N$ diagonal block matrix, where the generic block is $A_{0,J} = A_J - L_J C_J \in \mathbb{R}^{n_J \times n_J}$, for $J = 1, .., N$. Finally, let us introduce $L_E \triangleq \text{blkdiag}(L_J : J = 1, .., N)$, which is a $N \times N$ diagonal block matrix with

[1]The diagonal blocks $W_{I,I}$ are square diagonal matrices in $\mathbb{R}^{n_I \times n_I}$, whose s_I–th diagonal element, with $s_I = 1, ..., n_I$, is equal to the weight $W_s^{(I,I)}$ defined in Eq. (8.7) if $x_I^{(s_I)}$ is a shared variable, and is equal to 1 otherwise. The matrices $W_{I,J} \in \mathbb{R}^{n_I \times n_J}$, with $J \neq I$, have non-null elements only in positions (s_I, s_J) corresponding to shared variables x_s, and here they take the value of the consensus weight $W_s^{(I,J)}$. This results in W being a symmetrical, sparse and doubly–stochastic $n_E \times n_E$ matrix.

dimension $n_E \times p_E$, where $p_E \triangleq \sum_{J=1}^{N} p_J$.

The dynamics of $\epsilon_{x,E}$ can be derived:

$$\dot{\epsilon}_{x,E} = W\left[A_{0,E}\epsilon_{x,E} + \Delta f_E + \Delta g_E - L_E\eta_{yE}\right] \tag{10.6}$$

where $\eta_{y,E}(t)$ is a $p_E \times 1$ column vector collecting the uncertainty terms of the N subsystems: $\eta_{y,E} \triangleq \text{col}\,(\eta_{y,I} : J = 1, ..., N)$.

10.2.1 Convergence condition

In order to guarantee the convergence of the state estimation error, the matrix $WA_{0,E}$ has to be a stable matrix. We derived a sufficient condition assuring that all the eigenvalues of the matrix are in the negative semi-plane.

Proposition 10.2.1: If $A_{0,E}$ is a diagonalizable matrix and if W is made so that the elements on the diagonal $W_{i,i} > 0.5$, then $WA_{0,E}$ is a stable matrix.

Proof: Since $A_{0,E}$ is a diagonal block matrix where the single blocks $A_{0,I},\ I = 1, \dots N$ are stable matrices, it is a stable matrix in turn. If it is a diagonal or a diagonalized matrix, the elements on the diagonal are the negative eigenvalues $-\lambda_i$. Using Gerschgorin circles ([193]) on the columns, it is possible to demonstrate that all the eigenvalues of $WA_{0,E}$ are trapped in the collection of circles centered at $-W_{i,i}\lambda_i$, with radii $\sum_{k\neq i} W_{k,i}\lambda_i = (1 - W_{i,i})\lambda_i$, where $W_{k,i}$ is the element of W corresponding to the k-th row and i-th column, with $k, i = 1, \dots, n_E$. The condition $-W_{i,i}\lambda_i + (1 - W_{i,i})\lambda_i < 0$ assures that all the eigenvalues of $WA_{0,E}$ have negative real part. This condition is satisfied if $W_{i,i} > 0.5$ for all $i = 1, \dots, n_E$. ■

It is worth noting that assuming $A_{0,E}$ to be a diagonalizable matrix is not a restrictive assumption since we can choose $L_I,\ \forall I$ so that this assumption is guaranteed.

10.2.2 The detection threshold

After the convergence of the state estimation error has been proved, the solution of the differential equation (10.6) can be derived as:

$$\epsilon_{x,E}(t) = \int_0^t e^{WA_{0,E}(t-\tau)}\left[W\Delta f_E(\tau) + W\Delta g_E(\tau) - WL_E\eta_{y,E}(\tau)\right]d\tau + e^{WA_{0,E}(t)}\epsilon_{x,E}(0). \tag{10.7}$$

The extended output estimation error is then defined as:

$$\epsilon_{y,E} \triangleq C_E\epsilon_{x,E} + \eta_{y,E}, \tag{10.8}$$

where $C_E \triangleq \text{blkdiag}(C_J : J = 1, .., N)$ is a $N \times N$ diagonal block matrix, with dimension $p_E \times n_E$. From (10.6), (10.8) and the definition of C_E, the

following learning law for the adjustable parameter vector $\hat{\vartheta}_I$ of the adaptive approximator $\hat{g}_I$, $I \in 1, \ldots, N$ can be derived:

$$\dot{\hat{\vartheta}}_I = P_{\hat{\Theta}_I}\left[\Gamma_I H_I^\top W_{I,I}^\top C_I^\top \epsilon_{y,I}\right] \tag{10.9}$$

$$H_I^\top = \partial \hat{g}_I / \partial \hat{\vartheta}_I,$$

where $P_{\hat{\Theta}_I}$ is a projection operator restricting $\hat{\vartheta}_I$ within $\hat{\Theta}_I$ ([183]), Γ_I is a symmetric and positive definite learning rate matrix (see for details [190]). In the general form, the component-wise output estimation error can be bounded by the following threshold, that can be computed in a distributed way:

$$\begin{aligned}
&\left|\epsilon_{y,E}^{(k)}(t)\right| \leq \left|C_E^{(k)}\epsilon_{x,E}(t)\right| + \left|\eta_{y,E}^{(k)}(t)\right| \\
&\leq \left|C_E^{(k)}\Big\{\int_0^t e^{WA_{0,E}(t-\tau)}\Big[W\Delta f_E(\tau) + W\Delta g_E(\tau) - WL_E\eta_{y,E}(\tau)\Big]d\tau \right. \\
&\quad \left. + e^{WA_{0,E}t}\epsilon_{x,E}(0)\Big\}\right| + \bar{\eta}_{y,E}^{(k)}(t) \\
&\leq \left|C_E^{(k)}\right| \Big\{\int_0^t \left\|e^{WA_{0,E}(t-\tau)}\right\| W\left[\bar{\Delta} f_E(\tau) + \bar{\Delta} g_E(\tau) + |L_E|\,\bar{\eta}_{y,E}(\tau)\right] d\tau \\
&\quad + \left\|e^{WA_{0,E}t}\right\| \bar{\epsilon}_{x,E}(0)\Big\} + \bar{\eta}_{y,E}^{(k)}(t) \\
&\triangleq \bar{\epsilon}_{y,E}^{(k)}(t)
\end{aligned} \tag{10.10}$$

where

$$\bar{\Delta} f_E^{(s)}(t) = \max_{x^{(s)} \in R^{x^{(s)}}} \left\{\left|\Delta f_E^{(s)}(t)\right|\right\},$$

$$\bar{\epsilon}_{x,E}^{(s)}(0) = \max_{x^{(s)} \in R^{x^{(s)}}} \left\{\left|x^{(s)} - \hat{x}^{(s)}(0)\right|\right\},$$

for every $s = 1, ..., n_E$; Δg_I can be upper bounded by

$$\bar{\Delta} g_I \triangleq \|H_I\|\,\kappa_I(\hat{\vartheta}_I) + \bar{\nu}_I + \max_{\eta_{yI}} \max_{\varsigma_I} |\Delta \hat{g}_I|,$$

where κ_I is such that $\kappa_I(\hat{\vartheta}_I) \geq \left\|\tilde{\vartheta}_I\right\|$, and the extended upper bound $\bar{\Delta} g_E$ simply collects the upper bounds of the N subsystems.

The residual $\epsilon_{y,I}$, is compared, component by component, with the suitable detection threshold signal $\bar{\epsilon}_{y,I}$ in order to derive the local fault decision. The threshold in Eq. (10.10) guarantees that no false-positive alarms will be issued until T_0 because of the uncertainties. In this way, however, the effects of the faults may be "hidden" by the uncertainties in the system dynamics due to the conservativeness of the threshold. This is formalized in the following section in which a distributed detectability sufficient condition will be devised, characterizing the faults that can be detected by the proposed

FDI scheme.

10.2.3 Fault Detectability Analysis

Let us assume that at time $t = T_0$ a fault $\boldsymbol{\phi}$ occurs in the monolithic system. ϕ_E denotes the extended fault function vector collecting the N subsystems fault functions. After the occurrence of the fault, for $t > T_0$, the state estimation error dynamics becomes

$$\dot{\epsilon}_{x,E} = W\left[A_{0,E}\epsilon_{x,E} + \Delta f_E + \Delta g_E - L_E\eta_{y,E}(t)\right] + (1 - e^{-\alpha(t-T_0)})\phi_E$$

and the output estimation error equation for the k-th component is:

$$\begin{aligned}\epsilon^{(k)}_{y,E}(t) = C^{(k)}_E\epsilon_{x,E}(t) + \eta^{(k)}_{y,E}(t) = \\ C^{(k)}_E\Bigg\{\int_0^t e^{WA_{0,E}(t-\tau)}\Big[W(\Delta f_E(\tau) + \Delta g_E(\tau) - L_E\eta_{y,E}(\tau)) \\ + (1 - e^{-\alpha(\tau-T_0)})\phi_E(\tau)\Big]d\tau + e^{WA_{0,E}t}\epsilon_{x,E}(0)\Bigg\} + \eta^{(k)}_{y,E}(t) \quad (10.11)\end{aligned}$$

Now, we are able to state and prove a sufficient condition for the characterization, in a non-closed form, of a class of faults that can be detected by the proposed FDI methodology.

Theorem 10.2.2 (Fault Detectability): If there exists a time instant $t_1 > T_0$ such that the fault ϕ_E satisfies the inequality

$$\left|\int_{T_0}^{t_1} C^{(k)}_E e^{WA_{0,E}(t_1-\tau)}(1 - e^{-\alpha(\tau-T_0)})\phi_E(\tau)d\tau\right| > 2\bar{\epsilon}^{(k)}_{y,E}(t_1)$$

for at least one component $k \in \{1, ..., p_E\}$, then the fault will be detected at time t_1, that is $\left|\epsilon^{(k)}_{y,E}(t_1)\right| > \bar{\epsilon}^{(k)}_{y,E}(t_1)$.

Proof: At time instant $t_1 > T_0$, the output estimation error can be written as:

$$\begin{aligned}\epsilon^{(k)}_{y,E}(t_1) = \int_0^{t_1} C^{(k)}_E e^{WA_{0,E}(t_1-\tau)}W\left[\Delta f_E(\tau) + \Delta g_E(\tau) - L_E\eta_{y,E}(\tau)\right]d\tau \\ + C^{(k)}_E e^{WA_{0,E}t_1}\epsilon_{x,E}(0) + \eta^{(k)}_{y,E}(t_1) \\ + \int_{T_0}^{t_1} C^{(k)}_E e^{WA_{0,E}(t_1-\tau)}(1 - e^{-\alpha(\tau-T_0)})\phi_E(\tau)d\tau\end{aligned}$$

Using the triangle inequality we obtain:

$$\left|\epsilon_{y,E}^{(k)}(t_1)\right| \geq -\left|\int_0^{t_1} C_E^{(k)} e^{WA_{0,E}(t_1-\tau)} W\left[\Delta f_E(\tau) + \Delta g_E(\tau) - L_E \eta_{y,E}(\tau)\right] d\tau \right.$$
$$\left. + C_E^{(k)} e^{WA_{0,E}t_1}\epsilon_{x,E}(0) + \eta_{y,E}^{(k)}(t_1)\right|$$
$$+ \left|\int_{T_0}^{t_1} C_E^{(k)} e^{WA_{0,E}(t_1-\tau)}(1 - e^{-\alpha(\tau-T_0)})\phi_E(\tau)d\tau\right|$$

By recalling how the threshold was defined (Eq. 10.10), it is easy to see that the following inequality holds:

$$\left|\epsilon_{y,E}^{(k)}(t_1)\right| \geq -\bar{\epsilon}_{y,E}^{(k)}(t_1) + \left|\int_{T_0}^{t_1} C_E^{(k)} e^{WA_{0,E}(t_1-\tau)}(1 - e^{-\alpha(\tau-T_0)})\phi_E(\tau)d\tau\right| .$$

In this way the fault detection condition $\left|\epsilon_{y,E}^{(k)}(t_1)\right| > \bar{\epsilon}_{y,E}^{(k)}(t_1)$ is implied by the theorem hypothesis. ■

10.3 Distributed Isolation Architecture

After a fault is detected by any of the N LFDs, the *Global Fault Diagnoser* (GFD) receives the corresponding local fault decision and switches each LFD from fault detection to fault isolation operating mode, stopping the learning of the parameter $\hat{\vartheta}_I$. As in the previous sections, for isolation purposes we assume that the fault function $\boldsymbol{\phi}$ may belong to a known global fault set $\mathcal{F}$ or be unknown:

$$\mathcal{F} \triangleq \{\boldsymbol{\phi_1}(\boldsymbol{Cx}, \boldsymbol{u}), \ldots, \boldsymbol{\phi_{N_\mathcal{F}}}(\boldsymbol{Cx}, \boldsymbol{u})\}.$$

It is possible that not all the subsystems are affected by a given fault function $\boldsymbol{\phi_l}$, but only those contained in the corresponding fault influence set $\boldsymbol{\mathcal{U}_l}$ for the l-th fault function ϕ_l, with $l = 1, \ldots, \boldsymbol{N_\mathcal{F}}$. As a consequence, a local fault set $\mathcal{F}_I$ can be defined for each subsystem $\mathscr{S}_I$, collecting the local fault functions $\phi_{I,l}$ such that $I \in \boldsymbol{\mathcal{U}_l}$:

$$\mathcal{F}_I \triangleq \{\phi_{I,1}(C_I x_I, z_I, u_I), \ldots, \phi_{I,N_{\mathcal{F}_I}}(C_I x_I, z_I, u_I)\}.$$

It is worth noting that, following Assumption (10.1.5), the local fault functions depend only on the local variables x_I, z_I and u_I.

Besides the FDAE, in the isolation mode each LFD uses other $N_{\mathcal{F}_I}$ estimators, the FIEs, one for each fault in the local fault set $\mathcal{F}_I$, in order to locally isolate the fault that is acting on the subsystem I. In this way, it is not necessary that the I–th LFD knows the global fault influence set: it

is only able to isolate the *local* part of a fault that influences the subsystem $\mathscr{S}_I$. For each LFD $\mathscr{L}_I$, with $I = 1, ..., N$, the generic l–th FIE, with $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, monitors the corresponding fault function $\phi_{I,l}$, belonging to the local fault set $\mathcal{F}_I$. We assume that each fault function in $\mathcal{F}_I$ can be expressed as:

$$\phi_{I,l}(C_I x_I, z_I, u_I) = \Big[(\vartheta_{I,l,1})^\top H_{I,l,1}(C_I x_I, z_I, u_I), \\ \ldots, (\vartheta_{I,l,n_I})^\top H_{I,l,n_I}(C_I x_I, z_I, u_I)\Big]^\top , \quad (10.12)$$

where $H_{I,l,k} : \mathbb{R}^{p_I} \times \mathbb{R}^{q_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{q_{I,l,k}}$, with $k \in \{1, \ldots, n_I\}$, $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, are the known functions describing the functional structure of the fault and $\vartheta_{I,l,k} \in \Theta_{I,l,k} \subset \mathbb{R}^{q_{I,l,k}}$ are the unknown parameter vectors providing its "magnitude", where the parameter domains $\Theta_{I,l,k}$ are assumed to be origin–centered hyper–spheres with radius $M_{\Theta_{I,l,k}}$, without much loss of generality.

After $t = T_d$, the generic l–th FIE estimator is activated and monitors its subsystem, computing a local state estimate $\hat{x}_{I,l}$ and a local output estimate $\hat{y}_{I,l}$. The difference between the estimate $\hat{y}_{I,l}$ and the measurements y_I is the estimation error $\epsilon_{y,I,l} \triangleq y_I - \hat{y}_{I,l}$, used as a residual and compared, component by component, to an appropriate isolation threshold $\bar{\epsilon}_{y,I,l} \in \mathbb{R}^{p_I}_+$. The condition

$$|\epsilon^{(k)}_{y,I,l}(t)| \leq \bar{\epsilon}^{(k)}_{y,I,l}(t) \; \forall \, k = 1, \ldots, p_I \quad (10.13)$$

is associated to the l–th fault hypothesis

$$\mathscr{H}_{I,l} : \text{"The subsystem } \mathscr{S}_I \text{ is affected by the } l\text{–th fault"}. \quad (10.14)$$

Now, a detailed description of the FIEs is given. After the fault $\phi(t)$ has occurred, the s_I–th component of the I–th local state equation becomes

$$\dot{x}_I^{(s_I)} = A_I^{(s_I)} x_I + f_I^{(s_I)}(x_I, u_I) + g_I^{(s_I)}(C_I x_I, z_I, u_I) + \beta(t - T_0)\phi^{(s)}(Cx, u)\,.$$

The l-th FIE computes a local estimate, that, in the case of non-shared state variables, can be defined as:

$$\begin{cases} \dot{\hat{x}}_{I,l} = A_I \hat{x}_{I,l} + f_I(\hat{x}_{I,l}, u_I) + \hat{g}_I(y_I, v_I, u_I, \hat{\vartheta}_{I,0}) \\ \qquad\quad + L_I(y_I - \hat{y}_{I,l}) + \hat{\phi}_{I,l}(y_I, v_I, u_I, \hat{\vartheta}_{I,l}) \\ \hat{y}_{I,l} = C_I \hat{x}_{I,l}, \end{cases} \quad (10.15)$$

where $L_I \in \mathbb{R}^{n_I \times p_I}$ is the local output error gain, $\hat{\phi}^{(s_I)}_{I,l}(y_I, v_I, u_I, \hat{\vartheta}_{I,l}) \triangleq (\hat{\vartheta}_{I,l,s_I})^\top H_{I,l,s_I}(y_I, v_I, u_I)$ is the s_I–th component of a linearly-parameterized function that learns the structure of the l–th fault function $\phi_{I,l}$, where the vector $\hat{\vartheta}_{I,l} \triangleq \text{col}(\hat{\vartheta}_{I,l,k}, k \in \{1, \ldots, n_I\})$ contains its adjustable parame-

ters. We developed the following learning law, for every $I = 1, \ldots, N$, $l = 1, ..., N_{\mathcal{F}_I}$:

$$\dot{\hat{\vartheta}}_{I,l} = P_{\hat{\Theta}_{I,l}} \left[\Gamma_{I,l} H_{I,l}^{\top} W_{I,I}^{\top} C_I^{\top} \epsilon_{y,I,l} \right] \tag{10.16}$$
$$H_{I,l}^{\top} = \partial \hat{\phi}_{I,l} / \partial \hat{\vartheta}_{I,l}.$$

The dynamics of the l–th FIE estimator for the most general case of a distributed fault can be defined as

$$\dot{\hat{x}}_{I,l}^{(s_I)} = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[A_J^{(s_J)} \hat{x}_{J,l} + f_J^{(s_J)}(\hat{x}_{J,l}(t), u_J) + \hat{g}_J^{(s_J)}(y_J, u_J, v_J, \hat{\vartheta}_{J,0}) + L_J^{(s_J)}(y_{J,l} - \hat{y}_J) + \hat{\phi}_{J,l}^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,l}) \Big] \tag{10.17}$$

The i-th component of the estimation error, in the non-shared case, is

$$\dot{\epsilon}_{x,I,l}^{(i)} = A_{0,I}^{(i)} \epsilon_{x,I,l} + \Delta f_I^{(i)} + \Delta g_I^{(i)} - L_I^{(i)} \eta_{y,I} + (1 - e^{-\alpha(t-T_0)}) \phi^{(i)} - \hat{\phi}_{J,l}^{(i)}(y_J, v_J, u_J, \hat{\vartheta}_{J,l}), \tag{10.18}$$

On the other hand, the dynamics of the state estimation error component for shared variables can be described as:

$$\dot{\epsilon}_{x,I,l}^{(s_I)} = A_I^{(s_I)} x_I + f_I^{(s_I)}(x_I, u_I) + g_I^{(s_I)}(C_I x_I, z_I, u_I) + (1 - e^{-\alpha(t-T_0)}) \phi^{(s)} - \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[A_J^{(s_J)} \hat{x}_{J,l} + f_J^{(s_J)}(\hat{x}_{J,l}, u_J) + \hat{g}_J^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,0}) + L_J^{(s_J)}(y_J - \hat{y}_{J,l}) + \hat{\phi}_{J,l}^{(s_J)} \Big].$$

When we consider a matched fault

$$\phi^{(s)} = \phi_{J,l}^{(s_J)}(x_J, z_J, u_J, \vartheta_{J,l}), \forall\, J \in \mathcal{O}_{\boldsymbol{s}},$$

the error dynamics can then be rewritten as:

$$\dot{\epsilon}_{x,I,l}^{(s_I)} = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left[A_{0,J}^{(s_J)} \epsilon_{x,J,l} + \Delta f_J^{(s_J)} + \Delta g_J^{(s_J)} - L_J^{(s_J)} \eta_{y,J} + \Delta \phi_{J,l}^{(s_J)} \right] \tag{10.19}$$

where

$$\Delta \phi_{J,l}^{(s_J)} \triangleq (1 - e^{-\alpha(t-T_0)}) \phi^{(s)} - \hat{\phi}_{J,l}^{(s_J)}$$
$$= (1 - e^{-\alpha(t-T_0)}) (H_{J,l,s_J}(t)^{\top} \vartheta_{J,l,s_J} + \Delta H_{J,l,s_J}^{\top} \vartheta_{J,l,s_J}) - H_{J,l,s_J}(t)^{\top} \hat{\vartheta}_{J,l,s_J}$$

with $\Delta H_{J,l,s_J}^{\top} \triangleq H_{J,l,s_J}(x_J, z_J, u_J) - H_{J,l,s_J}(y_J, v_J, u_J)$. It can be rewritten

as

$$\Delta\phi_{J,l}^{(s_J)} = (1 - e^{-\alpha(t-T_0)})(H_{J,l,s_J}(t)^\top \tilde{\vartheta}_{J,l,s_J} + \Delta H_{J,l,s_J}(t)^\top \vartheta_{J,l,s_J}) \\ - e^{-\alpha(t-T_0)} H_{J,l,s_J}(t)^\top \hat{\vartheta}_{J,l,s_J}$$

if we introduce the parameter estimation error $\tilde{\vartheta}_{J,l,s_J} \triangleq \vartheta_{J,l,s_J} - \hat{\vartheta}_{J,l,s_J}$. Using the general formulation, we can express the dynamics of the estimation error in the case of a matched fault, as:

$$\dot{\epsilon}_{x,E,l} = W\left[A_{0,E}\,\epsilon_{x,E,l} + \Delta f_E + \Delta g_E - L_E\eta_{yE} + \Delta\phi_{E,l}\right].$$

We can now compute the state estimation error solution:

$$\epsilon_{x,E,l}(t) = \int_{T_d}^{t} e^{WA_{0,E}(t-\tau)} W\left[\Delta f_E(\tau) + \Delta g_E(\tau) - L_E\eta_{y,E}(\tau) + \Delta\phi_{E,l}(\tau)\right] d\tau \\ + e^{WA_{0,E}(t-T_d)}\epsilon_{x,E,l}(T_d). \quad (10.20)$$

The output estimation error in the case of a matched fault can be written componentwise as: $\epsilon_{y,E,l}^{(k)} \triangleq C_E^{(k)}\epsilon_{x,E,l} + \eta_{y,E}^{(k)}$, for all $k = 1, ..., p_E$. It can be bounded by:

$$\begin{aligned}\left|\epsilon_{y,E,l}^{(k)}(t)\right| &\leq \left|C_E^{(k)}\Big\{\int_{T_d}^{t} e^{WA_{0,E}(t-\tau)}\Big[W(\Delta f_E(\tau) + \Delta g_E(\tau) - L_E\eta_{y,E}(\tau) \right. \\ &\quad \left. + \Delta\phi_{E,l}(\tau))\Big]d\tau + e^{WA_{0,E}(t-T_d)}\epsilon_{x,E,l}(T_d)\Big\}\right| + \left|\eta_{y,E}^{(k)}(t)\right| \\ &\leq \left|C_E^{(k)}\right|\Big\{\int_{T_d}^{t}\left\|e^{WA_{0,E}(t-\tau)}\right\| W\left[\bar{\Delta} f_E(\tau) + \bar{\Delta} g_E(\tau) + |L_E|\,\bar{\eta}_{y,E}(\tau)\right. \\ &\quad \left. + \bar{\Delta}\phi_{E,l}(\tau)\right] d\tau + \left\|e^{WA_{0,E}(t-T_d)}\right\|\bar{\epsilon}_{x,E,l}(T_d)\Big\} + \bar{\eta}_{y,E}^{(k)}(t) \quad (10.21)\end{aligned}$$

where

$$\bar{\Delta}\phi_{E,l} = \mathrm{col}(\|H_{I,l,s_I}(t)\|\kappa_{I,l,s_I}(\hat{\vartheta}_{I,l,s_I}) + \bar{\Delta}H_{I,l,s_I}(t)\bar{\vartheta}_{I,l,s_I} \\ - e^{-\bar{\alpha}(t-T_d)}\|H_{I,l,s_I}(t)\|\,\|\hat{\vartheta}_{I,l,s_I}\|,\ s_I = 1, ..., n_I, I = 1, ..., N)\,.$$

where $\kappa_{I,l}(\hat{\vartheta}_{I,l}) \geq \left\|\tilde{\vartheta}_{I,l}\right\|$. The threshold (10.21) can be computed in a distributed way and, because of the way it is defined, guarantees that no matched fault will be excluded due to the presence of uncertainties or to the effect of the parameter estimation error $\tilde{\vartheta}_{I,l}$.

10.3.1 Fault isolability analysis

We now consider the case of a non–matched fault

$$\phi_I^{(s_I)}(x_I, z_I, u_I) = \phi_{I,\gamma}^{(s_I)}(x_I, z_I, u_I, \vartheta_{I,\gamma}),$$

with $\gamma \neq l$. In this case, the dynamics of the shared s_I–component of the estimation error for the l–th FIE of the I–th LFD can be written as

$$\begin{aligned}\dot{\epsilon}_{x,I,l}^{(s_I)} = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)}[A_{0,J}^{(s_J)}\epsilon_{x,J,l} + \Delta f_J^{(s_J)} + \Delta g_J^{(s_J)} - L_J^{(s_J)}\eta_{y,J} \\ + (1 - e^{-\alpha(t-T_0)})\phi_{I,\gamma}^{(s_I)}(x_I, z_I, u_I, \vartheta_{I,\gamma}) - \hat{\phi}_{J,l}^{(s_J)}(y_J, v_J, u_J, \hat{\vartheta}_{J,l})]\end{aligned}$$

and considering the vector $\epsilon_{x,E,l}$:

$$\dot{\epsilon}_{x,E,l} = W\left[A_{0,E}\,\epsilon_{x,E,l} + \Delta f_E + \Delta g_E - L_E\eta_{yE} + \Delta_l\phi_{E,\gamma}\right],$$

where the mismatch vector is introduced

$$\Delta_l\phi_{E,\gamma} \triangleq \mathrm{col}((1 - e^{-\alpha(t-T_0)})\phi_{I,\gamma}^{(s_I)} - \hat{\phi}_{I,l}^{(s_I)}, s_I = 1, ..., n_I, I = 1, ..., N).$$

The solution can then be written as

$$\begin{aligned}\epsilon_{x,E,l}(t) = \int_{T_d}^{t} e^{WA_{0,E}(t-\tau)}W\left[\Delta f_E(\tau) + \Delta g_E(\tau) - L_E\eta_{y,E}(\tau) + \Delta_l\phi_{E,\gamma}(\tau)\right]d\tau \\ + e^{WA_{0,E}(t-T_d)}\epsilon_{x,E,l}(T_d)\end{aligned}$$

and then, the output residual can be expressed componentwise by the following equation:

$$\begin{aligned}\epsilon_{y,E,l}^{(k)}(t) = \eta_{y,E}^{(k)}(t) + C_E^{(k)}\Big\{\int_{T_d}^{t} e^{WA_{0,E}(t-\tau)}W\Big[\Delta f_E(\tau) + \Delta g_E(\tau) \\ - L_E\eta_{y,E}(\tau) + \Delta_l\phi_{E,\gamma}(\tau)\Big]d\tau + e^{WA_{0,E}(t-T_d)}\epsilon_{x,E,l}(T_d)\Big\}.\end{aligned}$$

At this point, a sufficient condition for fault isolability can be proved.

Theorem 10.3.1 (Fault Isolability): Given a fault $\phi_{I,\gamma} \in \mathcal{F}_I$, if for each $l \in \{1, \ldots, N_{\mathcal{F}_I}\} \setminus \gamma$, the following inequality holds

$$\begin{aligned}\left|\int_{T_d}^{T_l} C_E^{(k)}e^{WA_{0,E}(T_l-\tau)}\Delta_l\phi_{E,\gamma}(\tau)d\tau\right| > \bar{\epsilon}_{y,E,l}^{(k)}(T_l) \\ + \left|C_E^{(k)}\right|\Big\{\int_{T_d}^{T_l}\left\|e^{WA_{0,E}(T_l-\tau)}\right\|W\left[\bar{\Delta}f_E(\tau) + \bar{\Delta}g_E(\tau)\right. \\ \left. + |L_E|\,\bar{\eta}_{y,E}(\tau)\right]d\tau + \left\|e^{WA_{0,E}(T_l-T_d)}\right\|\bar{\epsilon}_{x,E,l}(T_d)\Big\} + \bar{\eta}_{y,E}^{(k)}(T_l)\end{aligned}$$

at some time instant $T_l > T_d$, for some $k \in \{1, \ldots, p_I\}$, then the γ–th fault will be isolated. The local isolation time is upper-bounded by $\max_{l \in \{1,\ldots,N_{\mathcal{F}_I}\} \setminus \gamma} (T_l)$.

Proof: After fault detection, for $T_l > T_d$, we can bound the absolute value of the k–th component of the l–th FIE estimation error using the triangle inequality:

$$
\begin{aligned}
|\epsilon^{(k)}_{y,E,l}(T_l)| \geq & \left| \int_{T_d}^{T_l} C_E^{(k)} e^{WA_{0,E}(T_l-\tau)} \Delta_l \phi_{E,\gamma}(\tau) d\tau \right| \\
& - \left| C_E^{(k)} \int_{T_d}^{T_l} e^{WA_{0,E}(T_l-\tau)} \Big[W\Delta f_E(\tau) + W\Delta g_E(\tau) - WL_E \eta_{y,E}(\tau) \Big] d\tau \right| \\
& - \left| C_E^{(k)} e^{WA_{0,E}(T_l-T_d)} \epsilon_{x,E,l}(T_d) \right| - \left| \eta^{(k)}_{y,E}(T_l) \right|
\end{aligned}
$$

We want the inequality $|\epsilon^{(k)}_{y,E,l}(T_l)| > \bar{\epsilon}^{(k)}_{y,E,l}(T_l)$ to be satisfied in order to exclude the l–th fault. This results in the following inequality

$$
\begin{aligned}
& \left| \int_{T_d}^{T_l} C_E^{(k)} e^{WA_{0,E}(T_l-\tau)} \Delta_l \phi_{E,\gamma}(\tau) d\tau \right| > \bar{\epsilon}^{(k)}_{y,E,l}(T_l) \\
& + \left| C_E^{(k)} \int_{T_d}^{T_l} e^{WA_{0,E}(T_l-\tau)} [W\Delta f_E(\tau) + W\Delta g_E(\tau) - WL_E \eta_{y,E}(\tau) \right| d\tau] \\
& + \left| C_E^{(k)} e^{WA_{0,E}(T_l-T_d)} \epsilon_{x,E,l}(T_d) \right| + \left| \eta^{(k)}_{y,E}(T_l) \right|
\end{aligned}
$$

which is implied by the inequality in the hypothesis of the theorem. This fault is isolated in the sense of Definition 6.4.4 if the inequality holds for every fault function of $\mathcal{F}_I$ but the γ–th. ■

10.3.2 Global fault isolation logic

The global fault isolation logic is implemented by the Global Fault Diagnoser as expressed in the previous chapters.

Chapter 11

Conclusions

In this work, a comprehensive distributed architecture, suitable for the monitoring of modern complex systems, such as large-scale distributed or networked systems, Cyber-Physical Systems and Systems-of-Systems, is designed. The motivations for this thesis work are the renewed emphasis given to monitoring and fault-tolerant systems. The increased complexity in modern systems, in fact, implies the need for novel tools, able to consider all the different aspects and levels constituting complex systems. That's why we designed a comprehensive architecture, taking into account three different layers of the monitoring scheme:

- the physical environment, modeled as a generic large-scale uncertain non-linear continuous-time system. It is decomposed into some subsystems able to communicate with each other and influencing each other, in order to make the large-scale problem tractable;

- the sensor layer, composed by some sensor networks, measuring the variables of the physical system, filtering them by means of a distributed estimation method and communicating the filtered measurements to the diagnoser;

- the diagnosers layer, divided into Local Fault Diagnosers and the Global Fault Diagnoser. Each LFD monitors exactly one subsystem, basing on the local model, in a distributed way. The GFD coordinates the activity of the LFDs and produces a global fault decision about the healthy status of the monitored system (healthy or faulty).

The introduction of the sensor layer allows the decoupling of the physical and the sensing/computation topologies, bringing some advantages, such as scalability and reliability of the diagnosis architecture. Moreover, it allows to consider multi-rate systems and not synchronized measurements, having in mind realistic applications.

Two different communication networks are considered: the first layer communication network allows the sensor networks to communicate the filtered measurements to the diagnosers; the second-layer communication network connects the diagnosers. In this way they can communicate some variables permitting to consider and learn the uncertain influences between subsystems and to implement a kind of consensus protocol in order to improve diagnosers estimates. In particular, specifically designed methods are developed in order to take into account the issues emerging when dealing with communication networks and distributed systems, such as delays and packet dropouts. That's why we introduced a distributed delay compensation strategy, based on the use of Time Stamps and buffers and the definition of a time-varying consensus matrix. The goal of the novel time-varying matrix is twofold: it allows to manage communication delays, packet dropouts and interrupted links between diagnosers and permits to optimize the detectability skills by defining less conservative thresholds. Moreover, the diagnosers implement a clock-synchronization method and a re-synchronization mechanism in order to take into account multi-rate systems and sensor networks. The distributed fault detection and isolation schemes have been studied and analytical results regarding fault detectability, isolability and estimator convergence have been derived. Moreover, the presented DFDI discrete-time scheme has been extended to different scenarios. Specifically, we adapted the architecture to the continuous-time framework and we considered the case that the state is only partially measurable (multi Input multi Output case).

11.1 Future developments

As a future work, we are investigating the possibility to define less conservative detection thresholds. In [200] a filtering approach is proposed. Moreover, ongoing research aims at weakening some of the assumptions made in this work, such as, for instance, Assumptions 9.1.4 and 10.1.4 in the Input-Output case, requiring the decomposition to be such that the interconnection variables are measurable. In this context, we would like to study the system decomposition problem in order to propose a method able to define suitable decompositions in order to optimize detectability and isolability skills. In this work, in fact the decomposition of the system is supposed to be given. An interesting research topic could be to study how the decomposition influences the detectability and if it is possible to define some "optimal" decompositions, able to minimize, as example, the detection time or able to maximize the extent of the class of faults that can be detected, allowing to detect faults of "smaller magnitude". Besides, we are going to investigate the extension of the proposed deterministic (excluding the recently introduced bound on the estimation error) architecture to stochastic scenarios in

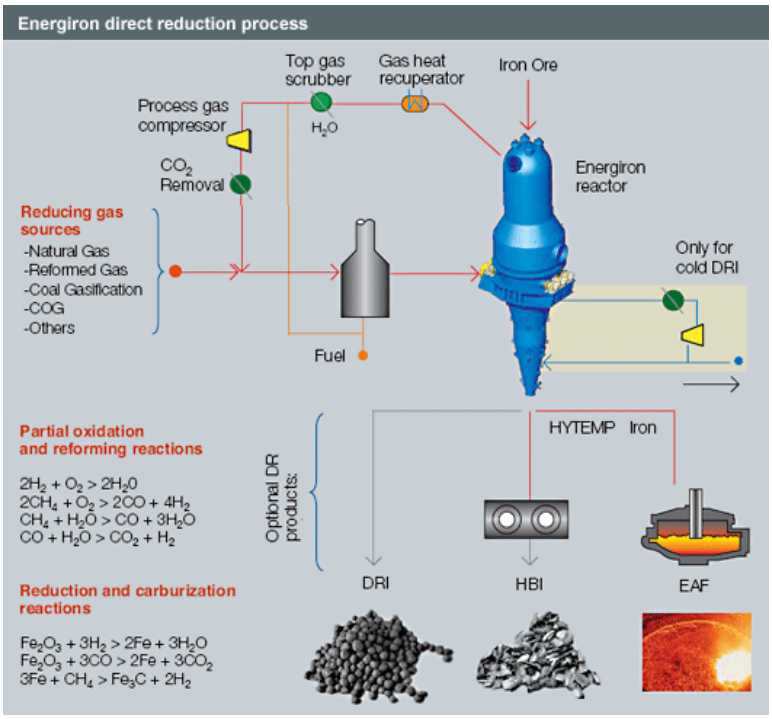

Figure 11.1: A simple scheme of the DRI plant.

order to define probabilistic thresholds with guaranteed confidence levels. A possibility could be to extend an approach based on LSCR (Leave-out Sign-dominant Correlation Regions), a system identification method introduced in [201], allowing to compute guaranteed confidence sets for the parameters of dynamical systems based on a finite number of data points. In [202] the LSCR principles are applied to change detection problems in a centralized architecture. It would be interesting to investigate the opportunities of this stochastic approach in a distributed architecture. In this way, the absence of false alarms, which is guaranteed with the proposed architecture, would not be assured, but the solution would be less conservative. Finally, the research effort goes on the validation of the proposed monitoring architecture with extensive simulations and on practically-relevant distributed use-cases. In cooperation with Danieli Automation, we are developing an architecture for the monitoring of a real industrial process, the DRI (Direct Reduction Iron) process. It is a complex chemical process, converting lump ore and/or iron oxide pellets into highly metallised, stable iron product through the chemical reactions with reducing gases. The reactions happen inside a shaft reactor (see in Figure 11.1 a streamlined scheme of the DRI plant, where only the main chemical reactions are written). The chosen approach models the reactor by discretizing it in a mono-dimensional way, along the height

(Figure 11.2). Because of the size of the DRI reactor, in fact, it should be modeled as a distributed parameter system. Instead of using the classical Partial Differential Equations approach for the discretization problem, we chose another, innovative, approach [203], based on an Algebraic Formulation of the physical laws governing the process, since it allows to directly lead to discrete equations without the need for intermediate PDE description. This approach is implemented by the *cells method* proposed in [204]. In the DRI case, the reactor has been divided in 160 cells. Each cell is modeled taking into account pellet and gas flows, mass/molar concentration and thermal energy transport, heat exchange between gas and solid phases, and chemical reactions. Different reactions happen at different height levels in the reactor, depending on gas and pellet temperature, pressure, velocity, density and composition. The influences between neighboring cells are considered. The numerical DRI model is based on the constitutive equations representing mass balance, moles balance and energy balance. At the moment, we are validating the 160 cells model with real plant data and we are going to study some faulty scenarios. The proposed monitoring architecture will be implemented by considering each cell as a subsystem. One of the main challenges, apart from the complexity of the chemical plant, is that there are only few sensors in the real process, measuring only some of the state variables (input and output of the reactor) and so the behavior of gas and pellet in the internal cells has to be reconstructed.

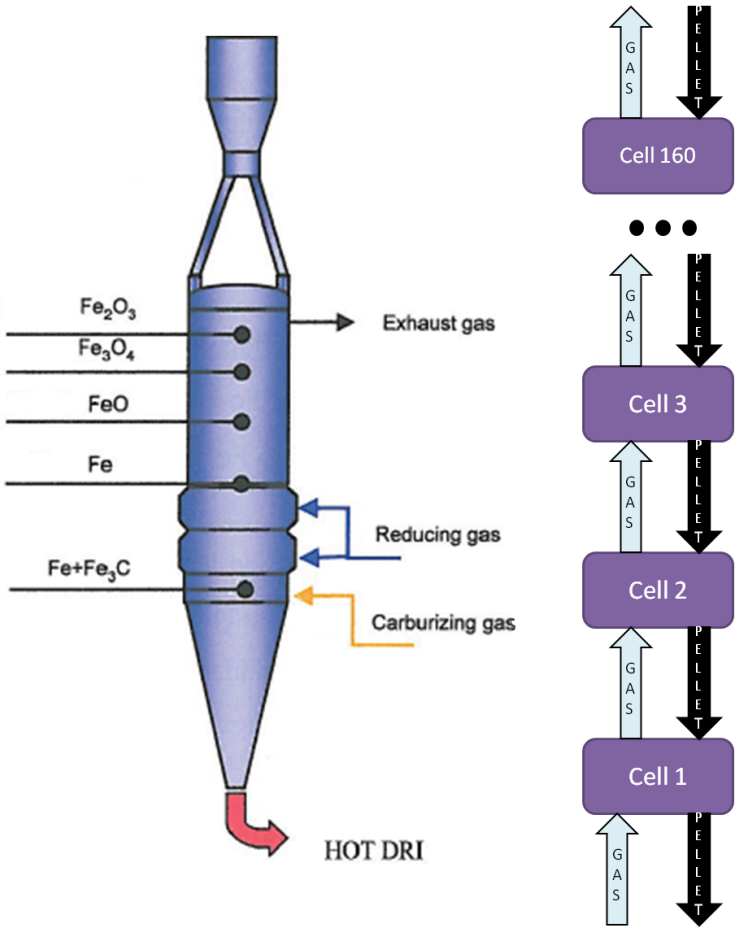

Figure 11.2: The DRI shaft reactor and the discretization scheme.

Bibliography

[1] T. Samad and T. Parisini, "Systems of systems," *The Impact of Control Technology (T.Samad and A.Annaswamy, eds.)*, 2011. [Online]. Available: www.ieeecss.org

[2] K. Baheti and H. Gill, "Cyber-physical systems," *The Impact of Control Technology (T.Samad and A.Annaswamy, eds.)*, 2011. [Online]. Available: www.ieeecss.org

[3] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[4] G. M. Milis, C. G. Panayiotou, and M. M. Polycarpou, "Towards a semantically enhanced control architecture," in *Intelligent Control (ISIC), 2012 IEEE International Symposium on*, oct. 2012, pp. 1195 –1200.

[5] R. M. Ferrari, "Fault diagnosis of distributed large-scale discrete-time nonlinear systems," *PhD Thesis, http://control.units.it/ferrari*, 2010.

[6] D. Bertsekas and J. Tsitsiklis, "Some aspects of parallel and distributed iterative algorithms-a survey," *Automatica*, vol. 27, no. 1, pp. 3–21, 1991.

[7] N. Lynch, *Distributed algorithms*. Morgan Kauffmann, 1996.

[8] R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis with overlapping decompositions: an adaptive approximation approach," *IEEE Trans. on Automatic Control*, vol. 54, no. 4, pp. 794–799, 2009.

[9] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault Tolerant Control*. Berlin: Springer, 2003.

[10] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag, 2006.

[11] J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Systems Magazine*, vol. 8, no. 6, pp. 3–11, 1988.

[12] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis. Part III: Process history based methods," *Computers & Chem. Eng.*, vol. 27, pp. 327–346, 2003.

[13] R. Beard, "Failure accomodation in linear systems through self–reorganization," *Technical Report MTV-71-1, Man Vehicle Laboratory, MIT, Cambridge, MA*, 1971.

[14] H. Jones, "Failure detection in linear systems," Ph.D. Thesis, Dept. of Aero and Astro, MIT, Cambridge, MA, 1973.

[15] R. Clark, "Instrument fault detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14, pp. 456—465, 1978.

[16] R. Isermann, "Process fault detection based on modeling and estimation methods. a survey." *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[17] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge–based redundancy – a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[18] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis. Part I: Quantitative model-based methods," *Computers & Chem. Eng.*, vol. 27, pp. 293–311, 2003.

[19] ——, "A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies," *Computers & Chem. Eng.*, vol. 27, pp. 313–326, 2003.

[20] J. Farrell, T. Berger, and B. Appleby, "Using learning techniques to accommodate unanticipated faults," *IEEE Control Systems Magazine*, vol. 13, pp. 40—49, 1993.

[21] A. Vemuri and M. M. Polycarpou, "On-line approximation methods for robust fault detection," *Proc. 13th IFAC World Congress*, vol. K, pp. 319–324, 1996.

[22] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 47, no. 4, pp. 576–593, 2002.

[23] M. Daigle, X. Koutsoukos, and G. Biswas, "Distributed diagnosis in formations of mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 353–369, 2007.

[24] N. Lechevin and C. Rabbath, "Decentralized detection of a class of non-abrupt faults with application to formations of unmanned airships," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 484–493, 2009.

[25] W. Li, W. H. Gui, Y. F. Xie, and S. X. Ding, "Decentralized fault detection system design for large-scale interconnected systems," in *Proc. of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess 2009)*, Barcelona, Spain, 2009, pp. 816–821.

[26] S. Stankovic, N. Ilic, Z. Djurovic, M. Stankovic, and K. Johansson, "Consensus based overlapping decentralized fault detection and isolation," *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pp. 570–575, 2010.

[27] R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis with overlapping decompositions and consensus filters," *Proc. American Control Conf. (ACC '07)*, pp. 693—698, 2007.

[28] X. Zhang, "Decentralized fault detection for a class of large-scale nonlinear uncertain systems," in *Proc. 2010 American Control Conference*, 2010, pp. 5650–5655.

[29] R. Ferrari, T. Parisini, and M. Polycarpou, "Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach," *Automatic Control, IEEE Transactions on*, vol. 57, no. 2, pp. 275 –290, 2012.

[30] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Trans. on Automatic Control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[31] D. M. Stipanovič, Inalhan, Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[32] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. on Automatic Control*, vol. 48, no. 9, pp. 1478–1495, 2003.

[33] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Trans. on Automatic Control*, vol. 51, no. 3, pp. 401—420, 2006.

[34] W. Dunbar and R. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[35] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[36] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach," *IEEE Trans. on Automatic Control*, vol. 53, no. 1, pp. 324–338, 2008.

[37] M. M. Polycarpou, G. Uber, Z. Wang, F. Shang, and Brdys, "Feedback control of water quality," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 68–87, 2002.

[38] M. Baglietto, T. Parisini, and R. Zoppoli, "Distributed-information neural control: the case of dynamic routing in traffic networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 3, pp. 485–502, 2001.

[39] S. Boccaletti, J. Kurths, G. Osipov, and D. Valladares, "The synchronization of chaotic systems," *Phys. Rep*, vol. 366, pp. 1–101, 2002.

[40] W. Wang and J.-J. E. Slotine, "A theoretical study of different leader roles in networks," *IEEE Trans. on Automatic Control*, vol. 51, no. 7, pp. 1156–1161, 2006.

[41] J. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[42] R. Raffard, C. J. Tomlin, and S. Boyd, "Distributed optimization for cooperative agents: application to formation flight," *Proc. IEEE Conference on Decision and Control*, vol. 3, pp. 2453–2349, 2004.

[43] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control," *Control Systems Magazine*, 2007.

[44] V. Kapila, A. G. Sparks, J. Buffington, and Q. Yan, "Spacecraft formation flying: Dynamics and control," *Journal of Guidance, Control and Dynamics*, vol. 23, no. 3, pp. 561–564, 2000.

[45] D. Scharf, F. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control. part ii: control," *Proc. American Control Conf. (ACC '04)*, vol. 4, pp. 2972–2985, 2004.

[46] J. Carpenter, "Decentralized control of satellite formations," *Int. J. Robust Nonlinear Control*, vol. 12, no. 2-3, pp. 141–161, 2002.

[47] J. Lavaei, A. Momeni, and A. G. Aghdam, "A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 268–278, 2008.

[48] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947—951, 2001.

[49] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 24—33, 2004.

[50] H. Tanner, G. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, 2004.

[51] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324–333, 2008.

[52] J. Cortés, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Trans. on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[53] S. Mastellone, D. M. Stipanovič, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[54] R. Teo and C. J. Tomlin, "Computing danger zones for provably safe closely spaced parallel approaches," *Journal of Guidance, Control and Dynamics*, vol. 26, no. 3, pp. 434–442, 2003.

[55] P. Li, L. Alvarez, and R. Horowitz, "Ahs safe control laws for platoon leaders," *IEEE Trans. on Automatic Control*, vol. 5, no. 6, pp. 614–628, 1997.

[56] S. Stankovič, M. Stanojevič, and D. Šiljak, "Decentralized overlapping control of a platoon of vehicles," *IEEE Trans. on Automatic Control*, vol. 8, no. 5, 2000.

[57] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25—34, 1987.

[58] T. Vicsek, Czirók, Ben-Jacob, and Cohen, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, 1995.

[59] T. Vicsek, "A question of scale," *Nature*, vol. 411, p. 421, 2001.

[60] ——, "The bigger picture," *Nature*, vol. 418, p. 131, 2002.

[61] I. Farkas, D. Helbing, and T. Vicsek, "Mexican waves in an excitable medium," *Nature(London)*, 2002.

[62] Y. Liu and K. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Trans. on Automatic Control*, vol. 49, no. 1, pp. 30–43, 2004.

[63] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 9, pp. 814—818, 2005.

[64] G. Palla, A. L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[65] H. V. D. Parunak, "" go to the ant": Engineering principles from natural multi-agent systems," *Annals of Operations Research*, vol. 75, no. 0, pp. 69–101, 1997.

[66] M. Ballerini, N. Cabibbo, R. Candelier, and A. Cavagna, "Empirical investigation of starling flocks: a benchmark study in collective animal behaviour," *Animal Behaviour*, 2008.

[67] A. Jadbabaie, J. Lin, and S. A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[68] E. Klavins, "Programmable self-assembly," *Control Systems Magazine*, vol. 27, no. 4, pp. 43–56, 2007.

[69] S. Martinez and J. Cortes, "Motion coordination with distributed information," *Control Systems Magazine*, 2007.

[70] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.

[71] Sinopoli, Sharp, Schenato, and Schaffert, "Distributed control applications within sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1235–46, 2003.

[72] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[73] A. Speranzon, C. Fischione, and K. Johansson, "Distributed and collaborative estimation over wireless sensor networks," *Proc. IEEE Conference on Decision and Control*, pp. 1025–1030, 2006.

[74] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, 2007.

[75] R. Olfati-Saber and R. Murray, "Consensus protocols for networks of dynamic agents," *Proc. American Control Conf. (ACC '03)*, vol. 2, pp. 951—956, 2003.

[76] ——, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[77] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," *Proc. American Control Conf.(ACC '05)*, vol. 3, no. 1859–1864, 2005.

[78] R. Olfati-Saber, E. Franco, Frazzoli, and J. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," *Lecture Notes in Control and Inform. Science*, vol. 331, pp. 169–182, 2006.

[79] M. Mehyar, D. Spanos, J. Pongsajapan, S. Low, and R. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 512–520, 2007.

[80] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[81] S. Roy, A. Saberi, and K. Herlugson, "A control-theoretic perspective on the design of distributed agreement protocols," *Int. J. Robust Nonlinear Control*, vol. 17, no. 1034-1066, 2007.

[82] A. Fagiolini, E. M. Visibelli, and A. Bicchi, "Logical consensus for distributed network agreement," *Proc. IEEE Conference on Decision and Control*, pp. 5250–5255, 2008.

[83] Q. Hui and W. M. Haddad, "Distributed nonlinear control algorithms for network consensus," *Automatica*, vol. 44, no. 9, pp. 2375–2381, 2008.

[84] M. S. Stankovič, S. Stankovič, and D. M. Stipanovič, "Consensus based multi-agent control structures," *Proc. IEEE Conference on Decision and Control*, p. 6, 2008.

[85] F. Garin and L. Schenato, "A survey on distributed estimation and control applications using linear consensus algorithms," in *Networked Control Systems*, ser. Lecture Notes in Control and Information Sciences, A. Bemporad, M. Heemels, and M. Johansson, Eds. Springer Berlin / Heidelberg, 2010, vol. 406, pp. 75–107.

[86] A. Speranzon, C. Fischione, K. Johansson, and A. Sangiovanni-Vincentelli, "A distributed minimum variance estimator for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 609–621, 2008.

[87] F. Cattivelli and A. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035 –1048, 2010.

[88] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622 –633, 2008.

[89] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *Decision and Control, 46th IEEE Conference on*, 2007, pp. 5492 –5498.

[90] ——, "Kalman-consensus filter : Optimality, stability, and performance," in *Decision and Control, 48th IEEE Conference on, held jointly with 28th Chinese Control Conference*, 2009, pp. 7036 –7042.

[91] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.

[92] R. Olfati-Saber and P. Jalalkamali, "Collaborative target tracking using distributed kalman filtering on mobile sensor networks," in *American Control Conference*, 2011, pp. 1100 –1105.

[93] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc wsns with noisy links;part i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350 –364, 2008.

[94] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Distributed moving horizon estimation for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462 –2475, 2010.

[95] S. Wang and E. Davidson, "On the stabilization of decentralized control systems," *IEEE Trans. on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.

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

[97] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Trans. on Automatic Control*, vol. 23, no. 2, pp. 108–128, 1978.

[98] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, "Diagnosis of large active systems," *Artificial Intelligence*, vol. 110, no. 1, pp. 135–189, 1999.

[99] Kurien, Koutsoukos, and Zhao, "Distributed diagnosis of networked, embedded systems," *Proc. of the 13th Int. Workshop on Principles of Diagnosis (DX-2002)*, pp. 179–188, 2002.

[100] Wu and C. N. Hadjicostis, "Distributed non-concurrent fault identification in discrete event systems," *Proc. CESA*, 2003.

[101] Rish, Brodie, Ma, Odintsova, and Beygelzimer, "Adaptive diagnosis in distributed systems," *IEEE Trans. on Neural Networks (special issue on Adaptive Learning Systems in Communication Networks)*, vol. 16, no. 10, pp. 1088–1109, 2005.

[102] E. Athanasopoulou and C. N. Hadjicostis, "Probabilistic approaches to fault detection in networked discrete event systems," *Neural Networks*, vol. 16, no. 5, pp. 1042—1051, 2005.

[103] T. Le and C. N. Hadjicostis, "Graphical inference methods for fault diagnosis based on information from unreliable sensors," *Proc. 9th Int. Conf. on Control, Automation, Robotics and Vision, 2006 ICARCV 2006*, p. 6, 2006.

[104] Y. Wang, T. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dynamic Systems*, vol. 2, pp. 233–263, 2007.

[105] F. Preparata, G. Metze, and R. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Transactions on Electronic Computers*, vol. EC-16, no. 6, pp. 848–854, 1967.

[106] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[107] J. Kuhl and S. Reddy, "Fault-diagnosis in fully distributed systems," *Twenty-Fifth Int. Symp. on Fault-Tolerant Computing, 1995, 'Highlights from Twenty-Five Years'*, 1995.

[108] T. D. Chandra and S. Toueg, "Unreliable failure detectors for reliable distributed systems," *J. Assoc. Comput. Mach.*, vol. 43, no. 2, pp. 225–267, 1996.

[109] Pike, Miner, and T. Wilfredo, "Model checking failed conjectures in theorem proving: A case study," *NASA Technical Report*, no. TM–2004–213278, 2004.

[110] X. Yang and Y. Tang, "Efficient fault identification of diagnosable systems under the comparison model," *IEEE Transactions on Computers*, vol. 56, no. 12, pp. 1612–1618, 2007.

[111] A. Fagiolini, G. Valenti, L. Pallottino, and G. Dini, "Decentralized intrusion detection for secure cooperative multi-agent systems," *Proc. IEEE Conference on Decision and Control*, 2007.

[112] C. Edwards, L. M. Fridman, and M.-W. Thein, "Fault reconstruction in a leader/follower spacecraft system using higher order sliding mode observers," *Proc. American Control Conf. (ACC '07)*, pp. 408—413, 2007.

[113] Q. Wu and M. Saif, "Robust fault detection and diagnosis for a multiple satellite formation flying system using second order sliding mode observer and wavelet networks," *Proc. American Control Conf. (ACC '07)*, 2007.

[114] N. Meskin, K. Khorasani, and C. A. Rabbath, "Fault consensus in a network of unmanned vehicles," in *Proc. of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess 2009)*, Barcelona, Spain, 2009, pp. 1001–1006.

[115] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry, "Cyber security analysis of state estimators in electric power systems," *Proc. IEEE Conference on Decision and Control*, 2010.

[116] S. Jafari, A. Ajorlou, A. G. Aghdam, and S. Tafazoli, "On the structural controllability of multi-agent systems subject to failure: A graph-theoretic approach," *Proc. IEEE Conference on Decision and Control*, 2010.

[117] Y. Murphey, J. Crossman, Chen, and Cardillo, "Automotive fault diagnosis–Part II: A distributed agent diagnostic system," *IEEE Trans. on Vehicular Technology*, vol. 52, no. 4, 2003.

[118] Roychoudhury, Biswas, K. Xenofon, and Abdelwahed, "Designing distributed diagnosers for complex physical systems," *Proc. 16th International Workshop on Principles of Diagnosis*, 2005.

[119] W. H. .Chung, J. L. Speyer, and R. H. Chen, "A decentralized fault detection filter," *J. of Dynamic Systems, Measurement and Control*, vol. 123, pp. 237–247, 2001.

[120] N. Lechevin, C. A. Rabbath, and E. Earon, "Towards decentralized fault detection in uav formations," *Proc. American Control Conf. (ACC '07)*, pp. 5759—5764, 2007.

[121] X. Zhang and Q. Zhang, "Distributed fault diagnosis in a class of interconnected nonlinear uncertain systems," *International Journal of Control*, vol. 85, no. 11, pp. 1644–1662, 2012.

[122] V. Reppa, M. Polycarpou, and C. Panayiotou, "Distributed sensor fault detection and isolation for nonlinear uncertain systems," in *Proc. of Safeprocess Conference*, 2012, pp. 1077–1082.

[123] A. Vemuri and M. Polycarpou, "Robust nonlinear fault diagnosis in input-output systems," *International Journal of Control*, vol. 68, no. 2, pp. 343–360, 1997.

[124] Q. Zhang, M. Basseville, and A. Benveniste, "Fault detection and isolation in nonlinear dynamic systems: A combined input–output and local approach," *Automatica*, vol. 34, no. 11, pp. 1359–1373, 1998.

[125] X. Zhang, M. M. Polycarpou, and T. Parisini, "Robust fault isolation for a class of non–linear input–output systems," *Int. Journal of Control*, vol. 74, no. 13, pp. 1295–1310, 2001.

[126] K. Subbarao and A. Vemuri, "Fault isolation using extrinsic curvature for multi-input-multi-output systems with nonlinear fault models," *Proc. American Control Conf. (ACC '07)*, 2007.

[127] Z. Zhang and I. Jaimoukha, "Fault detection and isolation for linear discrete-time systems using input/output measurement analysis," in *Proc. 48th IEEE Conf. on Decision and Control, held jointly with the 28th Chinese Control Conference.*, 2009, pp. 4908–4913.

[128] F. Boem, R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "A distributed fault detection methodology for a class of large-scale uncertain input-output discrete-time nonlinear systems," in *IEEE Proc. Conf. on Decision and Control and European Control Conf.*, no. 5-6, 2011, pp. 603–620.

[129] P. Tabuada, "Cyber physical systems: Position paper," in *NSF Workshop on Cyber-Physical Systems*, 2006.

[130] E. A. Lee, "Cyber-physical systems - are computing foundations adequate?" in *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, October 2006.

[131] E. Lee, "Cyber physical systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, may 2008, pp. 363 –369.

[132] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *Machine Learning in Cyber Trust*. Springer US, 2009, pp. 3–13.

[133] W. Wolf, "Cyber-physical systems," *Computer*, vol. 42, no. 3, pp. 88 –89, march 2009.

[134] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10. New York, NY, USA: ACM, 2010, pp. 731–736.

[135] A. A. Cardenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11. New York, NY, USA: ACM, 2011, pp. 355–366.

[136] F. Dorfler, F. Pasqualetti, and F. Bullo, "Distributed detection of cyber-physical attacks in power networks: A waveform relaxation approach," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, sept. 2011, pp. 1486 –1491.

[137] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems – part i: Models and fundamental limitations," *ArXiv e-prints*, Feb. 2012.

[138] ——, "Attack detection and identification in cyber-physical systems – part ii: Centralized and distributed monitor design," *ArXiv e-prints*, Feb. 2012.

[139] F. Pasqualetti, F. Dorfler, and F. Bullo, "Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design," in *Decision and Control and European Control Conference (CDC-ECC), 50th IEEE Conference on*, dec. 2011, pp. 2195 –2201.

[140] A. Teixeira, H. Sandberg, and K. Johansson, "Networked control systems under cyber attacks with applications to power networks," in *American Control Conference (ACC), 2010*, 30 2010-july 2 2010, pp. 3690 –3696.

[141] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757 – 2764, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109811004511

[142] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Online fault detection of sensor measurements," in *Sensors, 2003. Proceedings of IEEE*, vol. 2, oct. 2003, pp. 974 – 979 Vol.2.

[143] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, ser. DIWANS '06. New York, NY, USA: ACM, 2006, pp. 65–72. [Online]. Available: http://doi.acm.org/10.1145/1160972.1160985

[144] J. Liu, D. M. de la Peña, and P. D. Christofides, "Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements," *Automatica*, vol. 46, no. 1, pp. 52 – 61, 2010.

[145] E. Franco, R. Olfati-Saber, T. Parisini, and M. Polycarpou, "Distributed fault diagnosis using sensor networks and consensus-based filters," in *Decision and Control, 45th IEEE Conference on*, 2006, pp. 386 –391.

[146] F. Boem, Y. Xu, C. Fischione, and T. Parisini, "A distributed estimation method for sensor networks based on pareto optimization," in *Proc. of 51th Conference on Decision and Control*, 2012, pp. 775–781.

[147] A. Al-Nayeem, L. Sha, D. Cofer, and S. Miller, "Pattern-based composition and analysis of virtually synchronized real-time distributed systems," in *Cyber-Physical Systems (ICCPS), 2012 IEEE/ACM Third International Conference on*, april 2012, pp. 65 –74.

[148] D. Cofer, "Complexity-reducing design patterns for cyber-physical systems," *AFRL Technical Report*, 2011.

[149] A. Benveniste, "Loosely time-triggered architectures for cyber-physical systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '10. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2010, pp. 3–8.

[150] G. Pin and T. Parisini, "Networked predictive control of uncertain constrained nonlinear systems: Recursive feasibility and input-to-state stability analysis," *Automatic Control, IEEE Transactions on*, vol. 56, no. 1, pp. 72 –87, 2011.

[151] N. Kottenstette, X. Koutsoukos, J. Hall, J. Sztipanovits, and P. Antsaklis, "Passivity-based design of wireless networked control systems for robustness to time-varying delays," in *Real-Time Systems Symposium, 2008*, 30 2008-dec. 3 2008, pp. 15 –24.

[152] Y. Zheng, H. Fang, and H. Wang, "Takagi-sugeno fuzzy-model-based fault detection for networked control systems with markov delays," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 4, pp. 924 –929, 2006.

[153] A. Bemporad, "Predictive control of teleoperated constrained systems with unbounded communication delays," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 2, 1998, pp. 2133 –2138 vol.2.

[154] Z. Wang, D. Ho, and X. Liu, "Variance-constrained filtering for uncertain stochastic systems with missing measurements," *Automatic Control, IEEE Transactions on*, vol. 48, no. 7, pp. 1254 – 1258, 2003.

[155] Z. Wang, F. Yang, D. Ho, and X. Liu, "Robust control for networked systems with random packet losses," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 4, pp. 916 –924, 2007.

[156] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *Automatic Control, IEEE Transactions on*, vol. 50, no. 8, pp. 1177 – 1181, 2005.

[157] E. Franco, L. Magni, T. Parisini, M. Polycarpou, and D. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach," *Automatic Control, IEEE Transactions on*, vol. 53, no. 1, pp. 324 –338, 2008.

[158] Y. Wang, S. Ding, H. Ye, and G. Wang, "A new fault detection scheme for networked control systems subject to uncertain time-varying delay," *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 5258 –5268, 2008.

[159] X. He, Z. Wang, and D. Zhou, "Robust fault detection for networked systems with communication delay and data missing," *Automatica*, vol. 45, no. 11, pp. 2634 – 2639, 2009.

[160] P. Zhang, S. X. Ding, G. Z. Wang, and D. H. Zhou, "Fault detection for multirate sampled-data systems with time delays," *International Journal of Control*, vol. 75, no. 18, pp. 1457–1471, 2002.

[161] H. Ye and S. Ding, "Fault detection of networked control systems with network-induced delay," in *Control, Automation, Robotics and Vision Conference, 2004.*, vol. 1, 2004, pp. 294 – 297 Vol. 1.

[162] W. Qiu and R. Kumar, "Distributed diagnosis under bounded-delay communication of immediately forwarded local observations," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 3, pp. 628 –643, 2008.

[163] F. Boem, Y. Xu, C. Fischione, and T. Parisini, "A distributed estimation method for sensor networks based on pareto optimization," in *Proc. of 51th Conference on Decision and Control*, 2012.

[164] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu, "Tiny-sync: Tight time synchronization for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 3, 2007.

[165] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 2289 –2294.

[166] X.-l. Zhang, X.-q. Tang, and J.-h. Chen, "Time synchronization of hierarchical real-time networked cnc system based on ethernet/internet," *The International Journal of Advanced Manufacturing Technology*, vol. 36, pp. 1145–1156, 2008.

[167] P. Varutti and R. Findeisen, "On the synchronization problem for the stabilization of networked control systems over nondeterministic networks," in *American Control Conference.*, 2009, pp. 2216 –2221.

[168] P. L. Tang and C. de Silva, "Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control," *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 4, pp. 707 – 718, 2006.

[169] R. Patton, P. Frank, and D. Clark, *Fault Diagnosis in Dynamic Systems: Theory and Application.* Upper Saddle River, NJ, USA: Prentice Hall, 1989.

[170] J. M. Fowler and R. D'Andrea, "A formation flight experiment," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 35—43, 2003.

[171] M. M. Polycarpou and A. Helmicki, "Automated fault detection and accommodation: a learning systems approach," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 25, no. 11, pp. 1447–1458, 1995.

[172] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1999.

[173] M. Vidyasagar, "Decomposition techniques for large-scale systems with nonadditive interactions: Stability and stabilizability," *IEEE Trans. on Automatic Control*, vol. AC-25, no. 4, pp. 773–779, 1980.

[174] G. Chartrand and O. Oellermann, *Applied and Algorithmic Graph Theory*. McGraw-Hill College, 1992.

[175] S. Stankovič, M. S. Stankovič, and D. M. Stipanovič, "Consensus based overlapping decentralized estimator," *Proc. American Control Conf. (ACC '07)*, pp. 2744—2749, 2007.

[176] C. Langbort, R. Chandra, and R. D'Andrea, "Distributed control design for systems interconnected over an arbitrary graph," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1502–1519, 2004.

[177] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[178] S. C. Kadu, M. Bhushan, and R. Gudi, "Optimal sensor network design for multirate systems," *Journal of Process Control*, vol. 18, no. 6, pp. 594 – 609, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152407001473

[179] D. Jourdan and O. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Vehicular Technology Conference, IEEE 59th*, vol. 5, 2004, pp. 2466 – 2470.

[180] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.120.6062

[181] F. Boem, Y. Xu, C. Fischione, and T. Parisini, "Distributed fault detection using sensor networks and pareto estimation," in *Proc. of the European Control Conference (to appear)*, 2013.

[182] D. Carnevale, A. R. Teel, and D. Nesic, "A lyapunov proof of an improved maximum allowable transfer interval for networked control systems," *Automatic Control, IEEE Transactions on*, vol. 52, no. 5, pp. 892 –897, may 2007.

[183] M. M. Polycarpou, "On–line approximators for nonlinear system identification: a unified approach," in *Control and Dynamic Systems: Neural Network Systems Techniques and Applications*, X. Lenodes, Ed. New York: Academic, 1998, vol. 7, pp. 191–230.

[184] F. Boem, R. Ferrari, T. Parisini, and M. Polycarpou, "Distributed fault detection for uncertain nonlinear systems: a network delay compensation strategy," in *Proc. American Control Conference (to appear)*, 2013.

[185] M. Sichitiu and P. Bauer, "Stability of discrete time-variant linear delay systems and applications to network control," in *Electronics, Circuits and Systems, ICECS 2001. The 8th IEEE International Conference on*, vol. 2, 2001, pp. 985 –989 vol.2.

[186] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *Automatic Control, IEEE Transactions on*, vol. 50, no. 5, pp. 655 – 661, 2005.

[187] R. M. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis of large-scale discrete-time nonlinear systems: New results on the isolation problem," *Proc. IEEE Conference on Decision and Control*, pp. 1619–1626, 2010.

[188] R. Isermann, "Supervision, fault-detection and fault-diagnosis methodsŮan introduction," *Control engineering practice*, vol. 5, no. 5, pp. 639–652, 1997.

[189] B. Koppen-Seliger and S. Ding, "Fault detection and isolation of a three tank benchmark," in *Proc. European Control Conference*, 1999.

[190] F. Boem, R. M. G. Ferrari, and T. Parisini, "Distributed fault detection and isolation of continuous-time nonlinear systems," *Europ. J. of Control*, no. 5-6, pp. 603–620, 2011.

[191] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Proc. 4th Int. Symp. on Information Process. in Sensor Netw. (IPSN '05)*, pp. 63–70, 2005.

[192] L. Xiao, S. Boyd, and S. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

[193] C. D. Meyer, Ed., *Matrix analysis and applied linear algebra*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000.

[194] G. Goodwin and Sin, *Adaptive filtering prediction and control*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[195] C. R. Johnson, *Lectures on adaptive parameter estimation*. Upper Saddle River, NJ, USA: Prentice Hall, 1988.

[196] J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd ed. Prentice Hall, 1994.

[197] F. Lewis, Yeşildirek, and S. Jagannathan, *Neural Network Control of Robot Manipulators and Non-Linear Systems*. CRC Press, 1998.

[198] J. Farrell and M. M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy, and Traditional Adaptive Approximation Approaches*. Hoboken, NJ: Wiley-Interscience, 2006.

[199] R. M. Ferrari, T. Parisini, and M. M. Polycarpou, "A robust fault detection and isolation scheme for a class of uncertain input-output discrete-time nonlinear systems," *Proc. American Control Conf. (ACC '08)*, p. 6, 2008.

[200] F. Boem, R. M. G. Ferrari, C. Keliris, T. Parisini, and M. M. Polycarpou, "A distributed fault diagnosis approach for large-scale cyber-physical systems," *IEEE Transactions on Automatic Control (submitted to)*, 2013.

[201] M. Campi and E. Weyer, "Guaranteed non-asymptotic confidence regions in system identification," *Automatica*, vol. 41, no. 10, pp. 1751 – 1764, 2005.

[202] E. Weyer, S. Ko, and M. Campi, in *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2009, pp. 289 –294.

[203] R. Ferrari, T. Parisini, and M. Polycarpou, "An algebraic approach for robust fault detection of input-output elastodynamic distributed parameter systems," in *Proc. of the European Control Conference (to appear)*, 2013.

[204] E. Tonti, "A direct discrete formulation of field laws- the cell method," *CMES-Computer Modeling in Engineering and Sciences*, vol. 2, no. 2, pp. 237–258, 2001.