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

XX CICLO DEL
DOTTORATO DI RICERCA IN
INGEGNERIA DELL'INFORMAZIONE

Distributed Fault Detection and Isolation of Large-scale Nonlinear Systems: an Adaptive Approximation Approach

(Settore scientifico-disciplinare ING-INF/04)

DOTTORANDO
Riccardo Ferrari

COORDINATORE DEL COLLEGIO DEI DOCENTI
Chiar.mo Prof. **Roberto Vescovo**
Università degli Studi di Trieste

TUTORE E RELATORE
Chiar.mo Prof. **Thomas Parisini**
Università degli Studi di Trieste

CORRELATORE
Chiar.mo Prof. **Marios M. Polycarpou**
University of Cyprus

ANNO ACCADEMICO 2007/2008

Dedicated to the memory of
my grandfather Willy,
my first mentor.

Preface

The present thesis work has been submitted to the Department of Electrical, Electronic and Computer Engineering of the University of Trieste, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Information Engineering. The work has been supported by a grant financed by Danieli Automation S.p.A, whose subject was "Numerical Modeling in the Steel–making Industry".

The thesis work deals with the problem of fault diagnosis for distributed systems. Fault diagnosis is a key requirement in the design of modern and reliable systems. While the fault diagnosis problem has been extensively studied for centralized systems, and for some class of distributed systems such as multi–processor systems, a lack of research is apparent when looking for results about the fault diagnosis of distributed discrete and continuous–time systems. The objective of the present work is to fill this gap, by applying a divide et impera paradigm to actual fault diagnosis schemes in order to make the problem of fault diagnosis for distributed systems tractable, and the corresponding solutions robust and reliable themselves.

I wish here to profoundly thank my advisor, prof. Thomas Parisini, that supported and guided me with painstaking patience and efforts since my master's degree. He was a mentor, and a great friend during the last, intense years. He helped me avoid the pitfalls of research, and continuously motivated me in order to reach my objectives. Another expression of my gratitude is directed to my co–advisor, prof. Marios M. Polycarpou, whose experience and analytical skillfulness was invaluable during the development of all the theory providing the foundations for my work. His care and enduring support made my stay as a visiting student at the University of Cyprus a most profitable, and unforgettable experience.

My deep thankfulness goes also to Danieli Automation S.p.A., that made this research possible, and in particular to all the people of the R&D laboratory with whom I worked side by side during these years, on many practical problems too. If I ever developed an experimental attitude, it is thank to the countless hours I was allowed to spend in that laboratory, and thank to the skilled advice I was always given.

I want also to acknowledge the contribution of prof. Marcel Staroswiecky, with whom I had many fruitful discussions during a short stay at the Ecole

Normale Supérieure de Cachan, in Paris. A special thank goes to prof. Enzo Tonti, that instilled in me the true love for Science while guiding me in my master's degree thesis work, and while helping me during the first part of my Ph.D.

I want to thank all the fellow students with whom I shared my experience, many awesome moments and some difficult ones, during these years: Gilberto, Eric, Elisa, Daniele, Felice, Marco, Andrea, Federica, my great friends from the DEEI Department at the University of Trieste; and Demetrios, with whom I shared the office and much more during my stay in the ECE Department at the University of Cyprus. You made these years an experience worth to be remembered for my whole life.

Last, but not least, my sincere love goes to my family, that unselfishly supported me with continuous efforts in all these years, without showing any sign of fatigue. And my final "thank you" is for Elisa, she knows that she is my life and that everything I do, I do it for her.

Contents

List of Figures

Notation

In this chapter the main abbreviations and the mathematical symbols used in the remainder of the thesis will be summarized. When indexes are needed, we will make use of the symbols i, I, and j, J, and k, K. Lower–case indexes are used to denote vector components, while an upper–case subindex will always denote that a quantity belongs to a given subsystem or agent.

Graph theory specific definitions can be found in Chapter 3, while other seldom-used symbols are defined throughout the text as needed.

ARR	Analytical Redundancy Relation
DFDI	Distributed Fault Detection and Isolation (and Identification)
DOS	Dedicated Observer Scheme
FDAE	Fault Detection and Approximation Estimator
FDI	Fault Detection and Isolation (and Identification)
FIE	Fault Isolation Estimator
GOS	Generalized Observer Scheme
LFD	Local Fault Diagnoser
LTI	Linear Time Invariant
MFAE	Minimum Functional Approximation Error
RBF	Radial Basis Function, a kind of activation function for neural networks
$\lvert\cdot\rvert$	absolute value, taken component–wise if the argument is not a scalar, or cardinality if the argument is a set
$\lVert\cdot\rVert$	Euclidean vector norm
$\lVert\cdot\rVert_F$	Frobenius matrix norm

$\mathbb{R}$, $\mathbb{R}_+$	set of real numbers and of non–negative real numbers
$\mathbb{N}$, $\mathbb{N}_+$	set of natural numbers and of non–negative natural numbers
$\mathbb{Z}$, $\mathbb{Z}_+$	set of integer numbers and of non–negative integer numbers
$\mathbb{R}^n$, $\mathbb{N}^n$, $\mathbb{Z}^n$	n–dimensional space with real, natural and integer, respectively, coordinates
$\mathscr{S}$	a generic monolithic system
$\mathcal{A} \triangleq \{a_1, \ldots, a_N\}$	a non–ordered set or multiset
$\mathcal{A} \triangleq (a_1, \ldots, a_N)$	an ordered set or multiset
a	a vector is denoted with a lower–case symbol
$a^{(i)}$	i–th component of a vector
$a \triangleq \mathrm{col}(a^{(1)}, \ldots, a^{(N)})$	a vector built by using the col operator
$\bar{a}$	a bound, usually an upper bound, on the norm of a vector (or the absolute value of a scalar)
$\hat{a}$	an estimate of a
$\tilde{a} \triangleq a - \hat{a}$	estimation error on a
$x(t)$, $u(t)$	value at time t of the state and input vectors of the monolithic system $\mathscr{S}$
$f(\cdot)$	nominal dynamics of the monolithic system $\mathscr{S}$
$\eta(\cdot)$	uncertainty in the dynamics of the monolithic system $\mathscr{S}$
$\phi(\cdot)$	fault function affecting the dynamics of the monolithic system $\mathscr{S}$
$\beta(\cdot)$	time evolution of the fault magnitude
T_0	fault event instant
b	parameter describing the time evolution of an incipient fault
$\mathcal{F}$	fault class for the monolithic system $\mathscr{S}$

$\phi_j(\cdot)$	j–th element of the fault class $\mathcal{F}$
$\mathscr{H}_j$	fault hypothesis associated to ϕ_j
$H_{j,i}(\cdot)$	i–th structural function of ϕ_j
$\vartheta_{j,i}$	i–th parameter vector of ϕ_j
$\Theta_{j,i}$	domain to which $\vartheta_{j,i}$ belongs
$M_{\Theta_{j,i}}$	radius of the hyper-sphere that contains $\Theta_{j,i}$
$\mathcal{R} \triangleq \mathcal{R}^x \times \mathcal{R}^u$	stability regions for the monolithic system $\mathscr{S}$
$\hat{x}_0$	state estimation provided by the FDAE
λ	FDAE and FIE filter pole
$\epsilon_0 \triangleq x - \hat{x}_0$	state estimation error of the FDAE
$\bar{\epsilon}_0$	detection threshold on ϵ_0
T_d	fault detection instant
$\hat{\phi}_0(\cdot)$	on–line adaptive approximator for ϕ_0
ν_0	MFAE of $\hat{\phi}_0$
$\mathcal{P}_{\mathcal{A}}(\cdot)$	a projection operator on the domain $\mathcal{A}$
$\hat{x}_j$	state estimation provided by the j–th FIE
$\epsilon_j \triangleq x - \hat{x}_j$	state estimation error of the j–th FIE
$\bar{\epsilon}_j$	detection threshold on ϵ_j
$\Delta_{k,j}\phi(\cdot)$	mismatch function between the k–th and the j–th fault function
$\mathcal{D}$	decomposition of the monolithic system $\mathscr{S}$
$\mathscr{S}_I$	I–th subsystem contained in $\mathcal{D}$
$\mathcal{I}_I$	index set used in defining $\mathscr{S}_I$
$\mathcal{J}_I$	neighbors index set of subsystem $\mathscr{S}_I$
$\mathscr{L}_I$	I–th LFD
$\mathcal{O}_s$	overlap index sets of variable $x^{(s)}$
W_s	weighted adjacency matrix of the consensus protocol on $x^{(s)}$

x_I, u_I, z_I	local state, local input and interconnection vectors of $\mathscr{S}_I$
$f_I(\cdot)$, $g_I(\cdot)$	nominal and interconnection functions of $\mathscr{S}_I$
$\hat{g}_I(\cdot)$	on–line adaptive approximator for g_I
ν_I	MFAE for $\hat{g}_I$
ξ_I	measurement uncertainty on x_I
ζ_I	measurement uncertainty on z_I
χ_I	total uncertainty on x_I
$\hat{x}_{I,0}$, $\epsilon_{I,0}$, ...	all the other quantities pertaining to the I–th subsystem $\mathscr{S}_I$ are denoted as the ones pertaining to $\mathscr{S}$, but with an added leading subindex I
$\mathcal{S}$	global fault signature associated to the monolithic system $\mathscr{S}$
$\mathcal{S}_I$	local fault signature associated to the subsystem $\mathscr{S}_I$

Chapter 1

Introduction

How do we perceive the usefulness of technology? The most straightforward answer, is that we perceive it through all the things that technology can do for us: things that we are unable to do ourselves, either because we are physically unable to, or because we are unwilling to. For instance, a transportation system can bring us, and also a wealthy amount of goods, to destinations further away, and in a much shorter time than what we can do by relying on our limited physical strength alone. Furthermore, a personal computer can do more computations and store more informations, and do it faster than our mind. And yet, a washing machine or other household appliances can execute tasks that we consider too menial to be worth our time. These are fairly simple examples of technologies that we deem useful, because they enable us to accomplish things we would have been unable to, or simply because they give us more free time to engage in more creative activities.

If we consider more deeply our concept of usefulness, anyway, we would realize that what we have said constitutes just half of the answer. In fact a technology is not useful only because of the amount of things it can do for us, but only if it can also do those things exactly when we need, and as long as we need them. For instance, an automobile whose engine will suddenly stop and refuse to start again, leaving us in the middle of our journey, will not be useful. Moreover, a computer that will suddenly crash without having given us any prior notice, so that we could back-up our informations, will not be useful either. These are examples of technology that fails in providing the services we expect from it, thus causing some damage to us. But it must be understood that, depending on the situation, the amount of damage due to technical equipment failing may be unbearable. Should the engine of a plane, instead of that of a car, suddenly shut off this may lead to a very dangerous situation, that can even culminate in an air disaster, killing human beings. Should a computer system governing the functions of a complex and expensive industrial plant fail, this may result in extensive

damage and losses, and again may cause death or impairments to people.

The natural conclusion of this short semantic investigation, is that technology is useful only as long as it can provide a service to us in a *reliable* way, otherwise it is not only disadvantageous, but it may be dangerous as well. We stressed the word *reliable*, as in fact reliability is one of the key concerns in the design of modern dependable and safe technical systems. *Reliability* can be defined as the ability of a system to perform its intended function over a given period of time [1]. The inability to perform the intended function is called a *failure*, and it can be due to the effects of a fault. A *fault* is a change in the behavior of a system, or part of it, from the behavior that was set at design time. Logic then dictates that everything that cannot be accounted for the effects of a fault, should be blamed as wrong design, and though this appears to be a too common cause of failures, in this work we will concentrate only on faults.

A wrong conclusion that may be drawn from the definitions of failure and faults, is that in order to avoid failures a system must be designed so that faults will never happen, as faults will lead to failures. Luckily enough this is not true, otherwise no reliable system will ever be built, as the task of designing physical systems that do not undergo faults has been proven to be impossible. In fact experience tells us that not every fault will lead to a failure, as usually systems are robust enough so that they can withstand some faults while still providing their service, or at least a somewhat degraded version of their service. Such a system is called *fault tolerant*. With a proper design, fault tolerance may be an intrinsic property of some *redundant* systems, thus requiring no further effort. For instance, in a suspended bridge with a great number of tendons, if one tendon looses its ability to support its part of the load, that is if it experiences a fault, the other tendons will simply distribute between themselves the resulting extra load, thus avoiding the collapse of the bridge. Of course this kind of intrinsic fault tolerance works only for limited faults, and for some categories of systems.

The concept of *redundancy* is central to the development of fault tolerant systems. In the example about the bridge, the thing that makes the system fault tolerant is the presence of *physical redundancy*, that is the fact that critical components of the system, the tendons, are present in a greater number than in what is strictly necessary. And the key point in avoiding the failure, is the fact that after the fault occurrence in a component, that is after a tendon breaks or looses its ability to support its load, the system automatically "switches" to other healthy components. In the bridge example this switch does not require any actual action, thus making the process transparent and hiding its details. So we will briefly analyze an example where the robustness to faults is not an intrinsic property, but requires a well defined sequence of actions. Let us consider again an airplane, where for safety reasons all the critical components, that are flight control com-

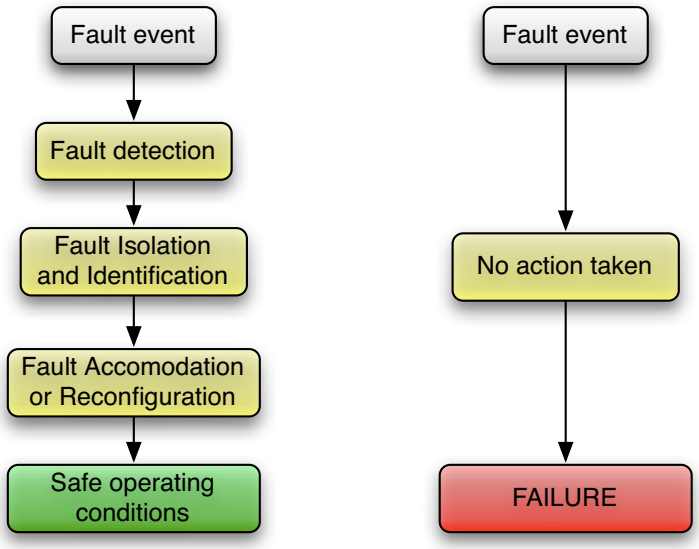


Figure 1.1: The possible effects of a fault event in the case of a fault tolerant system (left) and a non–fault tolerant system (right).

puters, actuators and sensors, present a threefold redundancy. If a critical component fails, the existence of a fault is detected by comparing the output of multiple redundant sensors, the component is isolated and taken off–line, and a spare, healthy copy of it is assigned to its task. This description uncovers the basic sequence of actions needed to implement a fault tolerant system: *detection* of a fault, *isolation and identification* of the fault and *fault accommodation* or *reconfiguration* of the system [1, 2] (Fig. 1.1).

The physical redundancy solution is highly expensive and can be justified only for critical, potentially life–threatening systems. A more affordable solution consists in the use of *analytic redundancy*, where the redundancy does not lay in having multiple physical copies of critical components, but in having a mathematical model of the healthy system. The mathematical model is used to detect and isolate faults, by comparing actual measurements to the prediction of the model, thanks to so–called *Analytical Redundancy Relations* (ARR) [3, 1]. After a successful fault diagnosis, it is not the system to be physically reconfigured to counter the effect of the fault, but it is the controller of the system that must change in order to *accommodate* the fault. The fault accommodation is possible if the system model, as well as a model of the fault, are available. A simple example may help in making this point clear. When driving a car, the car is the physical system that can fail, while the driver is its controller, that we assume here is not going to be affected by faults him- or herself. Of course an experienced driver has a mental model of the way the car behaves normally, so that by handling it he or she can tell whether something is not working properly. For instance,

if a tire is punctured and starts to go flat, a good driver will notice it and, having also a mental model of the effect of an almost flat tire, will change its way of driving in order to guarantee safety. Depending on the fault gravity, to a certain extent he or she will manage to have the car continue to provide its service, although degraded because probably the car will have to run on a much lower speed. As it should be clear, in this situation no physical redundancy did come to help, as an ordinary car does not have spare tires on each axle and waiting to be automatically deployed, but the mental model of the car and of that fault in the driver mind was used to detect and isolate the fault, and to devise a way of driving the car while facing the emergency.

In the present work the first two steps taken by a fault tolerant system, that is fault detection and isolation, will be dealt with. Specifically, a problem of *Fault Detection and Isolation* (FDI) with mathematical models, that is called *model–based fault diagnosis*, will be solved in a *distributed* way. The need for a distributed architecture is justified by the drawbacks of existing *centralized* fault diagnosis architectures when addressing actual large–scale and distributed systems. Now the fundamentals of the existing fault diagnosis methods will be summarized, and the motivations that lead to the interest in distributed architectures will be explained. Then the content of the following chapters will be anticipated.

1.1 Model–based Fault Diagnosis

Model–based fault diagnosis methods are a relatively recent accomplishment. Historically, the first methods for fault diagnosis of engineering systems did not rely on a full-mathematical model of the process to be monitored. The first known approach is the *limit checking*, that can be dated back as early as the development of the instrumentation for machines in the 19^{th} century [2], and relies only on a knowledge of the range in which each measured variable is allowed to vary. The event of a variable getting out of its bounds will be considered as due to a fault, and by analyzing which variables did cross their bounds an elementary fault isolation may be attained. Of course the success of this method is dependent upon the process working around a constant, well known set–point: in fact the event of a measured quantity getting out of its allowed range may be simply due to a human operator, or the control system, changing the operating conditions of the system. A more elaborate fault diagnosis approach was enabled by the availability of paper–drawn oscillograph, of band–pass filters and later of oscilloscopes during the first half of the 20th century, so that the behavior of the measured variables could be precisely analyzed in the time and frequency domain. These devices lead to the development of *signal–based* techniques, where known features of signals, such as spectral components or peculiar transients, were compared to nominal ones [2, 4]. As these methods

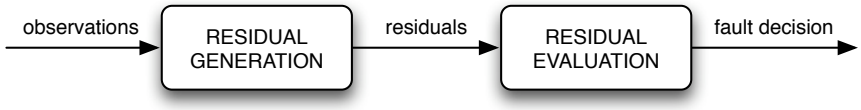


Figure 1.2: Three stages of model–based fault diagnosis, according to [14].

require some knowledge of previous behavior of the system during healthy operation, they belong to the wider class of *process history* fault diagnosis approaches [5]. An application of spectral analysis still common today is for the diagnosis of rotating machinery, where the appearance of unusual spectral components can be traced back to the impelling mechanical failure of components such as bearings or shafts [6, 7].

The model–based approach was finally made possible by the widespread use of process computers in engineering applications during the 1970s. Its foundations were set in the seminal works of Beard, Jones and Clark [8, 9, 10], among others (see the survey papers [11, 4, 12, 13]). The model–based approach is built on a mathematical model of the healthy behavior of the process that must be monitored. The fundamental idea is that by using the model some estimations of the measured variables can be computed, so that by comparing the estimations to the actual measurements a deviation due to a fault can be detected. The output of the comparison procedure is a number of signals called *residuals*, which ideally should be zero when no fault is present. 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 (Fig. 1.2).

One of the first applications of model–based FDI was in the chemical industry, were *parity relations* were used to compare theoretical mass and flow balance with actual measurements, in order to detect leakages in pipes [13, 2]. Parity relations are basically rearranged input–output models, by which residuals can be generated by comparison with the outputs from the actual process. Ideally, during healthy operating conditions the residual should be exactly zero, but model uncertainties and physical disturbances usually make residuals to be different from zero even when no fault is present, so that greater than zero thresholds must be used. To overcome this problem, much effort has been devoted to develop robust residuals, for instance by devising disturbance decoupling methods that can be applied to linear systems [15, 3], with [16] being a notable exception.

Another approach to model–based FDI is through the use of *diagnostic observers*, whose original idea can be traced back to the already cited works of Beard, Jones and Clark [8, 9, 10], and was later established by Frank and coworkers [17, 12]. Unlike parity relations approaches, in the observer–based approach a state–space model of the system to be monitored is used, so that state and output estimations can be computed. The estimation errors are

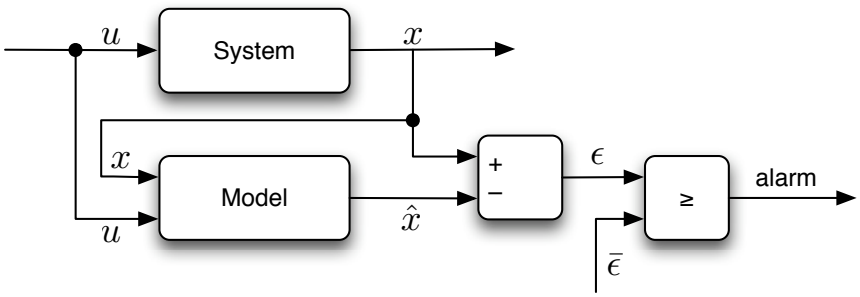


Figure 1.3: Basic scheme for model–based fault detection. Given a system with input u and measurable state x, a state estimate $\hat{x}$ is computed. The estimation error $\epsilon \triangleq x - \hat{x}$ is used as a detection residual and is compared against the threshold $\bar{\epsilon}$.

then used as residuals and compared to suitable thresholds for detection and isolation purposes (Fig. 1.3). The use of thresholds is needed, as in parity approaches, to avoid false–positive alarms due to the presence of modeling uncertainties and disturbances. The same robust disturbance decoupling techniques can be used with linear systems, and in fact the equivalence of parity relations and observer approaches has been proved [18]. For nonlinear systems, two approaches are possible when dealing with uncertainties and disturbances: the assumption that the latter are *structured*, and the use of *adaptive thresholds*. FDI problems for nonlinear systems with structured uncertainties were studied in [19, 20, 21, 22], for decoupling faults from unknown inputs. Unfortunately the assumption of structured uncertainties and disturbances is quite restrictive, and adaptive thresholds are the most promising solution for guaranteeing robustness without relying on very conservative fixed thresholds [23, 24, 25, 26], though the knowledge of a bound on the uncertainties and disturbances is needed. Another important approach that has been extensively used to represent modeling uncertainties in FDI schemes is the formulation of the problem in a stochastic framework [27].

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 prove to be a tantalizing task. A line of research tried to overcome this problem by using *qualitative models*, where only qualitative informations, such as sign and trend of measured variables, are used [28]. A more successful approach, anyway, is based on the use of adaptive on–line approximators, such as neural networks for instance, to learn on–line the unknown or uncertain parts of the system dynamical model, or the fault model if the fault accommodation problem is considered too [29, 30, 31, 32, 33, 34, 25, 35, 36, 37]. This learning approach enables the implementation of robust FDI schemes for nonlinear uncertain systems.

Finally, another way of solving the model–based FDI problem is by means of the so–called *parameter estimation approach*, where a parametric model of the system is used (see the surveys in [12, 2, 13]). During the monitoring, an on–line learning technique is used to adapt the model parameters to the observed measurements. If the parameters exit from their allowed nominal region that corresponds to the healthy system behavior, a fault is detected.

In this work the adaptive model–based approach will be used to develop a distributed Fault Detection, Isolation and Identification architecture for nonlinear and uncertain large–scale systems. *Fault Identification* is an extra step that is carried on after isolation, in order to quantify the extent to which a fault is present. For instance in the case of leakages in pipes or tanks, isolating the component where the leakage is present is not enough in order to eventually accommodate it, but the size of the leakage must be estimated too. Before describing the motivations that lead to addressing a distributed FDI architecture for large–scale systems, a definition of the FDI problem that we will solve, and an outline of the Generalized Observer Scheme (GOS) that we will use, will be given.

1.1.1 The FDI problem and the GOS solution

In a model–based FDI approach, basically two problems must be solved: the Fault Detection and the Fault Isolation problems [1, 12]. Now a definition of these two problem that will be followed in this work will be introduced.

Problem 1.1.1 (Fault Detection problem): Given

- A mathematical model of the system $\mathscr{S}$ to be monitored
- a sequence of measured system inputs and outputs

Test whether the following hypothesis is true or false

$$\mathscr{H}_0 : \text{"The system } \mathscr{S} \text{ is healthy"}.$$

□

Problem 1.1.2 (Fault Isolation problem): Given

- A mathematical model of the system $\mathscr{S}$ to be monitored
- a mathematical model of N possible faults that can occur to $\mathscr{S}$
- a sequence of measured system inputs and outputs

For all of the N following hypotheses, test whether they are true or false

$$\mathscr{H}_l : \text{"The system } \mathscr{S} \text{ is affected by the } l\text{–th fault"}, \; l \in \{1, \ldots, N\}.$$

□

In the model–based FDI literature, two schemes were devised in order to solve these problems: the *Dedicated Observer Scheme* (DOS) developed by Clark [10], and the *Generalized Observer Scheme* (GOS) developed by Frank (see [12] and the references by the same author therein). In both schemes for the isolation task 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.

In this work, a GOS scheme will be used, and the residuals and their thresholds will be designed so that false–positive alarms will be prevented. The scheme will make use of a detection observer called *Fault Detection and Approximation Estimator* (FDAE) that will provide an estimation error ϵ_0, and of N isolation observers called *Fault Isolation Estimators* (FIE) that will provide N estimation errors ϵ_l, $l \in \{1, \ldots, N\}$. Initially only the FDAE will be active in order to detect faults, by using the estimation error ϵ_0 as a residual. After a successful fault detection, the bank of N FIEs will be turned on and will use the estimation errors ϵ_l as residuals for solving the isolation problem (see Fig. 1.4). Because of the way residuals are designed in GOS schemes, a successful isolation decision will be reached if every but one hypothesis is falsified.

In the present work, it will be assumed that the full state x of the system is available as a measured output. It may seem quite a restrictive hypothesis, anyway, as noted in [36], many nonlinear control techniques need full state measurements, so demanding it for the FDI task is not a real limitation. In Chapter 4 this requirement will be slightly relaxed, while examples of FDI formulations for special classes of non-linear systems with only input-output measurements are presented in [35, 38].

1.2 Our motivation: large–scale and distributed systems

The GOS scheme presented in the last section was shown to possess interesting analytical properties [36], and theoretically can be applied to any system. Practical issues, anyway, must be solved when applying the scheme to systems of considerable size. In fact an acceptable solution to the FDI problem must be such that the fault decision can be provided in real–time, so that the larger possible amount of time is left to the fault accommodation phase before the fault event may lead to a failure. Actual FDI implementations are based on a *centralized* architecture, where a single computation node, that is a computer, is in charge of receiving all the necessary measurements and doing all the computations. In the case of systems large enough,

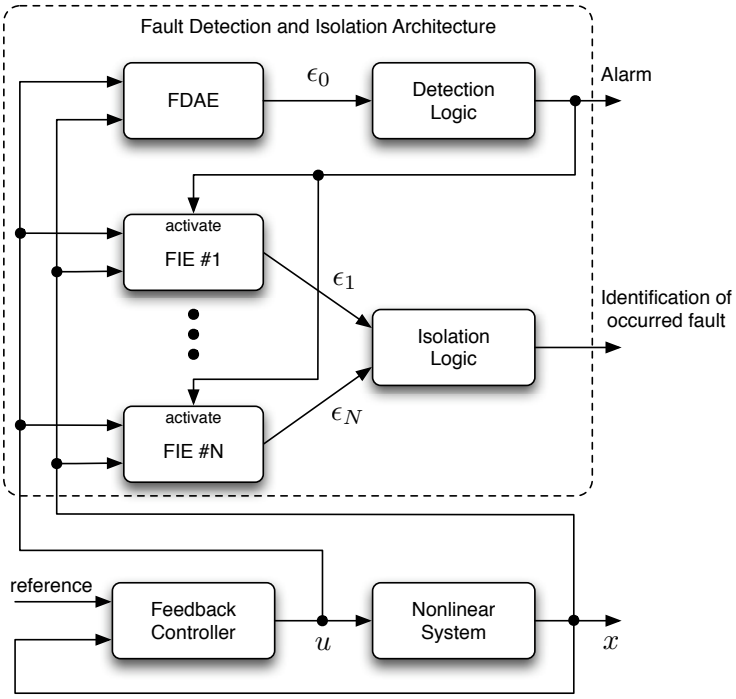


Figure 1.4: The GOS scheme on which the proposed FDI architecture will be based.

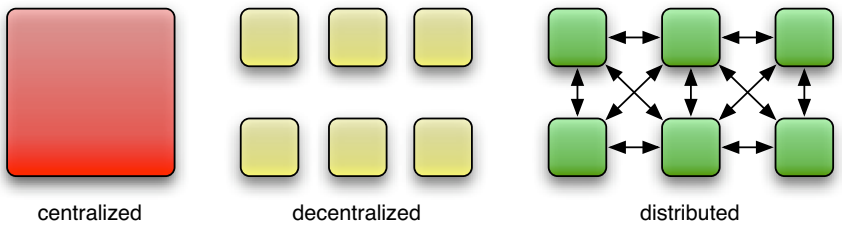


Figure 1.5: Pictorial representation of a centralized, a decentralized and a distributed system.

the task of computing in real–time all the estimations needed by the GOS scheme may be limited by the amount of computation power available at the computation node. Furthermore, if the measurements from the actual system are not taken from sensors directly wired to the computation node, but are carried through a communication network, the available bandwidth of the latter may be the bottleneck as well.

In order to understand these points, the concepts of *large–scale* and of *distributed* systems must be introduced. The first term applies to systems with a large number of state components, so that no feasible centralized architecture could be devised to solve estimation or control problems on it. The infeasibility, as suggested before, is due to the impossibility to find a computation and communication infrastructure fast enough to do all the computations and to receive all the measurements at a single site. Anyway the unfeasibility may be due to another feature of the system, that is the characteristic of being *distributed*. This term describes systems whose structure can be analyzed as being constituted by multiple subsystems that interact with neighboring subsystems. This is in contrast with the term *decentralized*, that applies to systems whose structure results to be made of multiple subsystems that do not interact with each other, and of course 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 looking at Fig. 1.5, where a pictorial representation is given.

It must be stressed that the terms centralized, decentralized and distributed can be used both when referring to physical systems and when referring to architectures. In this work, by the expression *system* or *physical system* we will denote the object that is being monitored against the presence of faults, while by *architecture* we will mean a combination of hardware and software used to implement and execute the fault diagnosis task.

The reason why a centralized architecture may prove to be infeasible for solving problems on a distributed system, may be made clear by considering the simple example in Fig. 1.6. The centralized architecture needs to convey

all the measurements from the various parts of the distributed system to a single location. This may be infeasible because of the considerations already brought up, should the distributed system be large–scale too. But another issue must be considered now: in many distributed systems of interest, the subsystems correspond to part of the system that are distributed in space, so that conveying all the measurements to a single geographical location would prove inconvenient. But most importantly, relying on a centralized architecture in many situations may be undesirable as it would lead to a safety threat. For instance, should the three physical blocks in Fig. 1.6 represent three airplanes, or more generically vehicles, moving in a formation [39, 40, 41], a centralized diagnosis architecture would result in an implementation on board of one of the three vehicles, or fixed at some ground station. Both these implementations would of course be highly unreliable and dangerous, as any single failure of the architecture itself will lead to the interruption of the diagnosis service for all the vehicles. The easiest way to overcome these drawbacks, is by the use of a decentralized architecture. In a decentralized architecture, as many local computing nodes as the number of subsystems are employed. Each node needs to receive the measurements from its corresponding subsystem, and will execute only the computations needed to solve the part of the problem pertaining to its subsystem. It is intuitive that this approach will reduce the computation power and the communication capacity needed by each node, with respect to a centralized architecture where only one node is present. But in a decentralized implementation an important weakness is hidden: in fact, as neighboring *nodes* are not supposed to communicate with each other, they will inevitably be unable to take into account in their solution the interactions between neighboring *subsystems*. In some engineering problems the coupling between subsystems is so weak that it can be ignored or considered as just a disturbance, but of course this does apply to only a subset of the cases of interest. The only consistently feasible architecture, then, is a distributed one, where, as in the decentralized case, as many computing nodes as the number of physical subsystems are present. But now, the nodes are allowed to interact with each other in a pattern that exactly mimics the pattern of physical interactions between subsystems. In this way, the nodes can exchange useful informations and measurements for implementing in their models the effect of the physical interconnections. Of course this exchange will lead to the need for a higher communication capacity than in the decentralized, or the centralized case, but this drawback is balanced by the decrease of the needed computation power and the increase in the architecture reliability and applicability.

Practical engineering examples of large–scale and/or distributed systems are abundant, and consists for example in large–scale communication or distribution networks, or multi–vehicle or multi–robots formations (see Fig. 1.7). These two classes of examples are useful to understand the nature

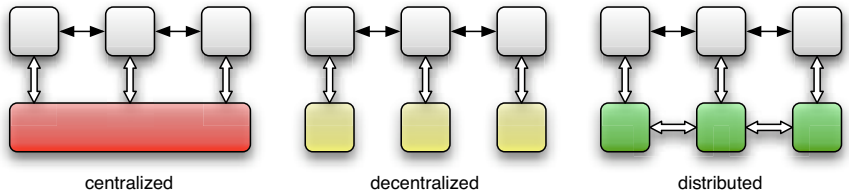
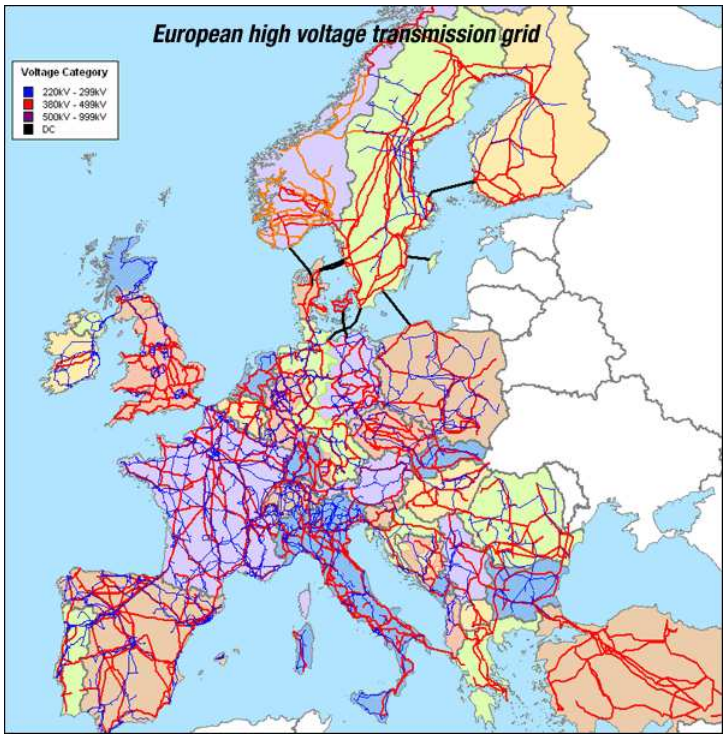


Figure 1.6: Pictorial representation of a centralized (red), a decentralized (yellow) and a distributed (green) architecture applied to a distributed system (white). Physical interaction between subsystems is represented by black arrows, while white thick arrows represent communication and measuring channels.

of the possible interactions between physical subsystems. In the first class the interactions are indeed physical or functional, as ultimately the system is *monolithic* and the boundaries between subsystems arise only because of the way we look at it. In the second class, the interaction is due to the various subsystems pursuing a common goal, that may consists in keeping a well–defined formation or performing a rendez–vous. This interaction of course is not physical, but nevertheless is a constitutive part of the way the resulting system works, and must therefore be diagnosed itself.

Of course the study of control, cooperation and 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, [42, 43, 44, 45, 41, 46, 47, 48, 49, 50] and the references cited therein). As far as the first class of interactions, that is the physical or functional one, is concerned, notable examples consists in large distribution and communication networks, such as drinking water distribution networks [51] and data [52] networks, and in coupled nonlinear systems synchronization [53]. The second class of interactions, that we may refer to as the "common goal" kind, includes the important topics of formation keeping and rendez-vous of Unmanned Aerial Vehicles (UAV) [39, 54], satellites [55, 56, 57], vehicles [58], and robots [59, 46, 60, 61]. Other notable examples occur when novel developments in transportation systems are considered, as in airplane formation and air traffic management [41, 40], and in Automatic Highway Systems (AHS) [62]. A most promising field of research has been enabled by the ground-breaking works of Reynolds and Vicsek on collective behavior [63, 64, 65, 66], that has lead to a number of powerful analyses and syntheses of particular distributed systems called *swarms* (see Fig. 1.7–(b)). Swarms occur when a large number of relatively unsophisticated individuals are provided with simple rules for interacting with other peers, and gathered together. The global effect of their interaction can lead to collective behaviours that are far more elaborate than what any single individual would be capable by acting alone. The most known example



(a)

(b)

Figure 1.7: Two examples of actual engineering system from the two classes of "physical interacting" and "common goal" distributed systems: (a) the European high voltage electrical distribution system, and (b) a robotic swarm.

of swarms in nature is given by the ants [67], whose colonies are able to execute remarkable tasks that are far away from the reach of a single insect. Popular swarms are made up of either social [68, 69, 70, 71], biological [63, 67, 65, 66, 70], robotic [72, 48, 73] or software individuals as in peer–to–peer networks [74], and their main advantage is that each individual is simple enough so that the overall deployment cost of a swarm is far lower than the one of a comparable centralized architecture, not to mention its higher reliability and robustness to failures and attacks. A closely related field of research comprises sensor networks [75, 43, 76]), and consensus problems [77, 78, 79, 80, 81, 82, 83, 84, 85, 86]. In sensor networks, the existence of many and possibly inexpensive sensing nodes is postulated, such that many measurements of the same group of variables, for instance the temperature distribution in a given environment, are available. The collective function of the sensor network is enabled by the fact that each node can communicate to neighboring nodes its measurements, so that each node can compute some form of average of its data and of the data of other nodes. The details of how the "average" is computed depends upon the peculiar consensus or gossip [87, 82] protocol that is employed. The big advantage earned by the use of sensor networks is that, under some hypotheses about the model of the sensing noise of each node, it can be proved that collectively the network can estimate the actual value of the measured variables in a way far more accurate than what a single node may do alone. Here a redundancy higher than what would be necessary in ideal conditions is tolerated, in order to counter the effects of measuring noise and uncertainties.

The *fil rouge* connecting all these examples is of course that they are distributed and/or large scale systems, with usually very complex global dynamics. The preferred method for dealing with them, as was suggested in the previous discussions, is through a *divide et impera* paradigm, where an excessively difficult problem is *decomposed* into smaller subproblems simpler enough to be solved with the existing computation and communication infrastructures. This approach is not new, as one could imagine. As far back as in the 1970s, researchers sought to develop so called "decentralized control" methods, described in the seminal paper [88], the well known book by Šiljak [89] and in the survey work of Sandell [90]. Since then there have been many enhancements in the design and analysis of decentralized and, later, distributed control and estimation schemes. On the other hand, one area where there has been much less research activity is in the design of fault diagnosis schemes specifically for distributed and large–scale systems. The fault diagnosis problem is a crucial problem in the safe operation of distributed systems, but the number of works that addressed it is still small. It is true that a considerable effort was aimed at developing distributed fault diagnosis algorithms suited to discrete event systems (see, for instance, [91, 92, 93, 94, 95, 96]), especially in the Computer Science literature where the problem of fault diagnosis for multi–processor systems is of

great importance [97, 98, 99, 100, 94, 101]. A notable contribution in the field of decentralized hybrid systems fault diagnosis is [102], although the fault detection scheme implemented does take into account only the jump part of the model, and not the continuous flow. An interesting scheme for the nonlinear fault detection of spacecraft formations, though it is neither distributed nor decentralized, was presented in [103]. But, as far as distributed discrete–time or continuous–time systems are concerned, only qualitative fault diagnosis schemes were attempted very recently [104, 105, 106, 107], or quantitative methods that were formulated for linear systems only [108, 109].

1.3 Objectives and outlines of the present work

Taking as a starting point the great interest and need for distributed fault diagnosis architectures, and the lack of suitable ones for systems described by continuous and discrete–time systems, this thesis will propose a distributed FDI for large–scale systems described by such models. The formulation will be taken from the works [110, 111, 112] by the same author, but will be greatly extended, and will use an adaptive approximation approach in order to address nonlinear uncertain systems. The FDI distributed problem will be solved by the application of a *divide et impera* paradigm, where the detection, isolation and identification tasks will be broken down and assigned to a network of *agents*, that we will call *Local Fault Diagnosers* (LFD). The LFDs will be allowed to communicate with each other, and also collaborate on the diagnosis of system components that may be shared between different diagnosers. Such diagnosers, each of which will have a different view on the system, will implement consensus techniques for reaching a common fault decision. The resulting architecture will be general enough to be applicable to nonlinear and uncertain systems of arbitrary size, without scalability issues.

Chapter 2 will summarize the results of [113], where a centralized and adaptive approximation approach for the FDI of nonlinear uncertain discrete–time systems is presented, that will serve as a basis for the development of a distributed FDI architecture. An illustrative example comprising both experimental and simulation data will be given. The central points in the move from a centralized to a distributed architecture will be addressed in Chapter 3, that will deal with the structural analysis of large–scale systems and the system and model decomposition problem. This will lead to the development of a distributed FDI scheme for discrete–time systems in Chapter 4. Finally, in Chapter 5 a simplified distributed FDI scheme for continuous–time systems is described, and concluding remarks are then drawn.

Chapter 2

Centralised Model-based Fault Diagnosis

The Distributed Fault Diagnosis and Identification (DFDI) architecture that will be developed in the present work, although will present an innovative design, will anyway be based on a well established scheme. This will be the Generalized Observer Scheme that was briefly presented in the introduction. This choice is motivated by the existence of an important amount of sound theoretical results in the literature about the application of this scheme to centralized nonlinear systems, such as [17, 12, 36, 37, 35], and by the inherent benefits offered by the scheme.

Now an outline of an Analytical Redundancy Relation based GOS scheme, along with some other useful definitions, will be given for the case of the FDI problem of a centralized system. This scheme and definitions will later be shown to be the basis of the DFDI architecture that will be built. This chapter will assume a discrete time uncertain and nonlinear model for the system under monitoring, and will concisely summarize results already present in the literature [113], in a way consistent with the other parts of this work.

2.1 Background and assumptions

It will be assumed that the nonlinear uncertain discrete-time dynamic system $\mathscr{S}$ under monitoring is described by the following equation

$$x(t+1) = f(x(t), u(t)) + \eta(x(t), u(t), t) + \beta(t - T_0)\phi(x(t), u(t)) \qquad (2.1)$$

where t is the discrete time instant, and we will assume that the sampling time is $T_s \in \mathbb{R}_+$. The terms $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote, respectively, the state and input vectors, while $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ represents the nominal healthy dynamics and $\eta : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \mapsto \mathbb{R}^n$ the uncertainty in the model

which may be caused by several factors such as, for instance, unmodeled dynamics, external disturbances, the possible discretization error and so on.

The term $\beta(t-T_0)\phi(x(t),u(t))$ denotes the changes in the system dynamics due to the occurrence of a fault. More specifically, the vector $\phi(x(t),u(t))$ represents the functional structure of the deviation in the state equation due to the fault and the function $\beta(t-T_0)$ characterizes the time profile of the fault, where T_0 is the unknown fault occurrence time. In this work, we shall consider either *abrupt* faults characterized by a "step-like" time-profile

$$\beta(t-T_0)=\begin{cases}0 & \text{if} \quad t<T_0\\ 1 & \text{if} \quad t\geq T_0\end{cases}, \tag{2.2}$$

or *incipient* faults characterized by an "exponential-like" time-profile

$$\beta(t-T_0)=\begin{cases}0 & \text{if} \quad t<T_0\\ 1-b^{-(t-T_0)} & \text{if} \quad t\geq T_0\end{cases}. \tag{2.3}$$

where $b>1$ is the unknown fault-evolution rate.

It is important to notice that the actual healthy part in model (2.1) is as general as possible. Neither a special structure is assumed for the nominal function f, neither the additive decomposition of the actual dynamics in the sum $f+\eta$ is a limitation. In fact, should any kind of uncertainty (additive, multiplicative, parametric, etc.) modify the nominal function f in an actual one f', then it would simply suffice to define the uncertainty term as $\eta(x(t),u(t),t)\triangleq f'(x(t),u(t))-f(x(t),u(t))$. This last remark on the generality of the additive decomposition holds, of course, for the effect of the fault function too.

For isolation purposes, we assume that there are $N_{\mathcal{F}}$ types of possible nonlinear fault functions; specifically, $\phi(x,u)$ belongs to a finite set of functions given by the following *fault class*

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

Each fault function in $\mathcal{F}$ is assumed to be in the form

$$\phi_l(x(t),u(t))=[(\vartheta_{l,1})^\top H_{l,1}(x(t),u(t)),\ldots,(\vartheta_{l,n})^\top H_{l,n}(x(t),u(t))]^\top,$$

where, for $i\in\{1,\ldots,n\}$, $l\in\{1,\ldots,N_{\mathcal{F}}\}$, the the "structure" of the fault is provided by *known* functions $H_{l,i}(x(t),u(t)):\mathbb{R}^n\times\mathbb{R}^m\mapsto\mathbb{R}^{q_{l,i}}$, and the *unknown* parameter vectors $\vartheta_{l,i}\in\Theta_{l,i}\subset\mathbb{R}^{q_{l,i}}$ provide its "magnitude"[1]. For the sake of simplicity and without much loss of generality, the parameter domains $\Theta_{l,i}$ are assumed to be origin–centered hyper–spheres. The following useful assumptions are needed.

[1]Which is referred to as the failure *mode* in the literature [12].

Assumption 2.1.1: At time $t = 0$ no faults act on the system. Moreover, the state variables $x(t)$ and control variables $u(t)$ remain bounded before and after the occurrence of a fault, i.e., there exist some stability regions $\mathcal{R} \triangleq \mathcal{R}^x \times \mathcal{R}^u \subset \mathbb{R}^n \times \mathbb{R}^m$, such that $(x(t), u(t)) \in \mathcal{R}^x \times \mathcal{R}^u$, $\forall\, t$. □

Assumption 2.1.2: The modeling uncertainty represented by the vector η in (2.1) is unstructured and possibly an unknown nonlinear function of x, u, and t, but it is bounded by some known functional $\bar{\eta}$, i.e.,

$$|\eta^{(i)}(x(t), u(t), t)| \leq \bar{\eta}^{(i)}(x(t), u(t), t), \quad \forall (x, u) \in \mathcal{R}, \forall t\,,$$

where, for each $i = 1, \ldots, n$, the bounding function $\bar{\eta}^{(i)}(x, u, t) > 0$ is known and bounded for all $(x, u) \in \mathcal{R}$. □

Assumption 2.1.3: The time profile parameter b is unknown but it is lower bounded by a known constant $\bar{b}$. □

As this work considers only the fault diagnosis problem and not the fault accommodation one, Ass. 2.1.1 is required for well–posedness. Ass. 2.1.2 and 2.1.3 make the problem analytically tractable, but are not a limitation in practical situations where some prior knowledge on the system operation is available.

2.2 Fault Detection and Isolation Architecture

In this section the proposed discrete–time *Fault Detection and Isolation* (FDI), to some extent analogous to the continuous one described in [36], will be described. The first service that the architecture must provide is the ability to detect faults. To this end, a nonlinear adaptive estimator named *Fault Detection and Approximation Estimator* (FDAE) will be started at time $t = 0$. This estimator is based on a model of the healthy system, and is able to provide an estimate $\hat{x}_0(t)$ of the system state. By using this estimate, the FDAE computes a residual and a threshold vector that guarantee the absence of false–positive alarms, that is no alarm will be fired before the actual fault occurrence time. Furthermore, as its name suggests the FDAE has another function. After the detection of a fault, in fact, an online adaptive approximator is turned on in order to learn the possibly unknown fault function ϕ.

The second service provided by the architecture is the ability to isolate and identify the occurred and detected fault. To solve this problem, the existence of a *fault class* $\mathcal{F}$ was assumed, that incorporates the existing knowledge about the peculiar ways of failing of the system being monitored. The ways of failing are described by a dynamical model of that faulty behavior, and this is accomplished through $N_{\mathcal{F}}$ parameterized fault functions contained in the class. The goal of the isolation service is to find which fault function does better represent the actual behavior of the system after

a fault has been detected, and to estimate the parameters vector of the fault function. The role of this vector is to assess the magnitude, or the gravity of the occurred fault. In order to implement the isolation service, a bank of $N_{\mathcal{F}}$ nonlinear adaptive estimators is employed, that are termed *Fault Isolation Estimators* (FIE). Each one is activated after a fault has been detected, and is tuned to a specific element of the fault class $\mathcal{F}$. It yields a state estimate $\hat{x_j} \in \mathbb{R}^n, j \in \{1, \ldots, N_{\mathcal{F}}\}$, where $N_{\mathcal{F}}$ is the number of nonlinear faults of the fault class $\mathcal{F}$. Each FIE computes its own residual and threshold vectors, and they are built so that a *Generalized Observer Scheme* is implemented.

Now the FDAE will be described, first during healthy operating conditions and then in faulty ones. After having provided an analytic result about fault detectability in Theorem (2.4.1), the isolability issue will be described in section (2.5) and the Isolability Theorem (2.6.1) will be proved.

2.3 Healthy behavior and Fault Detection and Approximation Estimator

At the time instant $t = 0$ the FDI architecture is started and, by Assumption 2.1.1, the system $\mathscr{S}$ is healthy. Until a fault is detected, the FDAE estimator is the only one to be enabled and provides a *state estimate* $\hat{x}_0$ of the state x. The difference between the estimate $\hat{x}_0$ and the actual measured state x will yield the following *estimation error*

$$\epsilon_0 \triangleq x - \hat{x}_0 \,,$$

which will be used as a residual and compared, component by component, to a suitable *detection threshold* $\bar{\epsilon}_0 \in \mathbb{R}^n_+$. The following condition

$$|\epsilon_0^{(k)}(t)| \leq \bar{\epsilon}_0^{(k)}(t) \; \forall \, k = 1, \ldots, n \tag{2.4}$$

will be associated to the *fault hypothesis*

$$\mathscr{H}_0 : \text{ "The system } \mathscr{S} \text{ is healthy"} \,.$$

Should condition (2.4) be unmet at some time instant t, the hypothesis $\mathscr{H}_0$ will be falsified and what will be called a *fault signature* will be noticed, leading to fault detection. In qualitative fault diagnosis schemes, such as [104], the fault signature is defined as a symbolic vector, that qualitatively describes the behavior of residuals and their derivatives after the occurrence of a fault. Instead, in quantitative schemes, such as [4, 12, 1], the fault signature represents the pattern of residuals that exhibit abnormal behavior after the occurrence of a fault. We will adhere to this last meaning, and we will introduce the following definition

Definition 2.3.1: The *fault signature* shown by the system $\mathscr{S}$ at time $t > 0$ is the index set $\mathcal{S} \triangleq \{k : \exists\, t_1,\, t \geq t_1 > 0,\, |\epsilon_0^{(k)}(t_1)| > \bar{\epsilon}_0^{(k)}(t_1)\}$ of the state components for which the hypothesis (2.4) did not hold for at least one time instant. □

Fault Detection Logic The fault detection logic can then be simply stated in terms of the signature $\mathcal{S}$: a fault affecting the system $\mathscr{S}$ will be detected at the first time instant such that $\mathcal{S}$ becomes non-empty. This time instant will be called the *fault detection time* T_d.

Definition 2.3.2: The *fault detection time* T_d is defined as $T_d \triangleq \min\{t : \exists\, k,\, k \in \{1, \ldots, n\} \,:\, |\epsilon_0^{(k)}(t)| > \bar{\epsilon}_0^{(k)}(t)\}$. □

Now the way the state estimate $\hat{x}_0$ is produced by means of the FDAE will be discussed. The FDAE is a nonlinear adaptive estimator based on the system model (2.1), and before the detection of a fault, for $0 \leq t < T_d$, its dynamics are selected as

$$\hat{x}_0(t+1) = \lambda(\hat{x}_0(t) - x(t)) + f(x(t), u(t)), \tag{2.5}$$

where $0 \leq \lambda < 1$ is a design parameter that fix the estimator poles. The state estimation error dynamics is described by the following difference equation

$$\epsilon_0(t+1) = \lambda\epsilon_0(t) + \eta(x(t), u(t), t) + \beta(t - T_0)\phi(x(t), u(t))\,.$$

By choosing $\hat{x}_0(0) = x(0)$, before the occurrence of a fault, for $0 \leq t \leq T_0$, the solution to the above equation is simply[2]

$$\epsilon_0(t) = \sum_{h=0}^{t-1} \lambda^{t-1-h}\eta(h)\,.$$

Recalling Assumption 2.1.2, it is straightforward to define the following threshold on the FDAE estimation error:

$$\bar{\epsilon}_0^{(i)}(t) \triangleq \sum_{h=0}^{t-1} \lambda^{t-1-h}\bar{\eta}^{(i)}(h) \geq |\epsilon_0^{(i)}(t)|,\ \forall t \leq T_0,\, i = 1, \ldots, n\,. \tag{2.6}$$

that guarantees no false–positives alarms will be issued prior to the fault occurrence time T_0, according to the fault detection logic described in this section.

[2] In the following, when there is no risk of ambiguity and for the sake of simplicity, a compact notation like, for instance, $\eta(t) \equiv \eta(x(t), u(t), t)$, will be used.

2.4 Faulty behavior and Fault Detectability

Threshold (2.6) guarantees that no false–positive alarms will be issued before T_0 because of the model uncertainty η. This of course, comes at the cost of the impossibility of detecting faults whose amplitude is "comparable" with the bound $\bar{\eta}$. This is formalized by the following

Theorem 2.4.1 (Fault Detectability): If there exist two time indexes $t_2 > t_1 \geq T_0$ such that the fault ϕ fulfills the following inequality for at least one component $i \in \{1, \ldots, n\}$

$$|\sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}(1 - b^{-(h-T_0)})\phi^{(i)}(h)| > 2\bar{\epsilon}_0^{(i)}(t_2)$$

then it will be detected at t_2, that is $|\epsilon_0^{(i)}(t_2)| > \bar{\epsilon}_0^{(i)}(t_2)$. □

Proof: At the time instant $t_2 > t_1 \geq T_0$ the i–th component of the state estimation error is

$$\epsilon_0^{(i)}(t_2) = \lambda^{t_2-t_1}\epsilon_0^{(i)}(t_1) + \sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}\eta^{(i)}(h) + \sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}(1 - b^{-(h-T_0)})\phi^{(i)}(h))$$

By the triangle inequality and $|\epsilon_0^{(i)}(t_1)| \leq \bar{\epsilon}_0^{(i)}(t_1)$, it follows that

$$|\epsilon_0^{(i)}(t_2)| \geq -\lambda^{t_2-t_1}\bar{\epsilon}_0^{(i)}(t_1) - |\sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}\eta^{(i)}(h)| + |\sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}(1 - b^{-(h-T_0)})\phi^{(i)}(h))|\,.$$

By recalling Assumption 2.1.2 and (2.6), it follows that

$$|\epsilon_0^{(i)}(t_2)| \geq -\lambda^{t_2-t_1}\sum_{h=0}^{t_1-1} \lambda^{t_1-1-h}\bar{\eta}^{(i)}(h) - \sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}\bar{\eta}^{(i)}(h) + |\sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}(1 - b^{-(h-T_0)})\phi^{(i)}(h)|$$

and hence

$$|\epsilon_0^{(i)}(t_2)| \geq -\bar{\epsilon}_0^{(i)}(t_2) + |\sum_{h=t_1}^{t_2-1} \lambda^{t_2-1-h}(1 - b^{-(h-t_0)})\phi^{(i)}(h)|\,.$$

If $\phi^{(i)}$ is such that the inequality in the hypothesis holds, then $|\epsilon_0^{(i)}(t_2)| > \bar{\epsilon}_0^{(i)}(t_2)$ and a fault will be detected. ■

Remark 2.4.1: Theorem 2.4.1 does not provide a closed-form expression for characterizing analytically the class of detectable faults. Anyway, suitable offline numerical tests can be carried on to check the theorem condition for different fault functions and different system operating conditions, in order to build an approximate class of detectable faults. □

After the detection of a fault at time $t = T_d$, the FDAE approximator is turned on and the dynamics (2.5) become

$$\hat{x}_0(t+1) = \lambda(\hat{x}_0(t) - x(t)) + f(x(t), u(t)) + \hat{\phi}_0(x(t), u(t), \hat{\vartheta}_0(t)) \,, \quad (2.7)$$

where $\hat{\phi}_0$ is an adaptive approximator and $\hat{\vartheta}_0(t) \in \hat{\Theta}_0 \subset \mathbb{R}^{q_0}$ denotes its parameters vector. The term adaptive approximator [114] may represent any nonlinear multivariable approximation model with adjustable parameters, such as neural networks, fuzzy logic networks, polynomials, spline functions, wavelet networks, etc. Again, for the sake of simplicity, $\hat{\Theta}_0$ is assumed to be an origin–centered hyper–sphere, with radius $M_{\hat{\Theta}_0}$.

In order for $\hat{\phi}_0$ to learn the fault function ϕ, its parameters vector is updated according to the following learning law:

$$\hat{\vartheta}_0(t+1) = \mathcal{P}_{\hat{\Theta}_0}(\hat{\vartheta}_0(t) + \gamma_0(t) H_0^\top(t) r_0(t+1)) \,,$$

where $H_0(t) \triangleq \partial\hat{\phi}_0(x(t), u(t), \hat{\vartheta}_0(t))/\partial\hat{\vartheta}_0 \in \mathbb{R}^{n \times q_0}$ is the gradient matrix of the on–line approximator with respect to its adjustable parameters, $r_0(t+1)$ is the signal

$$r_0(t+1) = \epsilon_0(t+1) - \lambda\epsilon_0(t) \,,$$

and $P_{\hat{\Theta}_0}$ is a *projection operator* [115]

$$\mathcal{P}_{\hat{\Theta}_0}(\hat{\vartheta}_0) \triangleq \begin{cases} \hat{\vartheta}_0 & \text{if } |\hat{\vartheta}_0| \leq M_{\hat{\Theta}_0} \\ \frac{M_{\hat{\Theta}_0}}{|\hat{\vartheta}_0|}\hat{\vartheta}_0 & \text{if } |\hat{\vartheta}_0| > M_{\hat{\Theta}_0} \end{cases} ,$$

The projection operator is one of the possible modifications to an adaptive approximator learning law, apart from the ϵ, σ and *dead–zone* modifications [114]. These modifications are needed to counter the effects of measuring or modeling uncertainties that make the approximation error be non–zero even when the parameter estimation error is zero or close to zero, and cause the phenomenon called *parameter drift*. It works by projecting at each time the updated parameter vector inside its allowable domain (fig. 2.1), thus assuring also the stability of the approximation scheme. It does not, anyway, sacrifice the convergence speed or the parameter estimation accuracy as the other kinds of modification.

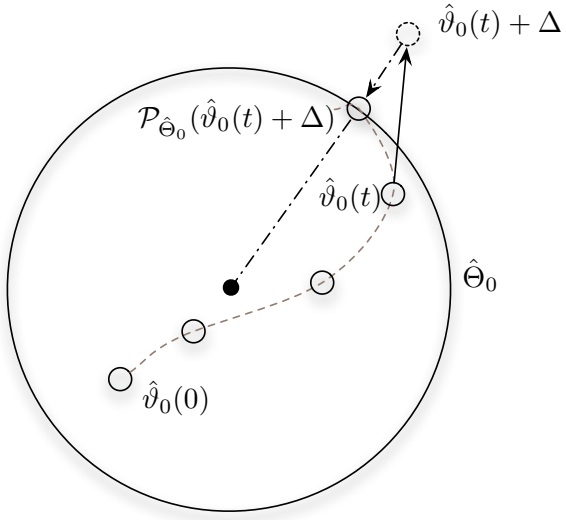


Figure 2.1: The projection operator $\mathcal{P}_{\hat{\Theta}_0}$: let us assume for the parameter $\hat{\vartheta}_0$ an admissible domain $\hat{\Theta}_0$, which center is the black dot. If at time t the learning law should predict a new value $\hat{\vartheta}_0 + \Delta$ that is outside $\hat{\Theta}_0$, the operator $\mathcal{P}_{\hat{\Theta}_0}$ will project it back on the domain boundary.

The learning rate $\gamma_0(t)$ is computed at each step as

$$\gamma_0(t) \triangleq \frac{\mu_0}{\varepsilon_0 + \|H_0(t)\|_F^2},\ \varepsilon_0 > 0,\ 0 < \mu_0 < 2$$

where $\|\cdot\|_F$ is the Frobenius norm and ε_0, μ_0 are design constants that guarantee the stability of the learning law [115, 116, 117, 118, 119].

2.5 Fault isolation logic

After a fault has been detected at time $t = T_d$, the $N_{\mathcal{F}}$ FIEs are activated in parallel to implement a kind of *Generalized Observer Scheme* [17, 36]. This scheme relies on each FIE being matched to a specific fault function belonging to the *fault class* $\mathcal{F}$ that represents the *a priori* knowledge about the possible way of failing of the system $\mathscr{S}$. A parallel with differential diagnostics procedures in medicine may be drawn in order to clarify the rationale behind this kind of scheme and the relative fault isolation logic. After fault detection, by considering only the signature[3] $\mathcal{S}$ the FDI scheme may conclude that the system is affected by either one of the known faults

[3]That would be called *syndrome* in medical language. Interestingly enough, the term syndrome is widely used in the Computer Science literature about fault diagnosis. For the specific problem of fault diagnosis of distributed computer systems, see the seminal paper of Preparata et al. [97].

in $\mathcal{F}$ or by an unknown fault. Which fault is the root cause cannot be discerned, as in general a signature would be such that more than a diagnosis can explain it. That is, more than one fault function in $\mathcal{F}$ may be such that it influences the variables referenced by the signature. Furthermore, even if the fault candidates would present in theory unique signatures, there is no guarantee that at detection time all the *analytic symptoms*[4] distinctive of a fault would be present.

It is at this point evident that no reliable diagnosis scheme could depend only on the analysis of the fault signature at detection time. To make a robust and correct fault decision, the FDI scheme then needs to conduct further tests that may lead to fault isolation, by mutually excluding the available fault candidates. This is related to physicians carrying on further tests in order to single out which of a number of diseases explain the actual medical syndrome. In the present problem, this is achieved by enabling the $N_{\mathcal{F}}$ FIEs to test in parallel the $N_{\mathcal{F}}$ fault hypotheses

$$\mathscr{H}_l : \text{"The system } \mathscr{S} \text{ is affected by the } l\text{–th fault"} ,$$

$l \in \{1, \ldots, N_{\mathcal{F}}\}$. To this end, analogously to the FDAE, the l–th FIE will provide its own *state estimate* $\hat{x}_l$ of the state x. The difference between the estimate $\hat{x}_l$ and the measured state x will yield the following *estimation error*

$$\epsilon_l \triangleq x - \hat{x}_l ,$$

which will play the role of a residual and will be compared, component by component, to a suitable *isolation threshold* $\bar{\epsilon}_l \in \mathbb{R}^n_+$. The following condition

$$|\epsilon_l^{(k)}(t)| \leq \bar{\epsilon}_l^{(k)}(t) \; \forall \, k = 1, \ldots, n \tag{2.8}$$

will be associated to the l–th fault hypothesis $\mathscr{H}_l$. Should this condition be unmet at some time instant t, the hypothesis will be falsified and the corresponding fault will be excluded as a possible cause of the fault signature, at the exclusion time $T_{e,l}$.

Definition 2.5.1: The *l–th fault exclusion time* $T_{e,l}$ is defined as $T_{e,l} \triangleq \min\{t : \exists \, k, \, k \in \{1, \ldots, n\}, \, |\epsilon_l^{(k)}(t)| > \bar{\epsilon}_l^{(k)}(t)\}$. □

Fault isolation logic The goal of the isolation logic is to exclude every but one of the faults belonging to the fault class $\mathcal{F}$, which will be said to be *isolated*.

Definition 2.5.2: A fault $\phi_p \in \mathcal{F}$ is *isolated* at time t iff $\forall l$, $l \in \{1, \ldots, N_{\mathcal{F}}\} \setminus p \, , T_{e,l} \leq t$ and $\nexists \, T_{e,p}$. Furthermore $T_{is,p} \triangleq \min\{T_{e,l}, \, l \in \{1, \ldots, N_{\mathcal{F}}\} \setminus p\}$ is the *fault isolation time*. □

[4] This term is defined in [2], and would correspond to the term *sign* in medical language.

Remark 2.5.1: It is worth noting that, if a fault has been isolated, we can conclude that it actually occurred if we assume a priori that only faults belonging to the class $\mathcal{F}$ may occur. Otherwise, it can only be said that it is not impossible that it occurred. If every fault in $\mathcal{F}$ is excluded, then it will be said that the proposed FDI architecture has isolated an unknown fault. In order to possibly add this fault to the class $\mathcal{F}$ of known fault, the FDAE on–line approximator is designed in order to be capable of learning any fault that can reasonably occur, and is started at T_d. □

2.6 FIE Estimators and Isolation Scheme

After a fault has been detected at time $t = T_d$, the bank of $N_{\mathcal{F}}$ FIEs is activated in order to isolate it. The dynamics of the state estimation of the l–th FIE, $l \in \{1, \ldots, N_{\mathcal{F}}\}$, is

$$\hat{x}_l(t+1) = \lambda(\hat{x}_l(t) - x(t)) + f(x(t), u(t)) + \hat{\phi}_l(x(t), u(t), \hat{\vartheta}_l(t))\,,\ l \in \{1, \ldots, N_{\mathcal{F}}\}\,, \quad (2.9)$$

where $\hat{\phi}_l(x(t), u(t), \hat{\vartheta}_l(t))$ is a linearly-parameterized function whose i–th component $\hat{\phi}_l^{(i)}(x(t), u(t), \hat{\vartheta}_l(t)) \triangleq (\hat{\vartheta}_{l,i})^\top H_{l,i}(x(t), u(t))$ matches the structure of $\phi_l^{(i)}$, with $\hat{\vartheta}_{l,i} \in \Theta_{l,i}$ and $\hat{\vartheta}_l \triangleq \mathrm{col}(\hat{\vartheta}_{l,i},\ i = 1, \ldots, n)$.

The learning law for $\hat{\vartheta}_{l,i}$ is analogous to the FDAE one:

$$\hat{\vartheta}_{l,i}(t+1) = \mathcal{P}_{\Theta_{l,i}}(\hat{\vartheta}_{l,i}(t) + \gamma_{l,i}(t) H_{l,i}(t) r_l^{(i)}(t+1))\,,$$

where $r_l^{(i)}(t+1)$ is given by

$$r_l^{(i)}(t+1) = \epsilon_l^{(i)}(t+1) - \lambda \epsilon_l^{(i)}(t)\,.$$

$P_{\Theta_{l,i}}$ is the projection operator on $\Theta_{l,i}$ and the learning rate $\gamma_{l,i}(t)$ is computed as

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

Remark 2.6.1: It is important to notice that, in spite of their similarity, the FDAE is built upon an on–line approximator that must be complex enough to be able to approximate any reasonable unknown fault, while the FIEs are designed to match a single fault function in $\mathcal{F}$. Anyway, although it is possible for a FIE to exactly match a fault function ϕ_l if $\hat{\vartheta}_{l,i}(t) = \vartheta_{l,i}$, $\forall i \in \{1, \ldots, n\}$, there is no guarantee that $\hat{\vartheta}_{l,i}(t)$ will converge to the true value $\vartheta_{l,i}$, as *persistence of excitation is not assumed in this work.* □

Assuming a matched fault, that is $\phi = \phi_l$, and with the initial condition $\hat{x}_l(T_d) = x(T_d)$, the solution to the i–th component of the estimation error dynamics equation is

$$\epsilon_l^{(i)}(t) = \sum_{h=Td}^{t-1} \lambda^{t-1-h}(\eta^{(i)}(h) + (1 - b^{-(h-T_0)})(\tilde{\vartheta}_{l,i})^\top H_{l,i}(h) \\ - b^{-(h-T_0)}(\hat{\vartheta}_{l,i})^\top H_{l,i}(h))\,,$$

where $\tilde{\vartheta}_{l,i}(t) \triangleq \vartheta_{l,i}(t) - \hat{\vartheta}_{l,i}(t)$ is the parameter estimation error. Owing to Ass. 2.1.2, the estimation error absolute value in the case of a matched fault can be upper bounded as

$$|\epsilon_l^{(i)}(t)| \leq \sum_{h=T_d}^{t-1} \lambda^{t-1-h}(\bar{\eta}^{(i)}(h) + (1 - b^{-(h-T_0)})\|\tilde{\vartheta}_{l,i}\|\|H_{l,i}(h)\| \\ + b^{-(h-T_0)}\|\hat{\vartheta}_{l,i}\|\|H_{l,i}(h)\|)\,.$$

The right hand side cannot be used as a threshold because b and $\tilde{\vartheta}_{l,i}$ are unknown. Anyway, the term $b^{-(t-T_0)}$ can be upper bounded by $\bar{b}^{-(t-T_d)}$ thanks to Ass. 2.1.3, while $\|\tilde{\vartheta}_{l,i}\|$ can be upper bounded by the function

$$\kappa_{l,i} \triangleq \|\hat{\vartheta}_{l,i}\| + M_{\Theta_{l,i}}\,.$$

Hence, we define the following threshold:

$$\bar{\epsilon}_l^{(i)}(t) \triangleq \sum_{h=T_d}^{t-1} \lambda^{t-1-h}(\bar{\eta}^{(i)}(h) + \kappa_{l,i}\|H_{l,i}(h)\| \\ + \bar{b}^{-(h-T_d)}\|\hat{\vartheta}_{l,i}\|\|H_{l,i}(h)\|) \geq |\epsilon_l^{(i)}(t)|\,, \quad (2.10)$$

that guarantees that if the fault $\phi_l \in \mathcal{F}$ occurs it will not be rejected by the corresponding FIE. Unfortunately, because of the model uncertainty η and of the parameter estimation error, there is no assurance that others FIEs will reject the fault ϕ_l so that it may be isolated. The following theorem gives a sufficient condition for a successful isolation decision.

Theorem 2.6.1 (Fault Isolability): Given a fault $\phi_p \in \mathcal{F}$, if for each $l \in \{1, \ldots, N_\mathcal{F}\} \setminus p$ there exists some time instant $t_l > T_d$ and some $i_l \in \{1, \ldots, n\}$ such that

$$\sum_{h=T_d}^{t_l-1} \lambda^{t_l-1-h}|\Delta_{p,l}\phi^{(i_l)}(h)| > \sum_{h=T_d}^{t_l-1} \lambda^{t_l-1-h}(2\bar{\eta}^{(i_l)}(h) + (\kappa_{l,i_l}(h) \\ + \bar{b}^{-(h-T_d)}\|\hat{\vartheta}_{l,i_l}\|)\|H_{l,i_l}(h)\|)\,,$$

where

$$\Delta_{p,l}\phi^{(i_l)}(t) \triangleq (1 - b^{-(t-T_0)})(\vartheta_{p,i_l})^\top H_{p,i_l}(t) - (\hat{\vartheta}_{l,i_l})^\top H_{l,i_l}(t), \\ \forall l,\, p \in \{1, \ldots, N_\mathcal{F}\},\, l \neq p$$

is the i_l–th component of the *fault mismatch function* between the p–th and the l–th faults, then the p–th fault will be isolated at time $\max_{l\in\{1,\dots,N_{\mathcal{F}}\}\setminus p}(t_l)$. □

Proof: Supposing that the p–th fault has occurred, the dynamics of the i_l–th component of the estimation error of the l–th FIE are described by

$$\epsilon_l^{(i_l)}(t+1) = \lambda\epsilon_l^{(i_l)}(t) + \eta^{(i_l)}(t) + \Delta\phi_{p,l}^{(i_l)}(t)\,,$$

so that for $t > T_d$ the solution to the above equation is

$$\epsilon_l^{(i_l)}(t) = \sum_{h=T_d}^{t-1} \lambda^{t-1-h}(\eta^{(i_l)}(h) + \Delta_{p,l}\phi^{(i_l)}(h)).$$

By using the triangular inequality we have

$$|\epsilon_l^{(i_l)}(t)| \geq \sum_{h=T_d}^{t-1} \lambda^{t-1-h}|\Delta_{p,l}\phi^{(i_l)}(h)| - \sum_{h=T_d}^{t-1} \lambda^{t-1-h}|\eta^{(i_l)}(h)|\,,$$

so that a sufficient condition for the l–th fault to be excluded is

$$\sum_{h=T_d}^{t-1} \lambda^{t-1-h}|\Delta_{p,l}\phi^{(i_l)}(h)| - \sum_{h=T_d}^{t-1} \lambda^{t-1-h}|\eta^{(i_l)}(h)| \geq \bar{\epsilon}_l^{(i_l)}(t)\,,$$

that is

$$\begin{aligned}\sum_{h=T_d}^{t-1} \lambda^{t-1-h}|\Delta_{p,l}\phi^{(i_l)}(h)| &\geq \sum_{h=T_d}^{t-1} \lambda^{t-1-h}|\eta^{(i_l)}(h)| \\ &+ \sum_{h=T_d}^{t-1} \lambda^{t-1-h}(\bar{\eta}^{(i_l)}(h) + \kappa_{l,i_l}\|H_{l,i_l}(h)\| + \bar{b}^{-(h-T_d)}\|\hat{\vartheta}_{l,i_l}\|\|H_{l,i_l}(h)\|)\,,\end{aligned}$$

which is implied by the inequality in the thesis. Requiring that this happens for each $l \neq p$ assures that each fault hypothesis but $\mathscr{H}_p$ is excluded, thus proving the theorem. ■

2.7 Illustrative example

A simple example is presented to illustrate the effectiveness of the proposed FDI scheme, based on the well-known three–tank problem (see Fig. 2.2). Both simulated and experimental data will be presented.

The experimental test bed consists of an AMIRA DTS200 three-tank system, connected to a DSpace acquisition and control card hosted by a

Siemens computer (see Fig. 2.3). All the tanks are cylinders with a cross-section $A^{(i)} = 0.156$ m^2, whilst every pipe has a cross-section $A_p^{(i)} = 5 \cdot 10^{-5}$ m^2 with outflow coefficient tuned to match the actual systems, $i \in \{1, 2, 3\}$. The tank levels are denoted by $x^{(i)}$, with $i \in \{1, 2, 3\}$, and are limited between 0 and 60 cm. The scalars $0 \leq u^{(i)} \leq 100$ ml/s, $i \in \{1, 2\}$, correspond to the inflows supplied by two pumps.

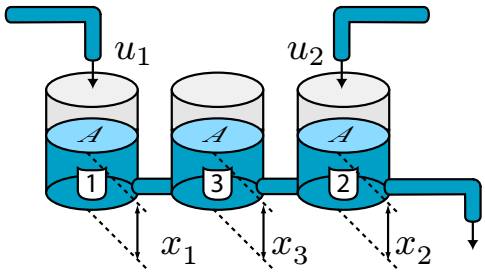


Figure 2.2: Structure of the three–tanks system under consideration.

Figure 2.3: The experimental three-tank system.

The tank discrete-time model will be obtained from the continuous-time version [36] by employing a simple forward Euler discretization with $T_s = 0.1$ s:

$$\begin{cases} x^{(1)}(t+1) = x^{(1)}(t) + \frac{T_s}{A^{(1)}}(c_p^{(1)} A_p^{(1)} \text{sign}(x^{(3)}(t) - x^{(1)}(t)) \times \\ \qquad \sqrt{2g|x^{(3)}(t) - x^{(1)}(t)|} + u^{(1)}(t)) \\ x^{(2)}(t+1) = x^{(2)}(t) + \frac{T_s}{A^{(2)}}(c_p^{(2)} A_p^{(2)} \text{sign}(x^{(3)}(t) - x^{(2)}(t)) \times \\ \qquad \sqrt{2g|x^{(3)}(t) - x^{(2)}(t)|} - c_p^{(3)} A_p^{(3)} \sqrt{2gx^{(2)}(t)} + u^{(2)}(t)) \\ x^{(3)}(t+1) = x^{(3)}(t) + \frac{T_s}{A^{(3)}}(c_p^{(1)} A_p^{(1)} \text{sign}(x^{(1)}(t) - x^{(3)}(t)) \times \\ \qquad \sqrt{2g|x^{(1)}(t) - x^{(3)}(t)|} - c_p^{(2)} A_p^{(2)} \text{sign}(x^{(3)}(t) - x^{(2)}(t)) \times \\ \qquad\qquad \sqrt{2g|x^{(3)}(t) - x^{(2)}(t)|}) \end{cases}$$

The FDAE on–line approximator $\hat{\phi}_0$ will consists of a 5–input, 3–output Radial Basis Function (RBF) neural network with one hidden layer of 3^5 fixed neurons equally spaced in the hyper–rectangle $[0,\, 10]^3 \times [0,\, 1]^2 \subset \mathbb{R}^5$. $\hat{\vartheta}_0$ will be a vector with $3{\cdot}3^5$ components containing the weights by which the hidden layer outputs are linearly combined in order to compute the network output.

Three FIEs will be employed, in order to match the following faults:

1. **Actuator fault in pump 1:** partial or full shutdown of the pump modeled as $u_f^{(1)} = u^{(1)}(1 - a^{(1)})$, where u_f represents the pumps flow in the faulty case and $0 \leq a^{(i)} \leq 1$, $i \in \{1, 2\}$.

2. **Leakage in tank 3:** circular hole of unknown radius $0 \leq \rho^{(3)} \leq 1$ in the tank bottom, so that the outflow due to the leak is $q_f^{(3)} = \pi(\rho^{(3)})^2 \sqrt{2gx^{(3)}(t)}$

3. **Actuator fault in pump 2:** same as 1 but related to pump number 2.

The resulting fault class $\mathcal{F}$ is

$$\mathcal{F} = \left\{ \begin{bmatrix} \vartheta_{1,1} H_{1,1}(t) \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \vartheta_{2,3} H_{2,3}(t) \end{bmatrix}, \begin{bmatrix} 0 \\ \vartheta_{3,2} H_{3,2}(t) \\ 0 \end{bmatrix} \right\},$$

where $\vartheta_{1,1} = a^{(1)}$, $H_{1,1}(t) = -\frac{T_s}{A^{(1)}} u^{(1)}(t)$, $\vartheta_{2,3} = \pi(\rho^{(3)})$, $H_{2,3}(t) = -\frac{T_s}{A^{(3)}} \cdot \sqrt{2gx^{(3)}(t)}$, $\vartheta_{3,2} = a^{(2)}$, $H_{3,2}(t) = -\frac{T_s}{A^{(2)}} u^{(2)}(t)$.

For both the FDAE and the FIEs all the auxiliary learning coefficients are equal to $\mu = 0.04$ and $\varepsilon = 0.001$, and the filter constant is $\lambda = 0.9$. The fault modeled is a leak with section $A_l = 2.612 \cdot 10^{-5}$ m^2 introduced in the third tank at time $T_0 = 45$ s. In the experimental case the leak is obtained by manually opening a drain valve, and the opening time is approximately equal to one second.

2.7.1 Simulated data

When building the actual model for simulating the three-tanks system, a random uncertainty no larger than 5%, 10% and 15% has been added, respectively, to the tanks cross section, the pipes cross section and the pipes outflow coefficient. After suitable offline simulations all the parameter domains were chosen to be hyper–spheres with unitary radius. The bound on the uncertainty function was set to the constant value $\bar{\eta}^{(i)} = T_s \cdot 0.002$, $i \in \{1, 2, 3\}$, while the bound on the time profile parameter was set to $\bar{b} = 1.01$.

Fig. 2.4 shows the results of a simulation where at $T_0 = 45$ s an incipient leak was introduced in tank 3, with a time profile described by $b = 1.05$. In Fig. 2.4(b) it can be seen that the fault is detected about 2 s later, and then is almost immediately isolated (Fig. 2.4(c)-(e)). The behavior of the estimation errors $\epsilon_2^{(1)}$ and $\epsilon_2^{(2)}$ is not reported, but anyway it is clear that they do not cross their corresponding thresholds as the fault function considered does not affect the dynamics of $x^{(1)}$ and $x^{(2)}$. In Fig. 2.4(f) the behavior of the third FIE parameter $\hat{\vartheta}_{2,3}$ is plotted: it can be seen that it approaches the value $\vartheta_{2,3} = 2.612 \cdot 10^{-5}$ m^2 corresponding to a complete pump shutdown. The offset is due to the fact that the FIE approximator is actually learning the fault function ϕ_1 plus the uncertainty η, rather than the fault alone.

2.7.2 Experimental data

In this case the nominal parameters have been used in building the model used by the FDI scheme. Anyway the uncertainty term η cannot be ignored, as in the experimental situation it accounts for the measurement errors introduced by the real level sensors. Because of the relatively high measurement errors, the uncertainty bounds have been set to the constant value $\bar{\eta}^{(i)} = T_s \cdot 0.004$, $i \in \{1, 2, 3\}$.

Fig. 2.5 shows the results of the application of the proposed FDI scheme to actual data recorded from the AMIRA test bench. As in the simulation case the fault is detected about 3 s later than T_0 and is isolated shortly thereafter. The parameter $\hat{\vartheta}_{2,3}$ shows an offset, too, that is due to the fact that the FIE approximator is trying to learn the effect of both the functions ϕ_2 and η. It is interesting to note, too, that the behavior of the FDAE error peak after the detection time is qualitatively the same in the simulated and experimental case, but only if its amplitude is considered relative to the threshold value. In fact, the peak value reached in the simulated case (about 2.4 mm) is different than the one reached in the experimental one (about 4.2 mm). This is due to the way the FDAE works: after detection, its on–line adaptive approximator is turned on and its effect is to decrease the estimation error. So, even if probably in both cases the estimation error should have grown to the same (larger) value, the presence of the FDAE approximator causes the error to drop shortly after the detection time T_d.

2.8 Concluding remarks

In this chapter a centralized FDI architecture for non–linear uncertain discrete–time systems, based on sound and proven techniques, was presented. The detection logic guarantees the absence of false–positive alarms, although this result comes at the cost of a reduced sensitivity to faults. In order to quantify this sensitivity, a fault detectability theorem was proved that can be used to characterize numerically the class of detectable faults. The same false–positive alarms guarantee is assured by the fault isolation logic, and similarly an isolability theorem was developed that characterizes isolable faults in terms of the magnitude of the difference, or *mismatch*, between them.

While the architecture presented so far is perfectly suited to solve the problem of FDI for "small" and centralized systems, this does not hold for large enough or distributed systems. The basis for a transition from a centralized to a distributed architecture will be given in the following chapter.

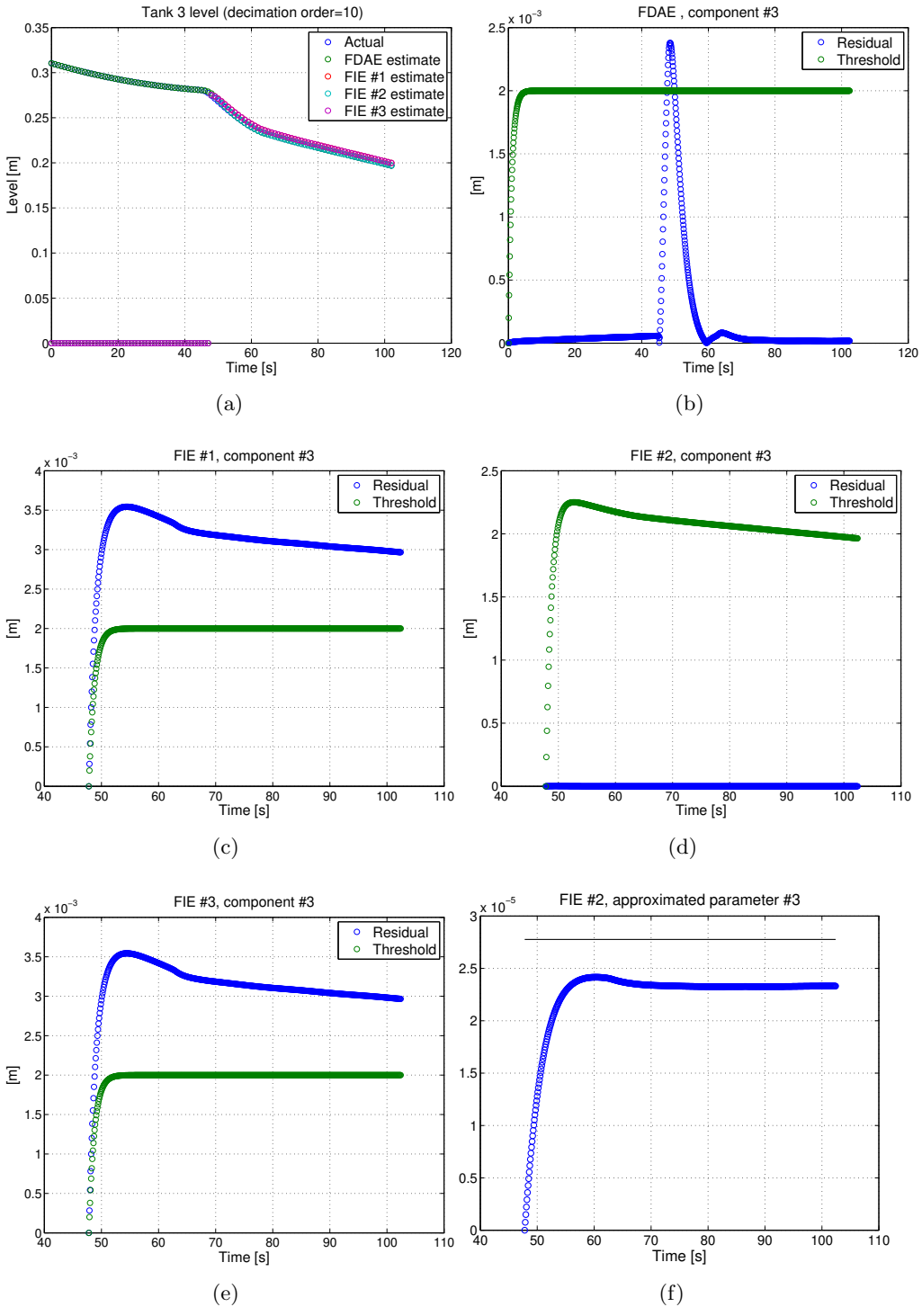


Figure 2.4: Time-behaviours of simulated signals related to tank no. 3 when an incipient leakage is introduced at time 45 s.

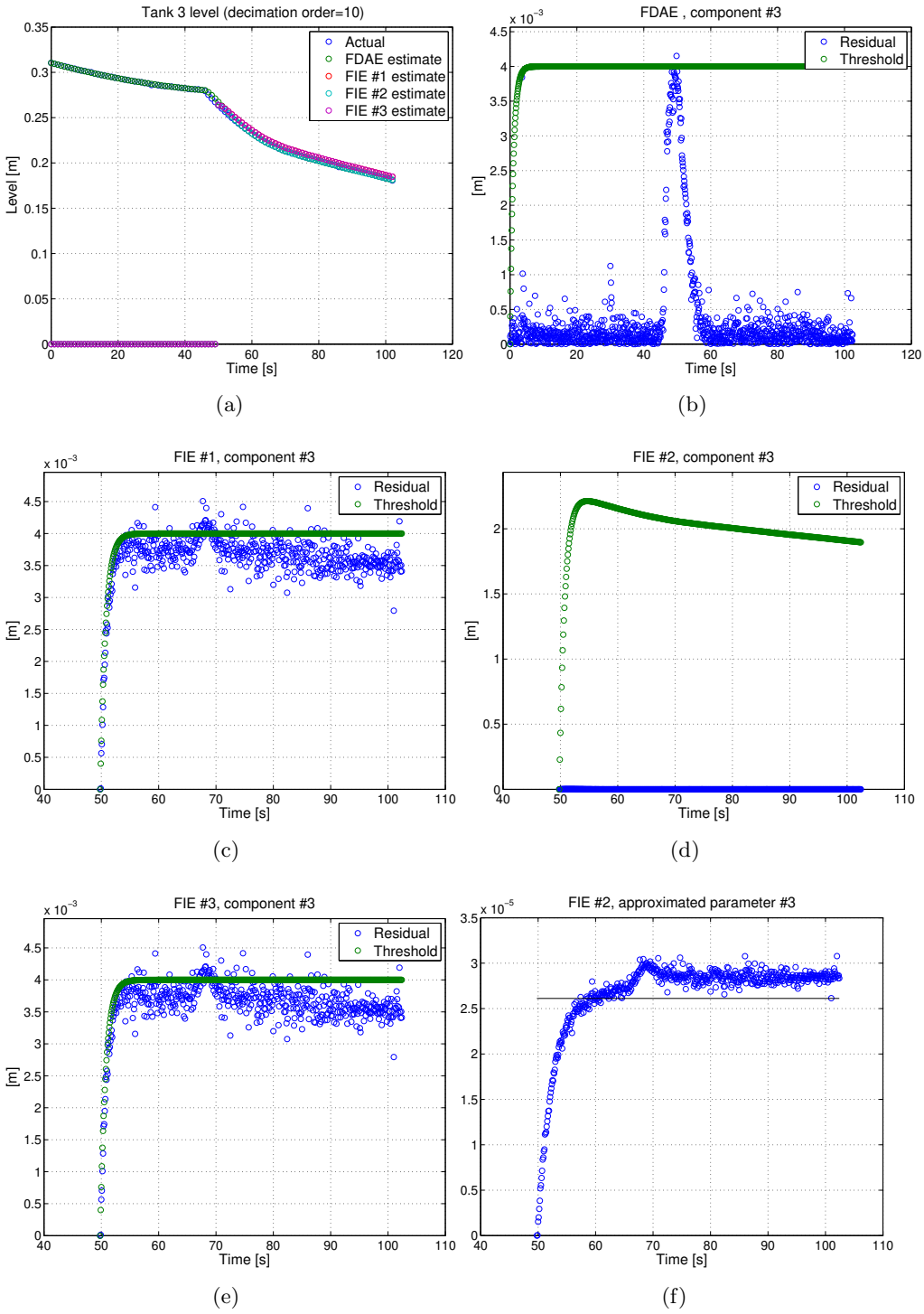


Figure 2.5: Time-behaviours of experimental signals related to tank no. 3 when an incipient leakage is introduced at time 45 s.

Chapter 3

From Centralized to Distributed Fault Diagnosis

The FDI architecture presented in Chapter 2 does represent an acceptable solution to the problem of fault diagnosis for discrete-time nonlinear systems. Uncertainties were accounted for in a robust way and the performance of the detection and isolation logics were analytically characterized. Similar results can easily be obtained for discrete-time nonlinear input-output systems [113], as well as for continuous one: in this last case see for instance [36, 35], where also theoretical results about fault detection time are given.

Anyway the solution proposed does stand on a fundamental, unstated assumption. This assumption is hidden in the FDAE and FIE estimator equations (2.5) and (2.9).

Assumption 3.0.1 (Unstated assumption about FDI schemes): Any useful solution to the FDI problem must be such that the fault decision d_I^{FD} can be provided in real–time. □

This should sound quite reasonable, as a late fault decision may come at a time when it is too late to steer a system away from a failure. Although the expression real–time does allow for some flexibility on the exact time when the fault decision about the health of the system at time t can be given, equations (2.5) and (2.9) do anyway imply that in the span of the sampling time T_s all the measurements and all the computations needed to evaluate the $N_{\mathcal{F}} + 1$ estimates of the next system state must be carried on. Furthermore, as it is usually understood in the literature, the equations assume that all the needed measurements are conveyed to a single location where the computations are done in a centralized way. While such a centralized implementation may work for reasonably small and slow systems, this may not hold for larger and faster systems because of the *limited available computation power*. With the term "large" we do not mean only systems with a large number of state variables, that would therefore need more computation power to provide the state estimation in time; we mean

also systems that are physically larger, so that the very task of conveying all the needed measurements in time to a single location where they can be processed may pose difficulties. Furthermore, the centralized approach would not be feasible also in all the situations where a centralized FDI architecture simply cannot be built, such as in sensor networks or in UAV formations. In these situations the choice of a central node for the task of diagnosing all the other nodes and itself would be arbitrary and not necessarily always optimal. Should that single node be prone to failures it would constitute a robustness issue too, in a situation where the robustness with respect to the failure of a single node is of fundamental importance.

It is then of paramount importance, for the successful application of model–based FDI schemes to real world large–scale systems, to reformulate the proposed architecture in a non-centralized way. This is the motivation that led to the present work. As with many problems in large–scale systems and in Computer Science, the solution will be based on the *divide et impera* paradigm. As a single "computation node" cannot be able to solve the FDI problem for a large-scale system, the solution will be to find an implementation involving multiple "computation nodes". These nodes will be called *agents*, and this term will denote here a hardware/software combination capable of:

- directly measuring physical variables;
- processing locally available information;
- communicating with other agents.

The task of subdividing the FDI problem in order to let more than one agent solve it will be called the *decomposition problem*. These agents will have only a limited view on the system, so that each agent will need to solve a computationally easier task than the one of monitoring the whole system. But having a limited view of the system does not bring only beneficial effects, but also detrimental ones. For properly solve its task, an agent will have to exchange informations between neighboring agents about parts of the system that it does not see, although these parts have a role in the dynamic behavior of the part assigned to it. So, actual solutions to the decomposition problem will always be a compromise, where less computation power needed by each agent is traded in for more communication capacity between them.

In this chapter we will first introduce a definition of what it is meant by centralized, decentralized, distributed and large–scale. Then a solution to the DFDI problem will be proposed, and a new FDI architecture will be finally introduced.

3.1 Large–scale, centralized, decentralized and distributed concepts

Before trying to develop a distributed architecture for FDI, we will first define some fundamental concepts that were informally introduced in Chapter 1. The most important thing to bear in mind is that a physical system is not inherently centralized, nor decentralized, nor distributed, nor large–scale. These adjectives make sense only when we have to solve a control or estimation problem about that system, so that we may find more convenient to *look* at the system in a way or in another. That is, our solution dictates in which kind of category the system would fall, and the concepts of being centralized, decentralized or distributed do apply to the control or estimation architecture in first place. Furthermore, even when a system may look obviously distributed itself because it consists of a number of interconnected subsystems, again this depends on the way we have chosen to draw the boundaries between subsystems.

As a consequence of these considerations, we will give the following definitions that apply both to physical systems and to control and estimation schemes, and that were inspired by [89, 39]

Definition 3.1.1: 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 variables belonging to it, without any interaction with other subsystems. □

Definition 3.1.2: 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 variables belonging to it, and by the influence of a proper subset of all the other subsystems. □

Definition 3.1.3: A system or architecture is *centralized* if it is neither decentralized nor distributed. □

Furthermore the following definition will be used to characterize *large–scale* systems:

Definition 3.1.4: A system is *large–scale* if no feasible centralized architecture can be devised to solve a problem on it. □

3.2 Structural analysis of large–scale systems

What will allow us to classify a system as large–scale and study how estimation and control problems on it can be solved, is its *structure*. The structure of the system is a way to represent how the different parts of the system interact with each other, although the structure does not provide the same level of details on the system as its dynamical model. The structure does

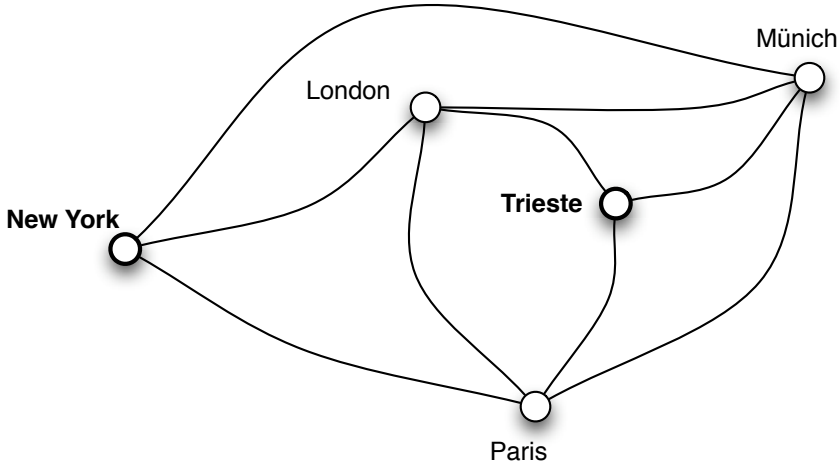


Figure 3.1: A simple graph showing the airline routes between Trieste, Paris, London, Münich and New York. One passenger wishing to fly from Trieste to New York can take a two legs flight through any of the airports of London, Paris and Münich. A passenger could also take three or even four legs flights between Trieste and New York without passing by the same airport more than once. Here the relation showed by the graph is "being connected by a direct flight". It is important to note that there is no need for any relation between the graph layout and the actual positions and distances of the cities, as the only information conveyed by the graph is the existence of relations between nodes. In fact, we chose to draw the graph this very way not to represent the actual geography, but to avoid edges crossing each other. A graph that can be drawn in such a way is called *planar*.

not tell us how a given state component influences another one, but tells us only that it does influence that variable.

Structural analysis does invariably involve the use of *graphs* [120], that are an intuitive and powerful pictorial representation of the relations between objects. Objects are represented as *nodes*, while the existence of a relation between two objects is represented by an *edge* between the corresponding nodes (Fig. 3.1). A kind of graph that provides the deepest insight into the system structure is the *bipartite graph* [1] (Fig. 3.2), where nodes are classified into two categories: states and input components on one side, and model equations on the other. An edge exists between one state or input component and one equation if and only if that component appears into that equation. No edges are allowed between nodes of the same category.

Although bipartite graphs offer a great deal of information, a simpler structural graph representation is preferred in most cases, as the present ones. This kind of graph is the *directed graph* or *digraph* [89], and is constituted by as many nodes as the state and input components: an *oriented*

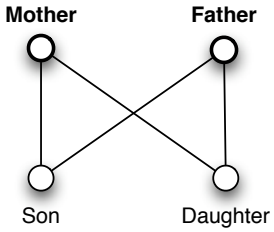


Figure 3.2: A simple bipartite graph where one category of nodes is constituted by parents, the other by sons, and the relationship of course is "being parent and son".

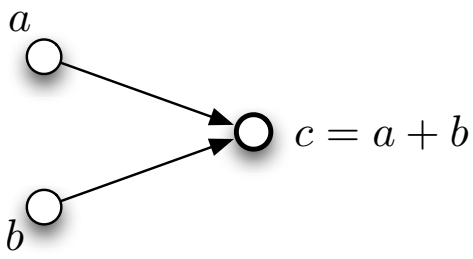


Figure 3.3: A simple directed graph that illustrates the sum operation between two inputs in order to get an output.

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 information about the causality (Fig. 3.3). In the following sections digraphs and their use in structural analysis are covered.

3.2.1 Some fundamental concepts on graphs

Before talking about digraphs, we will start with the more general case of a *graph* [120]

Definition 3.2.1: A graph $\mathcal{G} \triangleq (\mathcal{N}, \mathcal{E})$ is an ordered pair constituted by a non–empty set $\mathcal{N}$ of objects called *nodes* and by a possibly empty set $\mathcal{E}$ of two-elements subsets of elements of $\mathcal{N}$, called *edges*. $\mathcal{N}$ is called the *node set* of $\mathcal{G}$ and $\mathcal{E}$ is its *edge set*. □

According to Definition (3.2.1) edges do not have an orientation, and such a graph is called an *undirected graph*. Undirected graphs are useful when describing symmetric relations, such as the ones connecting two communicating nodes in a communication or a sensor networks.

Other important concepts are the *subgraph* and the *induced subgraph*, that will be used extensively when discussing the decomposition problem.

Definition 3.2.2: A graph $\mathcal{H} \triangleq (\mathcal{N}_\mathcal{H}, \mathcal{E}_\mathcal{H})$ is a *subgraph* of the graph $\mathcal{G} \triangleq (\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ iff $\mathcal{N}_\mathcal{H} \subseteq \mathcal{N}_\mathcal{G}$ and $\mathcal{E}_\mathcal{H} \subseteq \mathcal{E}_\mathcal{G}$. □

Definition 3.2.3: The subgraph $\mathcal{H} \triangleq (\mathcal{N}_\mathcal{H}, \mathcal{E}_\mathcal{H})$ *induced* on the graph $\mathcal{G} \triangleq (\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ by the node subset $\mathcal{K} \subseteq \mathcal{N}_\mathcal{G}$ is the maximal subgraph of $\mathcal{G}$ having node set $\mathcal{N}_\mathcal{H} = \mathcal{K}$, that is its edge set is $\mathcal{E}_\mathcal{H} = \{\{v_1, v_2\} : \{v_1, v_2\} \in \mathcal{E}_\mathcal{G},\, v_1 \in \mathcal{K},\, v_2 \in \mathcal{K}\}$. □

Apart from the edge set, there is another convenient way to describe how the nodes of a graph are connected by edges, by means of the *adjacency matrix*:

Definition 3.2.4: The *adjacency matrix* $A_\mathcal{G}$ of the graph $\mathcal{G} \triangleq (\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ is the $N \times N$ matrix

$$A_\mathcal{G}^{(i,j)} \triangleq \begin{cases} 1 & \text{if } \{v_i, v_j\} \in \mathcal{E}_\mathcal{G} \\ 0 & \text{else} \end{cases},$$

where $N \triangleq |\mathcal{N}_\mathcal{G}|$ is the number of nodes in the graph. □

Of course, the adjacency matrix of an undirected graph is symmetric. The adjacency matrix can be generalized so that it does not contains only zeros and ones. If we associate a weight, that is a non-negative real number, to each edge, we can relax the last definition so that $A_\mathcal{G}^{(i,j)}$ represents the weight of the edge connecting nodes v_1 and v_2:

$$A_\mathcal{G}^{(i,j)} \triangleq \begin{cases} \geq 0 & \text{if } \{v_i, v_j\} \in \mathcal{E}_\mathcal{G} \\ 0 & \text{else} \end{cases}.$$

A measure of how much a given node is connected to other nodes is called the node *degree*

Definition 3.2.5: The *degree* d_i of a node $v_i \in \mathcal{N}_\mathcal{G}$ of the graph $\mathcal{G}$ is the number of edges insisting on that node, that is $d_i \triangleq |\{\{v_i, v_j\} : v_j \in \mathcal{N}_\mathcal{G},\, \{v_i, v_j\} \in \mathcal{E}_\mathcal{G}\}|$. □

An interesting property of a graph is whether it is *connected* or not:

Definition 3.2.6: A graph $\mathcal{G} \triangleq (\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ is *connected* iff for each $v_1 \in \mathcal{N}_\mathcal{G}$ and for each $v_2 \in \mathcal{N}_\mathcal{G}$ there exists a sequence of edges[1] belonging to $\mathcal{E}_\mathcal{G}$ that starts in v_1 and ends in v_2. □

This means that in a connected graph it is possible to reach any node from any other node, by traversing edges belonging to the graph.

3.2.2 Directed graphs

A directed graph can be defined similarly to graphs, but requesting that the edges be endowed with an orientation. To distinguish between non–oriented and oriented edges, the latter will be called *arcs*.

[1] This sequence is called a *walk*. If it is such that no edges or nodes are repeated, it is called a *path* [120].

Definition 3.2.7: A *directed graph* $\mathcal{G} \triangleq (\mathcal{N}, \mathcal{E})$ is an ordered pair constituted by a non–empty set $\mathcal{N}$ of objects called *nodes* and by a possibly empty set $\mathcal{E}$ of ordered pairs of elements of $\mathcal{N}$, called *arcs*. $\mathcal{N}$ is called the *node set* of $\mathcal{G}$ and $\mathcal{E}$ is its *arc set*. □

As the arcs are ordered, we will use the notation (v_i, v_j) for the arc that leaves from node v_i and ends in node v_j.

The definitions of the subgraph, induced subgraph and adjacency matrix easily extend to directed graphs. A further distinction can be made regarding the property of being connected. A digraph is said to be *strongly connected* iff any two nodes can be connected by a sequence of arcs traversed in the direction of their orientation. A less stringent property is the one of being *weakly connected*, that holds when the underlying graph, that is obtained from the digraph by ignoring the arcs orientation, is connected.

The concept of a node degree is further specified by considering the number of arcs leaving from a node, called *out degree*, and the number of those ending in that node, called *in degree*.

Definition 3.2.8: The *out degree* od_i of a node $v_i \in \mathcal{N}_\mathcal{G}$ of the digraph $\mathcal{G}$ is the number of arcs leaving from that node, that is $od_i \triangleq |\{(v_i, v_j) : v_j \in \mathcal{N}_\mathcal{G}, (v_i, v_j) \in \mathcal{E}_\mathcal{G}\}|$. □

Definition 3.2.9: The *in degree* id_i of a node $v_i \in \mathcal{N}_\mathcal{G}$ of the digraph $\mathcal{G}$ is the number of arcs ending in that node, that is $od_i \triangleq |\{(v_j, v_i) : v_j \in \mathcal{N}_\mathcal{G}, (v_j, v_i) \in \mathcal{E}_\mathcal{G}\}|$. □

3.2.3 Structural graph of a system

Now we will show how the structure of a dynamical system can be represented through a directed graph. First of all we will give a rather broad and intuitive definition of what we mean by the *structure* of a dynamical system

Definition 3.2.10: The *structure* of a dynamical system $\mathscr{S}$ having a state vector $x \in \mathbb{R}^n$ and an input vector $u \in \mathbb{R}^m$ is the set of ordered pairs $\Sigma_\mathscr{S} \triangleq \{(x^{(i)}, x^{(j)}) : i, j \in \{1, \ldots, n\}, \text{"}x^{(i)} \text{ acts on } x^{(j)}\text{"}\} \cup \{(u^{(i)}, x^{(j)}) : i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}, \text{"}u^{(i)} \text{ acts on } x^{(j)}\text{"}\}$. □

The very general relation "acts on" has been used in Definition 3.2.10 in order not to bound it to a specific choice of system model. Anyway, as in this work we are considering dynamical systems described by differential or difference equations, the relation "acts on" will be equivalent to "appears in the state equation of".

As we have defined the structure as a set of ordered pairs of related state or input components, a natural choice for representing it is by the use of a directed graph, that will be called the *structural graph* [89].

Definition 3.2.11: The *structural graph* of a dynamical system $\mathscr{S}$, having a state vector $x \in \mathbb{R}^n$ and an input vector $u \in \mathbb{R}^m$, is the directed graph

$\mathcal{G} \triangleq \{\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G}\}$ having the node set $\mathcal{N}_\mathcal{G} \triangleq \{x^{(i)} : i \in \{1, \ldots, n\}\} \cup \{u^{(i)} : i \in \{1, \ldots, m\}\}$ and the system structure $\Sigma_\mathscr{S}$ as the arc set, that is $\mathcal{E}_\mathcal{G} = \Sigma_\mathscr{S}$. □

The structural graph has of course many advantages in the analysis of a large–scale system: in our case, among others, it will clearly show which measurements are needed to compute an estimate of a given variable. This will be useful, as explained later, in determining the amount of communication needed to compute the estimation of a subsystem state.

3.3 Divide et Impera: decomposition of large–scale systems

As stated previously, the task of solving the FDI problem for a large–scale systems in a centralized way, may be nor feasible, nor desirable. For this reason a natural solution, that will be followed in this work, is to *decompose* the original difficult problem into many subproblems that are easier to solve. As the model–based FDI problem involves at its fundamental level one or more estimators of the state vector x, the main step will consists in decomposing the estimation task. This will lead, for example, to having more than one fault detection estimator instead of a single one, each one estimating only a *subset* of the state vector. It is rather intuitive that by carefully choosing these subsets the number of states to be estimated and the number of measurements needed to compute them can be made smaller than any level should be deemed as acceptable, thus overcoming the limitations pointed out in the beginning of this chapter. How these subsets will be chosen constitutes the so–called *decomposition problem*. Distributed control and estimation problems in the literature are naturally based on a decomposition of the original large–scale system [90, 61]. When considered as a whole, the original system $\mathscr{S}$ will be called here the *monolithic system*.

Of course the decomposition should yield a set of FDI problems that are each one sufficiently simpler than the original centralized problem. This remark points to the need for a definition of the term "simpler" in the present contest. Although the solution to the decomposition problem itself is not the main subject of this work, anyway from what have been said so far each resulting subproblem should fulfill the following loosely defined constraints:

Computation constraint: the computation power needed to execute the estimation task of each subproblem should not exceed the computation power affordable by any single agent to which the subproblem may be assigned.

Communication constraint: the communication capacity needed to convey the measurements needed for the estimation task of each subproblem should not exceed the communication capacity affordable by any single agent to which the subproblem may be assigned.

While the reason for the second constraint may appear not completely clear now, it will at the end of Section 3.4, after the proposed distributed FDI architecture would have been laid down.

The outcome of the decomposition of a large–scale system $\mathscr{S}$ will be a description of its dynamic behavior in terms of the dynamics of a multiset[2] of N subsystems[3] $\mathscr{S}_I$, $I \in \{1, \ldots, N\}$. The first step in the decomposition will be to decompose the *structure* of $\mathscr{S}$, in order to select the components of the state and input vectors of $\mathscr{S}$ that will be assigned to each subsystem. The second step will be the decomposition of the dynamic model of $\mathscr{S}$, so that the dynamic equation of each subsystem can be derived. At the end of this process, the decomposition of the FDI problem between multiple agents, each one devoted to monitoring a a single subsystem, will be described.

3.3.1 System and structural graph decomposition

The idea of graph decomposition has been used in many fields [122] in order to apply the divide et impera paradigm. For instance in numerical methods involving the solution to partial differential equations (PDEs) [123, 124, 125, 126], in image processing [127], in operation research [128], and of course in large–scale system decomposition [89, 129].

To decompose a monolithic system $\mathscr{S}$ having a state vector $x \in \mathbb{R}^n$, an input vector $u \in \mathbb{R}^m$ and a structural graph $\mathcal{G} = (\mathcal{N}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, it is necessary to find a multiset of $N \geq 1$ 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 $u_I \in \mathbb{R}^{m_I}$. These local vectors will be constructed by taking components of the monolithic system vectors x and u, thanks to ordered sets $\mathcal{I}_I \triangleq (\mathcal{I}_I^{(1)}, \ldots, \mathcal{I}_I^{(n_I)})$ of indices[4], called *extraction index set* [111, 130, 112]:

Definition 3.3.1: The *local state* $x_I \in \mathbb{R}^{n_I}$ of a dynamical subsystem $\mathscr{S}_I$, arising from the decomposition of a monolithic system $\mathscr{S}$, is the vector $x_I \triangleq \mathrm{col}(x^{(j)} : j \in \mathcal{I}_I)$, where $\mathcal{I}_I$ is the subsystem extraction index set. □

Definition 3.3.2: The *local input* $u_I \in \mathbb{R}^{m_I}$ of a dynamical subsystem $\mathscr{S}_I$, arising from the decomposition of a monolithic system $\mathscr{S}$, is the vector $u_I \triangleq \mathrm{col}(u^{(k)} : (u^{(k)}, x^{(j)}) \in \mathcal{E}_\mathcal{G},\, j \in \mathcal{I}_I,\, k = 1, \ldots, m)$, where $\mathcal{I}_I$ is the subsystem extraction index set. □

It must be understood that when performing the "col" operation in the two previous definitions, the elements of the index set $\mathcal{I}_I$ are taken in the order they appear. Definition 3.3.2 was chosen so that the local input contains all the input components that affects at least one component of the local state vector. At this point, the structural graph of the I–th subsystem

[2] A multiset is a set that allows for repeated elements [121].

[3] In the subsequent analysis, a capital-case index will denote the specific subsystem under concern.

[4] By the term *index* we mean an element of $\mathbb{N}_+$.

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.

In order to be useful for our goal, the set of subsystems $\mathscr{S}_I$, that is the decomposition, cannot be arbitrary, but should respect a set of rules, stated in the following

Definition 3.3.3: A *decomposition* 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. $\mathcal{I}_I^{(j)} \leq n$, 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. $\bigcup_{I=1}^{N} \mathcal{I}_I = \{1, ..., n\}$. □

Point 1 prevents the definition of trivial empty subsystems, point 2 is necessary for well-posedness, point 3 avoids that resulting subsystems have isolated state components, while the most interesting is point 4. The last point requires that the decomposition covers the whole original monolithic system, but anyway does not require that for any two subsystems $\mathcal{I}_I \cap \mathcal{I}_J = \emptyset$, $I, J \in \{1, \ldots, N\}$. This will allow for a state component of $\mathscr{S}$ to be assigned to one or more subsystems, thus being "shared". Such a decomposition is called *overlapping decomposition*. Overlapping decompositions [131] were found to be rather a useful tool when addressing large–scale systems. In particular, problems of stability, control and estimation [132], and fault diagnosis [133] for large–scale linear system were successfully solved by using overlapping decompositions. The advantages of overlapping decompositions in the present setting will be discussed in Section 3.4.

As a result of overlaps, some components of x will be assigned to more than a subsystem thus giving rise to the concepts of *shared state variable* and *overlap index set*.

Definition 3.3.4: A shared state variable $x^{(s)}$ is a component of x such that $s \in \mathcal{I}_I \cap \mathcal{I}_J$, for some I, $J \in \{1, ... N\}$, $I \neq J$ and a given decomposition $\mathcal{D}$ of dimension N. □

Definition 3.3.5: The *overlap index set* of subsystems sharing a variable $x^{(s)}$ is the set $\mathcal{O}_s \triangleq \{I : s \in \mathcal{I}_I\}$, whose dimension is $N_s \triangleq |\mathcal{O}_s|$. □

In the following, the notation $x_I^{(s_I)}$, with $x_I^{(s_I)} \equiv x^{(s)}$, will be used to denote the fact that the s–th state component of the original large–scale

system, after the decomposition became the s_I–th of the I–th subsystem, $I \in \mathcal{O}_s$.

Now some more definitions are needed to characterize a structural decomposition. This is related to the fact that, generally, no subsystem will constitute an "island" that is completely independent from other subsystems[5]. Instead, some local state component will be "acted on" by variables belonging to other subsystems. The external variables influencing the dynamics of local state components of subsystem $\mathscr{S}_I$ will make up the vector of *interconnection variables* z_I

Definition 3.3.6: The *interconnection variables* vector $z_I \in \mathbb{R}^{p_I}$, $(p_I \leq n - n_I)$ of the subsystem $\mathscr{S}_I$ is the vector $z_I \triangleq \text{col}(x^{(k)} : (x^{(k)}, x^{(j)}) \in \mathcal{E}_\mathcal{G},\ j \in \mathcal{I}_I,\ k = 1, \ldots, n)$. □

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

Definition 3.3.7: The *neighbors index set* of a subsystem $\mathscr{S}_I$ is the set $\mathcal{J}_I \triangleq \{K : \exists\, (x^{(k)}, x^{(j)}) \in \mathcal{E}_\mathcal{G},\ k \in \mathcal{I}_K,\ j \in \mathcal{I}_I\}$. □

To gain some more insight into the afore-described decomposition approach, consider the simple example depicted in Fig. 3.4, where a specific decomposition of a system $\mathscr{S}$ into two overlapping subsystems $\mathscr{S}_1$ and $\mathscr{S}_2$ is considered.

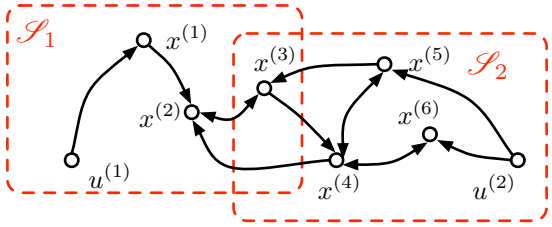


Figure 3.4: Example of decomposition of a large–scale system $\mathscr{S}$ into two overlapping subsystems $\mathscr{S}_1$ and $\mathscr{S}_2$ such that: $x_1 = [x^{(1)}\, x^{(2)}\, x^{(3)}]^\top$ and $x_2 = [x^{(3)}\, x^{(4)}\, x^{(5)}\, x^{(6)}]^\top$ are the local states, $u_1 = u^{(1)}$ and $u_2 = u^{(2)}$ the local inputs, $z_1 = [x^{(4)}\, x^{(5)}]^\top$ and $z_2 = x^{(2)}$ the interconnection variables, and $x^{(3)} \equiv x_1^{(3)} \equiv x_2^{(1)}$ is a shared variable with $\mathcal{O}_3 = \{1, 2\}$.

3.3.2 Model decomposition

The next step, after the structural decomposition, is to derive from the model of the monolithic system $\mathscr{S}$ the equations that describe the dynamic behavior of a given subsystem $\mathscr{S}_I$. These equations will be used by an agent monitoring $\mathscr{S}_I$ in order to provide the estimation tasks needed to solve the corresponding FDI problem.

[5]This is equivalent to say that generally the monolithic system $\mathscr{S}$ is not decentralized.

The main idea in model decomposition is to rewrite the model equations in order to separate the effect of the local variables from that of the interconnection variables, for instance by writing the dynamic equation right–hand side as the sum of two terms, one of which depends only on local variables and the other only on interconnection variables. The rationale for this is the assumption that for an agent monitoring the subsystem it does "cost" less to measure local variables than to get the non–local variables constituting the interconnection vector. This is the reason why in decentralized control or estimation schemes the problem of evaluating the effect of the "expensive" interconnection term is ignored altogether.

For linear systems, powerful model decomposition techniques and descriptions exist (see for instance the works published in recent years by D'Andrea et al. [134, 47]), that can be applied to systems showing either a regular or arbitrary structure. Of course these approaches take advantage of the linearity for separating the effects of local and interconnection variables. For non-linear systems, in general an additive decomposition into purely local and purely interconnection terms is not possible. Now this point will be clarified.

Starting from a simple and generic non–linear model for a discrete–time monolithic system $\mathscr{S}$

$$x(t+1) = f(x(t), u(t))$$

the simplest way to write the dynamics of the s_I–th component of the I–th subsystem is

$$x_I^{(s_I)}(t+1) = f^{(s)}(x(t), u(t))\,,$$

where it is understood that the s–th component of the monolithic state x corresponds to the s_I–th component of the local state x_I after the decomposition. Of course an agent that would use such a model for the estimation task will experience a reduced computational complexity with respect to a centralized implementation, as $n_I \leq n$, but will still need to directly get or to receive by other agents the measurements of all the components of the monolithic vectors x and u. This means that probably the computation constraint can be met, but not the communication constraint.

In order to limit the number of measurements needed by any agent, the structure of the system may come to an help. In fact, thanks to the structural analysis and the structural decomposition outlined before, it is clear that the dynamics of the local state x_I do not depend on the whole vectors x and u, but only on the local state x_I itself, the local input u_I and the interconnection vector z_I. For this reason, following [89], the dynamics of x_I can be split in the following way

$$x_I(t+1) = f_I(x_I(t), u_I(t)) + g_I(x_I(t), z_I(t), u_I(t))\,, \tag{3.1}$$

where $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ is the *local nominal function* and $g_I : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ is the *interconnection function*. Equation (3.1) is as

general as possible, although it looks similar to the additive decompositions of linear systems. It simply highlights, by the use of the function f, the linearly separable part of the local influence, while the function g_I accounts for all the non–linearly separable parts. Should the local dynamics be not linearly separable at all, it will simply means that f is always null.

Why, anyway, did we decompose the local dynamics into two parts, one of which depends solely on local variables? There is more than one reason. The first one, is that generally there are many components in a subsystem where does make sense to highlight the influence of the local variables alone, as this is the only influence. These are variables that in the structural graph are not the ending nodes of any arc coming from variables assigned to other subsystems. For this reason, these variables will be called *internal variables*, while the remaining will be called *input boundary variables*. The variables belonging to neighboring subsystems that influence at least an input boundary variable will be called *output boundary variables* (see Fig. 3.5).

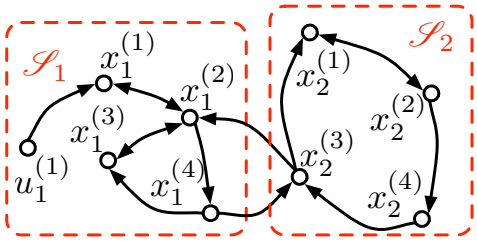


Figure 3.5: In this simple example, $x_1^{(2)}$ and $x_2^{(3)}$ are *input boundary variables*, $x_1^{(4)}$ and $x_2^{(3)}$ are *output boundary variables*, while the remaining ones are *internal variables*.

The second one does apply when we consider large–scale systems that are actually built by interconnecting a large number of subsystems. For instance, it may be a chemical plant built by interconnecting a large number of simple components such as pipes, pumps, valves and tanks. In this situation, it makes sense to choose a decomposition such that each subsystem $\mathscr{S}_I$ corresponds to an actual physical component: this kind of decomposition is called a *physical decomposition*, as opposed to a *mathematical decomposition* [89]. In a physical decomposition the local part of the model will account for the behavior of the component independently of the other components to which it is connected, whether the function g_I describes the effects of the interconnection. We will see in Section 3.4 why this may be important.

Now that the decomposition problem has been described, we will introduce a distributed FDI architecture that takes one of the possible decomposition solutions as the starting point to solve the FDI problem.

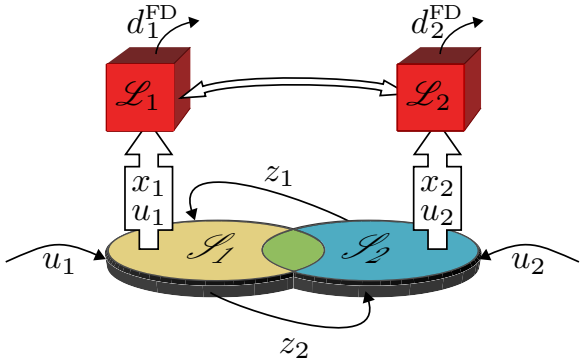


Figure 3.6: A scheme of the proposed DFDI architecture. In this example two subsystems $\mathscr{S}_1$ and $\mathscr{S}_2$ are represented as two ovals, a yellow and a blue one. Their green intersection represent their overlap, that is the part of the original system that they share. The subsystems inputs, as well as the interconnection variables are symbolized by black arrows. The two red cubes represent the two LFDs $\mathscr{L}_1$ and $\mathscr{L}_2$, that can take local measurements through the thick white arrows, and can communicate between each other thanks to the thinner white arrow.

3.4 The proposed distributed FDI architecture

"One subsystem, one diagnostic agent": this in short is the main idea about the distributed FDI architecture we are going to propose. The starting assumption is that a large–scale system, for which a centralized solution to the FDI problem is nor feasible nor desirable, has been decomposed into a multiset $\mathcal{D} \triangleq \{\mathscr{S}_1, ..., \mathscr{S}_N\}$ of N subsystems. The proposed *Distributed Fault Detection and Isolation* (DFDI) architecture will consists of a network of N agents called *Local Fault Diagnosers* (LFD), denoted by $\mathscr{L}_I$ and dedicated to monitor each of the subsystems $\mathscr{S}_I$ and provide a fault decision d_I^{FD} regarding their health.

Each LFD will possess a local and interconnection model of its subsystem, such as the simple one in equation (3.1). In order to use it for the task of computing an estimate of the local state, each LFD will be able to directly measure the local state x_I and the local input u_I. To populate the interconnection vector z_I, LFDs will communicate with their neighbors, exchanging the requested local measurements[6]. This scheme is exemplified in Fig. 3.6 for the case of only two subsystems.

Analogously to the centralized GOS implementation discussed in Chapter 2, the generic I–th LFD will provide a *local detection service* through a local FDAE estimator, that will be used to compute a local state estimate

[6]It will be assumed that the communication is fast enough so that the needed measurements can be exchanged in real time.

$\hat{x}_{I,0}$, a corresponding state estimation error $\epsilon_{I,0} \triangleq x_I - \hat{x}_{I,0}$ to be used as a residual, and a threshold $\bar{\epsilon}_{I,0}$. By assuming the existence of a local fault class $\mathcal{F}_I$, whose $N_{\mathcal{F}_I}$ fault functions represent the LFD knowledge about the peculiar way of failing of its own subsystem, a *local isolation service* will be provided. The isolation will be carried on thanks to $N_{\mathcal{F}_I}$ FIE estimators that will compute $N_{\mathcal{F}_I}$ further local state estimates $\hat{x}_{I,l}$, $l \in \{1, \ldots, N_{\mathcal{F}_I}\}$, along with as many state estimation errors $\epsilon_{I,l} \triangleq x_I - \hat{x}_{I,l}$ and thresholds $\bar{\epsilon}_{I,l}$. Of course the single–diagnoser fault detection and isolation logic presented in Chapter 2 will need to be refined in order to work in a multi–diagnoser setting, and this issue will be dealt with in Section 3.4.1. First some innovative traits of the proposed DFDI scheme will be presented.

The distributed solution to the FDI problem will present some innovative features with respect to the centralized one. The first one depends upon the fact that the interconnection function g_I will be assumed to be uncertain. This assumption makes the formulation more general, and it is especially useful in situations where the starting point is a physical decomposition. In such a situation each subsystem corresponds to a physical component, to which an LFD is logically attached. An LFD for a physical component can of course possess a nominal local model of its behavior when the component is operated in some known configuration, that is, for example, when it is operated in a test configuration defined by the component maker. But the effect of interconnecting the component to other pieces in a complex system cannot be completely modeled *a priori*. For this reason, the FDAE and FIE estimator will use an on–line adaptive approximator $\hat{g}$ in *lieu* of the uncertain interconnection function g. This on–line adaptive approximator will be implemented by the FDAE, that instead of using its adaptive capabilities for approximating a possibly unknown fault after detection, will approximate the interconnection before detection. The learning will be stopped as soon as a fault is detected in order to avoid the approximator learning the fault function as if it were part of the interconnection function. This will be addressed in details in Chapter 4, and covered again in Chapter 5.

The second difference between the distributed and centralized FDI scheme is that in the proposed DFDI architecture we allowed overlapping decompositions. This means that shared variables, belonging to more than one subsystem, will be monitored by more than one LFD too. This fact leads to the rather intuitive consideration that if something is monitored by "more eyes", or by "more than one point of view", better estimations and then better diagnoses can be obtained. In order to take advantage of this possibility, all the LFDs in the overlap set $\mathcal{O}_s$ of a shared variable $x^{(s)}$ will collaborate on the task of estimating it, and detecting and isolating faults by which it may be affected. This will be achieved by employing *consensus* techniques. Consensus and agreement problems pertains to the convergence of multiple estimates to a common value [79], and were extensively treated in the Computer Science literature concerning distributed fault diagnosis

of synchronous and asynchronous systems [99, 100]. In the control systems community, consensus filters have recently been the subject of significant research efforts, for instance in the framework of average–consensus on static and dynamic quantities by *sensor networks* [135, 82, 76, 87, 136]. In sensor networks [137, 43, 49] a high number of noisy measurements of the same variables are available, and the network itself is described by a graph where each node represents a sensor and each edge represents a communication channel linking two sensors. Each node of the network computes its own estimate of the measured variable by means of a linear or non–linear combination of its last measurement, its last estimate and the last estimates of neighboring nodes. By properly choosing the combination, that is described by both the graph and a *consensus protocol*, convergence to some desired function of the true variables value can be attained. The net result is to overcome the negative effect of the measurement noise on a single node, by letting all the nodes collaborate according to the graph describing the network. The simplest example of consensus corresponds to the case of a sensors, or agents, network where every node makes an initial noisy measurement of a single variable. The goal is to make the estimate of every node converge to a given function of the initial measurements, usually the average [138, 77]. Extensions to the basic scheme consider switching topology of the communication graph between nodes, variable communication time delays [78, 139], lossy communication channels [140], asynchronous communication [81], logical consensus [84], gossip algorithms [87] and nonlinear consensus protocols [78]. An interesting extension that addresses the performance of the consensus protocol [141, 136], consists in the use of a weighted adjacency matrix in order to optimize the convergence speed or the estimation error variance [138, 140, 136].

In the DFDI framework being developed, consensus techniques will be applied so that LFDs with lower uncertainties may help other LFDs in their overlap set to achieve the detectability, or isolability, condition. As the uncertainty, and its bound, is generally time–varying, one LFD cannot expect to have uncertainties low enough to always respect the detectability or isolability conditions by itself, so that the collaboration due to the consensus will generally prove beneficial to all the participating LFDs. How consensus techniques will be embedded in the estimation task of the proposed FDI architecture will be explained in Chapter 4 and, again, in Chapter 5.

3.4.1 Fault detection and isolation logic for multiple diagnosers

Now we will establish the way by which all the N LFDs will coordinate among themselves in order to detect and isolate faults affecting the system $\mathscr{S}$. First of all, it should be clear that if every LFD would be supposed to apply the FDI logic of Chapter 2 alone, without exchanging detection

and isolation decisions with other LFDs, this would not prevent them to correctly detect faults occurring to their subsystem. In fact, if we imagine to adapt the developments of Sections 2.3 and 2.4 to the generic I–th LFD, we can easily see that a properly defined local detection threshold $\bar{\epsilon}_{I,0}$ can be crossed by the local residual $\epsilon_{I,0}$ only if the fault is influencing some local variable. Immediately after detection the fault decision d_I^{FD} would be correct, even if the I–th LFD did not communicate it to the other LFDs. But not communicating it would anyway pose a risk, and now we will explain why.

Let us assume that the generic I–th LFD is the first to detect a fault, though in this hypothetical setting it does not know that it is the first. It can rightfully conclude that a fault did affect some components of its local state thus causing a non–empty signature. But, nothing can be deduced by the I–th LFD about what the fault is doing to other subsystems. We could simply hope that if a fault is itself distributed and is affecting other subsystems, other LFDs that are affected by the fault will detect it, even if the I–th LFD did not warn them. But of course this is not a good idea, as there will be situations in which the effect of the fault are widespread, but do not fulfill the detectability condition on other LFDs so that the fault goes unnoticed by them. These other LFDs will continue to work as their subsystems were healthy, posing a safety threat and unnecessarily learning the fault function with their FDAE approximator as if it were part of the interconnection function.

An isolation logic where the LFDs did not exchange their fault decisions would be even more problematic. Let us assume that the I–th LFD detects a fault influencing its subsystem, and accordingly starts trying to isolate it. All the hypothesis corresponding to the members of its local fault class $\mathcal{F}_I$ are simultaneously put to test thanks to its FIEs. Let us assume that at some time only one fault hypothesis remains unchallenged, so that the corresponding fault is isolated. Well, can it be said for sure that the actual fault is the one that has been isolated, or only that it is one that *locally* does exactly look as the isolated one? The answer is that without further information by other LFDs we cannot choose one or the other explanation.

This two examples shows the urgency for a DFDI logic where the LFDs exchange their fault decision. First of all, we will define again the concept of fault signature, but in the specific setting of distributed systems. Before detection the generic I–th LFD FDAE does test the following condition

$$|\epsilon_{I,0}^{(k)}(t)| \leq \bar{\epsilon}_{I,0}^{(k)}(t) \; \forall \, k = 1, \ldots, n_I \,, \tag{3.2}$$

that corresponds to the fault hypothesis.

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

Definition 3.4.1: The *local signature* shown by the subsystem $\mathscr{S}_I$, $I \in \{1, \ldots, N\}$ at time $t > 0$ is the index set $\mathcal{S}_I \triangleq \{k : \exists t_1, t \geq t_1 > 0, |\epsilon_{I,0}^{(k)}(t_1)| > \bar{\epsilon}_{I,0}^{(k)}(t_1)\}$ of the local state components for which the condition 3.2 did not hold for at least one time instant. □

Then, a *global signature* can be defined, too

Definition 3.4.2: The *global signature* shown by the system $\mathscr{S}$ at time $t > 0$ is the index set $\mathcal{S} \triangleq \{k : \exists t_1, t \geq t_1 > 0, \exists I \in \{1, \ldots, N\}, |\epsilon_{I,0}^{(j)}(t_1)| > \bar{\epsilon}_{I,0}^{(j)}(t_1), k \text{ is the } j\text{–th element of } \mathcal{I}_I\}$ of the state components for which the hypothesis 3.2 did not hold for at least one time instant and for at least one LFD. □

The detection logic for a single LFD can then be simply stated in the same way as in Chapter 2, with two differences:

- as soon as a LFD does detect a fault it will communicate this to all the other $N - 1$ LFDs[7]
- as soon as a LFD detects a fault or is informed about that, its FDAE approximator is *stopped* and its bank of FIEs is turned on.

Fault Detection Logic A fault affecting the subsystem $\mathscr{S}_I$ will be detected at the first time instant such that $\mathcal{S}_I$ becomes non-empty. This time instant will be called the *fault detection time* T_d.

As soon as a fault is detected in any of the N subsystems, the whole system will be declared as faulty and all the LFDs will start trying to isolate the fault. This behavior may be seen as paranoid and inefficient, as usually a fault sign in a subsystem will not mean that the whole system is influenced by the fault. But this is the only truly safe behavior: in fact "usually" is not enough when the goal is to provide safety against even a major but subtle failure. As it is always possible that a slowly developing but eventually catastrophic fault does show up in some limited parts of the system with only a local signature, the DFDI architecture should always act as this were the case.

Fault Isolation Logic We will now characterize faults with respect to their signature being distributed or not, as this will help explaining how the fault isolation logic for our DFDI scheme works. We will use an example, whose structural graph is depicted in Fig. 3.7 and shows an initially healthy system decomposed into three subsystems with some overlap.

[7] This is not a real threat to the communication constraint, as this one–to–all communication does happen only once and involves a simple binary information.

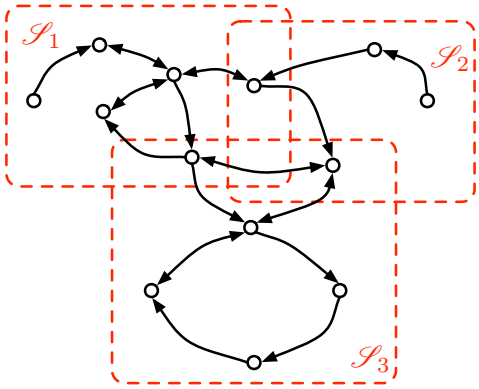


Figure 3.7: Healthy system decomposed into three subsystems.

Now the first and simplest of the possible fault scenarios will be analyzed.

Local Fault: The simplest situation is exemplified in Fig. 3.8. The structure of the fault is highlighted in green: green links represent part of the healthy dynamics changed by the fault, and green filled nodes represent variables affected by the fault. As can be seen, only the local signature $\mathcal{S}_1$ may become non-empty as this very fault affects only variables internal to subsystem $\mathscr{S}_1$, and that are not shared with any other subsystem. For this reason faults such this one will be termed *local faults* and their main feature, as far as regards our goal, is that the local and global signatures are the same and thus can be isolated by the corresponding LFD alone. An LFD, in fact, to isolate a local fault needs only to implement a fault isolation scheme as the one proposed in [113] and in Chapter 2, thus realizing a local GOS isolation scheme without further communication between neighboring LFDs except for the exchange of interconnection variables measurements.

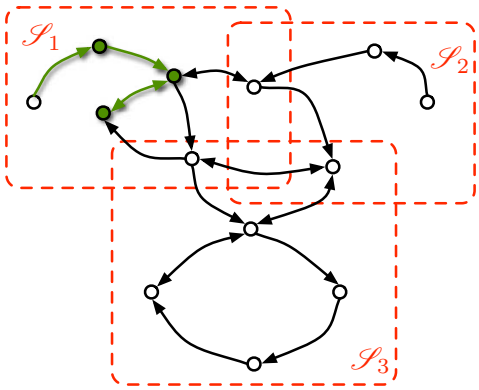


Figure 3.8: A local fault.

Distributed fault, non-overlapping signature: A more general situation

arises when links and variables in more than one subsystem are affected by the same single fault, so that the resulting global signature is distributed. Anyway it is assumed that shared variables are not affected, so that there can not be overlaps between signatures of neighboring subsystems[8] (Fig. 3.9).

It is clear that no single LFD alone can isolate such a fault, as any fault model belonging to its fault class may only explain the local signature, and not the global one. In order for this kind of fault to be isolated, the proposed DFDI isolation scheme will assume that every LFD concerned will have in its fault class a *local fault function* that is able to explain its local signature. Every LFD will then implement a GOS fault isolation scheme in a way similar to the previous case, but with the subtle difference that isolating the local part of a distributed fault will not be sufficient for a proper diagnosis. Only when all the LFDs involved will be able to isolate at some time their local part of the fault, then by communicating their successful diagnosis to each other they will collectively make a correct fault decision. It should be noted that, in general, this communication may not be limited to neighboring LFDs, as the global signature may be due to local signatures in subsystems located anywhere, but does anyway involve the exchange of only the information about the local diagnosis of a given distributed fault being locally isolated or not.

Although the scheme proposed will need the same local implementation in terms of FIEs as the previous one, that is the same as in the centralized case described in [113] and in Chapter 2, its important advantage is that it preserves scalability of the proposed DFDI scheme, and scalability is one of the important issues that motivated the present work. The scalability property holds as every LFD does need to know only local nominal models and local fault models related to its subsystem, and needs only to communicate limited information to other LFDs.

[8]This means, also, that there can be only a one-to-one relation between the global and all the non-empty local signatures.

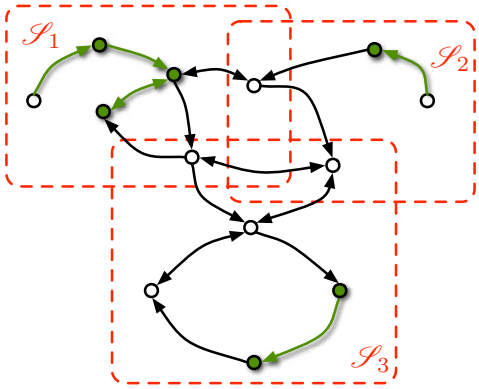


Figure 3.9: Distributed fault with non-overlapping signature.

Distributed fault, overlapping signature: In the most general situation, the signature of a distributed fault will generally include shared variables too (Fig. 3.10). As in the previous scenario, any single LFD will only be able to isolate the local part of the fault, and a correct fault decision can be made only collectively. But the fact that some variables interested by the signature are shared means that more than one LFD possesses a local model of the fault influence on those variables. This will enable the use of consensus techniques between LFDs sharing the same variable and possessing the same fault in their local class. As in the previous case, the distributed fault will be isolated only if all the LFDs will isolate their local part of the fault.

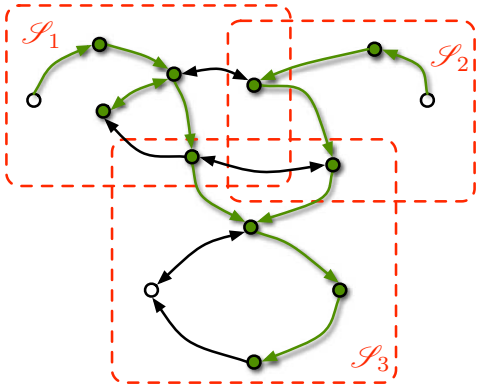


Figure 3.10: Distributed fault with overlapping signature.

The formal definitions regarding the fault detection and isolation logic for the proposed DFDI architecture will be given in Chapter 4.

3.5 Concluding remarks

In this chapter the motivations that justified the move from a centralized FDI architecture to a distributed one had been given. The way for achieving this move has been chosen to be through the divide et impera paradigm, and to this end the problem of structural analysis, and structure and model decomposition has been defined. A distributed FDI architecture, that is based on an overlapping decomposition that solves the decomposition problem, has then been proposed. Now, before presenting in details the architecture through two implementations of the DFDI scheme, in discrete and in continuous–time, an analysis of the consequences of the use of overlapping decomposition will be given.

Let us consider again the previous example, to which initially a centralized FDI scheme has now been applied (Fig. 3.11). The implementation of a DFDI scheme implies, as a starting point, a solution to the decomposition problem. How this solution may be found is not the subject of the present work, but anyway the effect of different kind of decomposition solutions on the fulfillment of the computation and communication constraints is of interest.

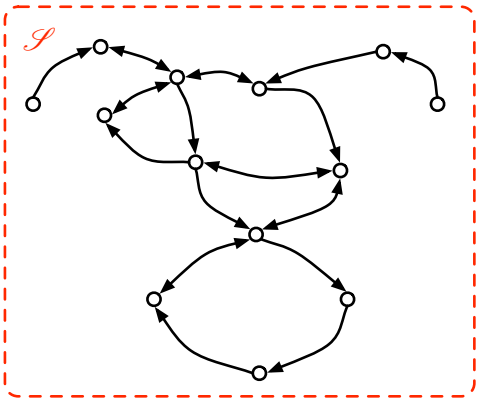


Figure 3.11: A trivial decomposition, equivalent to a monolithic system.

A limit situation is depicted in Fig. 3.12, and corresponds to a non–overlapping decomposition where each subsystem $\mathscr{S}_I$ contains exactly one variable. This solution will almost surely guarantees the fulfillment of the computation constraint, but on the other hand will probably not meet the communication constraint. In fact this solution requires a great communication effort because of the many arcs connecting different subsystems, that represents interconnection terms for which the measurement of interconnection variables from other subsystems is needed.

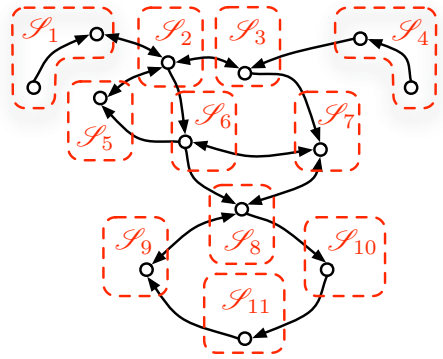


Figure 3.12: A limit decomposition in as may subsystems as the original system dimension.

A balanced situation is instead illustrated in Fig. 3.13, where the system $\mathscr{S}$ has been decomposed into subsystems of almost constant size, chosen so that the number of inbound arcs from neighboring subsystems is minimized. This is a non–overlapping decomposition and each variable is estimated exactly once, so that probably the computation constraint will be met, together with the communication one because of the low number of inbound arcs from neighbors.

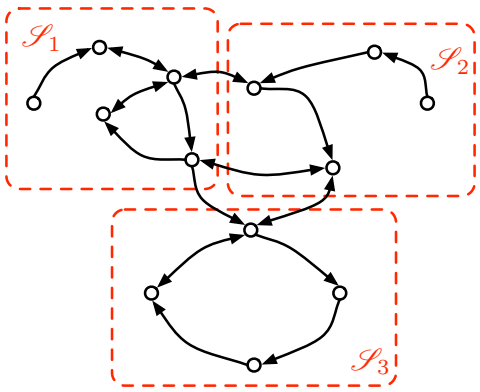


Figure 3.13: A supposedly "good" decomposition, without overlap.

A similar situation is the one of Fig. 3.14, that is obtained from the previous one by letting some overlap. The presence of overlap will call for a slightly greater combined computation power, but probably it still fulfills the computation constraint. Also some more communication is needed for the exchange of estimates and local model evaluations between the LFDs in each overlap set. The added value of overlap is that it can be arranged so that variables affected by greater model or measurement uncertainties are shared, thus possibly easing the requirements for the fault detectability or

isolability, as the next chapters will show.

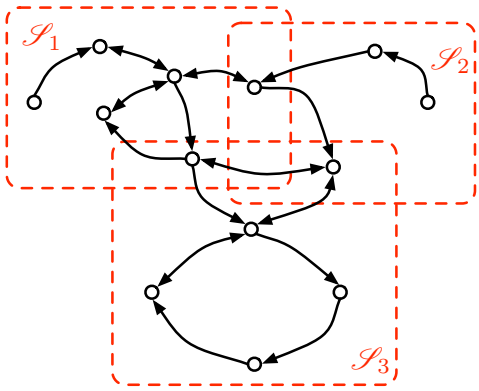


Figure 3.14: A supposedly "good" decomposition, with overlap.

The final, and again limit situation is presented in Fig. 3.15. Here all the subsystems correspond to the whole original system, thus leading to every variable being overlapped between all the diagnosers. The corresponding decomposition $\mathcal{D}$ contains N copies of the original system $\mathscr{S}$, and the necessity to account for this limit situation explains why we did use the concept of multiset earlier. Obviously this solution, for a large enough original system, will not meet the computation constraint: what should be this solution useful for, then? The answer is that such a solution does represent a well–studied case we talked about, that is one of a sensor or agents network where each node of the network, that is each LFD, monitors the same variables. This is important because, apart from the possibility in future developments to apply many of the analytical results in this field to DFDI problems, also shows how a DFDI functionality could be added to existing sensors networks architecture.

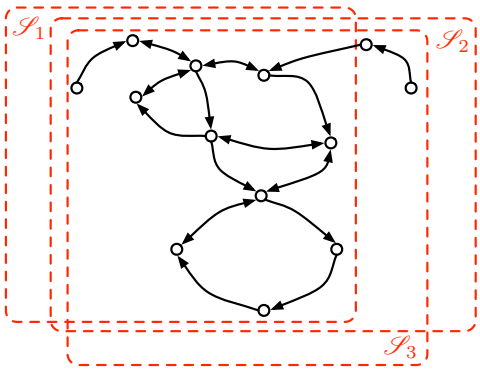


Figure 3.15: A limit decomposition where every subsystem is the original monolithic system.

Chapter 4

Distributed FDI for discrete-time systems

In this chapter an implementation of the DFDI architecture outlined in Chapter 3 will be developed for discrete–time systems. It will present some innovative features with respect to the centralized one of Chapter 2, namely the ability to approximate uncertain parts of the healthy model and the use of consensus techniques for improving the diagnosis capabilities on parts of the system shared by more than one diagnoser. Main analytical results regarding the robustness to uncertainties, the detectability and the isolability conditions, will be given. Finally, an illustrative simulation example consisting of an eleven tank system will be demonstrated.

4.1 Background and assumptions

Let us consider a nonlinear dynamic system $\mathscr{S}$ described by the following discrete–time model

$$\mathscr{S}: \; x(t+1) = f(x(t), u(t)) + \eta(t) + \beta(t - T_0)\phi(x(t), u(t)) \,, \tag{4.1}$$

where $t \in \mathbb{N}$ is the discrete time instant, $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the state and input vectors, respectively, and $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ represents the nominal healthy dynamics. Moreover, the function $\eta : \mathbb{N} \mapsto \mathbb{R}^n$ stands for the uncertainty in the state equation and includes external disturbances as well as modeling errors and possibly the discretization error. As to the faults affecting the nominal system modes, from a qualitative viewpoint, the term $\beta(t-T_0)\phi(x(t), u(t))$ represents the deviation in the system dynamics due to a fault. The term $\beta(t - T_0)$ characterizes the time profile of a fault that occurs at some *unknown* discrete-time instant T_0, and $\phi(x, u)$ denotes the nonlinear fault function.

This characterization allows both additive and multiplicative faults (since ϕ is a function of x and u), and even more general nonlinear faults. We let

the fault time profile $\beta(t-T_0)$ model either *abrupt* faults characterized by a "step-like" time–profile

$$\beta(t-T_0)=\begin{cases}0 & \text{if} \quad t<T_0\\ 1 & \text{if} \quad t\geq T_0\end{cases}, \tag{4.2}$$

or *incipient* faults characterized by an "exponential-like" time-profile

$$\beta(t-T_0)=\begin{cases}0 & \text{if} \quad t<T_0\\ 1-b^{-(t-T_0)} & \text{if} \quad t\geq T_0\end{cases}. \tag{4.3}$$

where $b>1$ denotes the unknown fault-evolution rate. Note that the fault time profile given by (4.3) only reflects the developing speed of the fault, while all its other basic features are captured by the function $\phi(x,u)$ described below.

Model in (4.1) may be impractical for fault detection and isolation (FDI), either because of its dimension, or because the system it represents is physically distributed, so that a centralized FDI architecture is neither possible nor desirable. This problem can be overcome by implementing a Distributed Fault Detection and Identification (DFDI) architecture as described in Chapter 3. To this end we will consider $\mathscr{S}$ as decomposed into N subsystems $\mathscr{S}_I$, $I=1,\ldots,N$, each characterized by a *local* state vector $x_I\in\mathbb{R}^{n_I}$, so that a separate monitoring system could be considered for each $\mathscr{S}_I$. To this end, the state equation of $\mathscr{S}_I$ can be modeled as

$$\mathscr{S}_I:\ x_I(t+1)=f_I^*(x(t),u(t))+\eta_I(t)+\beta(t-T_0)\phi_I(x(t),u(t))\,, \tag{4.4}$$

where the functions f_I^*, η_I and ϕ_I are built upon the components of f, η and ϕ that account for the dynamics of subsystem $\mathscr{S}_I$. The function f_I^* can then be conveniently split into two parts as follows:

$$\mathscr{S}_I:\ x_I(t+1)=f_I(x_I(t),u_I(t))+g_I(x_I(t),z_I(t),u_I(t))+\beta(t-T_0)\phi_I(x(t),u(t))\,, \tag{4.5}$$

where it has been supposed that the uncertainty term η_I affects only the interconnection part of the model and for this reason has been included in the function g_I. Specifically, $f_I:\mathbb{R}^{n_I}\times\mathbb{R}^{m_I}\mapsto\mathbb{R}^{n_I}$ is the *local nominal* function and $g_I:\mathbb{R}^{n_I}\times\mathbb{R}^{p_I}\times\mathbb{R}^{m_I}\mapsto\mathbb{R}^{n_I}$ represents the *interconnection function*; $u_I\in\mathbb{R}^{m_I}$, $(m_I\leq m)$ is the *local input*, and $z_I\in\mathbb{R}^{p_I}$, $(p_I\leq n-n_I)$ is the vector of *interconnection state variables,* that is constituted of variables through which other subsystems influence $\mathscr{S}_I$.

As in Chapter 2, for isolation purposes we assume that there are $N_{\mathcal{F}_I}$ known types of possible nonlinear fault functions describing the faults that may act on the I–th subsystem. As the known faults are assumed to retain the same structure of the healthy subsystem, their fault functions can be written in the form $\phi_I(x_I,z_I,u_I)$, and belong to a finite set given by

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

where, for the sake of simplicity, it will be assumed that the local fault functions from different LFDs that will try to isolate the same distributed fault will be given the same index[1]. This will make easier to formally define the isolation logic for distributed faults, that has been outlined in section 3.4.1.

Each fault function in $\mathcal{F}_I$ is assumed to be in the form

$$\phi_{I,l}(x_I(t), z_I(t), u_I(t)) = [(\vartheta_{I,l,1})^\top H_{I,l,1}(x_I(t), z_I(t), u_I(t)), \ldots, (\vartheta_{I,l,n_I})^\top H_{I,l,n_I}(x_I(t), z_I(t), u_I(t))]^\top, \quad (4.6)$$

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 "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". For the sake of simplicity and without much loss of generality, the parameter domains $\Theta_{I,l,k}$ are assumed to be origin–centered hyper–spheres with radius $M_{\Theta_{I,l,k}}$. The following assumptions are now needed.

Assumption 4.1.1: At time $t = 0$ no faults act on the system $\mathscr{S}$. Furthermore, 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 $(x_I(t), u_I(t)) \in \mathcal{R}_I^x \times \mathcal{R}_I^u$, $\forall\, I = 1, \ldots, N, \forall\, t \geq 0$. □

Clearly, as a consequence of Assumption 4.1.1, 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 variable z_I. The reason for introducing such a boundedness assumption is just a formal one. Since no fault accommodation is considered in this work, the feedback controller acting on the system $\mathscr{S}$ must be such that the measurable signals $x(t)$ and $u(t)$ remain bounded for all $t \geq 0$. However, it is important to state in advance that the design of the distributed FDI methodology will not depend on the specific structure of the controller that, accordingly, will not be detailed.

With reference to (4.5), it is worth noting that the interconnection function g_I includes the uncertainty represented by the term η_I. Therefore, in the sequel the following assumption will be needed.

Assumption 4.1.2: The interconnection function g_I is an unstructured and uncertain nonlinear function, but for each $k = 1, \ldots, n_I$, the k–th com-

[1] For example, let us assume the existence of three LFDs, and of four faults, where three of them are local faults and one is a distributed one influencing all the subsystems. Then the fault classes can be denoted as $\mathcal{F}_1 = \{\phi_{1,1}, \phi_{1,4}\}$, $\mathcal{F}_2 = \{\phi_{2,2}, \phi_{2,4}\}$, $\mathcal{F}_3 = \{\phi_{3,3}, \phi_{3,4}\}$, where $\phi_{1,1}$, $\phi_{2,2}$, $\phi_{3,3}$ are the local faults and $\phi_{I,4}$ with $I \in \{1, 2, 3\}$ are the three local models of the distributed fault. Of course, actual situations will need a more elaborate naming convention for faults, but this example should suffice to the present theoretical needs.

ponent of g_I is bounded by some known function[2], i.e.,

$$|g_I^{(k)}(t)| \leq \bar{g}_I^{(k)}(t)\,, \quad \forall\, x_I \in \mathcal{R}_I^x, \forall\, z_I \in \mathcal{R}_I^z, \forall\, u_I \in \mathcal{R}_I^u, \tag{4.7}$$

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

Assumption 4.1.3: The time profile parameter b is unknown but it is lower bounded by a known constant $\bar{b}$. □

Assumptions 4.1.2 and 4.1.3 make the problem analytically tractable, but are not a limitation in practical situations where some prior knowledge on the system operation is available.

4.2 Distributed Fault Detection and Identification Architecture

In this section, we will describe in details the proposed DFDI scheme outlined in Section 3.4. In general, the DFDI architecture is made of N communicating *Local Fault Diagnosers* (LFDs) $\mathscr{L}_I$, which are devoted to monitor each of the N subsystems a global system $\mathscr{S}$ has been decomposed into, by using a *Model based Analytical Redundancy Relation* approach [3, 1]. Each LFD $\mathscr{L}_I$ will provide a fault decision d_I^{FD} regarding the health of the corresponding subsystem $\mathscr{S}_I$, by relying on $N_{\mathcal{F}_I}+1$ nonlinear adaptive estimators of the local state x_I, with $I \in \{1, \ldots, N\}$. The first estimator, called *Fault Detection Approximation Estimator* (FDAE), will be based on the nominal healthy model (4.5) and will provide the detection capability. The remaining $N_{\mathcal{F}_I}$ estimators, called *Fault Isolation Estimators* (FIE) and meant to provide the isolation capability, will be based on models matched to each one of the $N_{\mathcal{F}_I}$ elements of the *fault class* $\mathcal{F}_I$.

Under normal operating conditions, that is from the instant the DFDI architecture is started at time $t = 0$ until a fault is detected, the FDAE is the only estimator that each LFD employs. After a fault is detected by any of the N LFDs, the FIEs of all the LFDs will be activated and will try to collectively isolate the occurred fault, by employing a kind of *Generalized Observer Scheme* [12, 17, 36].

Each LFD is allowed to take only local measurements of x_I and u_I, and to communicate with neighboring LFDs in $\mathcal{J}_I$ in order to populate the interconnection vector z_I. But, although an LFD will be able to measure exactly the input vector u_I, in order to slightly relax the full–state measurement assumption it will be assumed that it cannot directly measure x_I. Instead it will sense the following noisy version of x_I

$$y_I(t) \triangleq x_I(t) + \xi_I(t)\,,$$

[2]Again, when there is no risk of ambiguity and for the sake of simplicity, a compact notation like $g_I(t) \equiv g_I(x_I(t), z_I(t), u_I(t))$ will be used.

where ξ_I is an unknown function that represents the uncertainty associated to the process of measuring x_I by each LFD. From this fact, it follows also that instead of receiving the actual interconnection vector z_I, a LFD will receive from its neighbors the vector

$$v_I(t) \triangleq z_I(t) + \zeta_I(t),$$

where $\zeta_I(t)$ is made with the components of ξ_J, $J \in \mathcal{J}_I$ that affects the relevant components of the measurements y_J, $J \in \mathcal{J}_I$. The following further assumption is then needed

Assumption 4.2.1: The measuring uncertainties represented by the vectors ξ_I and ζ_I are unstructured and unknown, but for each $k = 1, \ldots, n_I$, the k–th component of ξ_I and of ζ_I are bounded by some known quantity, i.e.,

$$|\xi_I^{(k)}(t)| \leq \bar{\xi}_I^{(k)}, \quad |\zeta_I^{(k)}(t)| \leq \bar{\zeta}_I^{(k)}, \tag{4.8}$$

so that it is possible to define two compact region of interests such that $\xi_I(t) \in \mathcal{R}_I^{\xi}$ and $\zeta_I(t) \in \mathcal{R}_I^{\zeta}$. □

Under the assumptions made so far, a shared variable $x^{(s)}$ will be measured by distinct LFDs in the overlap set $\mathcal{O}_s$ with distinct uncertainties. Furthermore, because of Assumption 4.1.2, the interconnection part of the local model (4.5) will be affected by distinct uncertainties. Because of these considerations, it will be convenient for LFDs in the overlap set $\mathcal{O}_s$ to employ consensus techniques when implementing their FDAE and FIE estimators, as will be shown in Sections 4.3 and 4.6. In facts it may happen that a LFD might at some time experience uncertainties much higher than the ones of other LFDs in the overlap set, thus being disadvantaged in the detection and isolation task. By allowing LFDs to collaborate through consensus the effect of the unfavorable conditions can be reduced, thus providing on average an advantage to all the LFDs in the overlap sets. This rather intuitive point will be made clear in theorems 4.4.1 and 4.6.1.

In the next subsections, the design of the LFDs will be addressed according to the fault detection and identification methodology presented Chapters 2 and 3. First, we will present the detection task by considering the system under nominal (healthy) mode of behavior. Subsequently, the behavior of the system under faulty conditions will be analyzed, and the fault isolation mechanism will be described.

4.3 Healthy behavior and Fault Detection Estimator

At the time instant $t = 0$ the DFDI architecture is started and, by assumption, the system $\mathscr{S}$ is healthy. After each LFD is turned on, 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 will yield the following *estimation error*

$$\epsilon_{I,0} \triangleq y_I - \hat{x}_{I,0}\,,$$

which will play the role of a residual and will be compared, component by component, to a suitable *detection threshold* $\bar{\epsilon}_{I,0} \in \mathbb{R}_+^{n_I}$. As explained in Chapter 3, the following condition

$$|\epsilon_{I,0}^{(k)}(t)| \leq \bar{\epsilon}_{I,0}^{(k)}(t)\ \forall\, k = 1,\,\ldots,\,n_I \tag{4.9}$$

will be associated to the *fault hypothesis*

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

Should this condition be unmet at some time instant t, the hypothesis will be falsified and the subsystem will present a local fault signature $\mathcal{S}_I$. The fault detection logic for the I–th LFD can then be simply stated in terms of the local signature $\mathcal{S}_I$: a fault affecting the I–th subsystem will be detected by its LFD at the first time instant such that $\mathcal{S}_I$ becomes non-empty. This time instant will be called the *fault detection time* T_d

Definition 4.3.1: The *fault detection time* T_d is defined as $T_d \triangleq \min\{t :\ \exists\, I,\, I \in \{1,\ldots,N\},\, \exists\, k,\, k \in \{1,\ldots,n_I\}\ :\ |\epsilon_I^{(k)}(t)| > \bar{\epsilon}_I^{(k)}(t)\}$. □

This definition captures the fact that as soon as the fault is locally detected by one LFD, all the remaining LFDs are warned and the fault is globally detected. After a detection, all the LFDs switch to the faulty mode of behavior, as already explained in Chapter 3.

Now the way the state estimate $\hat{x}_I$ is produced by means of the FDAE will be discussed. The FDAE is a nonlinear adaptive estimator based on the subsystem model (4.5), and will be described now for the more general case of a shared variable $x^{(s)}$. As anticipated, when a variable is shared all the LFDs in its overlap set will employ consensus techniques in order to reach an agreement on its estimate, by exchanging their local estimates and their local models. In order to make the analysis as general as possible, it will be assumed that the consensus–related interactions between the LFDs in $\mathcal{O}_s$ depends on a communication graph $\mathcal{G}_s \triangleq (\mathcal{N}_s, \mathcal{E}_s)$. This will be useful for considering limit situations as the last one described in Section 3.5.

The estimator dynamics for the component $\hat{x}_{I,0}^{(s_I)}$ computed by the I–th LFD, $I \in \mathcal{O}_s$, before the detection of a fault, that is for $t < T_d$, takes the form

$$\begin{aligned}\hat{x}_{I,0}^{(s_I)}(t+1) = \lambda \left\{ \hat{x}_{I,0}^{(s_I)}(t) - y_I^{(s_I)}(t) + \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left[\hat{x}_{J,0}^{(s_J)}(t) - \hat{x}_{I,0}^{(s_I)}(t) \right] \right\} \\ + \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} [f_J^{(s_J)}(y_J(t), u_J(t)) + \hat{g}_J^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,0})]\end{aligned} \tag{4.10}$$

where $0 < \lambda < 1$, and $W_s = [W_s^{(I,J)}]$ is a weighted adjacency matrix needed for implementing a linear consensus protocol on x_s. The adjacency matrix W_s will be compatible with the consensus communication graph $\mathcal{G}_s$ associated to the LFDs in $\mathcal{O}_s$. In this work only doubly-stochastic adjacency matrices $W_s \in \mathbb{R}^{N_s \times N_s}$ will be considered, for instance the Metropolis adjacency matrices [140, 138] defined as

$$W_s^{(I,J)} = \begin{cases} 0 & (I,J) \notin \mathcal{E}_s \\ \frac{1}{1+\max\{d_s^{(I)}, d_s^{(J)}\}} & (I,J) \in \mathcal{E}_s,\, I \neq J \\ 1 - \sum_{K \neq I} W_s^{(I,K)} & I = J \end{cases}, \tag{4.11}$$

where $d_s^{(I)}$ is the degree of the I–th node in the communication graph $\mathcal{G}_s$. The term $\hat{g}_J^{(s_J)}$ is the s_J–th output of an adaptive approximator meant to learn the interconnection function g_J, and $\hat{\vartheta}_J \in \hat{\Theta}_J \subset \mathbb{R}^{q_J}$ denotes its adjustable parameters vector. For the sake of simplicity, $\hat{\Theta}_J$ is assumed to be an origin–centered hyper–sphere, with radius $M_{\hat{\Theta}_J}$.

In order for $\hat{g}_J$ to learn the interconnection function g_J, its parameters vector is updated according to the following learning law:

$$\hat{\vartheta}_{J,0}(t+1) = \mathcal{P}_{\hat{\Theta}_{J,0}}(\hat{\vartheta}_{J,0}(t) + \gamma_{J,0}(t) H_{J,0}^{\top}(t) r_{J,0}(t+1))\,,$$

where $H_{J,0}(t) \triangleq \partial \hat{g}_J(t)/\partial \hat{\vartheta}_{J,0} \in \mathbb{R}^{n_J \times q_J}$ is the gradient matrix of the on–line approximator with respect to its adjustable parameters, $r_{J,0}(t+1)$ is the signal

$$r_{J,0}(t+1) = \epsilon_{J,0}(k+1) - \lambda \epsilon_{J,0}(t),$$

and $P_{\hat{\Theta}_{J,0}}$ is a projection operator [115]

$$\mathcal{P}_{\hat{\Theta}_{J,0}}(\hat{\vartheta}_{J,0}) \triangleq \begin{cases} \hat{\vartheta}_{J,0} & \text{if } |\hat{\vartheta}_{J,0}| \leq M_{\hat{\Theta}_{J,0}} \\ \frac{M_{\hat{\Theta}_{J,0}}}{|\hat{\vartheta}_{J,0}|} \hat{\vartheta}_{J,0} & \text{if } |\hat{\vartheta}_{J,0}| > M_{\hat{\Theta}_{J,0}} \end{cases},$$

The learning rate $\gamma_{J,0}(t)$ is computed at each step as

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

where $\|\cdot\|_F$ is the Frobenius norm and $\varepsilon_{J,0}$, $\mu_{J,0}$ are design constants that guarantee the stability of the learning law [115, 116, 117, 118, 119].

It is worth noting that, in order to implement (4.10), the LFD $\mathscr{L}_I$ does not need the information about the expressions of $f_J^{(s_J)}$ and of $g_J^{(s_J)}$; instead, it suffices that $\mathscr{L}_J$, $J \in \mathcal{O}_s$, computes the term $f_J^{(s_J)} + g_J^{(s_J)}$ and communicates it to other LFDs in $\mathcal{O}_s$ alongside its actual state estimate $\hat{x}_{J,0}^{(s_J)}$.

Before the occurrence of a fault, for $t < T_0 < T_d$, the dynamics of the LFD estimation error component $\epsilon_{I,0}^{(s_I)}$can be written as

$$\epsilon_{I,0}^{(s_I)}(t+1) = \lambda \left\{ \epsilon_{I,0}^{(s_I)}(t) + \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left[\epsilon_{J,0}^{(s_J)}(t) - \epsilon_{I,0}^{(s_I)}(t) + \xi_I^{(s_I)}(t) - \xi_J^{(s_J)}(t) \right] \right\} + \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[f_J^{(s_J)}(x_J(t), u_J(t)) - f_J^{(s_J)}(y_J(t), u_J(t)) + g_J^{(s_J)}(t) - \hat{g}_J^{(s_J)}(t) \Big] + \xi_I^{(s_I)}(t+1)\,,$$

and, by remembering that $\sum_{I \neq J} W_s^{(I,J)} = 1 - W_s^{(I,I)}$ by assumption, it holds

$$\epsilon_{I,0}^{(s_I)}(t+1) = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \left\{ \lambda[\epsilon_{J,0}^{(s_J)}(t) - \xi_J^{(s_J)}(t)] + \Delta f_J^{(s_J)}(t) + \Delta g_J^{(s_J)}(t) \right\} + \lambda \xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,,$$

where by definition it is

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

The function Δf_I will generally assume a non–zero value because of the measurement uncertainty ξ_I, while there are many reasons for Δg_I doing the same: the local measurement uncertainty ξ_I, the measurement uncertainty of neighboring LFDs, and the uncertainty in the interconnection function g_I itself due to the fact that by definition it includes the original uncertainty term η_I. 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. This may be formalized by introducing an *optimal weight vector* $\hat{\vartheta}_{I,0}^*$ [33]

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

where $\mathcal{R}_I \triangleq \mathcal{R}_I^x \times \mathcal{R}_I^z \times \mathcal{R}_I^u$, and by introducing a *minimum functional approximation error* (MFAE)

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

By defining the *parameter estimation error* $\tilde{\vartheta}_{I,0} \triangleq \hat{\vartheta}_{I,0}^* - \hat{\vartheta}_{I,0}$ and the following function

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

it can be written

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

Thanks to (4.10), the dynamics of the LFD estimation error component $\epsilon_{I,0}^{(s_I)}$ before the occurrence of a fault, for $t < T_0 < T_d$, can be written as

$$\epsilon_{I,0}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)}\left[\lambda\epsilon_{J,0}^{(s_J)}(t) + \chi_J^{(s_J)}(t)\right] + \lambda\xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,, \tag{4.12}$$

where we introduced the following *total uncertainty* term

$$\chi_I^{(s_I)}(t) \triangleq \Delta f_I^{(s_I)}(t) - \lambda\xi_I^{(s_J)}(t) + \Delta g_I^{(s_I)}(t)\,.$$

In order to study the behavior of $\epsilon_{I,0}^{(s_I)}(t)$ and define the threshold $\bar{\epsilon}_{I,0}^{(s_I)}(t)$, it is convenient to introduce the following vectors related to the detection estimator of all the LFDs sharing the variable $x^{(s)}$

$$\epsilon_{s,0}(t) \triangleq \mathrm{col}(\epsilon_{I,0}^{(s_I)}, I\in\mathcal{O}_s),\ \chi_s(t) \triangleq \mathrm{col}(\chi_I^{(s_I)}, I\in\mathcal{O}_s),\ \xi_s(t) \triangleq \mathrm{col}(\xi_I^{(s_I)}, I\in\mathcal{O}_s)\,.$$

The FDAE estimation error dynamics of all the LFDs in $\mathcal{O}_s$ can then be written in a more useful and compact form

$$\epsilon_{s,0}(t+1) = W_s\left[\lambda\epsilon_{s,0}(t) + \chi_s(t)\right] + \lambda\xi_s(t) + \xi_s(t+1)\,. \tag{4.13}$$

As W_s is a doubly stochastic matrix by assumption, all its eigenvalues belongs to the unitary circle [142]. Then it follows that (4.13) represents the dynamics of a stable *Linear Time Invariant* (LTI) discrete–time systems with all the eigenvalues inside a circle of radius $\lambda < 1$. The solution to (4.13) is

$$\begin{aligned}\epsilon_{s,0}(t) &= \sum_{h=0}^{t-1}(\lambda W_s)^{t-1-h}\left[W_s\chi_s(h) + \lambda\xi_s(h) + \xi_s(h+1)\right] + \lambda^t W_s^t\epsilon_{s,0}\\ &= W_s\left\{\lambda\left[\sum_{h=0}^{t-2}(\lambda W_s)^{t-2-h}\left(W_s\chi_s(h) + \lambda\xi_s(h) + \xi_s(h+1)\right)\right.\right.\\ &\qquad \left.\left. +\lambda^{t-1}W_s^{t-1}\epsilon_{s,0}(0)\right] + \chi_s(t-1)\right\} + \lambda\xi_s(t-1) + \xi_s(t)\,,\end{aligned} \tag{4.14}$$

so that component-wise it reads

$$\begin{aligned}\epsilon_{I,0}^{(s_I)}(t) \equiv \epsilon_{s,0}^{(I)}(t) = w_{s,I}^\top\left\{\lambda\left[\sum_{h=0}^{t-2}(\lambda W_s)^{t-2-h}\left(W_s\chi_s(h) + \lambda\xi_s(h) + \xi_s(h+1)\right)\right.\right.\\ \left.\left. +\lambda^{t-1}W_s^{t-1}\epsilon_{s,0}\right] + \chi_s(t-1)\right\} + \lambda\xi_s(t-1) + \xi_s^{(I)}(t)\,,\end{aligned}$$

where $w_{s,I}^{\top}$ is a vector containing the I–th row of matrix W_s.

Now a threshold on the estimation error that guarantees no false–positive fault detections for $t < T_0$ will be derived. The absolute value of the estimation error for $t < T_0$ can be upper bounded by relying on the triangular inequality

$$
\begin{aligned}
|\epsilon_{I,0}^{(s_I)}(t+1)| &\leq \sum_{J\in\mathcal{O}_s} |W_s^{(I,J)} \left[\lambda\epsilon_{J,0}^{(s_J)}(t) + \chi_J^{(s_J)}(t)\right]| + |\lambda\xi_I^{(s_I)}(t)| + |\xi_I^{(s_I)}(t+1)|\,,\\
|\epsilon_{I,0}^{(s_I)}(t+1)| &\leq \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \left[\lambda|\epsilon_{J,0}^{(s_J)}(t)| + |\chi_J^{(s_J)}(t)|\right] + \lambda|\xi_I^{(s_I)}(t)| + |\xi_I^{(s_I)}(t+1)|\,,\\
|\epsilon_{I,0}^{(s_I)}(t+1)| &\leq \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \left[\lambda|\epsilon_{J,0}^{(s_J)}(t)| + \bar{\chi}_J^{(s_J)}(t)\right] + \lambda\bar{\xi}_I^{(s_I)}(t) + \bar{\xi}_I^{(s_I)}(t+1)\,,
\end{aligned}
\tag{4.15}
$$

where the upper bound on the total uncertainty term was defined as[3]

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

with the function $\kappa_{J,0}$ being such[4] $\kappa_{J,0}(\hat{\vartheta}_{J,0}) \geq \|\tilde{\vartheta}_{J,0}\|$.

The inequalities (4.15), by applying the absolute value component-wise so that $|\epsilon_{s,0}| \equiv \mathrm{col}(|\epsilon_{I,0}^{(s_I)}| : I \in \mathcal{O}_s)$, can be written as

$$
|\epsilon_{s,0}(t+1)| \leq W_s\left[\lambda|\epsilon_{s,0}(t)| + \bar{\chi}_s(t)\right] + \lambda\bar{\xi}_s(t) + \bar{\xi}_s(t+1)\,.
$$

By using the Comparison Lemma [143], then, the absolute value of each component of ϵ_s can be bounded by the corresponding component of $\bar{\epsilon}_s$, defined as the solution of the following equation

$$
\bar{\epsilon}_s(t+1) = W_s\left[\lambda\bar{\epsilon}_s(t) + \bar{\chi}_s(t)\right] + \lambda\bar{\xi}_s(t) + \bar{\xi}_s(t+1)\,, \tag{4.16}
$$

with initial conditions

$$
\bar{\epsilon}_s(0) \triangleq \mathrm{col}(\bar{\xi}_I^{(s_I)}(0) : I \in \mathcal{O}_s)\,.
$$

It is worth noting that the adaptive threshold defined in (4.13) can be easily computed by any LFD in $\mathcal{O}_s$ by means of linear filtering techniques [36]. As in Chapter 2, the main property of the threshold is its robustness with respect to all the modeling and measuring uncertainties, so that the absence of false–positive fault detections is guaranteed.

[3] Notations such as $\max_{\xi_J}$ are short for $\max_{\xi_J\in\mathcal{R}^{\xi_J}}$.

[4] As $\Theta_{J,0}$ is a compact the function $\kappa_{J,0}$ can always be defined, as it was done in Chapter 2.

For a non–shared component $x_{I,0}^{(j)}$ the estimator equation (4.10) and the error equation (4.13) simply become

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

$$\epsilon_{I,0}^{(j)}(t+1) = \left[\lambda\epsilon_{I,0}^{(j)}(t) + \chi_I^{(j)}(t)\right] + \lambda\xi_I^{(j)}(t) + \xi_I^{(j)}(t+1)\,,$$

and the threshold equation can be written as

$$\bar{\epsilon}_{I,0}^{(j)}(0) \triangleq \bar{\xi}_I^{(j)}(0), \quad \bar{\epsilon}_{I,0}^{(j)}(t+1) \triangleq \lambda\bar{\epsilon}_{I,0}^{(j)}(t) + \bar{\chi}_I^{(j)}(t) + \lambda\bar{\xi}_I^{(j)}(t) + \bar{\xi}_I^{(j)}(t+1)\,.$$

4.4 Faulty behavior and Fault Detectability

Now the behavior of the proposed DFDI architecture under faulty condition and its detection capabilities will be investigated. After the occurrence of a fault, for $t \geq T_0$, the error dynamics equation (4.13) for a shared component becomes

$$\epsilon_{s,0}(t+1) = W_s\left[\lambda\epsilon_{s,0}(t) + \chi_s(t)\right] + (1 - b^{-(t-T_0)})\phi_s(t) + \lambda\xi_s(t) + \xi_s(t+1)\,, \tag{4.17}$$

where $\phi_s(t) \in \mathbb{R}^{N_s}$ is a vector whose components are all equal to $\phi^{(s)}$. The following theorem gives a sufficient condition for the estimation error to cross its threshold, thus allowing the fault to be detected

Theorem 4.4.1 (Fault Detectability): If there exist a time index $t_1 > T_0$ and a subsystem $\mathscr{S}_I$ such that the fault ϕ_I fulfills the following inequality for at least one component $s_I \in \{1, \ldots, n_I\}$

$$\left|\sum_{h=T_0}^{t_1-1} \lambda^{t_1-1-h}(1 - b^{-(h-T_0)})\phi^{(s)}(h)\right| > 2\bar{\epsilon}_{I,0}^{(s_I)}(t_1)\,,$$

then it will be detected at t_1, that is $|\epsilon_{I,0}^{(s_I)}(t_1)| > \bar{\epsilon}_{I,0}^{(s_I)}(t_1)$. □

Proof: At the time index $t_1 > T_0$ the s_I–th component of the estimation error ϵ_I is equal to[5]

$$\begin{aligned}\epsilon_{I,0}^{(s_I)}(t_1) = w_{s,I}^{\top}\Bigg\{\lambda\Bigg[\sum_{h=0}^{t_1-2}(\lambda W_s)^{t_1-2-h}\left(W_s\chi_s(h) + \lambda\xi_s(h) + \xi_s(h+1)\right) \\ +\lambda^{t_1-1}W_s^{t_1-1}\epsilon_{s,0}(0)\Big] + \chi_s(t_1-1)\Bigg\} + \lambda\xi_s^{(I)}(t_1-1) + \xi_s^{(I)}(t_1) \\ + \sum_{h=T_0}^{t_1-1}\lambda^{t_1-1-h}(1 - b^{-(h-T_0)})\phi^{(s)}(h)\,,\end{aligned}$$

[5] As W_s is doubly stochastic and all the components of ϕ_s are equal to $\phi^{(s)}$, it holds $(W_s)^k\phi_s = \phi_s$ for all k.

so that, by using the triangle inequality, it holds

$$|\epsilon_{I,0}^{(s_I)}(t_1)| \geq -|w_{s,I}^{\top}\lambda \sum_{h=0}^{t_1-2}(\lambda W)_s^{t_1-2-h}(W_s\chi_s(h)+\lambda\xi_s(h)+\xi_s(h+1))| \\ - |\lambda^{t_1} w_{s,I}^{\top} W_s^{t_1-1}\epsilon_{s,0}(0)| - |w_{s,I}^{\top}\chi_s(t_1-1)| - |\lambda\xi_s(t_1)| - |\xi_s(t_1)| \\ + |\sum_{h=T_0}^{t_1-1}\lambda^{t_1-1-h}(1-b^{-(h-T_0)})\phi^{(s)}(h)|\,.$$

The threshold can be written as

$$\bar{\epsilon}_{I,0}^{(s_I)}(t_1) = w_{s,I}^{\top}\left\{\lambda\left[\sum_{h=0}^{t_1-2}(\lambda W_s)^{t_1-2-h}\left(W_s\bar{\chi}_s(h)+\lambda\bar{\xi}_s(h)+\bar{\xi}_s(h+1)\right)\right.\right. \\ \left.\left.+\lambda^{t_1-1}W_s^{t_1-1}\bar{\epsilon}_{s,0}(0)\right]+\bar{\chi}_s(t_1-1)\right\}+\lambda\bar{\xi}_s^{(I)}(t_1-1)+\bar{\xi}_s^{(I)}(t_1)\,,$$

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

$$|\epsilon_{I,0}^{(s_I)}(t_1)| \geq -\bar{\epsilon}_{I,0}^{(s_I)}(t_1) + |\sum_{h=T_0}^{t_1-1}\lambda^{t_1-1-h}(1-b^{-(h-T_0)})\phi^{(s)}(h)|\,,$$

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 4.4.1: Theorem 4.4.1, that easily translates to the case of non–shared variables, provides a way to check whether a fault will be detectable, that is whether at least one subsystem will show a non-empty signature. Anyway the fact that a fault function affects some state variables in a subsystem does not assure that a signature will include that variables: because of the way the fault detection logic was defined, the signature will be due only to the state variables for which the fault function fulfills the hypothesis of Theorem 4.4.1. On the other side, the scheme defined so far guarantees that a signature will never involve variables not influenced by the fault function ϕ_I. Thanks to the present choice of detection thresholds, the proposed DFDI scheme will not show the effect called *fault propagation* [1], where a fault determines the issuing of alarms on variables not directly affected by the fault[6]. □

[6] A more evocative name for this effect is the *Christmas Tree syndrome*, that can happen with FDI schemes based on simple limit checking. After a fault triggers a first alarm, it may happen that the changes in the operating point due to the fault will trigger almost every remaining alarms in a short time.

4.5 Fault isolation logic

After a fault has been detected at time T_d, the learning of the FDAE interconnection adaptive approximator $\hat{g}_I(t)$ of every LFD is stopped, that is $\hat{\vartheta}_{I,0}(t) = \hat{\vartheta}_{I,0}(T_d)\,,\ \forall\, t \geq T_d$, to prevent the interconnection approximator to learn part of the fault function ϕ_I too[7]. At the same time, each LFD will enable a bank of $N_{\mathcal{F}_I}$, $I = 1, \ldots, N$, Fault Isolation Estimators (FIEs) in order to implement a kind of *Generalized Observer Scheme* for the task of fault isolation, such as the one described in [36] and Chapter 2. This scheme relies on each FIE being matched to a specific fault function belonging to the *fault class* $\mathcal{F}_I$.

This is carried on by enabling the $N_{\mathcal{F}_I}$ FIEs that allow to test in parallel the $N_{\mathcal{F}_I}$ fault hypotheses

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

with $l = 1, \ldots, N_{\mathcal{F}_I}$. To this end, analogously to the FDAE, the l–th FIE will provide its own *local state estimate* $\hat{x}_{I,l}$ of the local state x_I. The difference between the estimate $\hat{x}_{I,l}$ and the measurements y_I will yield the following *estimation error*

$$\epsilon_{I,l} \triangleq y_I - \hat{x}_{I,l}\,,$$

which again will be used as a residual and compared, component by component, to a suitable *detection threshold* $\bar{\epsilon}_{I,l} \in \mathbb{R}^{n_I}_+$. The following condition

$$|\epsilon^{(k)}_{I,l}(t)| \leq \bar{\epsilon}^{(k)}_{I,l}(t)\ \forall\, k = 1,\ \ldots,\ n_I \tag{4.18}$$

will be associated to the l–th fault hypothesis. Should this condition be unmet at some time instant t, the hypothesis will be falsified and the corresponding fault will be excluded as a possible cause of the signature, at the exclusion time $T_{e,I,l}$.

Definition 4.5.1: The *l–th fault exclusion time* $T_{e,I,l}$ is defined as $T_{e,I,l} \triangleq \min\{t : \exists\, k,\ k \in \{1, \ldots, n_I\},\ |\epsilon^{(k)}_{I,l}(t)| > \bar{\epsilon}^{(k)}_{I,l}(t)\}$. □

Ideally, the goal of the isolation logic is to exclude every but one fault, which *may* be said to be *isolated*. In fact in the proposed DFDI setting a distinction should be drawn on the way local and distributed faults are isolated, according to the discussion in Section 3.4.1. If a fault is local, then having the corresponding LFD exclude every but that fault is sufficient for declaring it isolated. But, for distributed faults the isolation needs that all the LFDs having a local part of it in their fault classes did exclude all their other faults. To express this in a formal way, the Definition 2.5.2 that was used for centralized FDI schemes will be split as

[7] As stressed in Section 3.4.1, the learning of every LFD should be stopped, and not just the one of LFDs showing a non-empty signature. The fault may affect subsystems with empty signatures too and still be undetected there, because there it does not meet the detectability condition.

Definition 4.5.2: A fault $\phi_{I,p} \in \mathcal{F}_I$ is *locally isolated* at time t iff $\forall l$, $l \in \{1,\ldots,N_{\mathcal{F}_I}\}\setminus p\,, T_{e,I,l} \leq t$ and $\nexists\, T_{e,I,p}$. Furthermore $T_{lis,I,p} \triangleq \min\{T_{e,I,l},\, l \in \{1,\ldots,N_{\mathcal{F}_I}\}\setminus p\}$ is the *local fault isolation time.* □

Definition 4.5.3: A fault $\phi_{I,p} \in \mathcal{F}_I$ is *isolated* if for each LFD the corresponding local functions $\phi_{J,p}$ either has been isolated or do not exists, $J \in \{1,\ldots,N\}$. Furthermore $T_{is,I,p} \triangleq \min\{T_{lis,J,p},\, J \in \{1,\ldots,N\}\}$ is the *fault isolation time.* □

Remark 4.5.1: Again we should note that, if a fault has been isolated, we can conclude that it actually occurred if we assume a priori that only faults belonging to the class $\mathcal{F}_I$ may occur. Otherwise, it can only be said that it is not impossible that it occurred. If every fault in $\mathcal{F}_I$ is excluded, the following explanations may be given:

- an unknown fault, either local or distributed, has been isolated;
- the fault detection was triggered by a local or distributed fault of another subsystem.

□

4.6 Fault isolation and Fault Isolation Estimators

Now the Fault Isolation Estimators will be finally described. After the fault $\phi(t)$ has occurred, the state equation of the s_I–th component of the I–th subsystem becomes

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

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

$$\begin{aligned}\hat{x}_{I,l}^{(s_I)}(t+1) = \lambda \Bigg\{ \hat{x}_{I,l}^{(s_I)}(t) - y_I^{(s_I)}(t) + \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \left[\hat{x}_{J,l}^{(s_J)}(t) - \hat{x}_{I,l}^{(s_I)}(t)\right] \Bigg\} \\ + \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \Big[f_J^{(s_J)}(y_J(t), u_J(t)) + \hat{g}_J^{(s_J)}(t) \\ + \hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,l}) \Big]\,, \quad (4.19)\end{aligned}$$

where

$$\hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,l}) \triangleq (\vartheta_{J,l,s_J})^\top H_{J,l,s_J}(y_J(t), v_J(t), u_J(t))$$

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 $\hat{\vartheta}_{J,l} \triangleq \mathrm{col}(\vartheta_{J,l,k}, k \in \{1,\ldots,n_I\})$ has been introduced.

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

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

where $r_{J,l,k}(t+1)$ is the signal

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

and $P_{\hat{\Theta}_{J,l,k}}$ is a projection operator [115]

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

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

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

The corresponding estimation error dynamic equation is

$$\begin{aligned}\epsilon^{(s_I)}_{I,l}(t+1) = \lambda \Bigg\{ \epsilon^{(s_I)}_{I,l}(t) + \sum_{J \in \mathcal{O}_s} W^{(I,J)}_s \Big[\epsilon^{(s_J)}_{J,l}(t) - \epsilon^{(s_I)}_{I,l}(t) \\ + \xi^{(s_I)}_I(t) - \xi^{(s_J)}_J(t) \Big] \Bigg\} + \sum_{J \in \mathcal{O}_s} W^{(I,J)}_s \Big[\Delta f^{(s_J)}_J(T) + \Delta g^{(s_J)}_J(t) \\ + (1 - b^{-(t-T_0)})\phi^{(s)}(t) - \hat{\phi}^{(s_J)}_{J,l}(t) \Big] + \xi^{(s_I)}_I(t+1)\,,\end{aligned}$$

that is

$$\begin{aligned}\epsilon^{(s_I)}_{I,l}(t+1) = \sum_{J \in \mathcal{O}_s} W^{(I,J)}_s \Big[\lambda \epsilon^{(s_J)}_{J,l}(t) + \chi^{(s_J)}_J(t) + (1 - b^{-(t-T_0)})\phi^{(s)}(t) - \hat{\phi}^{(s_J)}_{J,l}(t) \Big] \\ + \lambda \xi^{(s_I)}_I(t) + \xi^{(s_I)}_I(t+1)\,.\end{aligned}$$

Now, supposing a matched fault, that is $\phi^{(s)}(t) = \phi^{(s_J)}_{J,l}(x_J(t), z_J(t), u_J(t), \vartheta_{J,l})$, $\forall\, J \in \mathcal{O}_s$, the error equation can be written as

$$\begin{aligned}\epsilon^{(s_I)}_{I,l}(t+1) = \sum_{J \in \mathcal{O}_s} W^{(I,J)}_s \Big[\lambda \epsilon^{(s_J)}_{J,l}(t) + \chi^{(s_J)}_J(t) \\ + (1 - b^{-(t-T_0)})(H_{J,l,s_J}(t)^{\top}\vartheta_{J,l,s_J} + \Delta H^{\top}_{J,l,s_J}\vartheta_{J,l,s_J}) - H_{J,l,s_J}(t)^{\top}\hat{\vartheta}_{J,l,s_J} \Big] \\ + \lambda \xi^{(s_I)}_I(t) + \xi^{(s_I)}_I(t+1)\,,\end{aligned}$$

where it was introduced

$$\Delta H^{\top}_{J,l,s_J}(t) \triangleq H_{J,l,s_J}(x_J(t), z_J(t), u_J(t)) - H_{J,l,s_J}(y_J(t), v_J(t), u_J(t))\,.$$

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

$$\begin{aligned}\epsilon_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \Big[\lambda\epsilon_{J,l}^{(s_J)}(t) + \chi_J^{(s_J)}(t) + (1-b^{-(t-T_0)})H_{J,l,s_J}(t)^\top\tilde{\vartheta}_{J,l,s_J} \\ +(1-b^{-(t-T_0)})\Delta H_{J,l,s_J}(t)^\top\vartheta_{J,l,s_J} - b^{-(t-T_0)}H_{J,l,s_J}(t)^\top\hat{\vartheta}_{J,l,s_J}\Big] \\ + \lambda\xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,,\end{aligned}$$

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

$$\begin{aligned}\bar{\epsilon}_{I,l}^{(s_I)}(t+1) = \sum_{J\in\mathcal{O}_s} W_s^{(I,J)} \Big[\lambda\bar{\epsilon}_{J,l}^{(s_J)}(t) + \bar{\chi}_J^{(s_J)}(t) + \|H_{J,l,s_J}(t)\|\kappa_{J,l,s_J}(\hat{\vartheta}_{J,l,s_J}) \\ +\bar{\Delta}H_{J,l,s_J}(t)\bar{\vartheta}_{J,l,s_J} - \bar{b}^{-(t-T_d)}\|H_{J,l,s_J}(t)\|\,\|\hat{\vartheta}_{J,l,s_J}\|\Big] \\ + \lambda\bar{\xi}_I^{(s_I)}(t) + \bar{\xi}_I^{(s_I)}(t+1)\,.\end{aligned}$$

The error and threshold solutions can be conveniently written by introducing the vectors

$$\epsilon_{s,l}(t) \triangleq \mathrm{col}(\epsilon_{I,l}^{(s_I)}, I\in\mathcal{O}_s),\ \chi_s(t) \triangleq \mathrm{col}(\chi_I^{(s_I)}, I\in\mathcal{O}_s),\ \bar{\epsilon}_{s,l}(t) \triangleq \mathrm{col}(\bar{\epsilon}_{I,l}^{(s_I)}, I\in\mathcal{O}_s)\,,$$

so that it holds

$$\begin{aligned}\epsilon_{s,l}(t+1) = W_s\Big[\lambda\epsilon_{s,l}(t) + \chi_s(t) + \mathrm{col}((1-b^{-(t-T_0)})H_{I,l,s_I}(t)^\top\tilde{\vartheta}_{I,l,s_I} \\ +(1-b^{-(t-T_0)})\Delta H_{I,l,s_I}(t)^\top\vartheta_{I,l,s_I} - b^{-(t-T_0)}H_{I,l,s_I}(t)^\top\hat{\vartheta}_{I,l,s_I}, I\in\mathcal{O}_s)\Big] \\ + \lambda\xi_s(t) + \xi_s(t+1)\,,\end{aligned}$$

$$\begin{aligned}\epsilon_{s,l}(t) = \sum_{h=T_d}^{t-1}(\lambda W_s)^{t-1-h}W_s\Big[\chi_s(h) + \mathrm{col}((1-b^{-(h-T_0)})H_{I,l,s_I}(h)^\top\tilde{\vartheta}_{I,l,s_I} \\ +(1-b^{-(h-T_0)})\Delta H_{I,l,s_I}(h)^\top\vartheta_{I,l,s_I} - b^{-(h-T_0)}H_{I,l,s_I}(h)^\top\hat{\vartheta}_{I,l,s_I}, I\in\mathcal{O}_s)\Big] \\ + \sum_{h=T_d}^{t-1}[(\lambda W_s)^{t-1-h}(\lambda\xi_s(h) + \xi_s(h+1))] + (\lambda W_s)^{t-T_d}\epsilon_{s,l}(T_d)\,,\end{aligned}$$

and component-wise it is

$$\epsilon_{I,l}^{(s_I)}(t) = w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \Big[\chi_s(h) + \mathrm{col}((1-b^{-(h-T_0)})H_{I,l,s_I}(h)^\top \tilde{\vartheta}_{I,l,s_I}$$
$$+(1-b^{-(h-T_0)})\Delta H_{I,l,s_I}(h)^\top \vartheta_{I,l,s_I} - b^{-(h-T_0)} H_{I,l,s_I}(h)^\top \hat{\vartheta}_{I,l,s_I}, I \in \mathcal{O}_s)\Big]$$
$$+ \lambda w_{s,I} \sum_{h=T_d}^{t-2} [(\lambda W_s)^{t-2-h}(\lambda \xi_s(h) + \xi_s(h+1))] + \lambda \xi_I^{(s_I)}(t-1) + \xi_I^{(s_I)}(t)$$
$$+ \lambda w_{s,I} (\lambda W_s)^{t-1-T_d} \epsilon_{s,l}(T_d)\,,$$

for the error solution and, analogously, it holds

$$\bar{\epsilon}_{I,l}^{(s_I)}(t) = w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \Big[\bar{\chi}_s(t) + \mathrm{col}(\|H_{I,l,s_I}(t)\| \kappa_{I,l,s_I}(\hat{\vartheta}_{I,l,s_I})$$
$$+\bar{\Delta} H_{I,l,s_I}(t)\bar{\vartheta}_{I,l,s_I} - \bar{b}^{-(t-T_d)} \|H_{I,l,s_I}(t)\|\, \|\hat{\vartheta}_{I,l,s_I}\|, I \in \mathcal{O}_s)\Big]$$
$$+ \lambda w_{s,I} \sum_{h=T_d}^{t-2} [(\lambda W_s)^{t-2-h}(\lambda \bar{\xi}_s(t) + \bar{\xi}_s(t+1))] + \lambda \bar{\xi}_I^{(s_I)}(t-1) + \bar{\xi}_I^{(s_I)}(t)$$
$$+ \lambda w_{s,I} (\lambda W_s)^{t-1-T_d} \bar{\epsilon}_{s,l}(T_d)\,,$$

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

Supposing a non matched fault instead, that is $\phi_I^{(s_I)}(x_I(t), z_I(t), u_I(t)) = \phi_{I,p}^{(s_I)}(x_I(t), z_I(t), u_I(t), \vartheta_{I,p})$ for some $I \in \mathcal{O}_s$ and with $p \neq l$, 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

$$\epsilon_{I,l}^{(s_I)}(t+1) = \sum_{J \in \mathcal{O}_s} W_s^{(I,J)} \Big[\lambda \epsilon_{J,l}^{(s_J)}(t) + \chi_J^{(s_J)}(t)$$
$$+(1-b^{-(t-T_0)})\phi_{I,p}^{(s_I)}(x_I(t), z_I(t), u_I(t), \vartheta_{I,p}) - \hat{\phi}_{J,l}^{(s_J)}(y_J(t), v_J(t), u_J(t), \hat{\vartheta}_{J,l})\Big]$$
$$+ \lambda \xi_I^{(s_I)}(t) + \xi_I^{(s_I)}(t+1)\,.$$

As shown before, a convenient way to study the behavior of the estimation error of the LFDs sharing the variable $x^{(s)}$ is to consider the vector $\epsilon_{s,l}$, whose dynamics are

$$\epsilon_{s,l}(t+1) = W_s\,[\lambda \epsilon_{s,l}(t) + \chi_s(t) + \Delta_{s,l}\phi_{I,p}(t)] + \lambda \xi_s(t) + \xi_s(t+1)\,,$$

where the following *mismatch vector* was introduced

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

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

$$\epsilon_{s,l}(t) = \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} W_s \left[\chi_s(h) + \Delta_{s,l}\phi_{I,p}(h)\right] + \sum_{h=T_d}^{t-1} \left[(\lambda W_s)^{t-1-h}(\lambda\xi_s(h) + \xi_s(h+1))\right] + (\lambda W_s)^{t-T_d}\epsilon_{s,l}(T_d)\,,$$

and component-wise it is

$$\epsilon_{I,l}^{(s_I)}(t) = w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \left[\chi_s(h) + \Delta_{s,l}\phi_{I,p}(h)\right] + \lambda w_{s,I} \sum_{h=T_d}^{t-2} \left[(\lambda W_s)^{t-2-h}(\lambda\xi_s(h) + \xi_s(h+1))\right] + \lambda\xi_I^{(s_I)}(t-1) + \xi_I^{(s_I)}(t) + \lambda w_{s,I}(\lambda W_s)^{t-1-T_d}\epsilon_{s,l}(T_d)\,.$$

Theorem 4.6.1 (Fault Isolability): Given a fault $\phi_{I,p} \in \mathcal{F}_I$, if for each $l \in \{1, \ldots, N_{\mathcal{F}_I}\} \setminus p$ there exists some time instant $T_l > T_d$ and some $s_I \in \{1, \ldots, n_I\}$ such that the following inequality holds

$$|w_{s,I} \sum_{h=T_d}^{T_l-1} (\lambda W_s)^{t-1-h}\Delta_{s,l}\phi_{I,p}(h)| > w_{s,I} \sum_{h=T_d}^{T_l-1} (\lambda W_s)^{t-1-h} \left[\bar{\chi}_s(h) + \mathrm{col}(\|H_{I,l,s_I}(t)\|\kappa_{I,l,s_I}(\hat{\vartheta}_{I,l,s_I}) + \bar{\Delta}H_{I,l,s_I}(t)\bar{\vartheta}_{I,l,s_I} - \bar{b}^{-(t-T_d)}\|H_{I,l,s_I}(t)\|\,\|\hat{\vartheta}_{I,l,s_I}\|, I \in \mathcal{O}_s)\right] + 2\left\{\lambda w_{s,I} \sum_{h=T_d}^{t-2} \left[(\lambda W_s)^{t-2-h}(\lambda\bar{\xi}_s(t) + \bar{\xi}_s(t+1))\right] + \lambda\bar{\xi}_I^{(s_I)}(t-1) + \bar{\xi}_I^{(s_I)}(t) + \lambda w_{s,I}(\lambda W_s)^{t-1-T_d}\bar{\epsilon}_{s,l}(T_d)\right\}\,,$$

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

Proof: By using the triangular inequality, the absolute value of the s_I–th component of the l–th FIE of the I–th LFD estimation error can be bounded for $t > T_d$ as

$$|\epsilon_{I,l}^{(s_I)}(t)| \geq |w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h}\Delta_{s,l}\phi_{I,p}(h)| - |w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h}\chi_s(h)| - |\lambda w_{s,I} \sum_{h=T_d}^{t-2} \left[(\lambda W_s)^{t-2-h}(\lambda\xi_s(h) + \xi_s(h+1))\right]| - |\lambda\xi_s^{(I)}(t-1)| - |\xi_s^{(I)}(t)| - |\lambda w_{s,I}(\lambda W_s)^{t-1-T_d}\epsilon_{s,l}(T_d)|\,,$$

and by using the known bounds on γ_s and ξ_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

$$|\epsilon_{I,l}^{(s_I)}(t)| \geq |w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \Delta_{s,l} \phi_{I,p}(h)|$$
$$- w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \bar{\chi}_s(h) - \lambda w_{s,I} \sum_{h=T_d}^{t-2} [(\lambda W_s)^{t-2-h} (\lambda \bar{\xi}_s(h) + \bar{\xi}_s(h+1))]$$
$$- \lambda \bar{\xi}_s^{(I)}(t-1)| - \bar{\xi}_s^{(I)}(t) - \lambda w_{s,I} (\lambda W_s)^{t-1-T_d} \bar{\epsilon}_{s,l}(T_d) \,.$$

In order for the l–th fault to be excluded it must hold $|\epsilon_{I,l}^{(s_I)}(t)| > \bar{\epsilon}_{I,l}^{(s_I)}(t)$, and this translates to the following

$$|w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \Delta_{s,l} \phi_{I,p}(h)| \geq \bar{\epsilon}_{I,l}(t) + w_{s,I} \sum_{h=T_d}^{t-1} (\lambda W_s)^{t-1-h} \bar{\chi}_s(h)$$
$$+ \lambda w_{s,I} \sum_{h=T_d}^{t-2} [(\lambda W_s)^{t-2-h} (\lambda \bar{\xi}_s(h) + \bar{\xi}_s(h+1))]$$
$$+ \lambda \bar{\xi}_s^{(I)}(t-1)| + \bar{\xi}_s^{(I)}(t) + \lambda w_{s,I} (\lambda W_s)^{t-1-T_d} \bar{\epsilon}_{s,l}(T_d) \,,$$

which is implied by the inequality in the theorem hypothesis. Should the inequality hold for every fault function of $\mathcal{F}_I$ but the p–th, then this fault will be isolated in the sense of Definition 4.5.2. ■

4.7 Illustrative example

In this example, depicted in Fig. 4.1, an eleven-tank system is monitored by three LFDs, according to the decomposition $\mathcal{D} = \{\mathscr{S}_1, \mathscr{S}_2, \mathscr{S}_3\}$, with index sets $\mathcal{I}_1 = [1\ 2\ 3\ 4\ 5]^\top$, $\mathcal{I}_2 = [4\ 5\ 6\ 7]^\top$ and $\mathcal{I}_3 = [5\ 8\ 9\ 10\ 11]^\top$. Three pumps are present, feeding the first, seventh and eleventh tank with the following 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 nominal tank sections are set according to the following vector $A = [\,1\ 0.5\ 1\ 1\ 2\ 2\ 1\ 1\ 0.5\ 0.5\ 0.5\,]$ m^2, while the interconnecting pipe cross-sections are nominally equal to $A_p = [\,0.2\ 0.22\ 0.38\ 0.2\ 0.16\ 0.18\ 0.24\ 0.2\ 0.18\ 0.14\ 0.42\ 0.2\,]$ m^2. Furthermore, to each tank are connected drain pipes whose nominal cross-section are $A_d = [\,0.025\ 0.0125\ 0.0225\ 0.0275\ 0.075\ 0.0375\ 0.025\ 0.03\ 0.01\ 0.0125\ 0.015\,]$ m^2. All the pipes outflow coefficients are unitary. When building the local models f_I of each LFD, anyway, the actual cross-sections used are affected by random uncertainties no larger than 5% and 8% of the nominal values, respectively for the tanks and for the pipes. The outflow coefficients are off

by no more than 10%. Furthermore the tank levels measurements y_I are affected by measuring uncertainties ξ_I whose components 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.

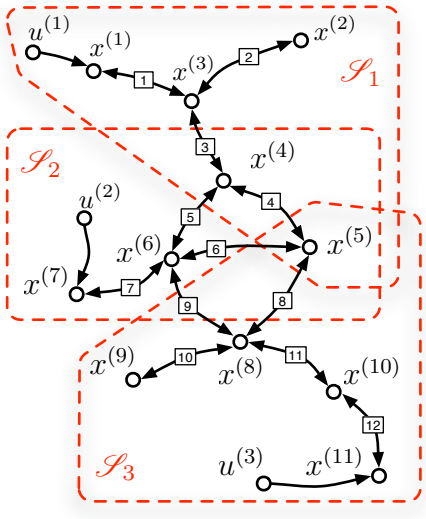


Figure 4.1: Structure of the eleven–tanks system under consideration. The square labels refer to the pipes number.

The adaptive approximators $\hat{g}_I$ of each LFD are implemented by RBF neural networks having 3 neurons along the range of each input dimension. The parameter domains Θ_I were chosen to be hyper-spheres with radii equal to $\begin{bmatrix} 2 & 3 & 2 \end{bmatrix} \cdot T_s$, $T_s = 0.1$ s being the sampling period. The learning rate auxiliary coefficients for the interconnection adaptive approximators were set to $\mu_{1,0} = 10^{-4}$, $\varepsilon_{1,0} = 10^{-3}$, $\mu_{2,0} = 0.5 \cdot 10^{-4}$, $\varepsilon_{2,0} = 10^{-3}$, $\mu_{3,0} = 0.5 \cdot 10^{-4}$, $\varepsilon_{3,0} = 10^{-3}$, while the filter constants were all set to $\lambda = 0.9$, and the total uncertainties were bounded by $\bar{\chi}_1 = \begin{bmatrix} 0.36 & 0.42 & 0.42 & 0.6 & 0.6 \end{bmatrix} \cdot T_s$, $\bar{\chi}_2 = \begin{bmatrix} 0.36 & 0.48 & 0.42 & 0.3 \end{bmatrix} \cdot T_s$, $\bar{\chi}_3 = \begin{bmatrix} 0.6 & 0.6 & 0.42 & 0.72 & 0.54 \end{bmatrix} \cdot T_s$. The weighting matrices for shared variables were $W_4 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$ and $W_5 = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$.

Three faults were modeled:

1. **Actuator fault in pump 1, 2 and 3:** partial or full shutdown of all the pumps modeled as $u_f^{(i)} = u^{(i)}(1 - a^{(i)})$, where u_f represents the pumps flow in the faulty case and $0 \leq a^{(i)} \leq 1$, $i \in \{1, 2, 3\}$.

2. **Leakage in tank 4, 5 and 6:** 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 $q_f^{(i)} = \pi(\rho^{(i)})^2\sqrt{2gx^{(i)}(t)}$, $i \in \{4,5,6\}$.
3. **Breakdown of pipes 3 (tanks 3↔4) and 5 (tanks 4↔6):** partial or complete breakdown of those pipes, so that a relative quota $0 \leq a_p^{(i)} \leq 1$, $i \in \{3,5\}$ of the water in the pipes is drained out of the tanks instead of flowing between them. This is equivalent to substituting the two pipes with four additional drain pipes, one connected to tank 3, two to tank 4 and one to tank 6.

All these cases represent distributed faults with distributed signatures, and the second and third ones feature overlapping signatures too. As can be easily seen, the local fault diagnosers experience the following local signatures:

- LFD no. 1 sees as local only the breakdown of pump 1, or the leakage in tanks 4 and 5, or the effect on tanks 3 and 4 of the breakdown of pipe 3;
- LFD no. 2 sees as local only the breakdown of pump 2, or the leakage in tanks 4, 5 and 6, or the effect on tanks 4 and 6 of the breakdown of pipe 5;
- LFD no. 3 sees as local only the breakdown of pump 3, or the leakage in tank 5.

The resulting fault classes $\mathcal{F}_I$ are then:

$$\mathcal{F}_1 = \left\{ \begin{bmatrix} \vartheta_{1,1,1}H_{1,1,1}(t) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vartheta_{1,2,4}H_{1,2,4}(t) \\ \vartheta_{1,2,5}H_{1,2,5}(t) \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \vartheta_{1,3,3}H_{1,3,3}(t) \\ \vartheta_{1,3,4}H_{1,3,4}(t) \\ 0 \end{bmatrix} \right\},$$

where $\vartheta_{1,1,1} = a^{(1)}$, $H_{1,1,1}(t) = -\frac{T_s}{A^{(1)}}u_1^{(1)}(t)$, $\vartheta_{1,2,4} = \pi(\rho^{(4)})$, $H_{1,2,4}(t) = -\frac{T_s}{A^{(4)}}\sqrt{2gx_1^{(4)}(t)}$, $\vartheta_{1,2,5} = \pi(\rho^{(5)})$, $H_{1,2,5}(t) = -\frac{T_s}{A^{(5)}}\sqrt{2gx_1^{(5)}(t)}$, $\vartheta_{1,3,3} = a_p^{(3)}$, $H_{1,3,3}(t) = -\frac{T_s}{A^{(3)}}a_p^{(3)}c_p^{(3)}A_p^{(3)}\cdot(\text{sign}(x_1^{(4)}(t)-x_1^{(3)}(t))\cdot\sqrt{2g|x_1^{(4)}(t) - x_1^{(3)}(t)|}+\sqrt{2gx_1^{(3)}(t)}$, $\vartheta_{1,3,4} = a_p^{(3)}$, $H_{1,3,4}(t) = -\frac{T_s}{A^{(4)}}a_p^{(3)}c_p^{(3)}A_p^{(3)}\cdot(\text{sign}(x_1^{(3)}(t)-x_1^{(4)}(t))\cdot\sqrt{2g|x_1^{(3)}(t) - x_1^{(4)}(t)|} + \sqrt{2gx_1^{(4)}(t)}$;

$$\mathcal{F}_2 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vartheta_{2,1,4}H_{2,1,4}(t) \end{bmatrix}, \begin{bmatrix} \vartheta_{2,2,1}H_{2,2,1}(t) \\ \vartheta_{2,2,2}H_{2,2,2}(t) \\ \vartheta_{2,2,3}H_{2,2,3}(t) \\ 0 \end{bmatrix}, \begin{bmatrix} \vartheta_{2,3,1}H_{2,3,1}(t) \\ 0 \\ \vartheta_{2,3,3}H_{2,3,3}(t) \\ 0 \end{bmatrix} \right\},$$

where $\vartheta_{2,1,4} = a^{(2)}$, $H_{2,1,4}(t) = -\frac{T_s}{A^{(7)}} u_2^{(1)}(t)$, $\vartheta_{2,2,1} = \pi(\rho^{(4)})$, $H_{2,2,1}(t) = -\frac{T_s}{A^{(4)}} \sqrt{2g x_2^{(1)}(t)}$, $\vartheta_{2,2,2} = \pi(\rho^{(5)})$, $H_{2,2,2}(t) = -\frac{T_s}{A^{(5)}} \sqrt{2g x_2^{(2)}(t)}$, $\vartheta_{2,2,3} = \pi(\rho^{(6)})$, $H_{2,2,3}(t) = -\frac{T_s}{A^{(6)}} \sqrt{2g x_2^{(3)}(t)}$, $\vartheta_{2,3,1} = a_p^{(5)}$, $H_{2,3,1}(t) = -\frac{T_s}{A^{(4)}} a_p^{(5)} c_p^{(5)} A_p^{(5)} \cdot (\text{sign}(x_2^{(3)}(t) - x_2^{(1)}(t)) \cdot \sqrt{2g|x_2^{(3)}(t) - x_2^{(1)}(t)|} + \sqrt{2g x_2^{(1)}(t)}$, $\vartheta_{2,3,3} = a_p^{(5)}$, $H_{2,3,3}(t) = -\frac{T_s}{A^{(6)}} a_p^{(5)} c_p^{(5)} A_p^{(5)} \cdot (\text{sign}(x_2^{(1)}(t) - x_2^{(3)}(t)) \cdot \sqrt{2g|x_2^{(1)}(t) - x_2^{(3)}(t)|} + \sqrt{2g x_2^{(3)}(t)}$;

$$\mathcal{F}_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vartheta_{3,1,5} H_{3,1,5}(t) \end{bmatrix}, \begin{bmatrix} \vartheta_{3,2,1} H_{3,2,1}(t) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\},$$

where $\vartheta_{3,1,5} = a^{(3)}$, $H_{3,1,5}(t) = -\frac{T_s}{A^{(11)}} u_3^{(1)}(t)$, $\vartheta_{3,2,1} = \pi(\rho^{(5)})$, $H_{3,2,1}(t) = -\frac{T_s}{A^{(5)}} \sqrt{2g x_3^{(1)}(t)}$.

In the present example a fault of the first kind, that is a pumps failure, will be modeled. At $T_0 = 750$ s an incipient fault with time constant $b = 1.05$ will start to affect the three pumps, reducing their effectiveness, respectively, by 25%, 35% and 20%. It must be stressed that this is considered as a single fault event leading to a distributed fault signature, and not as three local fault events happening at the same time. As the fault is distributed, the three LFDs will need to exchange their fault decisions in order to reach a correct diagnosis.

Figs. 4.2–4.4 shows the simulated behavior of the residuals and thresholds computed by the proposed DFDI architecture. As can be seen, the fault is detected at $T_d = 751$ s by the second LFD, that at $T_{lis,2,1} = 752$ s locally isolates the first fault. Later, at $T_{lis,1,1} = 758$ s the first LFD isolates the same fault, that is finally globally isolated at $T_{is,1} = 824$ s thanks to the third LFD. By looking at Figs. 4.2(e), 4.3(e) and 4.4(d) the good performances of the DFDI scheme in estimating the fault parameters can be appreciated: both the first and the second LFDs are very close to the true values of $\vartheta_{1,1,1} = 0.25$ and $\vartheta_{2,1,4} = 0.35$, while the third one is off by almost 20% with respect to $\vartheta_{3,1,5} = 0.20$. Anyway it must be acknowledged, similarly to what was pointed out in Chapter 2, that the FIEs approximators are actually learning the effect of the fault functions, of the uncertainties and of the interconnection approximator MFAE, rather than that of the fault function alone.

4.8 Concluding remarks

In this chapter, a novel distributed architecture for the FDI of nonlinear and uncertain discrete–time systems has been proposed. The architecture is based on an overlapping decomposition of an original monolithic system, and each resulting subsystem is assigned to a single Local Fault Diagnoser that monitors its health.

This contribution fills a gap in the present literature, where mainly schemes for discrete–event systems, for linear discrete–time or continuous time, or for systems described by qualitative non–linear models, were considered. Two notable features have been embedded in the proposed architecture: the capability of learning on–line the model uncertainties by the use of adaptive approximators, and the use of consensus techniques for computing possibly better estimates of the parts of the model shared by more than one Local Fault Diagnoser.

In order to characterize the performance of the DFDI architecture, classical analytical results for centralized FDI schemes were derived for the present case, namely the robustness of adaptive thresholds to model uncertainties, the fault detectability condition and the fault isolability conditions. Finally, a simulation example has been provided to show the effectiveness of the DFDI scheme.

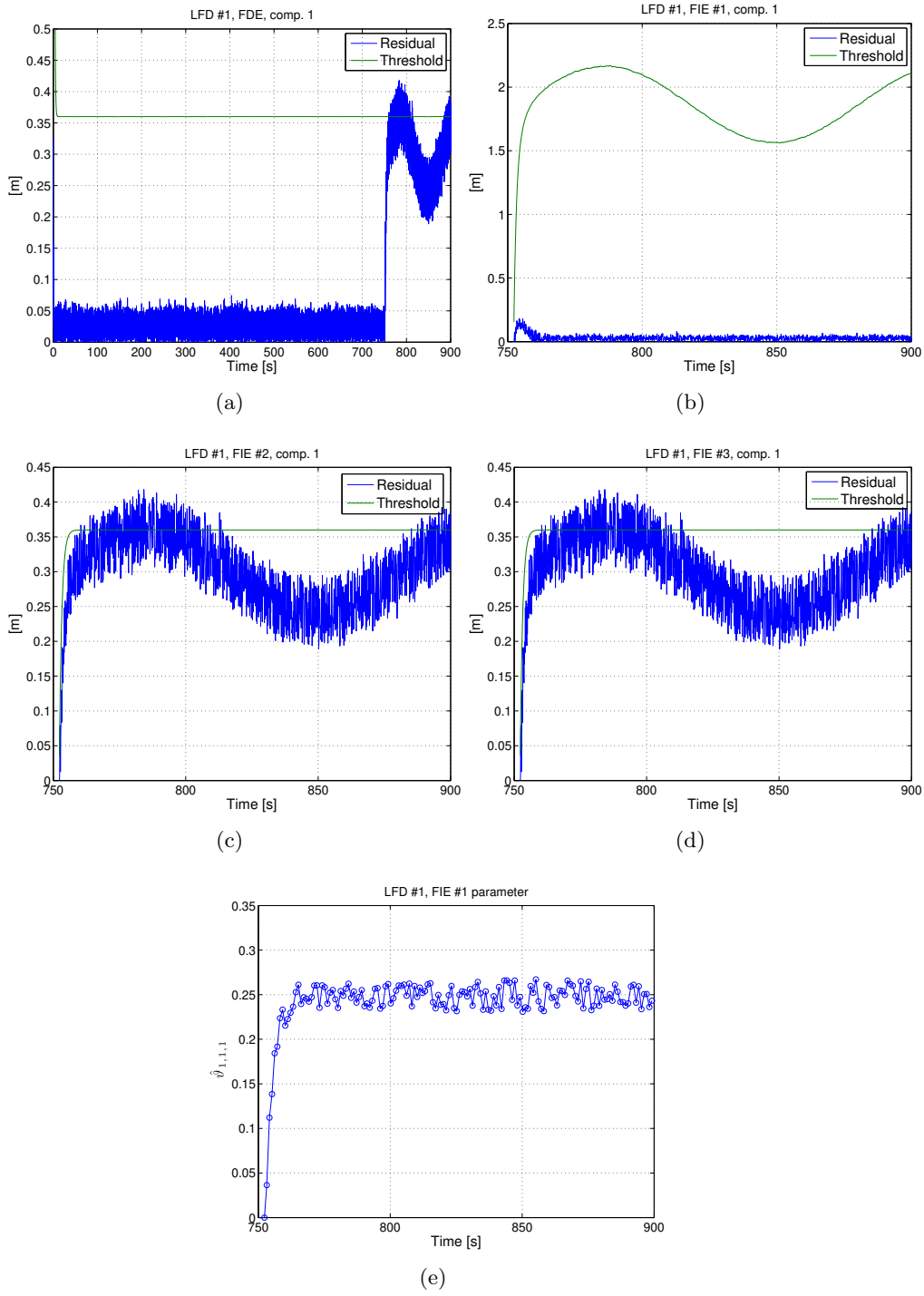
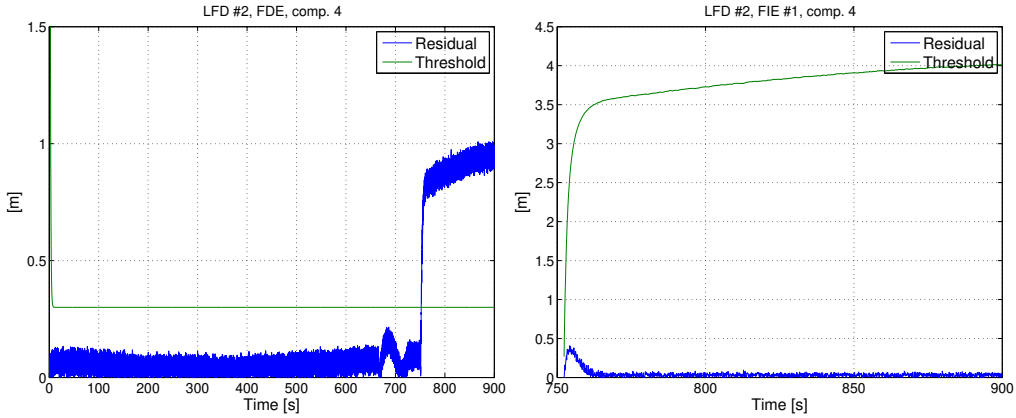
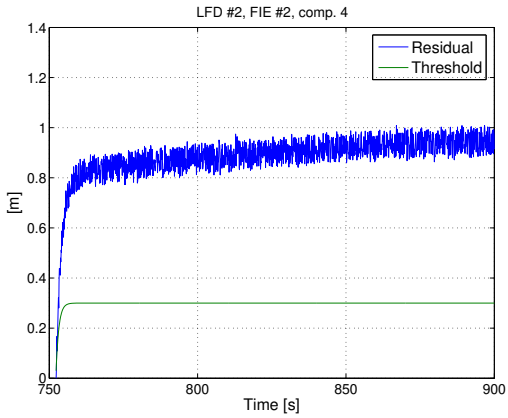


Figure 4.2: Time–behaviors of simulated signals related to tanks no. 1 when a leakage is introduced at time 750 s. Data for the estimated fault parameter have been decimated.

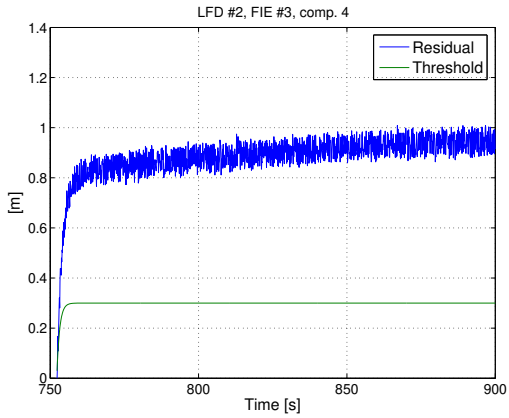


(a)

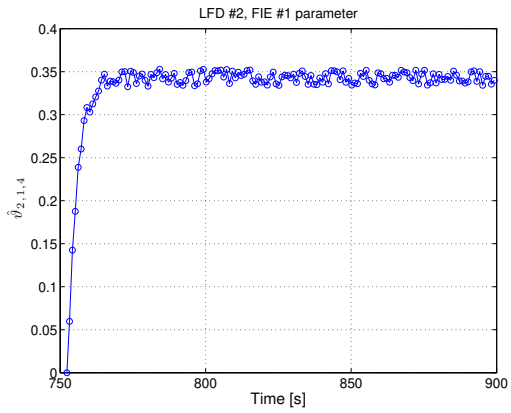
(b)



(c)



(d)



(e)

Figure 4.3: Time–behaviors of simulated signals related to tanks no. 7 when a leakage is introduced at time 750 s. Data for the estimated fault parameter have been decimated.

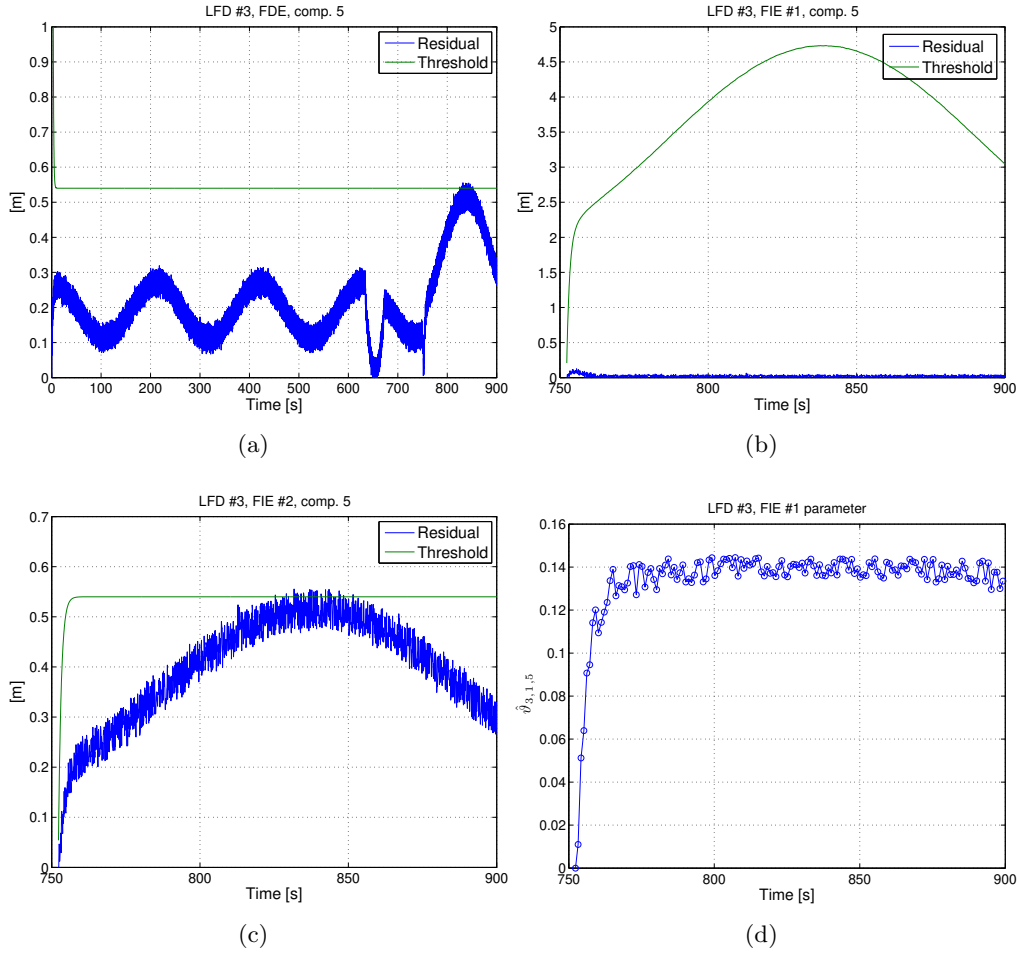


Figure 4.4: Time–behaviors of simulated signals related to tanks no. 11 when a leakage is introduced at time 750 s. Data for the estimated fault parameter have been decimated.

Chapter 5

Distributed FDI, specialization for continuous time systems

In this chapter it will be showed how the DFDI architecture developed in Chapter 4 for discrete-time systems, can be specialized for the continuous–time ones. In order to streamline the investigation, the following simplifications will be assumed: only the fault detection service will be provided, and only abrupt faults will be considered. Furthermore the consensus protocol on shared variable estimates will weigh equally the contributions from all the LFDs in the overlap set.

5.1 Background and assumptions

Let us consider a generic large–scale nonlinear system $\mathscr{S}$ described as (see [30])

$$\mathscr{S}: \dot{x} = f(x,u) + \beta(t-T_0)\phi(x,u)\,, \tag{5.1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the state and input vectors, respectively, and $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ represents the nominal healthy dynamics. As in Chapter 2, the term $\beta(t-T_0)\phi(x,u)$ denotes the changes in the system dynamics due to the occurrence of a fault. More specifically, the vector $\phi(x,u)$ represents the functional structure of the deviation in the state equation due to the fault and the function $\beta(t-T_0)$ characterizes the time profile of the fault, where T_0 is the unknown fault occurrence time. In this chapter, for the sake of simplicity we only consider the case of *abrupt* (sudden) faults and, accordingly, $\beta(\cdot)$ takes on the form of a step function, i.e., $\beta(t-T_0) = 0,\ \text{if } t < T_0\ \text{ and }\ \beta(t-T_0) = 1,\ \text{if } t \geq T_0\,.$

As discussed in Chapter 3, the model in (5.1) may be impractical for fault detection (FD), either because of its size, or because the system it

represents is physically distributed, so that a centralized FDI architecture is neither possible nor desirable. This problem will be overcome by considering $\mathscr{S}$ as decomposed into N subsystems $\mathscr{S}_I$, each characterized by a *local* state vector $x_I \in \mathbb{R}^{n_I}$, so that each $\mathscr{S}_I$ will be separately monitored. To this end, the dynamics of $\mathscr{S}_I$ can be modeled as

$$\mathscr{S}_I : \dot{x}_I = f_I^*(x,u) + \beta(t-T_0)\phi_{iI}(x,u)\,,$$

where the vectors f_I^* and ϕ_I are built upon the components of f and ϕ that account for the dynamics of subsystem $\mathscr{S}_I$. Again, following the decomposition procedures introduced in Chapter 3, f_I^* will be conveniently split into two parts:

$$\mathscr{S}_I : \dot{x}_I = f_I(x_I,u_I) + g_I(x_I,z_I,u_I) + \beta(t-T_0)\phi_I(x,u) \qquad (5.2)$$

with $f_I : \mathbb{R}^{n_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ being the *local nominal* function, $g_I : \mathbb{R}^{n_I} \times \mathbb{R}^{\bar{n}_I} \times \mathbb{R}^{m_I} \mapsto \mathbb{R}^{n_I}$ the *interconnection function*, $u_I \in \mathbb{R}^{m_I}$, $(m_I \leq m)$, the *local input*, and $z_I \in \mathbb{R}^{p_I}$, $(p_I \leq n - n_I)$, the vector of *interconnection state variables*. The decomposition $\mathcal{D} \triangleq \{\mathscr{S}_1, ..., \mathscr{S}_N\}$ will satisfy Definition 3.3.3 and will allow for overlaps between subsystems.
The following further assumptions are now needed, analogously to what has been assumed in Chapter 2.

Assumption 5.1.1: 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 $(x_I(t), u_I(t)) \in \mathcal{R}_I^x \times \mathcal{R}_I^u$, $\forall\, I = 1, \ldots, N, \forall\, t \geq 0$. □

Clearly, as a consequence of Assumption 5.1.1, 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. The first reason for introducing such a boundedness assumption is a formal one in order to make the problem of detecting faults well-posed. Moreover, from an application point of view, Assumption 5.1.1 does not turn out to be very restrictive as the difficult issue generally is the early detection of faults characterized by a relatively small magnitude. Indeed, since no fault accommodation is considered in this work, the feedback controller acting on the system $\mathscr{S}$ must be such that the measurable signals $x(t)$ and $u(t)$ remain bounded for all $t \geq 0$. However, it is important to note that the proposed *Distributed Fault Detection and Identification* (DFDI) design is not dependent on the structure of the controller.

Assumption 5.1.2: The decomposition $\mathcal{D}$ is given a priori and is such that, for each $\mathscr{S}_I$, the local nominal function f_I is perfectly known, whereas the interconnection term g_I is an uncertain function of x_I, z_I and u_I. For each $k = 1, \ldots, n_I$, the k–th component of g_I is bounded by some known functional, i.e.,

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

where the bounding function $\bar{g}_I^{(k)}(x_I, z_I, u_I) \geq 0$ is known, integrable, and bounded for all (x_I, z_I, u_I) in some compact region of interest $\bar{\mathcal{R}} \supseteq \mathcal{R}_I^x \times \mathcal{R}_I^z \times \mathcal{R}_I^u$. □

As explained in the previous chapters, Assumption 5.1.2 captures situations where each $\mathscr{S}_I$ corresponds to a known physical subsystem or a component, interacting through uncertain physical links as part of a complex large–scale system or to attain a higher goal (several application contexts can be found where such modeling approach turns out to be useful – see, for example, [51]). This uncertainty will be overcome in the following sections by employing an adaptive approximator $\hat{g}_I$ in lieu of g_I.

Remark 5.1.1: It is worth noting that the task of determining non-conservative bounding functions $\bar{g}_I^{(k)}(x_I, z_I, u_I) \geq 0$ may turn out to be rather difficult in practice, and has to be carried out by exploiting prior knowledge by plant technicians and extensive off-line simulation trials. □

5.2 Fault Detection Architecture

The simplified DFDI architecture will be based on the template portrayed in Section 3.4. It will be based on N agents called *Local Fault Diagnosers* (LFD), each monitoring a subsystem $\mathscr{S}_I$ originating from the decomposition $\mathcal{D}$ of the monolithic system $\mathscr{S}$. Each LFD will implement an estimator called *Fault Detection and Approximation Estimator* (FDAE) for providing the detection service. The FDAE will be based on a nonlinear adaptive estimator, and will compute an estimate of the local state x_I by directly measuring local variables and by exchanging with other LFDs the values of the interconnection and of the shared variables. The details of this estimator will now be given.

5.3 Healthy behavior and Fault Detection Estimator

The local fault detection algorithm is based on a nonlinear adaptive estimator built on the subsystem model (5.2), and for the I–th LFD it takes on the form

$$\dot{\hat{x}}_I^{(s_I)} = -\lambda\Bigg[\sum_{J\in\mathcal{O}_s}(\hat{x}_I^{(s_I)} - \hat{x}_J^{(s_J)}) + d_s(\hat{x}_I^{(s_I)} - x_I^{(s_I)})\Bigg] + \frac{1}{d_s}\sum_{J\in\mathcal{O}_s}[f_J^{(s_J)}(x_J, u_J) + \hat{g}_J^{(s_J)}(x_J, z_J, u_J, \hat{\vartheta}_J)] \quad (5.3)$$

for each $s_I = 1, \ldots, n_I$, where

- $\hat{x}_I^{(s_I)}$ denotes the estimate of the local state component $x_I^{(s_I)}$

- $x_I^{(s_I)}$ corresponds to the s–th component of the global state vector, that is $x_I^{(s_I)} \equiv x^{(s)}$
- $\mathcal{O}_s$ is the index set of the d_s LFDs sharing the variable $x^{(s)}$
- $\hat{g}_J(\cdot)$ is an adaptive approximator to be described later
- $\hat{\vartheta}_J$ is the vector of the adaptive approximator parameters
- $-\lambda < 0$ represents the value of the estimator poles.

As the entire state x_I is assumed to be measurable, it must be stressed that the estimate $\hat{x}_I^{(s_I)}$ is not used for estimation, but will be employed in the fault detection process for residual error generation and for adaptive approximation. A *consensus* mechanism is embedded in the estimator for shared components of the local state of $\mathscr{S}_I$, allowing LFDs in $\mathcal{O}_s$ to *share* their knowledge about the local and the approximated interconnection part of the model. In particular the consensus is attained by the summations over $\mathcal{O}_s$, where the estimates and the models of each LFD are weighted by a constant value.

It is worth noting that, in order to implement (5.3), the LFD $\mathscr{L}_I$ does not need the information about the expressions of $f_J^{(s_J)}$ and of $\hat{g}_J^{(s_J)}$. Instead, it suffices that $\mathscr{L}_J$, $J \in \mathcal{O}_s$, computes the term $f_J^{(s_J)} + \hat{g}_J^{(s_J)}$ and communicates it to other LFDs in $\mathcal{O}_s$ alongside its actual state estimate $\hat{x}_J^{(s_J)}$. Furthermore, each $\mathscr{L}_J$, with $J \in \mathcal{J}_I$, must communicate to $\mathscr{L}_I$ its values of the local state components needed to populate the interconnection state vector z_I.

Clearly, for non–shared state components the overlap index set is a singleton and (5.3) simplifies to an estimator without consensus as follows:

$$\dot{\hat{x}}_I^{(k)} = -\lambda(\hat{x}_I^{(k)} - x_I^{(k)}) + f_I^{(k)}(x_I, u_I) + \hat{g}_I^{(k)}(x_I, z_I, u_I, \hat{\vartheta}_I)\,.$$

Since it is assumed that, for each $\mathscr{S}_I$, the interconnection function g_I is uncertain (or unknown), a key point in the proposed scheme is that each LFD will adaptively learn the uncertain function g_I using a linearly parameterized adaptive approximator $\hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I) : \mathbb{R}^{n_I} \times \mathbb{R}^{p_I} \times \mathbb{R}^{m_I} \times \mathbb{R}^{q_I} \mapsto \mathbb{R}^{n_I}$ of the form

$$\hat{g}_I^{(k)}(x_I, z_I, u_I, \hat{\vartheta}_I) = \sum_{l=1}^{r_I} c_{I,k}^{(l)} \varphi_{I,l}(x_I, z_I, u_I)\,,$$

where $\varphi_{I,l}(\cdot)$ are given parameterized basis functions, $c_{I,k} \in \mathbb{R}^{r_I}$ are the parameters to be determined, i.e., $\hat{\vartheta}_I \in \mathbb{R}^{q_I}$, $\hat{\vartheta}_I \triangleq \mathrm{col}\,(c_{I,k}\colon\ k = 1, \ldots, n_I)$. Here the term adaptive approximator [114] may represent any linear–in–the–parameters, but otherwise nonlinear multi-variable approximation model, such as neural networks, fuzzy logic networks, polynomials, spline functions, wavelet networks, etc. By introducing the gradient matrix $H_I \triangleq$

$\partial \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I)/\partial \hat{\vartheta}_I$ with respect to the adjustable parameter vector [33], the approximator output can be written as $\hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I) = H_I \hat{\vartheta}_I$.

Using adaptive parameter estimation techniques, the learning law for the parameter vector takes on the form:

$$\dot{\hat{\vartheta}}_I \triangleq \mathcal{P}\left(\Gamma_I H_I^\top \epsilon_I\right) ,$$

where $\mathcal{P}$ is a *projection operator* [30] that restricts $\hat{\vartheta}_I$ to a pre–defined compact and convex set $\Theta_I \subset \mathbb{R}^{q_I}$, $\Gamma_I \in \mathbb{R}^{q_I \times q_I}$ is a symmetric and positive definite learning rate matrix and $\epsilon_I(t) \triangleq x_I(t) - \hat{x}_I(t)$ is the estimation error, which plays a double role: it provides a measure of the residual error for fault detection purposes and it also provides the error measure used for adaptively learning the unknown interconnection term g_I.

In general, the approximated interconnection term $\hat{g}_I$ cannot be expected to perfectly match the true term g_I. This can be formalized by introducing an *optimal weight vector* $\hat{\vartheta}_I^*$ within the compact convex set Θ_I [33]:

$$\hat{\vartheta}_I^* \triangleq \arg \min_{\hat{\vartheta}_I \in \Theta_I} \max_{\mathcal{R}} \|g_I(x_I, z_I, u_I) - \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I)\| \tag{5.4}$$

and the corresponding *minimum functional approximation error* (MFAE)

$$\nu_I(t) \triangleq g_I(x_I(t), z_I(t), u_I(t)) - \hat{g}_I(x_I(t), z_I(t), u_I(t), \hat{\vartheta}_I^*) . \tag{5.5}$$

By introducing the *parameter estimation error* $\tilde{\vartheta}_I \triangleq \hat{\vartheta}_I - \hat{\vartheta}_I^*$, the dynamics of the generic estimation error component for $t < T_0$ can be written as:

$$\dot{\epsilon}_I^{(s_I)} = \frac{1}{d_s} \sum_{J \in \mathcal{O}_s} (-h_{J,s_J}^\top \tilde{\vartheta}_J + \nu_J^{(s_J)}) - \lambda \left[\sum_{J \in \mathcal{O}_s} (\epsilon_I^{(s_I)} - \epsilon_J^{(s_J)}) + d_s \epsilon_I^{(s_I)} \right] ,$$

where the notation $h_{J,s_J}^\top$ stands for the s_J–th row of the gradient matrix H_J. The solution of the above equation can be written as

$$\epsilon_I^{(s_I)}(t) = \frac{1}{d_s} \left[\int_0^t \mathrm{e}^{-\lambda d_s (t-\tau)} \sum_{J \in \mathcal{O}_s} (-h_{J,s_J}^\top \tilde{\vartheta}_J + \nu_J^{(s_J)}) \mathrm{d}\tau \right.$$
$$\left. + \mathrm{e}^{-2\lambda d_s t} \sum_{J \in \mathcal{O}_s} (\mathrm{e}^{\lambda d_s t} - 1 + d_s \delta_{IJ}) \epsilon_J^{(s_J)}(0) \right] ,$$

where δ_{IJ} is the Kronecker delta function defined as $\delta_{IJ} = 1$ if $I = J$ and $\delta_{IJ} = 0$ otherwise. Again, this expression can be simplified in the case of a non–shared state component as follows:

$$\epsilon_I^{(k)}(t) = \int_0^t \mathrm{e}^{-\lambda(t-\tau)} (-h_{I,k}^\top \tilde{\vartheta}_k + \nu_I^{(k)}) \mathrm{d}\tau + \mathrm{e}^{-\lambda t} \epsilon_I^{(k)}(0) .$$

This solution shows that, because of the parameter estimation error $\tilde{\vartheta}_I$ and of the MFAE, the estimation error will be nonzero even in the absence of a fault. By applying the triangle inequality, it can be shown that the absolute value of ϵ_I can be upper-bounded as follows:

$$|\epsilon_I^{(s_I)}(t)| \leq \bar{\epsilon}_I^{(s_I)}(t) \triangleq \frac{1}{d_s}\left[\int_0^t \mathrm{e}^{-\lambda d_s(t-\tau)} \sum_{J\in\mathcal{O}_s} (\kappa_J(\tau)\|h_{J,s_J}^{\top}\| + \bar{\nu}_J^{(s_J)})\mathrm{d}\tau + \mathrm{e}^{-2\lambda d_s t} \sum_{J\in\mathcal{O}_s} (\mathrm{e}^{\lambda d_s t} - 1 + d_s\delta_{IJ})|\epsilon_J^{(s_J)}(0)|\right], \quad (5.6)$$

where $\kappa_I(t) \geq \|\tilde{\vartheta}_I\|$ depends on the geometric properties of the set Θ_I. For instance, letting the parameter set Θ_I be a hyper-sphere centered in the origin and with radius equal to M_I, we have $\kappa_I(t) \triangleq M_I + \|\hat{\vartheta}_I\|$. Moreover

$$|\nu_I^{(k)}(t)| \leq \bar{\nu}_I^{(k)}(t)\,,$$

where

$$\bar{\nu}_I^{(k)}(t) \triangleq \bar{g}_I^{(k)}\left(x_I(t), z_I(t), u_I(t)\right) + K_{\hat{g}}\|\mathrm{col}\,\left(x_I(t), z_I(t), u_I(t)\right)\|\,,$$

and $K_{\hat{g}}$ denotes the Lipschitz constant of the adaptive approximator on the compact set $\bar{\mathcal{R}}$ introduced in Assumption 5.1.2. The bound described by (5.6) represents an adaptive threshold on the state estimation error that can be easily implemented by linear filtering techniques [36]. The bound $\bar{\epsilon}_I(t)$ will be exploited in the next section in the fault detection context.

The fault detection logic for the generic LFD is exactly equivalent to the centralized one presented in section 2.3. The following condition

$$|\epsilon_I^{(k)}(t)| \leq \bar{\epsilon}_I^{(k)}(t)\ \forall\, k = 1, \ldots, n \quad (5.7)$$

will be associated to the *fault hypothesis*

$$\mathscr{H}_I : \text{"The subsystem } \mathscr{S}_I \text{ is healthy"}\,.$$

Should condition (2.4) be unmet at some time instant t, the hypothesis $\mathscr{H}_I$ will be falsified and a *local fault signature* will be noticed on subsystem $\mathscr{S}_I$

Definition 5.3.1: The *local fault signature* shown by the subsystem $\mathscr{S}_I$ at time $t > 0$ is the index set $\mathcal{S}_I \triangleq \{k : \exists\, t_1,\, t \geq t_1 > 0,\, |\epsilon_I^{(k)}(t_1)| > \bar{\epsilon}_I^{(k)}(t_1)\}$ of the state components for which the hypothesis (5.7) did not hold for at least one time instant. □

Fault Detection Logic The fault detection logic can again be simply stated in terms of the signature $\mathcal{S}_I$: a fault affecting the subsystem $\mathscr{S}_I$ will be detected at the first time instant such that $\mathcal{S}_I$ becomes non-empty. The difference with the centralized case is that the *fault detection time* T_d will be defined as the first time instant at which at least one subsystem has been detected to be faulty.

Definition 5.3.2: The *fault detection time* T_d is defined as $T_d \triangleq \min\{t : \exists\, I,\, I \in \{1,\dots,N\},\, \exists\, k,\, k \in \{1,\dots,n_I\}\, :,\, |\epsilon_I^{(k)}(t)| > \bar{\epsilon}_I^{(k)}(t)\}$. □

5.4 Faulty behavior and Fault Detectability

It must be acknowledged that the adaptive threshold $\bar{\epsilon}_I(t)$ defined by (5.6) is designed to avoid false positives, since a certain level of estimation error is always present due to the uncertainty in the learning process. Of course, this may lead to certain faults being undetectable if they cause an estimation error small enough to be indistinguishable from the estimation error due to the uncertainty. This intuitive point will be formalized in Theorem 5.4.1. First, analogously to (5.4) and (5.5), the following quantities are defined

$$\hat{\vartheta}_I^{*f} \triangleq \arg \min_{\hat{\vartheta}_I \in \Theta_I} \max_{\bar{\mathcal{R}}} \|g_I\left(x_I, z_I, u_I\right) + \phi_I\left(x, u\right) - \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I)\|$$

$$\nu_I^f(t) \triangleq g_I\left(x_I, z_I, u_I\right) + \phi_I\left(x, u\right) - \hat{g}_I(x_I, z_I, u_I, \hat{\vartheta}_I^{*f})$$

as well as the *mismatch function*

$$\Delta\phi_I^{(k)}(t) \triangleq h_{I,k}^{\top}\left(\hat{\vartheta}_I^{*f} - \hat{\vartheta}_I^{*}\right)^{(k)} + \nu_I^{f(k)} - \nu_I^{(k)}\,.$$

Now, we can state the following result

Theorem 5.4.1 (Fault Detectability): Given a variable $x^{(s)}$ with an overlap set $\mathcal{O}_s$, suppose that for some time–interval $[t_1, t_2]$ the corresponding components of the mismatch functions $\Delta\phi_J(t)$, $J \in \mathcal{O}_s$, fulfill the following inequality for at least the I–th LFD, $I \in \mathcal{O}_s$:

$$\begin{aligned}\int_{t_1}^{t_2} \mathrm{e}^{-\lambda d_s(t-\tau)} \sum_{J\in\mathcal{O}_s} |\Delta\phi_J^{(s_J)}|\mathrm{d}\tau \geq 2\int_0^{t_2} \mathrm{e}^{-\lambda d_s(t-\tau)} \\ \times \sum_{J\in\mathcal{O}_s} (\kappa_J(\tau)\|h_{J,s_J}^{\top}\| + \bar{\nu}_J^{(s_J)})\mathrm{d}\tau \\ + 2\,d_s\mathrm{e}^{-2\lambda d_s t} \sum_{J\in\mathcal{O}_s} (\mathrm{e}^{\lambda d_s t} - 1 + d_s\delta_{IJ})|\epsilon_J^{(s_J)}(0)|.\end{aligned}$$

Then, a fault will be detected at time–instant $t = t_2$ by the I–th LFD, that is $|\epsilon_I^{(s_I)}(t_2)| > \bar{\epsilon}_I^{(s_I)}(t_2)$. Moreover, t_1 is an upper bound on the fault occurrence time T_0. □

Proof: Following the proof of Theorem 3.2 in [110], the error dynamics at the generic time instant t can be written as follows:

$$\dot{\epsilon}_I^{(s_I)} = \begin{cases} -\lambda\left(\sum_{J\in\mathcal{O}_s}(\epsilon_I^{(s_I)} - \epsilon_J^{(s_J)}) + d_s\epsilon_I^{(s_I)}\right) \\ +\frac{1}{d_s}\sum_{J\in\mathcal{O}_s}(-h_{J,s_J}^{\top}\tilde{\vartheta}_J + \nu_J^{(s_J)}(t)) \qquad \text{if } t < T_0 \\ -\lambda\left(\sum_{J\in\mathcal{O}_s}(\epsilon_I^{(s_I)} - \epsilon_J^{(s_J)}) + d_s\epsilon_I^{(s_I)}\right) \\ +\frac{1}{d_s}\sum_{J\in\mathcal{O}_s}(-h_{J,s_J}^{\top}\tilde{\vartheta}_J^{f} + \nu_J^{f(s_J)}(t)) \qquad \text{if } t \geq T_0 \end{cases},$$

where $\tilde{\vartheta}_J^f \triangleq \hat{\vartheta}_J - \hat{\vartheta}_I^{*f}$. By using the mismatch function, the solution to the error dynamics is

$$\begin{aligned}\epsilon_I^{(s_I)}(t) = \frac{1}{d_s}\int_0^t \mathrm{e}^{-\lambda d_s(t-\tau)} \sum_{J\in\mathcal{O}_s}(-h_{J,s_J}^\top \tilde{\vartheta}_J + \nu_J^{(s_J)})\mathrm{d}\tau \\ + \frac{1}{d_s}\int_{T_0}^t \mathrm{e}^{-\lambda d_s(t-\tau)} \sum_{J\in\mathcal{O}_s}\Delta\phi_J^{(s_J)}\mathrm{d}\tau \\ + \frac{1}{d_s}\mathrm{e}^{-2\lambda d_s t}\sum_{J\in\mathcal{O}_s}(\mathrm{e}^{\lambda d_s t} - 1 + d_s\delta_{IJ})\epsilon_J^{(s_J)}(0)\end{aligned}$$

so that the application of the triangle inequality leads to:

$$\begin{aligned}|\epsilon_I^{(s_I)}(t)| \geq \frac{1}{d_s}\left|\int_{T_0}^t \mathrm{e}^{-\lambda d_s(t-\tau)}\sum_{j\in\mathcal{O}_s}\xi_J^{(s_J)}\mathrm{d}\tau\right| \\ - \frac{1}{d_s}\int_0^t \mathrm{e}^{-\lambda d_s(t-\tau)}\sum_{j\in\mathcal{O}_s}|-z_{j,s_J}^\top\tilde{\vartheta}_J + \nu_J^{(s_J)}|\mathrm{d}\tau \\ - \frac{1}{d_s}\mathrm{e}^{-2\lambda d_s t}\sum_{j\in\mathcal{O}_s}(\mathrm{e}^{\lambda d_s t} - 1 + d_s\delta_{ij})|\epsilon_J^{(s_J)}(0)|\,.\end{aligned}$$

A sufficient condition for the previous inequality to hold is

$$\begin{aligned}|\epsilon_I^{(s_I)}(t)| \geq \frac{1}{d_s}\int_{T_0}^t \mathrm{e}^{-\lambda d_s(t-\tau)}\sum_{J\in\mathcal{O}_s}|\Delta\phi_J^{(s_J)}|\mathrm{d}\tau \\ - \frac{1}{d_s}\int_0^t \mathrm{e}^{-\lambda d_s(t-\tau)}\sum_{J\in\mathcal{O}_s}|-h_{J,s_J}^\top\tilde{\vartheta}_J + \nu_J^{(s_J)}|\mathrm{d}\tau \\ - \frac{1}{d_s}\mathrm{e}^{-2\lambda d_s t}\sum_{J\in\mathcal{O}_s}(\mathrm{e}^{\lambda d_s t} - 1 + d_s\delta_{IJ})|\epsilon_J^{(s_J)}(0)|\,,\end{aligned}$$

and, if $|\epsilon_I^{(s_I)}(t_2)| > \bar{\epsilon}_I^{(s_I)}(t_2)$ for some $t_2 > T_0$, then a fault is detected. This translates into the following inequality

$$\begin{aligned}\int_{T_0}^{t_2}\mathrm{e}^{-\lambda d_s(t-\tau)}\sum_{J\in\mathcal{O}_s}|\Delta\phi_J^{(s_J)}|\mathrm{d}\tau \geq \int_0^{t_2}\mathrm{e}^{-\lambda d_s(t-\tau)} \\ \times\sum_{J\in\mathcal{O}_s}(\kappa_J(\tau)\|h_{J,s_J}^\top\| + \bar{\nu}_J^{(s_J)})\mathrm{d}\tau \\ + \int_0^{t_2}\mathrm{e}^{-\lambda d_s(t-\tau)}\sum_{J\in\mathcal{O}_s}|-h_{J,s_J}^\top\tilde{\vartheta}_J + \nu_J^{(s_J)}|\mathrm{d}\tau \\ + 2\,d_s\mathrm{e}^{-2\lambda d_s t}\sum_{J\in\mathcal{O}_s}(\mathrm{e}^{\lambda d_s t} - 1 + d_s\delta_{IJ})|\epsilon_J^{(s_J)}(0)|\,,\end{aligned}$$

which implies the thesis when $t_1 \geq T_0$, thus proving the theorem. ■

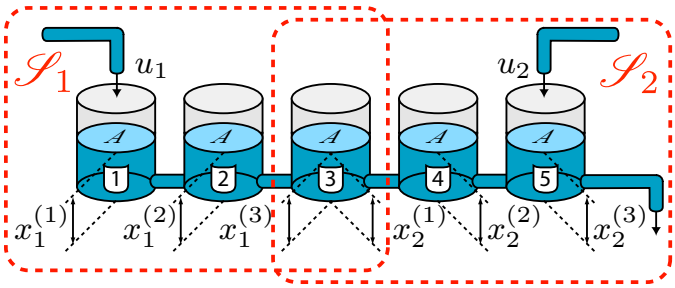


Figure 5.1: Structure of the five-tanks system under consideration.

Remark 5.4.1: It is worth noting that in a distributed fault-diagnosis system without overlap where there are no shared variables, a detection decision may be difficult to reach in presence of a low mismatch $\Delta\phi_J^{(s_J)}$ and/or high uncertainties $\kappa_J(\tau)\|h_{J,s_J}^{\top}\| + \bar{\nu}_J^{(s_J)}$. On the other hand, we expect a consensus mechanism like the one proposed in this chapter to be of benefit in such a scenario. However, due to the generality of the framework considered here (we do not make any assumption on the structural/geometric properties of the faults with respect to the structure of the distributed plant, and we do not assume persistence of excitation) proving that the proposed consensus-based methodology in general performs better than a consensus–less one turns out to be difficult and is beyond the scope of the present work. □

5.5 Illustrative example

Now, a simple example to illustrate the effectiveness of the proposed DFDI scheme will be presented. It is based on the well-known three-tank problem, extended to encompass a five-tank string and two LFDs (see Fig. 5.5). The two LFDs monitor three tanks each, while sharing the third tank. Clearly, here 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 (for details about the dynamical equations of a multi-tank system the reader is referred for example to [36]). All the tanks are cylinders with a cross-section $A = 1$ m^2, whilst every pipe has a cross-section $A_p = 0.1$ m^2 and unitary outflow coefficient. The tank levels are denoted by $x_1^{(i)}$ and $x_2^{(i)}$, with $i = 1, 2, 3$, and are limited between 0 and 10 m. The scalars $0 \leq u_I \leq 1$ m^3/s, $i = 1, 2$, correspond to the inflows supplied by two pumps.

The interconnection variables being $z_1 = x_2^{(2)}$ and $z_2 = x_1^{(2)}$, $g_1(x_1, z_1, u_1)$ and $g_2(x_2, z_2, u_2)$ should be 5-inputs, 3-outputs functions. Because of the topology of this specific example, both g_1 and g_2 have only one non-zero

output component and depend only on $(x_1^{(2)}, x_2^{(1)})$ and $(x_2^{(2)}, x_1^{(3)})$ 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 $\hat{g}_1$ is implemented with 49 basis functions, while the network $\hat{g}_2$ is made of 4 basis functions only. In both cases the basis functions are equally spaced over the square $[0, 10]^2$; the learning rate matrices are $\Gamma_I = \text{diag}(0.75)$ and the estimator constants are $\lambda_I = 1.5$. After suitable offline simulations the parameter Θ_1 and Θ_2 domains are chosen to be hyper-spheres with radii equal to 0.75 and 1.5, respectively. The non-zero bounds on the approximation error are set to $\bar{\nu}_1^{(3)} = 0.025$ and $\bar{\nu}_2^{(1)} = 0.2$. Finally, the inflows are $u_1 = 0.2 \cdot \cos(0.3t) + 0.3$ and $u_2 = 0.25 \cdot \cos(0.5t) + 0.3$; the nominal tank initial levels were 8, 6.5, 5, 3.5 and 3 m, while the estimated ones are 15% higher and 15% lower, respectively for the first and the second LFD.

Fig. 5.5 shows the results of a simulation where at $T_0 = 20$ s an abrupt leakage with cross-section $A_l = 0.15\text{m}^2$ was introduced in tank 3, first when a consensus–filter is employed and then when it is not. In this respect, it can be observed that the LFD based on the network with fewer neurons (hence with more limited approximation capabilities) does not reach a detection decision in the absence of the consensus mechanism whereas a decision is reached in the presence of consensus using the information provided by the other LFD based on a networks with a much larger number of basis functions. The much better performance when the consensus mechanism is used is also due to the fact that the consensus equation dampens the difference between the estimates and the true values and also the difference among the two estimates. This can be seen very clearly by comparing the initial transient behaviors in Figs. 5.5–(a) and 5.5–(b).

5.6 Concluding remarks

In this final chapter a DFDI architecture for continuous–time systems has been described. It featured some simplifications with respect to full-fledged architecture: it was able to only detect faults as it lacked isolation capabilities, considered only abrupt faults and the consensus protocol it applied for shared variables employed constant weights. Anyway the fundamental analytical results were provided, namely: the robustness of detection threshold with respect to uncertainties and the detectability condition. Finally, simulation results were provided to shows the usefulness of the consensus techniques when detecting faults influencing shared variables.

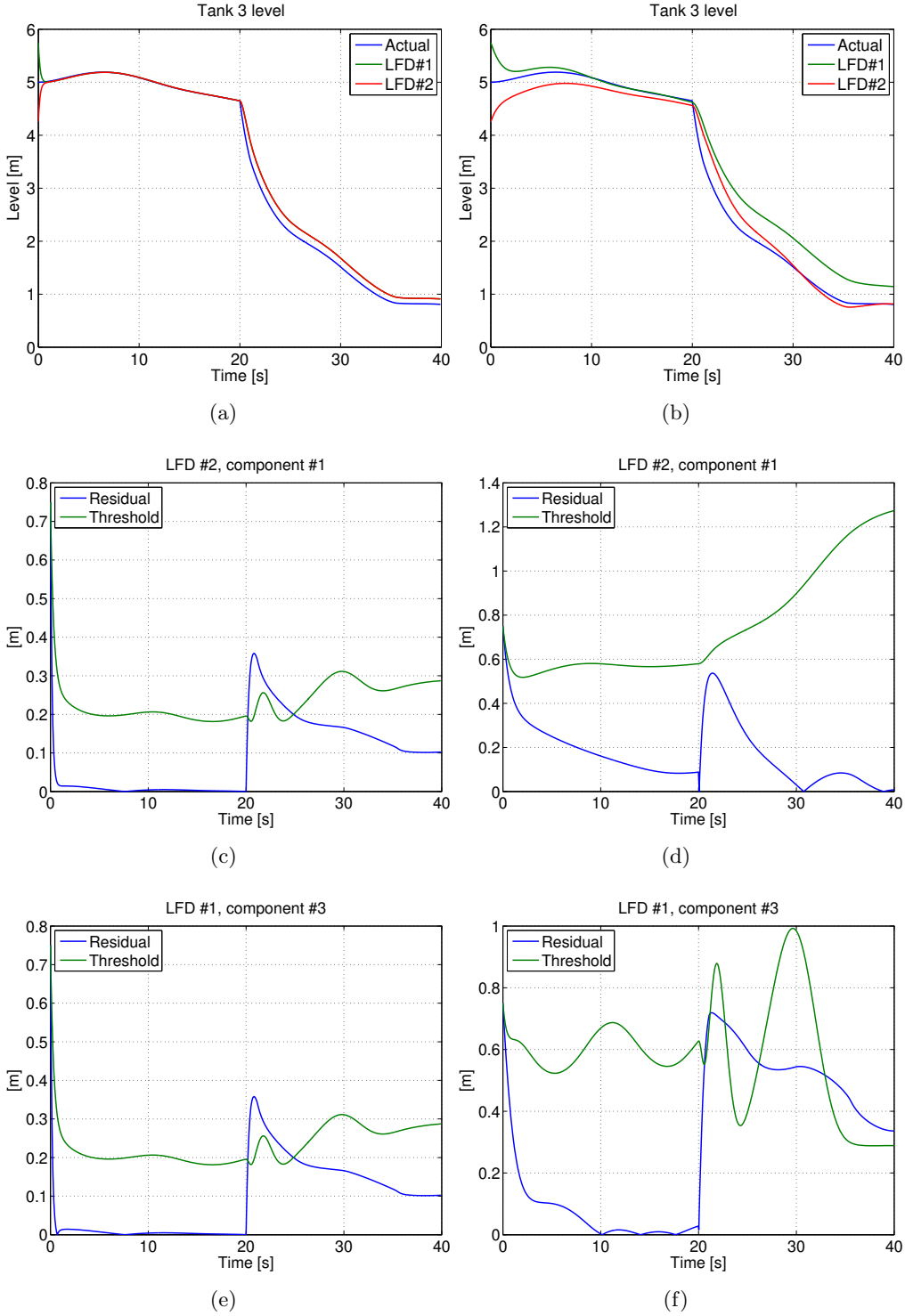


Figure 5.2: Time-behaviors of signals related to tanks no. 3 when a leakage is introduced at time 20 s, with (a, c, e) and without (b, d, f) consensus.

Chapter 6

Concluding Remarks

In this thesis work a general distributed architecture for the Fault Diagnosis of uncertain, nonlinear large–scale systems was developed. The diagnosis logic was inspired a centralized Generalized Observer Scheme, developed in [113] and reviewed in Chapter 2, that solves the *Fault Detection and Isolation* (FDI) problem for nonlinear uncertain discrete–time systems. The typical drawback of centralized schemes, that is the lack of scalability, was overcome by the application of a divide et impera paradigm. In this way, an infeasible FDI problem for a large–scale and monolithic system is decomposed into a number of subproblems, each one simple enough to be solved by available computation and communication infrastructures. Furthermore, the decomposition approach is useful in all the situations were a centralized architecture is simply undesirable, for instance in distributed systems such as sensor networks or multi–vehicle formations. In these kinds of systems, a central node, that would implement the FDI service for the whole network or formation, simply cannot be chosen, and its very existence would nevertheless pose a safety threat.

The starting point in the application of this paradigm, is the availability of a solution to the so–called *decomposition problem*, where the structure and the mathematical model of a monolithic large–scale system $\mathscr{S}$ are decomposed so that a number of subsystems $\mathscr{S}_I$ are obtained. The FDI problem for each subsystem $\mathscr{S}_I$ is then assigned to a single agent, called *Local Fault Diagnoser* (LFD) and denoted by $\mathscr{L}_I$. Each subsystem $\mathscr{S}_I$ represents a part of the original system $\mathscr{S}$, without the parts being required to be mutually disjoint, thus allowing for *overlapping decompositions*. The overlaps in the decomposition lead to portions of $\mathscr{S}$ being shared among different subsystems, so that more than one LFD will monitor it. This redundancy is the key for trying to ease the fault detectability and isolability conditions on shared variables, thanks to the use of consensus techniques between all the involved LFDs.

A distinctive trait of the proposed scheme is that it featured an adaptive

approximation strategy for dealing with the modeling uncertainties. In fact it was assumed that the parts of the original system dynamics that, after the decomposition, account for the *interconnection* between different subsystems, were uncertain. This assumption is particularly suited to situations were the decomposition is *physical* [89], so that each subsystem corresponds to an actual physical component whose connection to other subsystems cannot be completely known *a priori*. The uncertain interconnection part of each subsystem dynamics is learned on–line, before a fault is detected, by using an adaptive approximator, such as a neural network.

The performance of the proposed scheme for both discrete and continuous–time large–scale systems were investigated by deriving the Detectability Theorems 4.4.1 and 5.4.1, and the Isolability Theorem 4.6.1, that respectively pose the conditions for a fault being detected, and then being isolated amongst a class of known faults defined a priori. Furthermore, the absence of false positive alarms due to modeling uncertainties is guaranteed by the use of adaptive thresholds, and is proved for both discrete–time and continuous–time systems.

6.1 Main original contributions

The main original contributions of the present work can be summarized in the following points.

Development of a distributed architecture for quantitative model–based FDI of nonlinear systems This is the most important contribution of this work, as to the best of the author knowledge this is the first formulation of this kind. In fact, while many distributed schemes have been devised for linear discrete–time or continuous time systems, and many qualitative schemes for nonlinear systems, no distributed quantitative scheme has been previously proposed for nonlinear systems. The proposed scheme does not pose any restriction on the form of the nonlinear model assumed for the system, nor does require the subsystems to be connected in a particular way. Furthermore uncertainties are allowed in the model, and not only special care is taken to make the detection and isolation procedures robust to them, but also an adaptive approximator is provided for learning them. And apart from the fault detection and isolation, fault identification has been dealt with as well.

These features are the most important, as they make the proposed scheme completely general, and applicable to engineering systems of arbitrary size without scalability issues[1].

[1]The only kind of system where the proposed architecture would not be conveniently applicable, would be a large–scale and centralized system. That is, a system with a large number of state variables whose dynamics depend upon all the system variables: for

Use of consensus techniques for distributed FDI Another original feature is the use of consensus in the distributed FDI of discrete–time or continuous–time systems. While in discrete–event systems fault diagnosis, especially in the Computer Science community, consensus techniques are quite common, for other kinds of systems they are rather novel, and have been previously used for instance in [144] for the distributed diagnosis of linear discrete–time systems. In this work consensus is implemented in an innovative way in order to tackle the distributed estimation of general non–linear systems. The effects on the performance of the proposed scheme of the use of consensus, furthermore, is analytically studied and the results are included in the Detectability Theorems 4.4.1 and 5.4.1, and in the Isolability Theorem 4.6.1.

6.1.1 Published results

The main contributions presented in this work have been already published in the scientific literature, or are in the process of being submitted. In order of relevance, the publications by the present author on this subject are

- Riccardo M.G Ferrari, Thomas Parisini, Marios M Polycarpou, "Distributed Fault Detection and Isolation: an Adaptive Approximation Approach," *IEEE Transactions on Automatic Control*, (to be submitted).

- Riccardo M.G Ferrari, Thomas Parisini, Marios M Polycarpou, "Distributed Fault Diagnosis with Overlapping Decompositions: an Adaptive Approximation Approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, 2009.

- Riccardo M.G. Ferrari, Thomas Parisini, and Marios M. Polycarpou, "A Robust Fault Detection and Isolation Scheme for a Class of Uncertain Input-output Discrete-time Nonlinear Systems," in *Proc. of American Control Conference 2008 (ACC '08)*, Seattle, June 11-13, 2008.

- Riccardo M.G. Ferrari, Thomas Parisini, and Marios M. Polycarpou, "A Fault Detection and Isolation Scheme for Nonlinear Uncertain Discrete-Time Systems," in *Proc. of Conference on Decision and Control 2007 (CDC '07)*, New Orleans, December 12-14, 2007.

- Riccardo M.G. Ferrari, Thomas Parisini, and Marios M. Polycarpou, "Distributed fault diagnosis with overlapping decompositions and con-

instance it may be a system representing a large number of particles interacting with each other through the gravitational attraction. Anyway, this is a limit case never encountered in actual large–scale engineering systems, whose structure is never completely centralized.

sensus filters," in *Proc. of American Control Conference 2007 (ACC '07)*, New York, July 11-13, 2007.

- Riccardo M.G. Ferrari, Thomas Parisini, and Marios M. Polycarpou, "A Fault Detection Scheme for Distributed Nonlinear Uncertain Systems," in *Proc. of Joint CCA, ISIC and CACSD Conference 2006*, Munich, October 4-11, 2006.

6.2 Future developments

The use of discrete–time models in Chapter 4 opens the door to a lot of possible extensions, in order to take into account many practical issues of actual distributed systems, such as sensor networks and multi–vehicle formations. As briefly discussed in Chapter 3, the proposed distributed FDI formulation is already suited to deal with such systems, although it was always assumed that no delays are present either in the model of the system being monitored, or in the communication channels linking the local fault diagnosers. But the very use of discrete–time models make the inclusion of delays much easier than in the case of continuous–time ones, and indeed for distributed control problems this has already be done successfully, for instance in [54].

Another aspect where the present formulation should be improved is related to the adaptive approximation of interconnection uncertainties. In fact, in the scheme developed so far the allowed domains for the parameters to be learned were assumed to be origin–centered hyper-spheres, and this can lead to high detection threshold because of the parametric uncertainty added by the adaptive approximator itself. Furthermore, not only more general domains should be considered, but some mechanism for stopping the learning should be devised. Unfortunately no persistence of excitation was assumed, so there is no guarantee that the learned parameters will tend to the optimal one, making the approximation error tend to zero. For this reason, a learning–stopping mechanism should be included so that the actual decrease in uncertainty due to the learning up to the present time may be evaluated, and a decision whether to stop the learning be taken. Then, the residual amount of uncertainty may be estimated, thus leading to lower detection and isolation thresholds.

An extension much needed is related to the use of input–output models, so that the requirement of full state measurements may no longer be needed. Similar extensions were already published for centralized continuous and discrete–time systems [35, 38].

Finally a very interesting issue that should be addressed, is how to optimally solve the decomposition problem. In fact in this work we assumed that a solution was already given, and we implicitly assumed that it was a "good" solution, that is a solution that met the computation and communication constraints introduced in Chapter 3. Anyway, in a real–life design

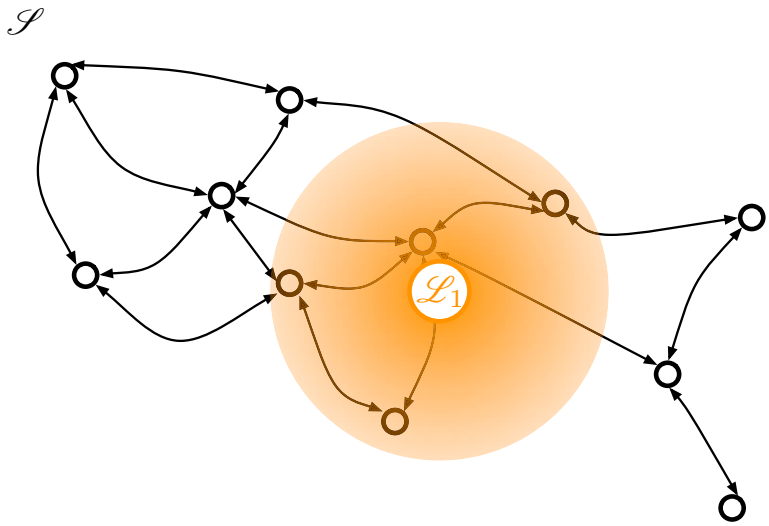


Figure 6.1: An example where the large scale system $\mathscr{S}$ structure is represented by the black graph, and the sensing radius of the only node $\mathscr{L}_1$ of a sensor network is represented by color shades.

of a DFDI system this assumption should not be taken for granted, and almost surely the designer would face the task of decomposing an existing large–scale system for diagnosis purposes. The problem of optimal decomposition was already extensively treated in the literature, for instance in parallel computation problems when simulating large–scale PDEs systems [123, 124, 125, 126]. Anyway, the decomposition problem can be made more interesting by adding some features typical of sensor networks, a kind of infrastructure where DFDI applications are particularly appealing. In fact, a characteristic feature of the nodes of a sensor network, is that the sensing radius of each node is limited. For instance, in Fig. 6.1 it can be seen that a sensor network made by a single node will never be able to cover the large–scale system $\mathscr{S}$, whose structural graph is drawn so that the position of each node corresponds to the physical position of the variable it represents. For this reason, probably a sensor network with at least three nodes should be employed, as shown in Fig. 6.2. The sensing radius boundedness may be represented by a measuring uncertainty that depends on the distance between the measuring node, and the physical location of the variable measured, in some realistic way. Such an interpretation may be valuable when the large–scale system to be monitored for faults is constituted by a large area, such as a marine environment where a fault may consists in an increasing pollutants concentration, or a wood area that may be subject to fires, or a mountainous region where avalanches are probable. In all these examples, a sensor network made by remote–sensing units with communication capabilities may be an ideal solution.

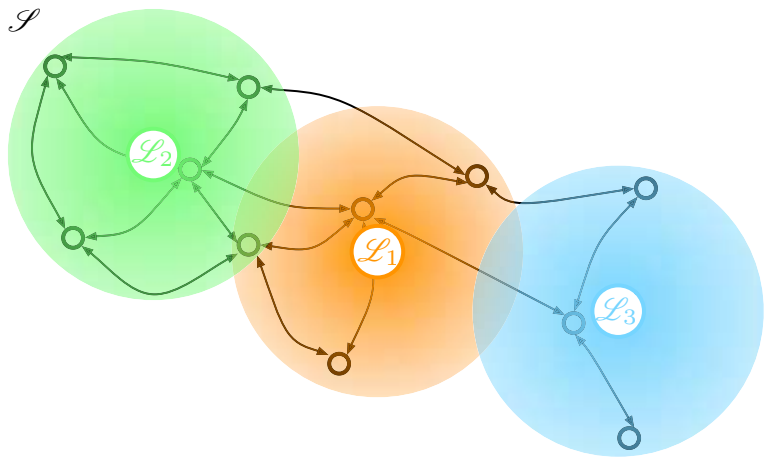


Figure 6.2: The same example of fig. 6.1, but now the sensor network is constituted by three nodes that guarantee a complete cover on $\mathscr{S}$.

Bibliography

[1] Blanke, Kinnaert, Lunze, and M. Staroswiecki, *Diagnosis and Fault Tolerant Control*. Berlin: Springer, 2003.

[2] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006.

[3] Chen and R. Patton, *Robust model-based fault diagnosis for dynamic systems*. Springer, 1999.

[4] J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Syst. Mag.*, vol. 8, no. 6, pp. 3–11, 1988.

[5] 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.

[6] J. Bangura, R. Povinelli, N. Demerdash, and R. Brown, "Diagnostics of eccentricities and bar/end-ring connector breakages in polyphase induction motors through a combination of time-series data mining and time-stepping coupled fe–state-space techniques," *Industry Applications*, vol. 39, no. 4, pp. 1005—1013, 2003.

[7] J. Stack, T. Habetler, and R. Harley, "Fault-signature modeling and detection of inner-race bearing faults," *Industry Applications*, vol. 42, no. 1, pp. 61—68, 2006.

[8] R. Beard, "Failure accomodation in linear systems through self–reorganization," *Technical Report MTV-71-1, Man Vehicle Laboratory, MIT, Cambridge, MA*, 1971.

[9] H. Jones, "Failure detection in linear systems," Ph.D. Thesis, Dept. of Aero and Astro, MIT, Cambridge, MA, 1973.

[10] R. Clark, "Instrument fault detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 14, pp. 456—465, 1978.

[11] R. Isermann, "Process fault detection based on modeling and estimation methods. a survey." *Automatica J. IFAC*, vol. 20, no. 4, pp. 387–404, 1984.

[12] P. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge–based redundancy – a survey and some new results," *Automatica J. IFAC*, vol. 26, no. 3, pp. 459–474, 1990.

[13] 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.

[14] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems.* New York: Marcel Dekker, 1998.

[15] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. Autom. Control*, vol. 29, no. 7, pp. 603—614, 1984.

[16] C. D. Persis and A. Isidori, "A geometric approach to nonlinear fault detection and isolation," *IEEE Trans. Autom. Control*, vol. 46, no. 6, pp. 853–865, 2001.

[17] R. Patton, P. Frank, and Clark, *Fault Diagnosis in Dynamic Systems: Theory and Application.* Prentice Hall, 1989.

[18] J. Gertler, "Analytical redundancy methods in fault detection and isolation," *Proc. IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*, pp. 9—21, 1991.

[19] J. Wünnenberg, "Observer–based fault detection in dynamic systems," Ph.D. Thesis, Universitaet Duisburg, Germany, 1990.

[20] R. Seliger and P. Frank, "Robust component fault detection and isolation in nonlinear dynamic systems using nonlinear unknown input observers," *Proc. IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes*, pp. 313—318, 1991.

[21] M. Kinnaert, "Robust fault detection based on observers for bi-linear systems," *Automatica J. IFAC*, vol. 35, pp. 1829—1842, 1999.

[22] C. D. Persis and A. Isidori, "On the problem of residual generation for fault detection in nonlinear systems and some related facts," *Proc. European Control Conference*, 1999.

[23] A. Emami-Naeini, M. Akhter, and S. Rock, "Effect of model uncertainty on failure detection: the threshold selector," *IEEE Trans. Autom. Control*, vol. 33, pp. 1106—1115, 1988.

[24] H. Wang and S. Daley, "Actuator fault diagnosis: an adaptive observer-based technique," *IEEE Trans. Autom. Control*, vol. 41, no. 7, pp. 1073—1078, 1996.

[25] A. T. Vemuri and M. M. Polycarpou, "Robust nonlinear fault diagnosis in input–output systems," *Internat. J. Control*, vol. 68, no. 2, pp. 343—360, 1997.

[26] H. Wang, Z. Huang, and S. Daley, "On the use of adaptive updating rules for actuator and sensor fault diagnosis," *Automatica J. IFAC*, vol. 33, pp. 217—225, 1997.

[27] M. Basseville and I. V. Nikiforov, "Detection of abrupt changes: Theory and application," *Prentice Hall*, vol. New York, 1993.

[28] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies," *Computers & Chem. Eng.*, vol. 27, pp. 313–326, 2003.

[29] J. Farrell, T. Berger, and B. Appleby, "Using learning techniques to accommodate unanticipated faults," *IEEE Control Syst. Mag.*, vol. 13, pp. 40—49, 1993.

[30] M. M. Polycarpou and A. Helmicki, "Automated fault detection and accommodation: a learning systems approach," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 11, pp. 1447–1458, 1995.

[31] M. M. Polycarpou and A. Trunov, "Learning approach to nonlinear fault diagnosis: detectability analysis," *IEEE Trans. Autom. Control*, vol. 45, pp. 806–812, 2000.

[32] M. M. Polycarpou, "Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach," *IEEE Trans. Autom. Control*, vol. 46, no. 5, pp. 736–742, 2001.

[33] 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.

[34] A. T. Vemuri and M. M. Polycarpou, "Neural-network-based robust fault diagnosis in robotic systems," *IEEE Trans. Neural Netw.*, vol. 8, no. 6, pp. 1410–1420, 1997.

[35] X. Zhang, M. M. Polycarpou, and T. Parisini, "Robust fault isolation for a class of non–linear input–output systems," *Internat. J. Control*, vol. 74, no. 13, pp. 1295–1310, 2001.

[36] ——, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 576–593, 2002.

[37] X. Zhang, T. Parisini, and M. M. Polycarpou, "Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1259–1274, 2004.

[38] 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., 2008 (ACC '08)*, p. 6, 2008.

[39] J. M. Fowler and R. D'Andrea, "A formation flight experiment," *IEEE Control Syst. Mag.*, vol. 23, no. 5, pp. 35—43, 2003.

[40] 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.

[41] D. M. Stipanovič, Inalhan, Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica J. IFAC*, vol. 40, no. 8, pp. 1285–1296, 2004.

[42] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Trans. Autom. Control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[43] Sinopoli, Sharp, Schenato, and Schaffert, "Distributed control applications within sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1235–46, 2003.

[44] W. Dunbar and R. Murray, "Receding horizon control of multi-vehicle formations: A distributed implementation," *Proc. 43th IEEE Conf. on Decision and Control (CDC '04)*, 2004.

[45] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of discrete-time agents with delayed information exchange: A receding-horizon approach," *Proc. 43th IEEE Conf. on Decision and Control (CDC '04)*, pp. 4274—4279, 2004.

[46] 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.

[47] Langbort, R. Chandra, and R. D'Andrea, "Distributed control design for systems interconnected over an arbitrary graph," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1502–1519, 2004.

[48] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401—420, 2006.

[49] J. Baillieul and P. Antsaklis, "Control and communication challenges in networked real-time systems," *Proc. IEEE*, vol. 95, no. 1, pp. 9–28, 2007.

[50] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[51] M. M. Polycarpou, G. Uber, Z. Wang, F. Shang, and Brdys, "Feedback control of water quality," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 68–87, 2002.

[52] M. Baglietto, T. Parisini, and R. Zoppoli, "Distributed-information neural control: the case of dynamic routing in traffic networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 3, pp. 485–502, 2001.

[53] S. Boccaletti, J. Kurths, G. Osipov, and D. Valladares, "The synchronization of chaotic systems," *Phys. Rep*, vol. 366, pp. 1–101, 2002.

[54] 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. Autom. Control*, vol. 53, no. 1, pp. 324–338, 2008.

[55] V. Kapila, A. G. Sparks, J. Buffington, and Q. Yan, "Spacecraft formation flying: Dynamics and control," *Proc. American Control Conf., 1999 (ACC '99)*, pp. 4137–4141, 1999.

[56] C. Sabol, R. Burns, and C. McLaughlin, "Satellite formation flying design and evolution," *Journal of Spacecraft and Rockets*, vol. 38, no. 2, 2001.

[57] G. Yang, Q. Yang, V. Kapila, D. Palmer, and R. Vaidyanathan, "Fuel optimal manoeuvres for multiple spacecraft formation reconfiguration using multi-agent optimization," *Int. J. Robust Nonlinear Control*, vol. 12, pp. 243–283, 2002.

[58] S. Stankovič, M. Stanojevič, and D. Šiljak, "Decentralized overlapping control of a platoon of vehicles," *IEEE Trans. Autom. Control*, vol. 8, no. 5, 2000.

[59] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 947—951, 2001.

[60] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," *Proc. of Robotics: Science and Syst.*, 2006.

[61] M. Earl and R. D'Andrea, "A decomposition approach to multi-vehicle cooperative control," *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 276—291, 2007.

[62] P. Li, L. Alvarez, and R. Horowitz, "Ahs safe control laws for platoon leaders," *IEEE Trans. Autom. Control*, vol. 5, no. 6, pp. 614–628, 1997.

[63] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25—34, 1987.

[64] 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.

[65] T. Vicsek, "A question of scale," *Nature*, vol. 411, p. 421, 2001.

[66] ——, "The bigger picture," *Nature*, vol. 418, p. 131, 2002.

[67] 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.

[68] I. Farkas, D. Helbing, and T. Vicsek, "Mexican waves in an excitable medium," *Nature(London)*, 2002.

[69] Y. Liu and K. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 30–43, 2004.

[70] 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.

[71] G. Palla, A.-L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[72] A. Jadbabaie, J. Lin, and S. A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[73] W. Wang and J.-J. E. Slotine, "A theoretical study of different leader roles in networks," *IEEE Trans. Autom. Control*, vol. 51, no. 7, pp. 1156–1161, 2006.

[74] 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.

[75] O. Imer, S. Yuksel, and T. Basar, "Optimal control of lti systems over unreliable communication links," *Automatica J. IFAC*, vol. 42, no. 9, pp. 1429–1439, 2006.

[76] A. Speranzon, C. Fischione, and K. Johansson, "Distributed and collaborative estimation over wireless sensor networks," *Proc. 45th IEEE Conf. on Decision and Control (CDC '06)*, pp. 1025–1030, 2006.

[77] R. Olfati-Saber and R. Murray, "Consensus protocols for networks of dynamic agents," *Proc. American Control Conf., 2003 (ACC '03)*, vol. 2, pp. 951—956, 2003.

[78] ——, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[79] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," *Proc. American Control Conf., 2005 (ACC '05)*, vol. 3, no. 1859–1864, 2005.

[80] R. Olfati-Saber, E. Franco, Frazzoli, and J. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," *Lecture Notes in Control and Inform. Sci.*, vol. 331, pp. 169–182, 2006.

[81] M. Mehyar, D. Spanos, J. Pongsajapan, S. Low, and R. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 512–520, 2007.

[82] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[83] 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.

[84] A. Fagiolini, E. M. Visibelli, and A. Bicchi, "Logical consensus for distributed network agreement," *Proc. 47th IEEE Conf. on Decision and Control (CDC '08)*, pp. 5250—5255, 2008.

[85] Q. Hui and W. M. Haddad, "Distributed nonlinear control algorithms for network consensus," *Automatica J. IFAC*, vol. 44, no. 9, pp. 2375–2381, 2008.

[86] M. S. Stankovič, S. Stankovič, and D. M. Stipanovič, "Consensus based multi-agent control structures," *Proc. 47th IEEE Conf. on Decision and Control (CDC '08)*, p. 6, 2008.

[87] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," *Proceedings of IEEE Infocom*, pp. 1653–1666, 2005.

[88] S. Wang and E. Davidson, "On the stabilization of decentralized control systems," *IEEE Trans. Autom. Control*, vol. 18, no. 5, pp. 473–478, 1973.

[89] D. Šiljak, *Large-Scale Dynamic Systems: Stability and Structure*. North Holland, 1978.

[90] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Trans. Autom. Control*, vol. 23, no. 2, pp. 108–128, 1978.

[91] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella, "Diagnosis of large active systems," *Artificial Intelligence*, no. 110, pp. 135–189, 1999.

[92] 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.

[93] Wu and C. N. Hadjicostis, "Distributed non-concurrent fault identification in discrete event systems," *Proc. CESA 2003*, 2003.

[94] 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.

[95] 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.

[96] 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.

[97] F. Preparata, G. Metze, and R. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 6, pp. 848–854, 1967.

[98] 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.

[99] 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.

[100] Pike, Miner, and T. Wilfredo, "Model checking failed conjectures in theorem proving: A case study," *NASA Technical Report*, no. TM–2004–213278, 2004.

[101] X. Yang and Y. Tang, "Efficient fault identification of diagnosable systems under the comparison model," *IEEE Trans. Comput.*, vol. 56, no. 12, pp. 1612–1618, 2007.

[102] A. Fagiolini, G. Valenti, L. Pallottino, and G. Dini, "Decentralized intrusion detection for secure cooperative multi-agent systems," *Proc. 46th IEEE Conf. on Decision and Control (CDC '07)*, 2007.

[103] 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., 2007 (ACC '07)*, pp. 408—413, 2007.

[104] P. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 29, no. 6, pp. 554—565, 1999.

[105] 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.

[106] Roychoudhury, Biswas, K. Xenofon, and Abdelwahed, "Designing distributed diagnosers for complex physical systems," *Proc. 16th International Workshop on Principles of Diagnosis*, 2005.

[107] M. Daigle, X. Koutsoukos, and G. Biswas, "Distributed diagnosis in formations of mobile robots," *IEEE Trans. Robot.*, 2007.

[108] W. H. Chung, J. L. Speyer, and R. H. Chen, "A decentralized fault detection filter," *J. of Dynamic Syst., Meas., and Control*, vol. 123, pp. 237–247, 2001.

[109] N. Lechevin, C. A. Rabbath, and E. Earon, "Towards decentralized fault detection in uav formations," *Proc. American Control Conf., 2007 (ACC '07)*, pp. 5759—5764, 2007.

[110] R. M. Ferrari, T. Parisini, and M. M. Polycarpou, "A fault detection scheme for distributed nonlinear uncertain systems," *Proc. 2006 IEEE International Symposium on Intelligent Control (ISIC '06). Munich, Germany, 4-6 October 2006.*, pp. 2742–2747, 2006.

[111] ——, "Distributed fault diagnosis with overlapping decompositions and consensus filters," *Proc. American Control Conf., 2007 (ACC '07)*, pp. 693—698, 2007.

[112] ——, "Distributed fault diagnosis with overlapping decompositions: an adaptive approximation approach," *IEEE Trans. Autom. Control*, vol. (to appear), 2009.

[113] ——, "A fault detection and isolation scheme for nonlinear uncertain discrete-time sytems," *Proc. 46th IEEE Conf. on Decision and Control (CDC '07)*, pp. 1009–1014, 2007.

[114] J. Farrell and M. M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy, and Traditional Adaptive Approximation Approaches*. Hoboken, NJ: Wiley-Interscience, 2006.

[115] M. M. Polycarpou, "On–line approximators for nonlinear system identification: A unified approach," *Control and Dynamic Systems: Neural Network Systems Techniques and Applications*, vol. 7, pp. 191–230, 1998.

[116] C. Johnson, *Lectures on adaptive parameter estimation*. Upper Saddle River, NJ, USA: Prentice Hall, 1988.

[117] Astrom and Wittenmark, *Adaptive Control*, 2nd ed. Prentice Hall, 1994.

[118] F. Lewis, Yeşildirek, and Jagannathan, *Neural Network Control of Robot Manipulators and Non-Linear Systems*. CRC Press, 1998.

[119] G. Goodwin and Sin, *Adaptive filtering prediction and control*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[120] Chartrand and O. Oellermann, *Applied and Algorithmic Graph Theory*. McGraw-Hill College, 1992.

[121] K. P. Bogart, *Introductory Combinatorics*. Marshfield, MA, USA: Pitman Publishing, Inc., 1983.

[122] 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.

[123] X. Cai and Y. Saad, "Overlapping domain decomposition algorithms for general sparse matrices," *Numerical Linear Algebra with Applications*, vol. 3, no. 3, pp. 221—237, 1996.

[124] B. F. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 2004.

[125] J. Lagnese and G. Leugering, *Domain Decomposition Methods in Optimal Control of Partial Differential Equations*. Birkhäuser, 2004.

[126] H. Simon, "Partitioning of unstructured problems for parallel processing," *Computing Systems in Engineering*, vol. 2, pp. 135—148, 1991.

[127] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888—905, 2000.

[128] D. Johnson, C. Aragon, L. McGeoch, and C. Schevon, "Optimization by simulated annealing: an experimental evaluation. Part I: Graph partitioning," *Operations Research*, vol. 37, no. 6, pp. 865–892, 1989.

[129] M. Vidyasagar, "Decomposition techniques for large-scale systems with nonadditive interactions: Stability and stabilizability," *IEEE Trans. Autom. Control*, vol. AC-25, no. 4, pp. 773–779, 1980.

[130] S. Stankovič, M. S. Stankovič, and D. M. Stipanovič, "Consensus based overlapping decentralized estimator," *Proc. American Control Conf., 2007 (ACC '07)*, pp. 2744—2749, 2007.

[131] Ikeda and D. Šiljak, "Overlapping decompositions, expansions, and contractions of dynamic systems," *Large Scale Systems*, vol. 1, pp. 29–38, 1980.

[132] M. Hodžič and D. Šiljak, "Decentralized estimation and control with overlapping information sets," *IEEE Trans. Autom. Control*, vol. AC-31, no. 1, 1986.

[133] M. Singh, D. Li, Y. Chen, M. Hassan, and Q. Pan, "New approach to failure detection in large-scale systems," *IEE Proceedings D. Control Theory and Applications*, vol. 130, pp. 243–249, 1983.

[134] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Autom. Control*, vol. 48, no. 9, pp. 1478–1495, 2003.

[135] R. Olfati-Saber and J. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," *Proc. 44th IEEE Conf. on Decision and Control and Eur. Control Conf. 2005 (CDC '05 ECC '05)*, pp. 6698–6703, 2005.

[136] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667—689, 2004.

[137] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine*, vol. 40, no. 8, pp. 102—114, 2002.

[138] L. Xiao, S. Boyd, and S. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, pp. 33–46, 2007.

[139] F. Xiao and L. Wang, "Consensus protocols for discrete-time multi-agent systems with time-varying delays," *Automatica J. IFAC*, vol. 44, no. 10, pp. 2577–2582, 2008.

[140] 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.

[141] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, pp. 65—78, 2004.

[142] M. Hazewinkel, *Encyclopaedia of mathematics*. Springer-Verlag, 2002.

[143] L. Grujic and D. Šiljak, "On stability of discrete composite systems," *IEEE Trans. Autom. Control*, vol. 18, no. 5, pp. 522– 524, 1973.

[144] E. Franco, R. Olfati-Saber, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis using sensor networks and consensus-based filters," *Proc. 45th IEEE Conf. on Decision and Control (CDC '06)*, 2006.

Index