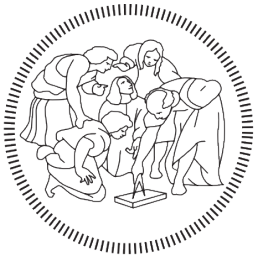

POLITECNICO
MILANO 1863

Scuola di Ingegneria Industriale e dell'Informazione
Master of Science in Automation and Control Engineering

Model based approaches for fault detection of stochastic large scale systems

Relatore: Prof. Marcello Farina

Tesi di Laurea di:
Andrea Caspani Matr. 877304

Anno Accademico 2018-2019

to Jessica

Abstract

As modern engineering systems are becoming increasingly more complex, high performances, reliability, and safety properties play a fundamental role in their technological development. Abreast of the hardware development of monitoring systems, which brings about new sensors and higher computational capabilities as time goes by, there is a constant research over smart algorithms capable to exploit the generated data. In this scenario a main role is played by fault detection and isolation (FDI) algorithms.
A large number of different FDI algorithms can be found in literature, most of them belong to two main categories: signal-based algorithms and model-based ones.
In this Thesis a recently-proposed scheme for model-based fault detection will be tested. More specifically, the proposed algorithm belongs to the class of observer-based methods. Also, a new observer-based scheme for fault isolation, based on fault models, will be introduced, described and tested.
Both algorithms have been designed to be implemented in a distributed fashion based on partitioned system models, in order to be used in a large scale system scenario.
In details, the original contributions of this Thesis are the following.
First we will implement the aforementioned model-based partition-based fault detection algorithm on two different large scale systems. A chemical plant, which consists in a distributed system where subsystems own strong relationships with their neighbouring system, and a power network system, which instead shows the case where the relationships between the subsystems are weaker. This is done with the purpose of highlighting the main advantages and limitations of the proposed algorithm in realistic case studies.
Second, a new algorithm that exploits fault models (assuming that faults are persistent, when present) is proposed. Both a centralized and a distributed version of the algorithm are implemented. By means of a suitable filtering process, the value taken by the fault at each time instant is estimated, and such value is then compared with analytically computed thresholds. This is done with the purpose of combining together the main advantages of the two considered approaches.

Strengths and weaknesses of the proposed methods are analysed in detail, combining both the simulation results and the theoretical explanations.

Sommario

L'esponenziale aumento di complessità nei moderni impianti tecnologici ha reso l'affidabilità e la sicurezza, così come la costante ricerca di migliorie nelle prestazioni, un elemento di fondamentale importanza per il loro sviluppo tecnologico. Di pari passo con lo sviluppo hardware, che ci rifornisce quotidianamente di nuova sensoristica e di migliori capacità di calcolo, lo sviluppo di algoritmi in grado di sfuttare al massimo un sempre maggior numero di dati rappresenta un importante settore di ricerca. In questo scenario si collocano gli algoritmi che si occupano di rivelazione di malfunzionamenti (i.e., fault detection) e della loro identificazione (i.e., fault isolation).
Molti algoritmi di questo tipo sono diponibili in letteratura, e, nella maggior parte dei casi, possono essere raggruppati in due macro categorie: algoritmi signal-based ed algoritmi model-based.
In questa Tesi sono esaminate le performance di un algoritmo di fault detection di tipo model based sviluppato recentemente, in particolare ci riferiremo alla sottoclasse degli algoritmi cosiddetti observer-based. Successivamente è proposto un nuovo algoritmo per la fault isolation anch'esso di tipo observer based che sfrutta malfunzionamenti (i.e., fault) modellizzati matematicamente.
Entrambi gli algoritmi sono stati implementati per garantirne l'applicabilità a casi studio distribuiti, in modo da essere utilizzabili su impianti reali di grandi dimensioni.
I contributi originali di questa Tesi sono i seguenti.
Per prima cosa è stato implementato l'algoritmo per la fault detection sopracitato e ne sono state valutate le prestazioni, considerando due impianti di grandi dimensioni: un impianto chimico ed un modello di rete elettrica. Il primo è stato scelto perchè rappresenta un caso di sistema distribuito dove i legami tra sottosistemi sono molto forti, il secondo per il motivo opposto. Questa scelta è giustificata dal fatto che si intende mostrare i vantaggi e gli svantaggi dell'algoritmo in scenari di tipo realistico.
In secondo luogo è stato sviluppato un nuovo algoritmo per l'isolazione dei fault che sfrutta i modelli dei guasti (supponendo che essi siano persistenti, quando presenti). Mediante un processo di stima il valore della variabile associata ad uno

specifico guasto é approssimata ad ogni estante di tempo. Il valore così ottenuto viene poi confrontato con delle specifiche soglie, a loro volta calcolate mediante metodi analitici.
I principali punti di forza, così come le debolezze di tali approcci sono analizzati in dettaglio, combinando i risultati delle simulazioni alle spiegazioni teoriche.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As modern engineering systems are becoming increasingly more complex, high performances, reliability, and safety properties play a fundamental role in their technological development. Abreast of the hardware development of monitoring systems, which brings about new sensors and higher computational capabilities as time goes by, there is a constant research over smart algorithms capable to exploit the generated data. In this scenario a main role is played by fault detection and isolation (FDI) algorithms.
Considering a general system or plant, a fault can be defined as any relevant deviation from its standard behaviour. A fault affecting a system highlights the presence of some kind of malfunctioning; it is then crucial to rapidly detect and locate it. Fault detection can be defined as the process which, by means of a suitable manipulation of the available data (including model of the system, outputs and inputs), infers the health state of the system, identifying the presence of any possible fault. Fault isolation refers to step of localization of a detected fault on the plant. In few words, fault detection warn us about the presence of a malfunctioning, while fault isolation locates it.

1.1 Fault detection and isolation

A large number of different FDI algorithms can be found in literature (e.g., in [6], in [2] and in [13]). Most of them belong to two main categories: signal-based algorithms and model-based ones.

Signal-based FDI algorithms (e.g., the one in [8]) are essentially data-driven, meaning that they exploit exclusively data collected by sensors, inputs and outputs of the system, and make use of statistical and logical analysis. First of all, considering the plant to be in "healthy" state (no Faults occurrence), data are collected

for a period of time. These data are used to define the features characterizing the system behaviour when no fault is acting. In this way, observing the plant behaviour at work, and comparing new plant data, collected online, with the derived features, it is possible to detect a unhealthy performance, and then possibly declare and locate the fault.

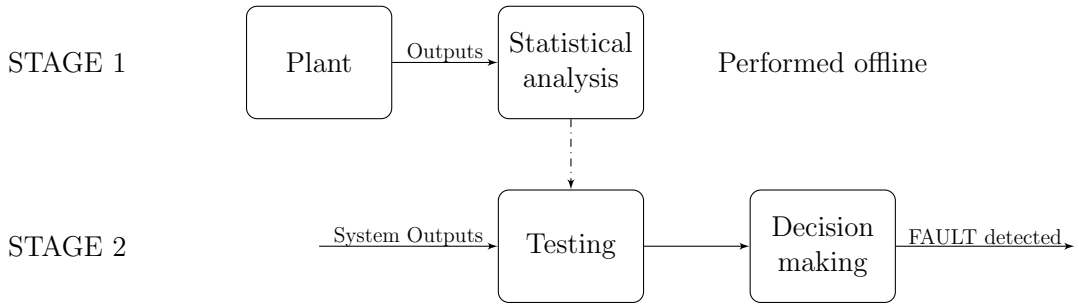

Figure 1.1: Stages of signal-based fault detection and isolation.

Signal-based methods strongly rely on the analysis of the data pre-generated by the system. This fact makes such approaches vulnerable in the acquisition phase, considering that even a small deviation of the outputs will influence the statistical analysis, and then the testing phase. Also, such methods are rigid to changes in the plant, meaning that every time the system is modified, the whole acquisition process must be afresh performed.

Model-based FDI algorithms (e.g., the ones in [1] and [7]) instead make fundamental use, as the name says, of the mathematical model of the considered plant, derived from physical equations and relationships. The available model is exploited to compute confidence levels characterizing the behaviour of the outputs in absence of faults.

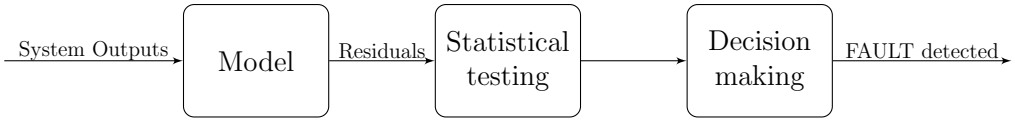

Figure 1.2: Stages of model-based fault detection and isolation.

Model-based methods do not require any offline acquisition of the data. Once the mathematical model of the system is derived, specific signals are generated combining information coming from plant sensors and analytically obtained values coming from the system model, e.g. observer residuals. The detection is then performed

comparing these residuals with specific analytically computed bounds. Model-based approaches are less data-dependent and are less rigid to system changes with respect to signal-based algorithms.
In this thesis a recently-proposed scheme for model-based fault detection will be tested. More specifically, the proposed algorithm belongs to the class of observer-based methods.

1.2 Observer based fault detection

Given the mathematical model of the system and the plant output vector y, it is possible to use an observer to compute an output vector estimate $\hat{y}$.
Consider the general linear time invariant (LTI) system affected by noise

$$\begin{aligned} x(k+1) &= Ax(k) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{1.1}$$

where $x(k)$ is the state vector, $y(k)$ the output vector, $u(k)$ the inputs, $w(k)$ and $v(k)$ are stochastic noises applied to the state and to the outputs respectively, and A, C are the system matrices.
The related Luenberger observer is given by the equations

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + L(y - \hat{y}) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{1.2}$$

where $\hat{x}(k)$ and $\hat{y}(k)$ are the estimated state and the estimated output vectors respectively, while L is the gain matrix.
As a fault occurs, provided that certain structural conditions are verified, the system outputs y will be affected in some way. Since the observer is derived from the standard (i.e., nominal) model of the plant (i.e. when faults are absent), it is not designed to account for possible anomalous behaviours. This means that the estimation gets worse as the fault affects the system, or roughly speaking that the estimation error "increases". Analysing the trend of the residual $r(k) = y(k) - \hat{y}(k)$ it is then possible to monitor the health status of the system, and then to possibly declare fault.

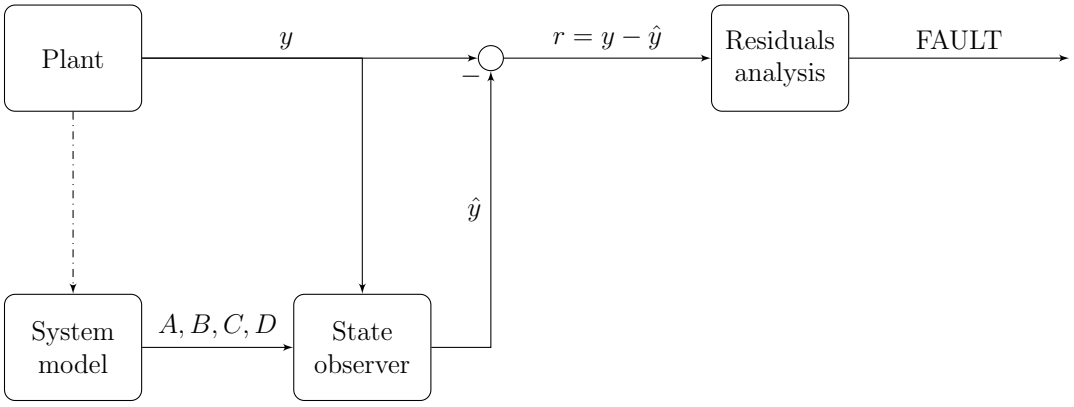

Figure 1.3: Stages of observer-based fault detection and isolation.

Note however that residuals, in case of faults, are affected both by the noise components $v(k)$ and $w(k)$ and by faults at the same time. The algorithm should then be able to distinguish a fault event analysing a noisy signal. To do so, at each time step k the residual $r(k)$ is compared with a threshold $\rho(k)$. The latter is computed analytically from the mathematical model and observer matrices, in order to guarantee a desired level of false alarm and missed detection rate. A possible fault detection test is the following.

$$\begin{cases} \text{if } |r(k)| < \rho(k), \text{then no fault is detected} \\ \text{otherwise fault is detected} \end{cases}$$

This kind of residuals monitoring gives no information about a revealed fault, it is then not suitable for fault isolation. In this thesis a modified (i.e., distributed) version of such approach is considered, which exploits estimated fault signals and returns just a partial isolation (i.e. faults can be located but there is no information about their properties).

1.3 Observer based detection and isolation based on fault models

An alternative approach to the one described in Chapter 1.2 is based on explicit fault models. Common fault models are described in Section 1.3.1.

1.3.1 Fault models

As discussed, a fault is any relevant deviation from the standard behaviour of a general system or plant. This definition includes a large number of different criticalities possibly affecting our system, as described below.

Consider the general linearized and discretized equations (1.2) governing the behaviour of an engineering system. Some major categories of faults can be mathematically modelled to improve the performances of our FDI algorithms:

- **Additive faults** are modelled as signals directly added on the system equations, either on the state or on the outputs, i.e.,

$$\begin{aligned} x(k+1) &= Ax(k) + B_f \boldsymbol{f_{add}(k)} + w(k) \\ y(k) &= Cx(k) + D_f \boldsymbol{f_{add}(k)} + v(k) \end{aligned} \tag{1.3}$$

  The elements of vector $f_{add}(k)$ are set to zero in "healthy" system conditions while, when a fault occurs, they take non zero values.
  Faults of this kind represent by a bias acting directly on the system equations, for example:

  - leakages in tanks, pipes, etc., which bring about direct losses of liquid flowrates, and subsequently affect the related state variables;
  - broken output sensor, which return a biased status of the system, that may lead to a wrong control action aimed to balance the bias.

- **Multiplicative faults** are modelled as perturbations on the system parameters, e.g.,

$$x(k+1) = (I + \boldsymbol{f_{mul}(k)})Ax(k) + w(k) \tag{1.4}$$

  When the elements of matrix $f_{mul}(k)$ are different from zero a shift in the outputs measurements occurs,which is linearly dependent upon the known inputs. Also the system dynamic response can be compromised.
  A change in the parameter of a system can be related to different processes. Some examples can be:

  - change in the inertia of a power generation unit;
  - flow rate reduction due to deposits in water pipes.

1.3.2 Fault detection and isolation

The fault models introduced above can be used in the isolation process. Consider, for example the system affected by additive noise in equation (1.3) (although the same line of reasoning can be applied to model (1.4)).
The main idea consists of considering, in the estimation process, the fault vector $f_{add}(k)$ as a component of the state vector. Specifically $f_{add}(k)$ is considered as a constant state of the system, i.e., with model

$$f_{add}(k+1) = f_{add}(k)$$

An enlarged system including faults is then obtained. Afterwards an enlarged predictor is used to estimate simultaneously the state vector x and the fault vector f_{add}, returning the estimated vectors $\hat{x}$ and $\hat{f}_{add}$, respectively.
Similarly to the fault detection case, the isolation is carried on comparing at each time instant k each component of $\hat{f}_{add}(k)$ with a suitable analytical threshold. Such threshold is rigorously computed from the expected variance of the estimated fault vector, as better specified in the following chapters.

1.4 Fault detection and isolation for large scale systems

The early detection and the diagnosis of incipient faults are crucial tasks especially while dealing with large plants. Large scale systems (LSSs) such as complex process plants, high-performance ships, space vehicles, and many others engineering apparatuses, rely on automatic diagnostic systems such as FDI to properly react to failure situations. This allows operations ranging from immediate emergency actions to the long-term modification of the maintenance schedule.
The aim of this thesis is to devise FDI schemes, with a special focus on their implementation over large scale systems. Complex technological plants, such as chemical systems or power networks, are characterized by the large number of components and sensors. In these cases centralized approaches (for control and monitoring) are prone to possible problems: from one side, computational problems can be encountered in algorithmic implementations, as the computational complexity grows exponentially with the number of variables; on the other side, such approaches are rigid, meaning that any change in the plant requires to adapt the whole (control and monitoring) algorithm to the new system configuration.
Both issues can be solved devising and implementing distributed algorithms.
First, the large scale system is divided (i.e., partitioned) in smaller components or subsystems, taking into account of all the relationships between them. Physically, a subsystem is a self-contained part of the plant, with its own inputs and outputs, but where couplings with other subsystems are possible. For example, the chemical plant proposed in [11] is composed by two reactors and a separator.

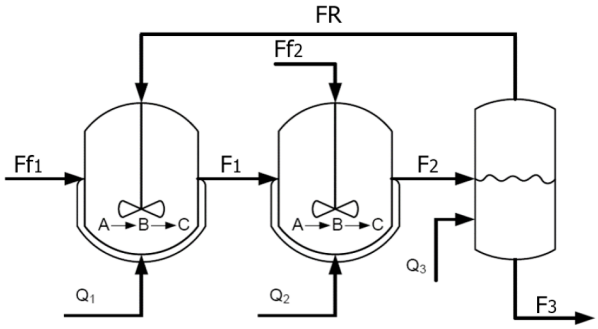

Figure 1.4: Chemical system scheme from [11]

The liquid flows in the first reactor, then in the second and at last in the separator. There is also a recirculation flow from the separator to the first reactor that makes the system strongly coupled. Here it is easy to define, as subsystems, the reactors and the separator. Graphically, the system can be represented as a graph connecting the subsystems as follows.

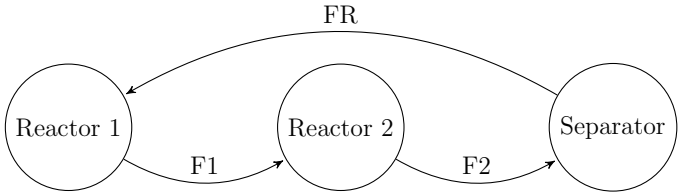

Figure 1.5: Coupling graph of the chemical system in [11]

Here circles represent the subsystems, while arrows represent the coupling relationships between them. The direction of an arrow from i to j means that some variables from subsystem i are used by subsystem j and not vice-versa.
Once given such new system (*decomposed*) representation, it is then possible to redefine the monitoring problem in a distributed way. Since each subsystem is modelled and treated as a single system where external contributions are due to the relationships between the subsystems, the computational load is split out in a number of local estimators (meaning an overall linear growth of the computational complexity of the scale of the system and no more exponential with respect to the system size).
Distributed approaches are less rigid than purely centralized ones, as they allow the implementation of plug and play policies. When a subsystem is damaged, it can be removed or modified without the need to rearrange the whole control process, but just the local features.

1.5 Contribution of this thesis

This thesis is developed following the main research results in [3] and [8] about model-based distributed fault detection.
In [3] and [1], a distributed model-based fault detection method grounded on a partition-based Luenberger observer is developed. For each subsystem, a local observer is computed and information is exchanged only between the neighbouring observers. The covariance of the residuals is also computed locally and it is used to obtain consistent fault detection thresholds.
The fault detection test is faced in two ways: the first one, proposed in [3], consists in comparing, for each time instant, the output prediction error with the aforementioned thresholds; the second one (discussed in [1]) resorts to a method based on the analysis of the moving averages of the residuals, which are compared to a analytically obtained thresholds. While the application of the first method to an academic example highlights the presence of an often restrictive trade off between false-alarm and missed-detection rates, the second approach seems to overcome these significant drawbacks: it both reduces significantly the false-alarm rate and, improves the detection rate of even faults of small entity. The algorithm does not isolate faults in a strict sense and faults are not explicitly modelled inside the observer equations. Indeed it allows to locate the subsystem affected by a fault, but not the specific fault and its properties.
In [8], a partition-based model-based FDI scheme based on moving horizon estimation is instead developed. This algorithm is able to estimate both the state variables and the possible faults, since faults are explicitly modelled as states of the system. Nevertheless, this algorithm is used only for deterministic systems, since there is no definition of bounds or thresholds on the residual variance. Moreover, the implementation is based on the parallelization of an optimization problem and requires an iterative information spread in an all-to-all fashion. This method is tested with a chemical plant case study, showing the effectiveness of a partition based estimation facing a strongly interconnected net of subsystems.

The original contributions of this thesis are the following.
First we will implement the model-based partition-based fault detection algorithm proposed in [1] on two different large scale plants: the chemical system in [11], which consists in a distributed system where subsystems own strong relationships with their neighbouring system, and the power network system in [3] which instead shows the case where relationships between the subsystems are weaker. This is done with the purpose of highlighting the main advantages and limitations of the algorithm proposed in [1] in realistic case studies.
A new algorithm is then proposed that, similarly to [8], uses fault models (assuming that faults are persistent, when present). Both a centralized and a distributed

version of the algorithm are implemented. By means of a suitable filtering process, the value taken by the fault at each time instant is estimated. Such value is compared with some thresholds, in the distributed case those are also computed using an approach similar to the one used in [1] and [3]. This is done with the purpose of combining together the main advantages of the two considered approaches.

Chapter 2

Observer-based distributed fault detection

In this chapter the model-based fault detection method proposed in [1] is described along its main properties. Focus is placed on the distributed partition-based filter, on the computation of the residuals covariance matrices, and on two different fault detection testing methods: the one which tests single residuals (discussed in [3]) and the one testing moving window averages of the residuals (see [1]).

2.1 Distributed system model

The linear discrete-time model of a large scale system, characterized by the interconnection of M susbsystems is the following.

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k+1) + \sum_{j\neq i} A_{ij}x_j(k) + w_i(k) \\ y_i(k) &= C_i x_i(k) + v_i(k) \end{aligned} \tag{2.1}$$

The variable $x_i(k) \in \mathbb{R}^{n_i}$ is the state and $y_i(k) \in \mathbb{R}^{p_i}$ is the output of the subsystem. Signals $w_i(k) \in \mathbb{R}^{n_i}$ and $v_i(k) \in \mathbb{R}^{p_i}$ are zero mean white noises, for $i = 1, ..., M$, such that $\mathbb{E}[w_i(k)w_j^T(k)] = Q_i\delta_{ij}$, $\mathbb{E}[v_i(k)v_j^T(k)] = R_i\delta_{ij}$, $E[w_i(k)v_j^T(k)] = 0$ for all $i, j = 1, ..., M$ and $h, k \geq 0$. In the above notation δ_{ij} is the *Kronercker delta* function, i.e. $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.
Crucially important for our scope is the definition of the sets of the predecessors $\mathcal{N}_i = \{j | A_{ij} \neq 0\}$ and of the successors $\mathcal{S}_i = \{j | i \in \mathcal{N}\}$ of subsystem i. The definition of these sets allows for the graph-based representation of the large scale system aready introduced in the previous chapter, see e.g., Figure 1.5 and Figure 2.1 below.

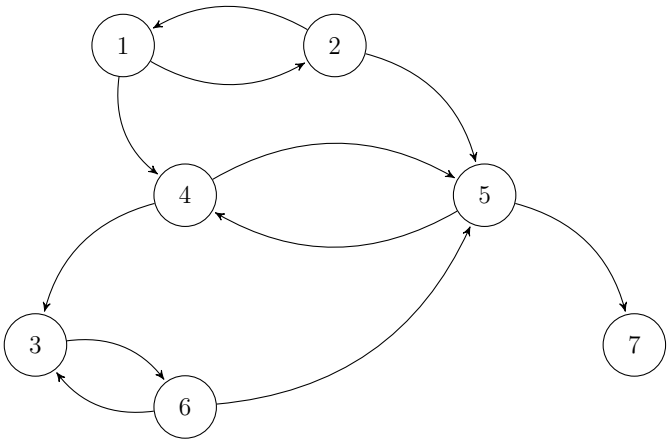

Figure 2.1: Example of a large scale system coupling graph.

Collecting the equations (2.1) for all $i =, ..., M$ we obtain the collective model:

$$\begin{aligned} x(k+1) &= Ax(k+1) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{2.2}$$

where $x^T(k) = [x_1^T(k) \dots x_M^T(k)]^T$, $y^T(k) = [y_1^T(k) \dots y_M^T(k)]^T$, $Q = diag(Q_1 \dots Q_M)$ and $R = diag(R_1 \dots R_M)$ are the covariance matrices of noises w and v respectively, $R = diag(C_1 \dots C_M)$ and

$$A = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MM} \end{bmatrix}$$

Remark. Often the mathematical model of an engineering system is originally given by a set of continuous time equations. Discrete-time models such as (2.1) and (2.2) can be obtained through a discretization process. However, commonly used discretization methods may impair the sparsity structure of the system.
Consider for example the chemical plant introduced in Section 1.4. Its graph representation shows the relationships between the subsystems with directed arrows. As introduced in the previous chapter, an arrow directed from subsystem i to subsystem j denotes a "structural" coupling term, meaning that variables of subsystem i directly affect the behaviour of subsystem j behaviour and not vice-versa. These coupling terms are directly reflected on the structure of matrix A, whose blocks elements A_{ji} are non zero only in presence of a direct influence of subsystem i on subsystem j (i.e., a directed arrow in the coupling graph from i to j).

$$A = \begin{pmatrix} A_{11} & 0 & A_{13} \\ A_{21} & A_{22} & 0 \\ 0 & A_{32} & A_{33} \end{pmatrix}$$

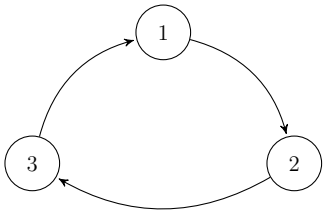

Figure 2.2: Sparse matrix A and related coupling graph. Diagonal blocks A_{ii} account for the internal structure of the i-th subsystem, non diagonal elements A_{ij}, $i \neq j$, are related to the relationships between the subsystems

To overcome the risk of loss in sparsity, which is of crucial interest in the development of distributed algorithms, a block-wise discretization such the one proposed [5] will be adopted in this thesis. This method, named *Mixed Euler-ZOH*, consists of integrating the state equation of each differential subsystem i by considering the state $x_j(t)$, for $j \neq i$, as constant during the interval. The diagonal elements (i.e., A_{ii}) are then equal to the exact discretization of each subsystem, while the sparsity structure of A is maintained (i.e., $A_{ji} \neq 0$ only if exists direct relation from subsystem i to subsystem j).

2.2 Distributed partition-based predictor

In this work, the state of the real system is estimated thanks to a Luenberger predictor. In the standard centralized case, its equations are the following:

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + L(y(k) - C\hat{x}(k)) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{2.3}$$

This predictor exploits the information given by the system outputs and estimates its states, while gain matrix L is computed in order to guarantee the Schur stability of matrix $(A - LC)$. In general, matrix L, if designed with classical tools (e.g., Kalman filter theory, pole placement), is full, i.e., does not have the same block structure of matrix A. The idea in this thesis is to face the filtering problem in a distributed and scalable way e.g., as proposed in [3] and [4]. The main cornerstones of such approaches are the following.

- **Distributed design**. Conditions on stability and convergence of the solution are enforced locally by each subsystem, exploiting only the pieces of information from the neighbouring system. Local computations allow a scalable evolution of the LSS, where plug-in and unplugging of subsystems do not require a whole system reconfiguration.

- **Distributed implementation**. As for the design, also the online implementation of the predictor is performed locally: each subsystem needs only information from the neighbours to work. This reduces the whole computational load, since the estimation process does not need a wide spread of information as in the purely centralized case.

The equations of the partition-based Luenberger filter proposed in [3] are the following:

$$\begin{aligned} \hat{x}_i(k+1) &= \sum_{j \in \mathcal{N}_i} \left(A_{ij}\hat{x}_j(k) + L_{ij}(y_j(k) - C_j\hat{x}_j(k)) \right) \\ \hat{y}_i(k) &= C_i\hat{x}_i(k) \end{aligned} \tag{2.4}$$

where $\hat{x}_i$ and $\hat{y}_i$ are the estimated state and the estimated output of subsystem i, respectively.
Gains L_{ij} are computed in order to guarantee the convergence of the solution and, to maintain sparsity, we set $L_{ij} \neq 0$ only if $A_{ij} \neq 0$. This is possible, for example, by designing the predictor as in [3] (i.e., distributed filter) or with linear matrix inequalities (LMI). Details about the latter approach are given in Section 2.2.2.

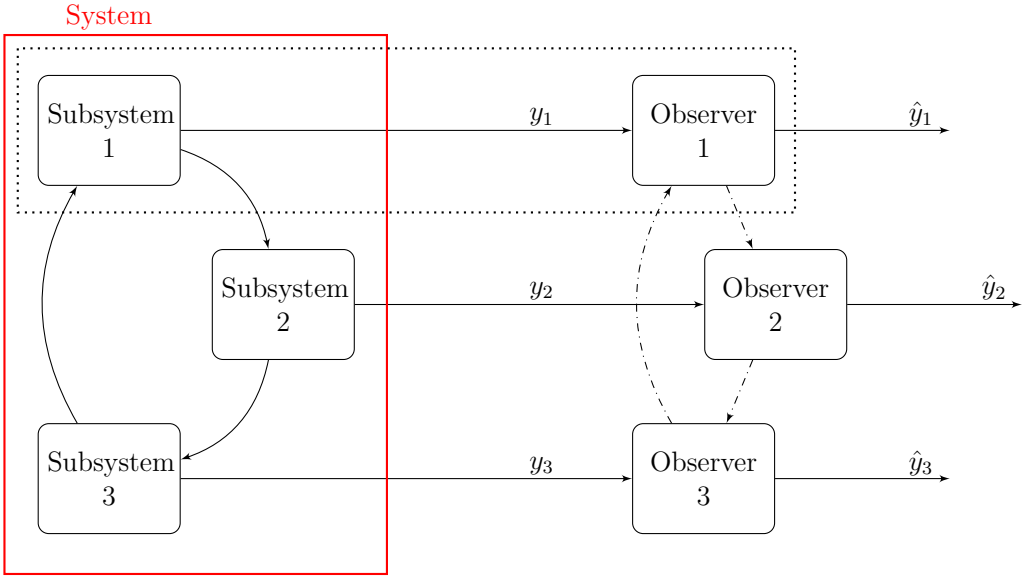

Figure 2.3: Example of distributed estimation scheme.

As in the picture, the i-subsystem (for all $i = 1, \ldots, M$) has a related local observer which estimates its own state and output vectors x_i and y_i. Each observer collects output measurements y_j, from the associated subsystem, and the estimated vectors $\hat{x}_j$, from the set of its predecessors $\mathcal{N}_i$, in order to perform the estimation. Simultaneously the estimated vector $\hat{x}_i$ is sent to the set of the successors $\mathcal{S}_i$ of each subsystem.

2.2.1 Estimation error

The local estimation error associated to the i-th local observer is defined as $e_i(k) = x_i(k) - \hat{x}_i(k)$ and its dynamics is given by

$$e_i(k+1) = \sum_{j \in N_i} \{(A_{ij} - L_{ij}C_j)e_j(k) - L_{ij}v_j(k)\} + w_i(k) \tag{2.5}$$

Considering the overall error equation:

$$e(k+1) = (A + LC)e(k) - Lv(k) + w(k) \tag{2.6}$$

where $e(k) = [e_1^T(k), \ldots, e_M^T(k)]^T$, L si the block matrix having the (i,j)-th element equal to L_{ij} for $i, j = 1, \ldots, M$, while A, C and noises vectors v and w, and their related covariance matrices Q and R, refer to the collective model presented in (2.2).
Defining $F = A - LC$, the covariance matrix of the collective estimation error vector e, i.e., $\Pi(k) := \mathbb{E}[e(k)e^T(k)]$, evolves according to the following rule:

$$\Pi(k+1) = F\Pi(k)F + LRL^T + Q \tag{2.7}$$

Since the subsystems are connected to each other, it is not possible to compute the variance of $e_i(k)$, namely $\Pi_{ii}(k)$, as well as its evolution over time, in a purely local and distributed way. In fact, the evolution of $\Pi(k)$, as well as that of $e_i(k)$, can be computed only in a centralized way. However, in [3], a method has been proposed, which allows to to compute, in a distributed fashion, a block diagonal upper bound to $\Pi(k)$, at each time instant k. We name as $B_i(k)$ this bound associated to the i-th subsystem only (i.e., $B_i(k)$ is an upper bound to $\Pi_{ii}(k)$, covariance matrix of the i-th subsystem). The evolution of $B_i(k)$ is obtained according to the following rule.

$$B_i(k+1) = Q_i + \sum_{j \in \mathcal{N}_i} [(\tilde{A}_{ij} - L_{ij}\tilde{C}_j)B_i(k)(\tilde{A}_{ij} - L_{ij}\tilde{C}_j)^T + L_{ij}\tilde{R}_j L_{ij}^T] \tag{2.8}$$

where $\tilde{A}_{ij} = \sqrt{\zeta_i}A_{ij}$, $\tilde{C}_i = \sqrt{\zeta_i}C_i$, $\tilde{R}_i = \sqrt{\zeta_i}R_i$ and $\zeta_i = |\mathcal{S}_i|$ for all $i, j = 1, ..., M$.
$B_i(k)$ will be used in the following sections to compute reliable bounds of the residuals variance.

2.2.2 Distributed partition-based predictor design through Linear Matrix Inequalities

In this subsection we show how to properly design the Luenberger gain matrix L using linear matrix inequalities (LMI).

The problem consists of computing a structured matrix L, not only such that matrix $F = A - LC$ is Schur stable (conferring stability to the error dynamics (2.6)), but also by allowing the recursive set of equations (2.8) to converge to a steady state solution. To provide that both are verified, in [3] the following result is proved.

Theorem 1. *Define*

$$\mathbb{F} = \tilde{F} \odot \tilde{F} = \begin{bmatrix} \tilde{F}_{11} \otimes \tilde{F}_{11} & \dots & \tilde{F}_{1M} \otimes \tilde{F}_{1M} \\ \vdots & \ddots & \vdots \\ \tilde{F}_{M1} \otimes \tilde{F}_{M1} & \dots & \tilde{F}_{MM} \otimes \tilde{F}_{MM} \end{bmatrix} \tag{2.9}$$

where $\odot$ denotes the Khatri-Rao product, while $\otimes$ denotes the Kroneker product. Also $\tilde{F} = (\tilde{A} - L\tilde{C})$ and $\tilde{F}_{ij} = (\tilde{A}_{ij} - L\tilde{C}_j)$ where $\tilde{A}_{ij} = \sqrt{\zeta_i}A_{ij}$, $\tilde{C}_i = \sqrt{\zeta_i}C_i$, $\tilde{R}_i = \sqrt{\zeta_i}R_i$ and $\zeta_i = |\mathcal{S}_i|$ for all $i, j = 1, ..., M$.

If matrix $\mathbb{F}$ is Schur stable, then

(i) There exists, for all $i = 1, \dots, M$ a matrix $\bar{B}_i \geq 0$, independent of the initial conditions of (2.8), such that $B_i(k) \to \bar{B}_i$ as $k \to +\infty$;

(ii) $A - LC$ is Schur stable.

However, it has not been possible to use the condition provided by *Theorem 1* for LMI-based design of the gain matrix L. For this reason, in this thesis, the adopted procedure is the following:

1. Design L structured in such a way that

   $$\tilde{F} = (\tilde{A} - L\tilde{C})$$

   is Schur stable;

2. Check if $\mathbb{F}$ is also Schur stable.

To solve *Step 1* we rely on the following, well known, result:

Theorem 2. *$\tilde{F}$ is Schur stable in and only if $\exists$ symmetric matrix $P > 0$ such that*

$$\tilde{F}^T P \tilde{F} - P < 0 \tag{2.10}$$

Where given a generic symmetric matrix Z, the condition $Z > 0$ means that matrix Z is positive definite, while $Z < 0$ means that Z is negative definite.

Note that (2.10) can be expanded as follows

$$\tilde{F}^T P \tilde{F} - P < 0 \tag{2.11a}$$
$$(\tilde{A} - L\tilde{C})^T P (\tilde{A} - L\tilde{C}) - P < 0 \tag{2.11b}$$
$$\tilde{A}^T P \tilde{A} - \tilde{C}^T L^T P \tilde{A} - \tilde{A}^T P L \tilde{C} + \tilde{C}^T L^T P L \tilde{C} - P < 0 \tag{2.11c}$$

Define $K = PL$, in this way we rewrite (2.11c) as

$$\tilde{A}^T P P^{-1} P \tilde{A} + \tilde{C}^T K^T P^{-1} K \tilde{C} - P - \tilde{C}^T K^T \tilde{A} - \tilde{A}^T K \tilde{C} < 0 \tag{2.11d}$$

In turn (2.11d) is equivalent to

$$-(P + \tilde{C}^T K^T \tilde{A} + \tilde{A}^T K \tilde{C}) + \begin{bmatrix} \tilde{A}^T P & \tilde{C}^T K^T \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix} \begin{bmatrix} P\tilde{A} \\ K\tilde{C} \end{bmatrix} < 0 \tag{2.11e}$$

Resorting to the Schur complement, (2.11e) reduces to the following LMI

$$\begin{bmatrix} P + \tilde{C}^T K^T \tilde{A} + \tilde{A}^T K \tilde{C} & \tilde{A}^T P & \tilde{C}^T K^T \\ P\tilde{A} & P & 0 \\ K\tilde{C} & 0 & P \end{bmatrix} > 0 \tag{2.12}$$

As a solution to the LMI (feasibility) problem defined by (2.12) and $P > 0$ is available, we can compute

$$L = P^{-1} K$$

Note that, however, in order to impose the proper sparsity block structure to matrix L, the following additive constraints must be imposed on the matrices P and K:

(a) P is block diagonal, i.e.,

$$P = diag(P_1, \ldots, P_M)$$

where $P_i \in \mathbb{R}^{n_i \times n_i}$;

(b) K has the same block structure of A, i.e.,

$$K = \begin{bmatrix} K_{11} & \ldots & K_{1M} \\ \vdots & \ddots & \vdots \\ K_{M1} & \ldots & K_{MM} \end{bmatrix} \tag{2.13}$$

where $K_{ij} \in \mathbb{R}^{n_i \times p_j}$, $\forall i, j$, and $K_{ij} \neq 0$ only if $A_{ij} \neq 0$

Note that this procedure clearly does not guarantee to find a gain matrix L guaranteeing both the stability of the error dynamics, and the convergence of equations (2.8). In our examples it has been sufficient to design a structured L matrix solving a feasibility problem for *Theorem 2*. The obtained L could already guarantee $\mathbb{F}$ Schur stability.
Further studies may be devoted to establish a relationship between the Schur stability of $\tilde{F}$ and the one of $\mathbb{F}$.

2.3 Single residual testing

2.3.1 Main rationale of the algorithm

The Fault detection algorithm here considered is based on the analysis of the residuals

$$r_i(k) = y_i(k) - \hat{y}_i(k) \tag{2.14}$$

The residual $r_i(k)$ represents the deviation of the estimated output with respect to the actual one, updated to the k-th time instant. Residuals can be computed locally using data generated by the state estimator and the subsystem outputs.
In normal working plant conditions, $r_i(k)$ behaves as a stochastic signal with zero mean and defined variance, as shown for example in Fig. 2.4.

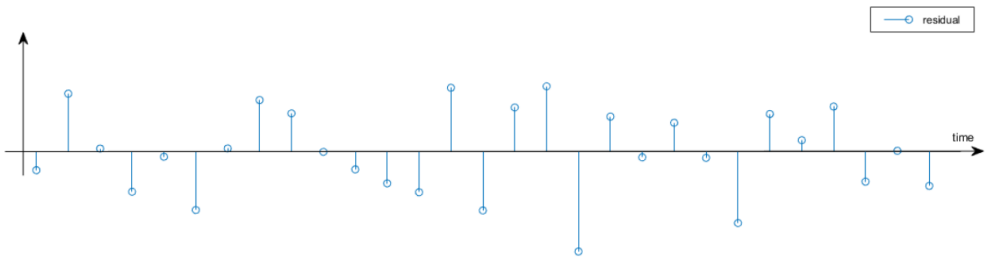

Figure 2.4: Residual behaviour in standard plant working conditions

When a fault affects the plant, it will corrupt also the associated residuals behaviour (see e.g., Fig. 2.5),since the observer model is no more equal to the real one. It is then reasonable to compute suitable thresholds embodying the standard trends of $r_i(k)$, and declare fault when a residual exceed its associated bound.

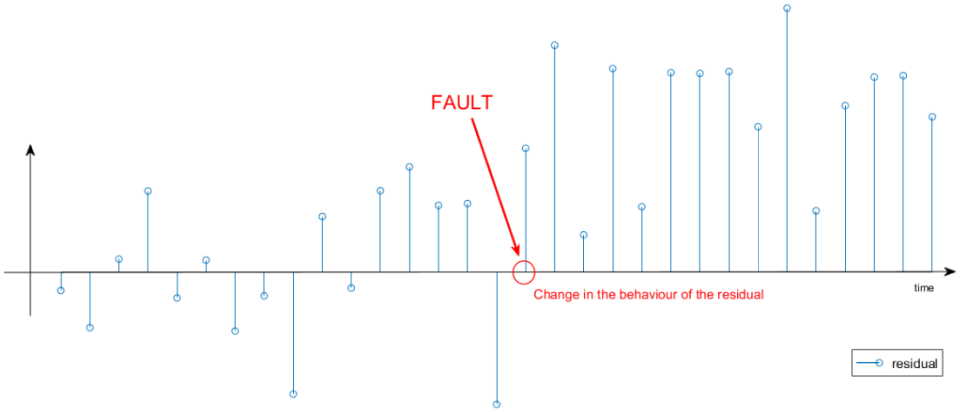

Figure 2.5: Residual behaviour when the plant is affected by a fault

The general idea of the fault detection algorithm in [1] is to compute these thresholds using the covariance of $r_i(k)$, i.e.,

$$\Sigma_i(k) = \mathbb{E}[r_i(k) r_i(k)^T]$$

Defining with $r_{i,l}(k)$, the l-th entry of $r_i(k)$, we have that $\sigma^2_{i,l}$, the l-th diagonal entry of $\Sigma_i(k)$ is the variance of $r_{i,l}(k)$. Relying on the Gaussianity of $r_{i,l}(k)$ (which follows from the Gaussianity of noises $w_j(k)$ and $v_j(k)$, $j = 1, ..., M$ and from the linearity of both state and observer equations, (2.1) and (2.4), respectively), for any $p \in (0, 1]$, we can define a scalar $\alpha \geq 0$ such that

$$P(|r_{i,j}(k)|/\sigma_{i,l}(k) \geq \alpha) = p \qquad i.e \quad \int_{-\alpha}^{\alpha} f(x)dx = 1 - p$$

where f is the probability distribution of a zero mean Gaussian variable with unitary variance. It follows that $|r_{i,l}(k)| > \rho_{i,l}(k) = \alpha\sigma_{i,l}(k)$ with probability p in nominal (i.e, non-faulty) conditions. This criterion is used to detect faults, and the following rule is applied at any time instant.

$$\begin{cases} \text{if } |r_{i,l}(k)| < \rho_{i,l}(k), \text{then no fault is detected} \\ \text{otherwise fault is detected} \end{cases}$$

Is therefore clear that p represent the FA (false alarm) rate and will be denoted with p_{FA}.

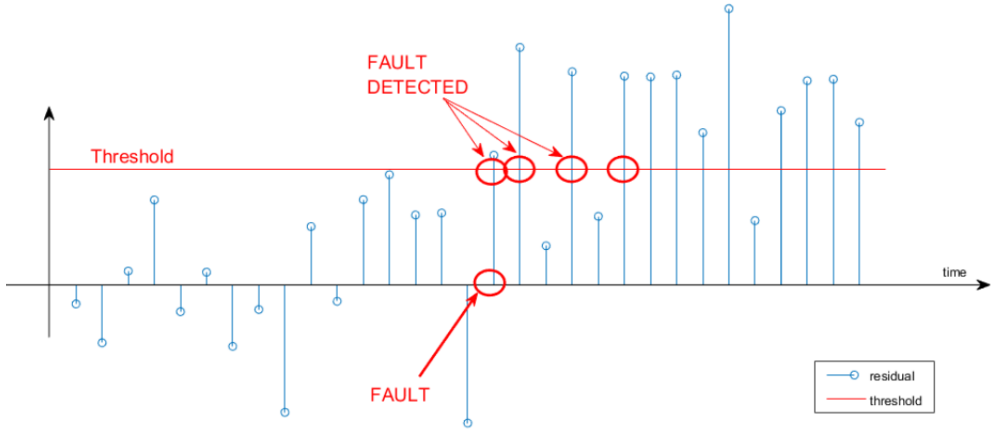

Figure 2.6: Residual fault detection test

2.3.2 Distributed threshold computation, approximation of the residual variance

Given system (2.1), the predictor equations (2.4), and the estimation error evolution over time (2.5), it is possible to compute the residual vector of the i-th subsystem $r_i(k)$ and its variance $\Sigma_i(k)$. Respectively:

$$r_i(k) = C_i e_i(k) + v_i(k) \tag{2.15}$$

$$\Sigma_i(k) = C_i \Pi_{ii}(k) C_i^T + R_i \tag{2.16}$$

where $\Pi_{ii}(k) = \mathbb{E}[e_i(k)e_i(k)^T]$ is the variance of the estimation error treated in the previous Section 2.2.1.
Since, as we already explained, Π_{ii} cannot be computed in a purely local way, also the real value of $\Sigma_i(k)$ needs to be determined considering the overall scheme. However, in view of the linear relationship between the residual $r_i(k)$ and the state estimation error e_i, in [1] the local upper bound introduced in (2.8) is used to compute an analytical upper bound to $\Sigma_i(k)$.

$$\Sigma_i^B(k) = C_i B_i(k) C_i^T + R_i \tag{2.17}$$

The l-th diagonal element $(\sigma_{i,l}^B)^2$ of matrix $\Sigma_i^B(k)$ is used in place of the actual variance $\sigma_{i,l}^2(k)$ of the residual $r_{i,l}(k)$ in the computation of threshold $\rho_{i,l}(k)$. Once given a desired false alarm rate ρ, we have

$$\rho_{i,l}(k) = \alpha \sigma_{i,l}^B \tag{2.18}$$

Note that the thresholds computed for a given p_{FA} (i.e., to obtain a given false alarm rate) are upper bounds to the tight thresholds values, since upper bounds to the actual residual variances are used. In this way, in non faulty conditions, $|r_{i,l}(k)| < \rho_{i,l}(k)$ with probability smaller than p_{FA}, meaning smaller probability to have a false alarm, but at the same time a smaller probability to detect faults.

2.4 Moving window average of residuals testing

As discussed in [1], the choice of a single residual (i.e., one at each time step) may lead to unsatisfactory performances facing small-amplitude faults. Considering, on the other hand, the mean value of a number n of subsequent residuals should give us more opportunity to detect small faults, while maintaining a good FA rate. This approach, proposed in [1], exploits the analytical upper bound $\boldsymbol{B}(k) = diag(B_1(k), ..., B_M(k))$ derived above to define a proper upper bound to the covariance of a mean of residuals. At each time step the average of the residual values computed over a sliding window is tested. Specifically, we define

$$\bar{r}_{i,l}^{(m)}(k) = \frac{1}{m}\sum_{j=0}^{m-1} r_{i,l}(k-j) \tag{2.19}$$

which is a Gaussian variable with zero mean and variance $(\bar{\sigma}_{i,l}^{(m)}(k))^2 = \mathbb{E}[(\bar{r}_{i,l}^{(m)}(k))^2]$, in view of the assumpions introduced in the previous sections. Considering then $\bar{r}_{i,l}^{(m)}(k)$ as a single observation is then easy to derive a FD test based on thresholds crossing as before.

$$\begin{cases} \text{if } |\bar{r}_{i,l}^{(m)}(k)| < \bar{\rho}_{i,l}^{(m)}(k), \text{then no fault is detected} \\ \text{otherwise fault is detected} \end{cases}$$

where $\bar{\rho}_{i,l}^{(m)}(k) = \alpha(\sigma_{i,l}^{(m)})(k)$. The choice of the average of m values rather than a single one is justified by the fact that $\bar{\sigma}_{i,l}^{(m)}(k) \leq \sigma_{i,l}(k)$. This means, that in principle, we can obtain the same false alarm rate with a smaller threshold which should allow the detection of smaller-amplitude faults.

In order to compute an analytical upper bound to $\bar{\sigma}_{i,l}^{(m)}(k)$, we exploit the one computed for the single residual case introduced in Section 2.3.2, i.e., $\Sigma_i^B(k)$. Denote with $(\sigma_{i,l}^B(k))^2$ the l-th diagonal entry of matrix $\Sigma_i^B(k)$. Considering that

$$(\bar{\sigma}_{i,l}^{(m)}(k))^2 = \frac{1}{m^2}\sum_{j,h=0}^{m-1} \gamma_{i,l}(k-j, k-h) \tag{2.20}$$

where $\gamma_{i,l}(k-j,k-h) = \mathbb{E}[r_{i,l}(k-j)r_{i,l}(k-h)]$. A reliable and conservative upper bound to $\gamma_{i,l}(k-j,k-h)$ for all $j,h=0,...,m$ is provided by the following proposition.

Proposition 1. It holds that both $\gamma_{i,l}(k-j,k-h) \leq \gamma_{i,l}^{B,1}(k-j,k-h)$ and for $j \neq h$, $\gamma_{i,l}(k-j,k-h) \leq \gamma_{i,l}^{B,2}(k-j,k-h)$, where

$$\bar{\gamma}_{i,l}^{B,1}(k-j,k-h) = \frac{1}{2}((\sigma_{i,l}^{B}(k-j))^2 + (\sigma_{i,l}^{B}(k-h))^2) \tag{2.21a}$$

$$\begin{aligned}\bar{\gamma}_{i,l}^{B,2}(k-j,k-h) = \frac{1}{2}\,\|C_{i,l}\|\,\|diag(\boldsymbol{B}(k-max(h,j)))\\ + C_{i,l}B_i(k-max(h,j))C_{i,l}^T\mathbb{1}_n\|\mu\lambda^{|h-j|} + \|C_{i,l}\|\,\left\|L_i^c R_{i,l}^c\right\|\mu\lambda^{|h-j|-1}\end{aligned} \tag{2.21b}$$

where L_i^c is the i-th block column of L, $R_{i,l}^c$ is the l-th column of matrix R_i and $C_{i,l}$ is the l-th row of matrix C_i. Scalars μ and $\lambda \in [0,1)$ are defined in such a way that $\|F^j\| \leq \mu\lambda^j$ for all $j > 0$. In view of *Proposition 1* the upper bound to $(\sigma_{i,l}^{(m)}(k))^2$ is defined as:

$$(\sigma_{i,l}^{(m),B}(k))^2 = \frac{1}{m^2}\sum_{j,h=0}^{m-1}\bar{\gamma}_{i,l}^{B}(k-j,k-h) \tag{2.22}$$

if $j=h$, $\bar{\gamma}_{i,l}^{B}(k-j,k-h) = \bar{\gamma}_{i,l}^{B,1}(k-j,k-h)$, else if $j \neq h$

$$\bar{\gamma}_{i,l}^{B}(k-j,k-h) = min\{\bar{\gamma}_{i,l}^{B,1}(k-j,k-h); \bar{\gamma}_{i,l}^{B,2}(k-j,k-h)\} \tag{2.23}$$

The term (2.23) allows to play with the tradeoff between guaranteed false alarm and missed detection rates. The obtained approximation also exploits the stationarity of $r_{i,l}(k)$: since the correlation between $r_{i,l}(k-j)$ and $r_{i,l}(k)$ asymptotically tends to zero, the contributions of "old" (i.e., for higher values of j) measurements become negligible in the computation of $(\sigma_{i,l}^{(m),B}(k))^2$.

Chapter 3

Application of distributed fault detection algorithms to selected case studies

In this chapter we show some significant simulation results obtained applying the model-based distributed fault detection algorithm described in Section 2 on two selected case studies: the chemical plant described in [11] and the power network system introduced in [3].

3.1 Chemical plant case study

The first large scale system considered in this chapter is inspired by the chemical plant used for simulation studies in [11] and sketched in Figure 3.1 .

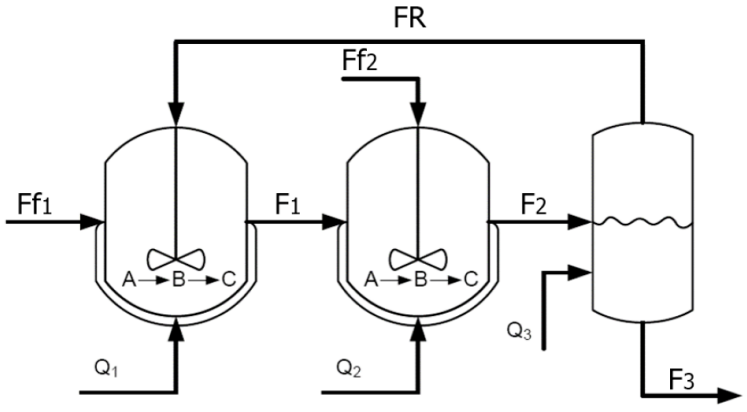

Figure 3.1: Chemical system scheme from [11]

The plant is composed of two reactors and a separator. The reactant liquid A feeds the two reactors, where it is converted into the products B and C. There is also a recirculation flow from the separator to the first reactor that makes the system strongly coupled.
The centralized mathematical model of the plant is composed of twelve non linear continuous time differential state equations, which account for the levels of the liquid inside the reactors and the separators, H_i, their temperature, T_i, and the concentrations of chemical products A, x_{Ai} and B, x_{Bi}, with $i = 1, 2, 3$.
The non linear model of the plant is

$$
\begin{aligned}
\dot{H}_1 &= \frac{1}{\rho A_1}(F_{f1} + F_R - F_1) \\
\dot{x}_{A1} &= \frac{1}{\rho A_1 H_1}(F_{f1}x_{A0} + F_R x_{AR} - F_1 x_{A1}) - k_{A1}x_{A1} \\
\dot{x}_{B1} &= \frac{1}{\rho A_1 H_1}(F_R x_{BR} - F_1 x_{B1}) + k_{A1}x_{A1} - k_{B1}x_{B1} \\
\dot{T}_1 &= \frac{1}{\rho A_1 H_1}(F_{f1}T_0 + F_R T_R - F_1 T_1) \\
&\quad - \frac{1}{C_p}(K_{A1}x_{A1}\Delta H_A + k_{B1}x_{B1}\Delta H_B) + \frac{Q_1}{\rho A_1 C_p H_1} \\
\dot{H}_2 &= \frac{1}{\rho A_2}(F_{f2} + F_1 - F_2) \\
\dot{x}_{A2} &= \frac{1}{\rho A_2 H_2}(F_{f2}x_{A0} + F_1 x_{A1} - F_2 x_{A2}) - k_{A2}x_{A2} \\
\dot{x}_{B2} &= \frac{1}{\rho A_2 H_2}(F_1 x_{B1} - F_2 x_{B2}) + k_{A2}x_{A2} - k_{B2}x_{B2} \\
\dot{T}_2 &= \frac{1}{\rho A_2 H_2}(F_{f2}T_0 + F_1 T_1 - F_2 T_2) \\
&\quad - \frac{1}{C_p}(K_{A2}x_{A2}\Delta H_A + k_{B2}x_{B2}\Delta H_B) + \frac{Q_2}{\rho A_2 C_p H_2} \\
\dot{H}_3 &= \frac{1}{\rho A_3}(F_2 - F_D - F_R - F_3) \\
\dot{x}_{A3} &= \frac{1}{\rho A_3 H_3}(F_2 x_{A2}(F_D + F_R)x_{AR} - F_3 x_{A3}) \\
\dot{x}_{B3} &= \frac{1}{\rho A_3 H_3}(F_2 x_{A2}(F_D + F_R)x_{BR} - F_3 x_{B3}) \\
\dot{T}_3 &= \frac{1}{\rho A_3 H_3}(F_2 T_2 - (F_D + F_R)T_R - F_3 T_3) + \frac{Q_3}{\rho A_3 C_p H_3}
\end{aligned}
\tag{3.1}
$$

where, for $i = 1, 2, 3$

$$F_i = k_{vi} H_i \qquad k_{Ai} = k_A \exp\left(-\frac{E_A}{RT_i}\right) \qquad k_{Bi} = k_B \exp\left(-\frac{E_B}{RT_i}\right)$$

Also,

$$F_D = 0.01 F_R \qquad x_{AR} = \frac{\alpha_A x_{A3}}{\bar{x}_3} \qquad x_{BR} = \frac{\alpha_B x_{B3}}{\bar{x}_3}$$
$$\bar{x}_3 = \alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3} \qquad x_{C3} = (1 - x_{a3} - x_{B3})$$

The state vector collecting the twelve state variables is the following

$$x = \begin{bmatrix} H_1 & x_{A1} & x_{B1} & T_1 & H_2 & x_{A2} & x_{B2} & T_2 & H_3 & x_{A3} & x_{B3} & T_3 \end{bmatrix}^T \tag{3.2}$$

The input vector u is composed of the flowrates feeding the reactors and the separator, and of the supplied heat.

$$u = \begin{bmatrix} F_{f1} & Q_1 & F_{f2} & Q_2 & F_R & Q_3 \end{bmatrix}^T \tag{3.3}$$

The output vector of the original system in [11] corresponds with the state (i.e., $y = x$). Since our main goal is to test the estimation capabilities of the proposed algorithm, we decided to reduce the number of the available (i.e., measured) outputs. The considered output vector is

$$y = \begin{bmatrix} H_1 & T_1 & H_2 & T_2 & H_3 & x_{A3} & x_{B3} & T_3 \end{bmatrix}^T \tag{3.4}$$

The system has been linearized around the equilibrium working point defined by the constant input vector $\bar{u} = \begin{bmatrix} 8.33 & 10 & 0.5 & 10 & 66.2 & 10 \end{bmatrix}^T$. Subsequently, the system has been discretized, considering a sampling time of $0.1s$, by means of the method in [5] which, as already mentioned, maintains the structure of the system matrices.
The standard centralized, linearized and discretized equations, fed by the equilibrium input vector $\bar{u}$, are then in the following form

$$\begin{cases} x(k+1) = Ax(k) + B\bar{u} + v(k) \\ y(k) = Cx(k) + w(k) \end{cases} \tag{3.5}$$

where $A \in \mathbb{R}^{12\times12}$ and $C \in \mathbb{R}^{8\times8}$ are the centralized, linearized and discretized system matrices, $B \in \mathbb{R}^{12\times6}$ is the collective matrix related to inputs $\bar{u}$, while

$v(k)$ and $w(k)$ are zero mean white noises applied to the state and to the output, with covariance matrices $Q \in \mathbb{R}^{12\times12}$ and $R \in \mathbb{R}^{8\times8}$ respectively. The matrix Q has been chosen considering diagonal entries $\sigma^2_{w,ii} = 5 \times 10^{-2}\bar{x}_i$, for $i = 1, \ldots, 12$, where $\bar{x}_i$ is the value of the related state in nominal conditions. The same reasoning has been done for the diagonal entries $\sigma^2_{v,ii}$ of matrix R.

In the Table 3.1 the steady state and parameters' values are reported.

Parameter	Value	Units	Parameter	Value	Units
H_1	29.8	m	A_1	3	m^2
x_{A1}	0.542	wt(%)	A_2	3	m^2
x_{B1}	0.393	wt(%)	A_3	1	m^2
T_1	315	K	ρ	0.15	kg/m^2
H_2	30	m	C_p	25	$kJ/kg\,K$
x_{A2}	0.503	wt(%)	k_{v1}	2.5	$kg/m\,s$
x_{B2}	0.421	wt(%)	k_{v2}	2.5	$kg/m\,s$
T_2	315	K	k_{v3}	2.5	$kg/m\,s$
H_3	3.27	m	x_{A0}	1	wt(%)
x_{A3}	0.238	wt(%)	T_0	313	K
x_{B3}	0.570	wt(%)	k_A	0.02	$1/s$
T_3	315	K	k_B	0.018	$1/s$
F_{f1}	8.33	kg/s	E_A/R	-100	K
$Q1$	10	kJ/s	E_B/R	-150	K
F_{f2}	0.5	kg/s	ΔH_A	-40	kJ/kg
$Q2$	10	kJ/s	ΔH_B	-50	kJ/kg
F_R	66.2	kg/s	α_A	3.5	
$Q3$	10	kJ/s	α_B	1.1	
			α_C	0.5	

Table 3.1: Table of the parameters of the chemical plant in [11]

In view of the structure of the system (see e.g., the graph structure in Fig. 1.5) and having discretized the system by means of the method in [5], A, B and C have the following structure:

$$A = \begin{bmatrix} A_{11} & 0 & A_{13} \\ A_{12} & A_{22} & 0 \\ 0 & A_{32} & A_{33} \end{bmatrix} \qquad B = \begin{bmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{bmatrix} \qquad C = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix}$$

Figure 3.2: Structure of collective matrices A, B and C. $A_{ij} \in \mathbb{R}^{4\times4}$, $\forall i, j = 1, 2, 3$, $B_i \in \mathbb{R}^{4\times2}$, $\forall i, j = 1, 2, 3$, while $C_1, C_2 \in \mathbb{R}^{2\times4}$ and $C_3 \in \mathbb{R}^{4\times4}$

This particular structure of the system matrices allows to partition the discrete time system. In particular, we define

$$x_1 = \begin{bmatrix} H_1 \\ x_{A1} \\ x_{B1} \\ T_1 \end{bmatrix} \quad x_2 = \begin{bmatrix} H_2 \\ x_{A2} \\ x_{B2} \\ T_2 \end{bmatrix} \quad x_3 = \begin{bmatrix} H_3 \\ x_{A3} \\ x_{B3} \\ T_3 \end{bmatrix}$$

$$u_1 = \begin{bmatrix} F_{f1} \\ Q_1 \end{bmatrix} \quad u_2 = \begin{bmatrix} F_{f2} \\ Q_2 \end{bmatrix} \quad u_3 = \begin{bmatrix} F_R \\ Q_3 \end{bmatrix} \tag{3.6}$$

$$y_1 = \begin{bmatrix} H_1 \\ T_1 \end{bmatrix} \quad y_2 = \begin{bmatrix} H_2 \\ T_2 \end{bmatrix} \quad y_3 = \begin{bmatrix} H_3 \\ x_{A3} \\ x_{B3} \\ T_3 \end{bmatrix}$$

which are the partitioned system state, input and output vectors, respectively.
Also, the noise vectors $v(k)$ and $w(k)$ can be decomposed into three subvectors each acting on a single subsystem (i.e., $v(k)^T = \begin{bmatrix} v_1^T(k) & v_2^T(k) & v_3^T(k) \end{bmatrix}$ and $w(k)^T = \begin{bmatrix} w_1^T(k) & w_2^T(k) & w_3^T(k) \end{bmatrix}$ respectively) whose dimensions are equal to the ones of the related state or input.
The submodels resulting from the partition process are the following.

$$\begin{cases} x_1(k+1) = A_{11}x_1(k) + A_{13}x_3(k) + B_1\bar{u}_1 + v_1(k) \\ \quad y_1(k) = C_1x_1(k) + w_1(k) \\ \\ x_2(k+1) = A_{22}x_2(k) + A_{21}x_1(k) + B_2\bar{u}_2 + v_2(k) \\ \quad y_2(k) = C_2x_2(k) + w_2(k) \\ \\ x_3(k+1) = A_{33}x_3(k) + A_{32}x_2(k) + B_3\bar{u}_3 + v_3(k) \\ \quad y_3(k) = C_3x_3(k) + w_3(k) \end{cases} \tag{3.7}$$

3.1.1 Distributed predictor

The LMI-based design method discussed in Section 2.2.2 is used to compute the Luenberger observer gains L_{ij} which guarantee Schur stability of matrix $F = (A - LC)$ and to confer a proper structure to the collective matrix L equal to the one of A, i.e., as in Fig.3.2. The resulting distributed Luenberger predictor has

equations

$$\begin{cases} \hat{x}_1(k+1) = A_{11}\hat{x}_1(k) + L_{11}(y_1(k) - C_1\hat{x}_1(k)) \\ \qquad\qquad + A_{13}\hat{x}_3(k) + L_{13}(y_3(k) - C_3\hat{x}_3(k)) + B_1\bar{u}_1 \\ \hat{x}_2(k+1) = A_{22}\hat{x}_2(k) + L_{22}(y_2(k) - C_2\hat{x}_2(k)) \\ \qquad\qquad + A_{21}\hat{x}_1(k) + L_{21}(y_1(k) - C_1\hat{x}_1(k)) + B_2\bar{u}_2 \\ \hat{x}_3(k+1) = A_{33}\hat{x}_3(k) + L_{33}(y_3(k) - C_3\hat{x}_3(k)) \\ \qquad\qquad + A_{32}\hat{x}_2(k) + L_{32}(y_2(k) - C_2\hat{x}_2(k)) + B_3\bar{u}_3 \end{cases} \tag{3.8}$$

3.1.2 Threshold computation

Once the gain matrices are computed, the ingredients needed to compute the analytical upper bounds to the residuals variance $(\sigma^B_{i,l})^2$, introduced in equation (2.8) are available. In Table 3.2 those analytical bounds are compared with the corresponding empirical variance of the residuals (calculated over 10000 samples), in case of single residual analysis (i.e., $m = 1$).

Single Residual $m = 1$		$(\sigma^B_{i,l})^2$	$(\sigma_{EMP,i})^2$
Subsystem 1	r_{H_1}	0.0200	0.0199
	r_{T_1}	0.8217	0.5164
Subsystem 2	r_{H_2}	0.0266	0.0223
	r_{T_2}	1.1515	0.6575
Subsystem 3	r_{H_3}	0.0120	0.0064
	$r_{x_{A_3}}$	0.0508	0.0179
	$r_{x_{B_3}}$	0.0004	0.0002
	r_{T_3}	91.1323	32.5589

Table 3.2: Comparison between analytical values of the residual variance $(\sigma^B_{i,l})^2$ and the empirical value $\sigma_{EMP,i}$ for the single residual

In Tables 3.3 and 3.4 the same values are shown, considering moving sliding windows with dimensions $m = 20$ and $m = 40$, respectively.

Sliding window average, $m = 20$	$10^{-2}\times$	$(\sigma_{i,l}^{(20),B})^2$	$(\sigma_{EMP,i}(20))^2$
Subsystem 1	r_{H_1}	1.8658	0.1184
	r_{T_1}	64.8809	2.2606
Subsystem 2	r_{H_2}	2.4532	0.1416
	r_{T_2}	88.8313	2.9365
Subsystem 3	r_{H_3}	1.1325	0.0470
	$r_{x_{A_3}}$	4.5997	0.1774
	$r_{x_{B_3}}$	0.0375	0.0012
	r_{T_3}	5388.3516	106.1951

Table 3.3: Comparison between the analytical upper bound to the variance of the mean over a moving sliding window of 20 residuals $(\sigma_{i,l}^{(20),B})^2$ and the empirical value $(\sigma_{EMP,i}^{(20)})^2$

Sliding window average, $m = 40$	$10^{-2}\times$	$(\sigma_{i,l}^{(40),B})^2$	$(\sigma_{EMP,i}(40))^2$
Subsystem 1	r_{H_1}	1.2130	0.0580
	r_{T_1}	38.5784	1.0787
Subsystem 2	r_{H_2}	1.5813	0.0679
	r_{T_2}	52.4205	1.4119
Subsystem 3	r_{H_3}	0.07459	0.0229
	$r_{x_{A_3}}$	2.9184	0.0927
	$r_{x_{B_3}}$	0.0275	0.0006
	r_{T_3}	2998.9128	48.5497

Table 3.4: Comparison between the analytical upper bound to the variance of the mean over a moving sliding window of 40 residuals $(\sigma_{i,l}^{(40),B})^2$ and the empirical value $(\sigma_{EMP,i}^{(40)})^2$

3.1.3 Simulation results

In this section the simulation results obtained with the chemical plant model are reported. Persistent additive faults of different amplitudes occur at time step $k = 100$. Small faults correspond to $\sim 1\%$ of the value taken, in normal working conditions, by the state variable directly affected by the fault. Medium faults correspond instead to $\sim 2.5\%$ of the same value. Faults act directly on the 4^{th} state equation of the second subsystem, which represent the temperature of the second chemical reactor.

For each proposed case we display two types of diagrams. Figures 3.3, 3.5, 3.7, 3.9, 3.11, 3.13 show, for each experiment type, the rate of plants where a fault is detected for each time instant k. The corresponding rates are shown for each subsystem output residual and for each subsystem. These results are obtained considering the analytical threshold only and 1000 Montecarlo runs.

The plots in Figures 3.4, 3.6, 3.8, 3.10, 3.12, 3.14, instead, show the values taken by the residual (or the mean of residuals over sliding windows) at each simulation time instant, compared to the corresponding thresholds, both analytical and empirical. In this case 30 Montecarlo runs only were considered. Three different residual testing scenarios are taken into account: single residual testing and moving window averages on windows of lengths $m = 20$ and $m = 40$.

In the computation of the thresholds, the false alarm (FA) rate has been set to 0.0002, i.e, $\alpha = 3.7190$. The related thresholds are indeed the values $\rho_{i,l}^{(m)}$, where i is the index of the substystem, l is the index related to the state of the subsystem, while m is the length of the sliding window (omitted when considering the single residual). These values are compared with the corresponding empirical values $\rho_{EMP,\,i,l}^{(m)}$, which were obtained simulating the system in standard steady-state conditions for 10000 time instants.

While the results obtained with medium and large faults show very good detection capabilities in all the considered cases, the results obtained with the small additive fault are of particular interest and somehow counterintuitive. Remarkably, from Figure 3.3 we notice that the detection capabilities of our scheme are good when we test single residuals but, when we test averages of $m = 20$ residuals, these capabilities decrease, see Figure 3.5. Interestingly, on the other hand, when $m = 40$, the algorithm performances are very good, although with a slight delay. This peculiar behaviour will be discussed in Section 3.3.

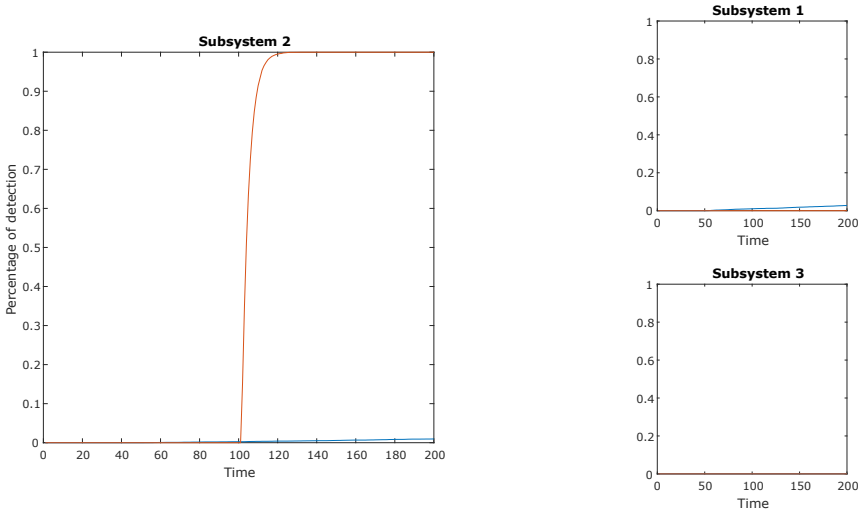

Figure 3.3: Small fault affecting subsystem 2, single residual testing

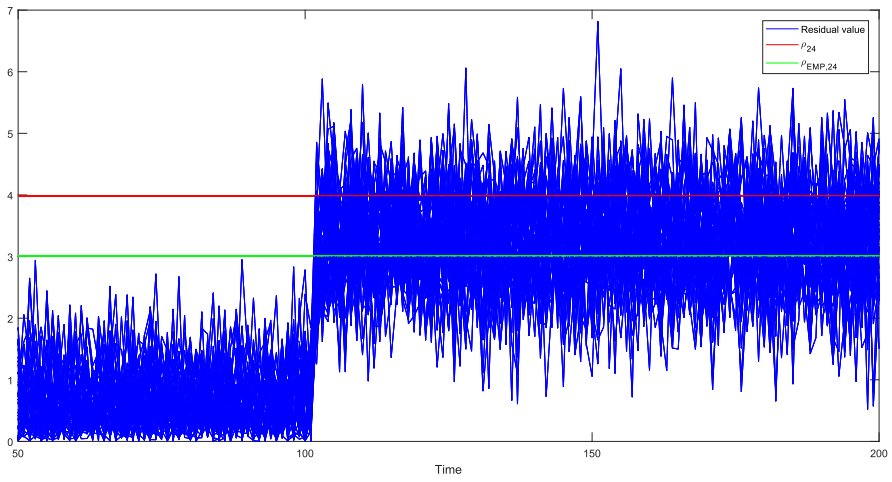

Figure 3.4: Single residual testing, behaviour of the residuals, compared to $\rho_{i,l}$ and $\rho_{EMP,\,i,l}$

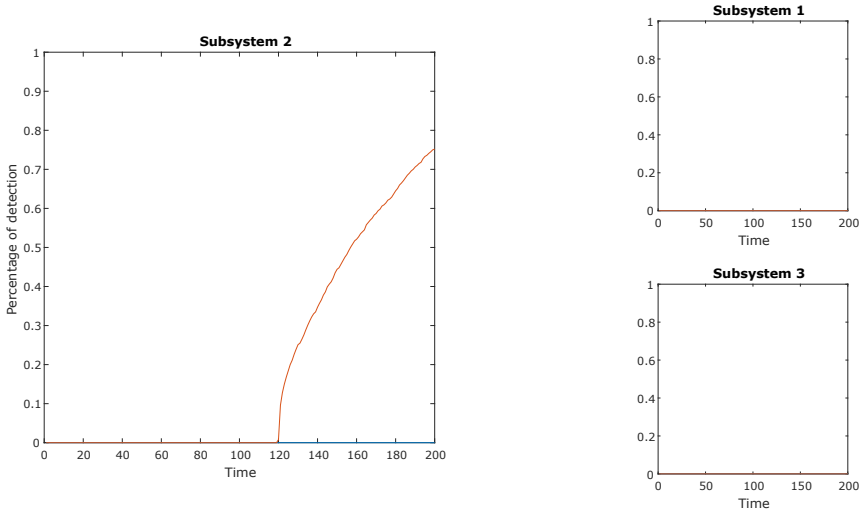

Figure 3.5: Small fault affecting subsystem 2, testing over a sliding window of 20 residuals

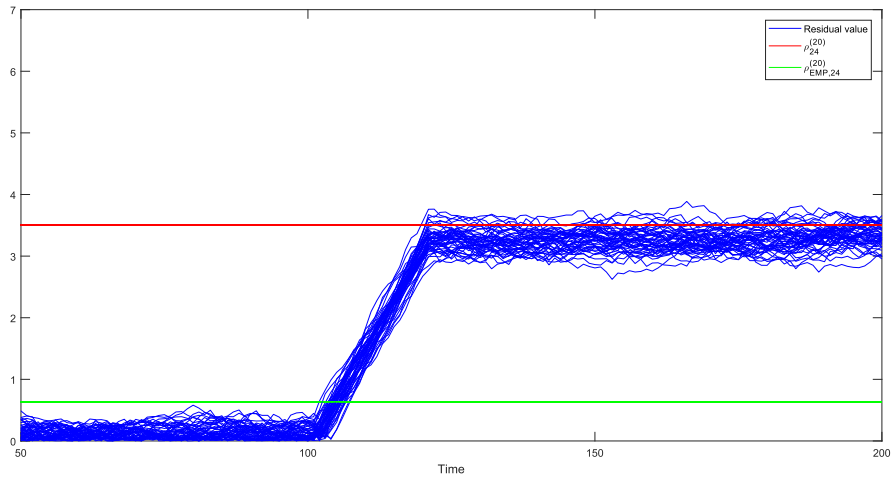

Figure 3.6: Moving sliding window mean of 20 residual testing, behaviour of the residuals, compared to $\rho_{i,l}^{(20)}$ and $\rho_{EMP,\,i,l}^{(20)}$

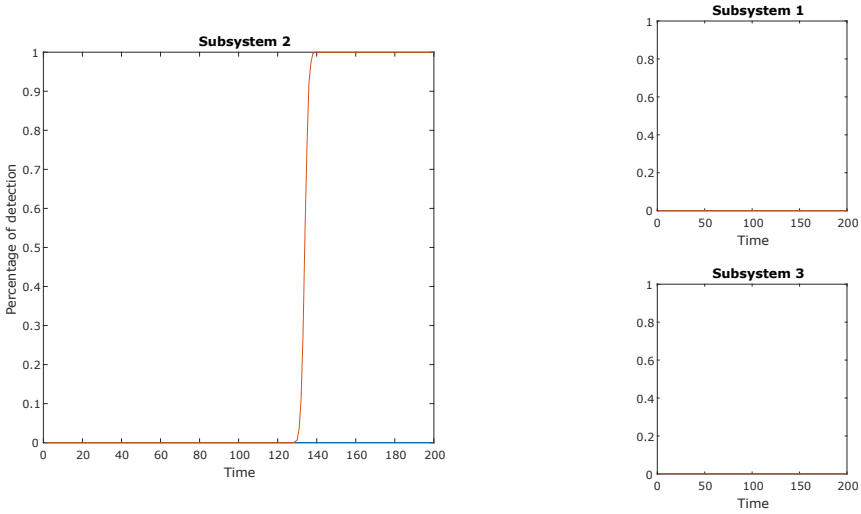

Figure 3.7: Small fault affecting subsystem 2, testing over a sliding window of 40 residuals

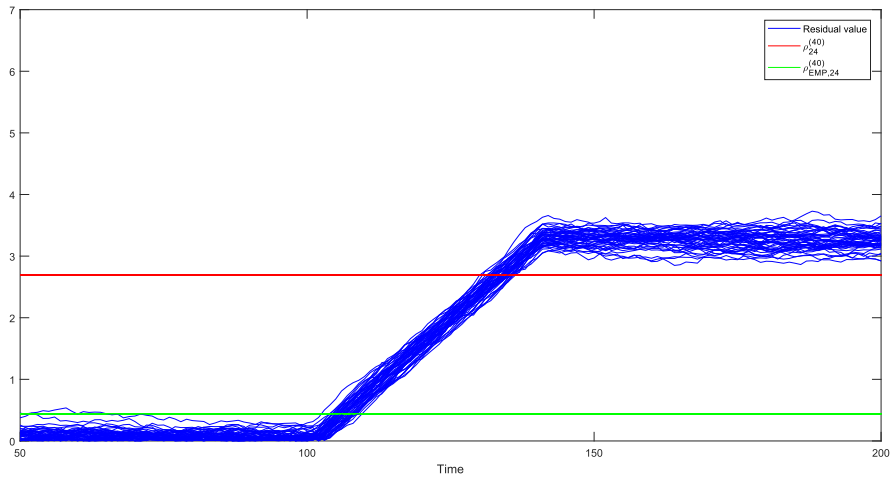

Figure 3.8: Moving sliding window mean of 40 residual testing, behaviour of the residuals, compared to $\rho_{i,l}^{(40)}$ and $\rho_{EMP,\,i,l}^{(40)}$

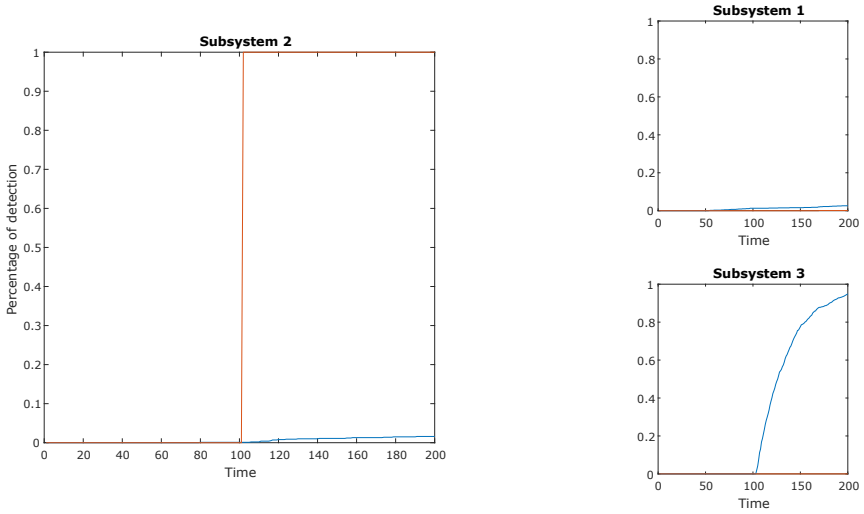

Figure 3.9: Medium fault affecting subsystem 2, single residual testing

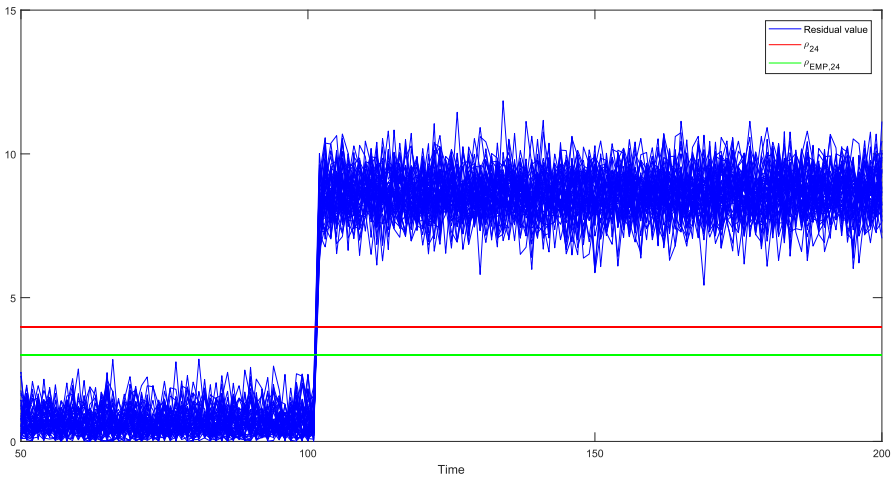

Figure 3.10: Single residual testing, behaviour of the residuals, compared to $\rho_{i,l}$ and $\rho_{EMP,\,i,l}$

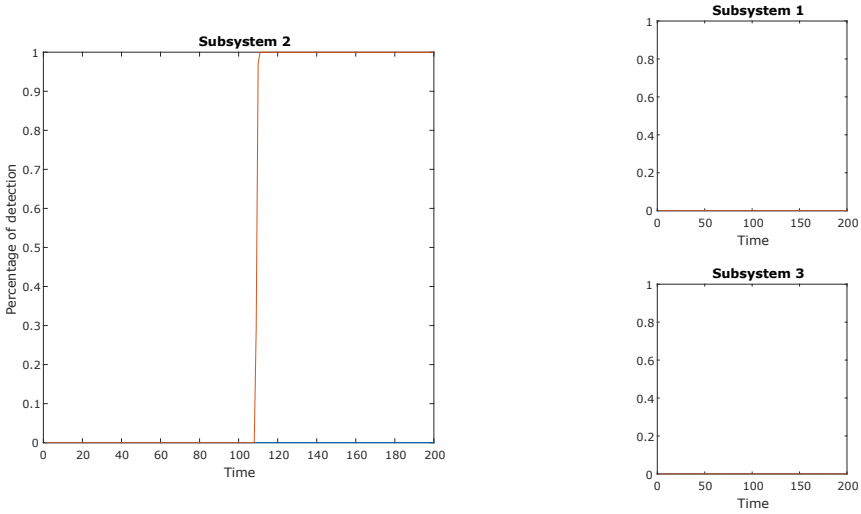

Figure 3.11: Medium fault affecting subsystem 2, testing over a sliding window of 20 residuals

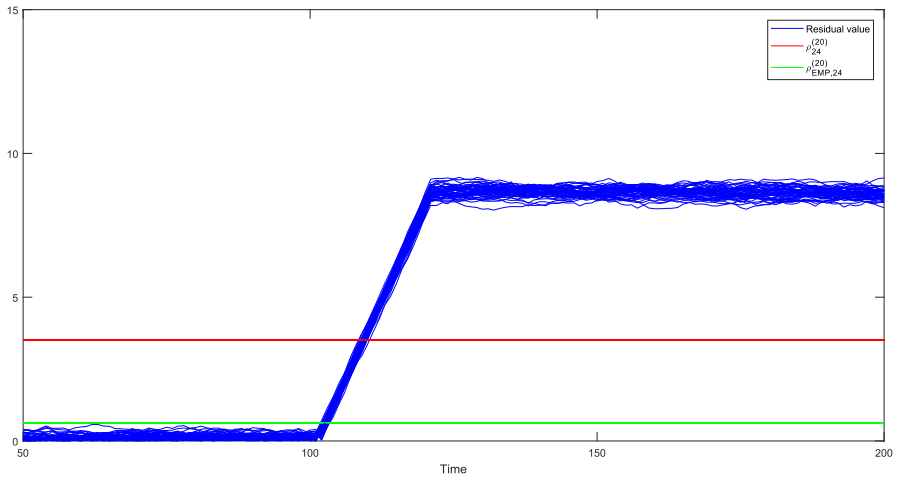

Figure 3.12: Moving sliding window mean of 20 residual testing, behaviour of the residuals, compared to $\rho_{i,l}^{(20)}$ and $\rho_{EMP,\,i,l}^{(20)}$

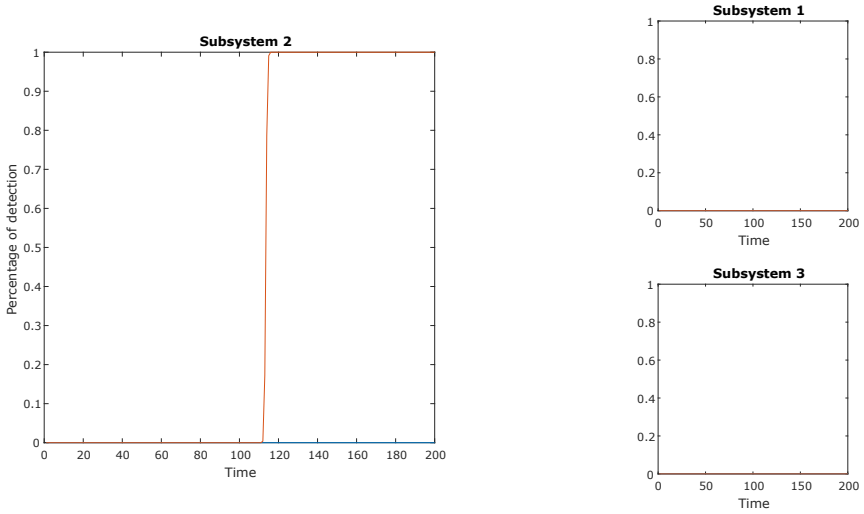

Figure 3.13: Medium fault affecting subsystem 2, testing over a sliding window of 40 residuals

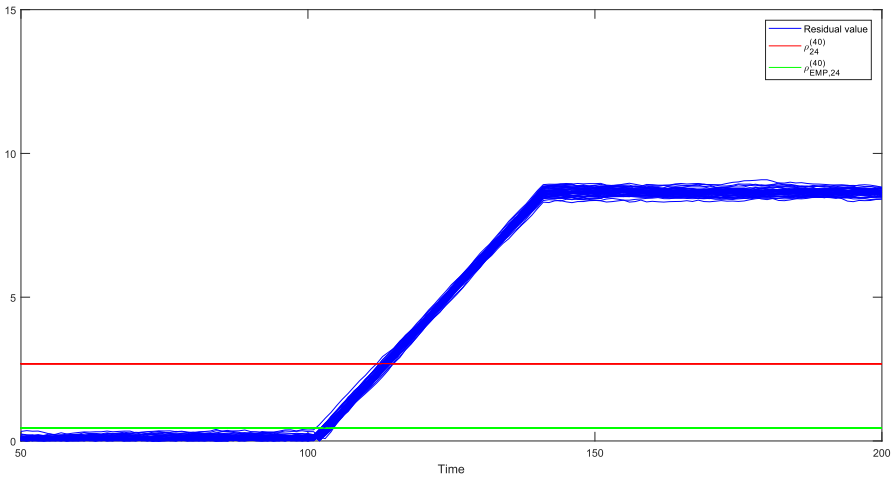

Figure 3.14: Moving sliding window mean of 40 residual testing, behaviour of the residuals, compared to $\rho_{i,l}^{(40)}$ and $\rho_{EMP,\,i,l}^{(40)}$

3.2 Power network system case study

Large scale systems which can benefit from partition-based FDI are power network systems (PNS). In Fig. 3.15 there is a general representation with all the main elements composing a standard PNS.

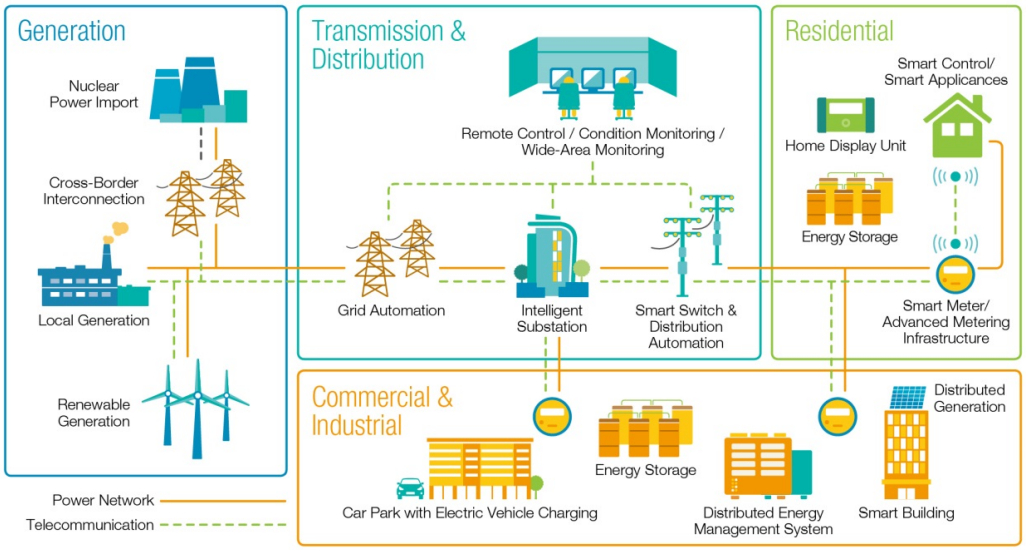

Figure 3.15: Example of a Power network system from [10]

In the examined case, we consider the PNS model introduced in [3]. It is composed of 5 power generation areas coupled through tie lines. Tie lines allows for power exchange between neighbouring areas: if the load profile (i.e., required current) of an area grows too much, exceeding the power generated in that area, neighbouring areas can contribute to satisfy the request. In our case we consider electrical power to be generated by thermal power stations equipped with single stage turbines. The coupling graph of the examined power network system is shown in Figure 3.2.

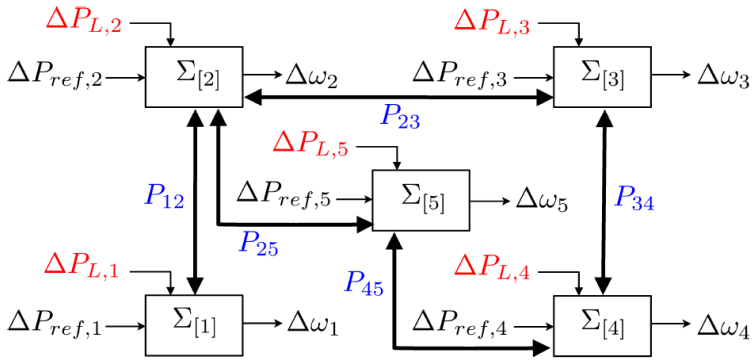

Figure 3.16: Coupling graph of the PNS in [3]

Neighbouring relationships are symmetric since they are directly related to power exchanges in electric lines (i.e., electric power flows in both directions). The dynamics of each power generation area $\Sigma_{[i]}$, $i = 1, ..., 5$, equipped with primary control and linearized around the equilibrium value for all variables, is described by the a linear time invariant model of the general type

$$\dot{x}_i(t) = A^c_{ii}x_i(t) + B^c_i u_i + L^c_i \Delta P_{L_i} + \sum_{j \in \tilde{\mathcal{N}}_i} A^c_{ij} x_j \tag{3.9}$$

The state $x_i = (\Delta\theta_i, \Delta\omega_i, \Delta P_{m_i}, \Delta P_{v_i})$ includes the angular deviation of the rotor with respect the stationary reference axis on the stator $\Delta\theta_i$, the speed deviation of rotating mass from the nominal value $\Delta\omega_i$, the deviation of the mechanical power from its nominal value ΔP_{m_i} and the deviation of the steam valve position from the nominal value ΔP_{v_i}. The matrices of the system are

$$A^c_{ii} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{\sum_{j\in\tilde{N}_i}}{2H_i} & -\frac{D_i}{2H_i} & \frac{1}{2H_i} & 0 \\ 0 & 0 & -\frac{1}{T_{t_i}} & \frac{1}{T_{t_i}} \\ 0 & -\frac{1}{R_i T_{g_i}} & 0 & -\frac{1}{T_{g_i}} \end{bmatrix}$$

$$B^c_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{T_{g_i}} \end{bmatrix}, \quad A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{P_{ij}}{2H_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad L^c_i = \begin{bmatrix} 0 \\ -\frac{1}{2H_i} \\ 0 \\ 0 \end{bmatrix}$$

We denote with $\tilde{\mathcal{N}}_i = \mathcal{N}_i \setminus i$ the set of the strict predecessors.
In Table 3.5 the meaning of each parameter and constant is reported.

$\Delta\theta_i$	Deviation of the angular displacement of the rotor with respect to the stationary reference axis on the stator
$\Delta\omega_i$	Speed deviation of rotating mass from nominal value
ΔP_{m_i}	Deviation of the mechanical power from nominal value (p.u.)
ΔP_{v_i}	Deviation of the steam valve position from nominal value (p.u.)
ΔP_{ref_i}	Deviation of the reference set power from nominal value (p.u.)
ΔP_{L_i}	Deviation of the nonfrequency-sensitive load change from nominal value (p.u.)
H_i	Inertia constant defined as $H_i = \frac{\text{kinetic energy at rated speed}}{\text{machine rating}}$ (typically values in range $[1-10]$ sec)
R_i	Speed regulation
D_i	Defined as $\frac{\text{percent change in load}}{\text{change in frequency}}$
T_{t_i}	Prime mover time constant (typically values in range $[0.2-2]$ sec)
T_{g_i}	Governor time constant (typically values in range $[0.1-0.6]$ sec)
P_{ij}	Slope of the power angle curve at the initial operating angle between area i and area j

Table 3.5: Variables of a generation area with typical value ranges. (p.u.) stands for "per unit", from [9].

The control input of each area is $\bar{u}_i = \Delta P_{ref_i}$, while ΔP_L is the local power load. Since the control design is out of the scope of this work, we set $\Delta P_{ref} = \Delta P_{L_i}$ and constant $\Delta P_{L_i} = \begin{bmatrix} 0.22 & 0.12 & 0.10 & 0.08 & -0.1 \end{bmatrix}$. In view of this, the load L_i^c and the input B_i^c components can be merged in a single factor $\tilde{B}_i$.

$$\tilde{B}_i = B_i^c + L_i^c = \begin{bmatrix} 0 \\ -\frac{1}{2H_i} \\ 0 \\ \frac{1}{T_{g_i}} \end{bmatrix}$$

The angular deviation $\Delta\theta_i$ and the angular speed deviation $\Delta\omega_i$, as well as the load profile ΔP_{L_i}, are considered measurable. In view of this, the output transformation is

$$y_i(t) = C_i x_i(t)$$

where matrix C_i is

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The state matrix A of the collective model has the following sparse structure

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & A_{25} \\ 0 & A_{32} & A_{33} & A_{34} & 0 \\ 0 & 0 & A_{43} & A_{44} & A_{45} \\ 0 & A_{52} & 0 & A_{54} & A_{55} \end{bmatrix} \tag{3.10}$$

Matrices $\tilde{B}$ and C of the collective model are

$$\tilde{B} = \begin{bmatrix} \tilde{B}_1 & 0 & 0 & 0 & 0 \\ 0 & \tilde{B}_2 & 0 & 0 & 0 \\ 0 & 0 & \tilde{B}_3 & 0 & 0 \\ 0 & 0 & 0 & \tilde{B}_4 & 0 \\ 0 & 0 & 0 & 0 & \tilde{B}_5 \end{bmatrix} \quad C = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & 0 \\ 0 & 0 & C_3 & 0 & 0 \\ 0 & 0 & 0 & C_4 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix} \tag{3.11}$$

Considering A, $\tilde{B}$ and C we obtain the centralized collective continuous time model of our PNS, composed by 5 generation areas

$$\begin{cases} \dot{x}(t) = Ax(t) + \tilde{B}\bar{u} \\ y(t) = Cx(t) \end{cases} \tag{3.12}$$

where the collective input vector $\bar{u}^T = \begin{bmatrix} \bar{u}_1^T & \bar{u}_2^T & \bar{u}_3^T & \bar{u}_4^T & \bar{u}_5^T \end{bmatrix}$.

In order to preserve the sparse structure of matrix (3.10), together with the related possibility to compute the partitioned distributed model, we discretize the system by means of the method in [5] with a sampling interval of $0.1s$. Furthermore, Gaussian white noises, $w(k)$ and $v(k)$, are added both on the states and on the measurements, respectively. We recall that noises $w_i(k) \in \mathbb{R}^{n_i}$ and $v_i(k) \in \mathbb{R}^{p_i}$ are zero-mean white noises, for $i = 1, \ldots, 5$, with related covariance matrices Q_i and R_i, respectively. Similarly to [3], we set:

$$Q_i = \begin{bmatrix} 0.9 \times 10^{-6} & 0 & 0 & 0 \\ 0 & 0.9 \times 10^{-6} & 0 & 0 \\ 0 & 0 & 1 \times 10^{-3} & 0 \\ 0 & 0 & 0 & 1 \times 10^{-3} \end{bmatrix}$$

$$R_i = \begin{bmatrix} 0.9 \times 10^{-6} & 0 \\ 0 & 0.9 \times 10^{-6} \end{bmatrix}$$

The standard centralized and discretized equations, denoted with subscript "d", are then in the following form

$$\begin{cases} x(k+1) = A_d x(k) + \tilde{B}_d \bar{u} + w(k) \\ y(k) = C_d x(k) + v(k) \end{cases} \tag{3.13}$$

where $w(k)^T = \begin{bmatrix} w_1^T(k) & w_2^T(k) & w_3^T(k) & w_4^T(k) & w_5^T \end{bmatrix}^T$ is the collective state noise vector, and $v(k)^T = \begin{bmatrix} v_1^T(k) & v_2^T(k) & v_3^T(k) & v_4^T(k) & v_5^T \end{bmatrix}^T$ is the collective output noise vector.

Model (3.13), in view of the structure of matrices (3.10) and (3.11), which is maintained also after the discretization process, can be partitioned in five subsystems accounting for their relationships as follows

$$\begin{cases} x_1(k+1) = A_{d,11} x_1(k) + A_{d,12} x_2(k) + \tilde{B}_{d,1} \bar{u}_1 + w_1(k) \\ \quad y_1(k) = C_{d,1} x_1(k) + v_1(k) \\ \\ x_2(k+1) = A_{d,22} x_2(k) + A_{d,21} x_1(k) + A_{d,23} x_3(k) + A_{d,25} x_5(k) + \tilde{B}_{d,2} \bar{u}_2 + w_2(k) \\ \quad y_2(k) = C_{d,2} x_2(k) + v_2(k) \\ \\ x_3(k+1) = A_{d,33} x_3(k) + A_{d,32} x_2(k) + A_{d,34} x_4(k) + \tilde{B}_{d,3} \bar{u}_3 + w_3(k) \\ \quad y_3(k) = C_{d,3} x_3(k) + v_3(k) \\ \\ x_4(k+1) = A_{d,44} x_4(k) + A_{d,43} x_3(k) + A_{d,45} x_5(k) + \tilde{B}_{d,4} \bar{u}_4 + w_4(k) \\ \quad y_4(k) = C_{d,4} x_4(k) + v_4(k) \\ \\ x_5(k+1) = A_{d,55} x_5(k) + A_{d,52} x_2(k) + A_{d,54} x_4(k) + \tilde{B}_{d,5} \bar{u}_5 + w_5(k) \\ \quad y_5(k) = C_{d,5} x_5(k) + v_5(k) \end{cases} \tag{3.14}$$

3.2.1 Distributed predictor

The Luenberger observer gains L_{ij}, guaranteeing both Schur stability of matrix $F = (A_d - LC)$, and a structure to the collective matrix L equal to the one of A_d, are computed according to LMI-based design method described in Section 2.2.2. The obtained distributed Luenberger predictor result in a set of 5 equations, similarly to the one in (3.8), structured as the partitioned system (3.14).

3.2.2 Threshold computation

Once the gain matrices are computed, all the ingredients needed to compute the analytical upper bounds to the residuals variance $(\sigma_{i,l}^B)^2$, introduced in equation (2.8) are available. In Tables 3.6, 3.7, and 3.8 the analytical bounds are compared with the corresponding empirical variances of the residuals (calculated over 10000 samples), in case $m = 1$, $m = 20$, and $m = 40$, respectively.

Single Residual $m = 1$	$10^{-4}\times$	$(\sigma_{i,l}^B)^2$	$(\sigma_{EMP,i})^2$
Subsystem 1	$r_{\Delta\theta_1}$	0.0325	0.0218
	$r_{\Delta\omega_1}$	0.0663	0.0238
Subsystem 2	$r_{\Delta\theta_2}$	0.0503	0.0237
	$r_{\Delta\omega_2}$	0.1678	0.0278
Subsystem 3	$r_{\Delta\theta_3}$	0.0414	0.0229
	$r_{\Delta\omega_3}$	0.1353	0.0254
Subsystem 4	$r_{\Delta\theta_4}$	0.0411	0.0227
	$r_{\Delta\omega_4}$	0.1132	0.0332
Subsystem 5	$r_{\Delta\theta_5}$	0.0411	0.0226
	$r_{\Delta\omega_5}$	0.1185	0.0314

Table 3.6: Comparison between analytical values of the residual variance $(\sigma_{i,l}^B)^2$ and the empirical value $\sigma_{EMP,i}$ for the single residual

Sliding window average, $m = 20$	$10^{-5}\times$	$(\sigma_{i,l}^{(20),B})^2$	$(\sigma_{EMP,i}(20))^2$
Subsystem 1	$r_{\Delta\theta_1}$	0.2730	0.0117
	$r_{\Delta\omega_1}$	0.5376	0.0046
Subsystem 2	$r_{\Delta\theta_2}$	0.4130	0.0094
	$r_{\Delta\omega_2}$	1.2976	0.0023
Subsystem 3	$r_{\Delta\theta_3}$	0.3444	0.0102
	$r_{\Delta\omega_3}$	1.0622	0.0059
Subsystem 4	$r_{\Delta\theta_4}$	0.3418	0.0104
	$r_{\Delta\omega_4}$	0.9004	0.0524
Subsystem 5	$r_{\Delta\theta_5}$	0.3417	0.0094
	$r_{\Delta\omega_5}$	0.9402	0.0224

Table 3.7: Comparison between the analytical upper bound to the variance of the mean over a moving sliding window of 20 residuals $(\sigma_{i,l}^{(20),B})^2$ and the empirical value $(\sigma_{EMP,i}^{(20)})^2$

Sliding window average, $m = 40$	$10^{-5}\times$	$(\sigma_{i,l}^{(40),B})^2$	$(\sigma_{EMP,i}(40))^2$
Subsystem 1	$r_{\Delta\theta_1}$	0.1660	0.0053
	$r_{\Delta\omega_1}$	0.3223	0.0018
Subsystem 2	$r_{\Delta\theta_2}$	0.2489	0.0041
	$r_{\Delta\omega_2}$	0.7655	0.0010
Subsystem 3	$r_{\Delta\theta_3}$	0.2085	0.0041
	$r_{\Delta\omega_3}$	0.6295	0.0026
Subsystem 4	$r_{\Delta\theta_4}$	0.2070	0.0051
	$r_{\Delta\omega_4}$	0.5359	0.0507
Subsystem 5	$r_{\Delta\theta_5}$	0.2069	0.0042
	$r_{\Delta\omega_5}$	0.5591	0.0209

Table 3.8: Comparison between the analytical upper bound to the variance of the mean over a moving sliding window of 40 residuals $(\sigma_{i,l}^{(40),B})^2$ and the empirical value $(\sigma_{EMP,i}^{(40)})^2$

3.2.3 Simulation results

In this section the simulation results obtained with the power network system model are reported.
In the simulations whose results are shown in Figures 3.17- 3.22, a persistent additive fault of amplitude 1, which occurs at time step $k = 100$, is introduced on the third state of the subsystem 2. Secondly, in the simulations whose results are shown in Figures 3.23- 3.28, a realistic fault, which commonly occurs in PNS, is applied. More specifically, the governor time constants T_{g_1} is increased from 0.1 to 2 at time $k = 100$. Note that T_g defines the frequency of the control action closing the loop between the generated power and the required one. Therefore, an increase in its value leads to slower response of the power load and a slower recovery of the network system, compromising the entire system performance.
For each proposed case we display two types of diagrams. Figures 3.17, 3.19, 3.21, 3.23, 3.25, 3.27 show, for each experiment type, the rate of networks where a fault is detected for each time instant k. The corresponding rates are shown for each subsystem output residual and for each subsystem. These results are obtained considering the analytical threshold only 1000 Montecarlo runs.
The plots in Figures 3.18, 3.20, 3.22, 3.24, 3.26, and 3.28, instead, show the values taken by the residual (or the mean of residuals over sliding windows) at each simulation time instant, compared to the corresponding thresholds, both analytical and empirical. In this case 30 Montecarlo runs only were considered. Three different residual testing scenarios are taken into account: single residual testing and moving window averages on windows of lengths $m = 20$ and $m = 40$.
In the computation of the thresholds, the false alarm (FA) rate has been set to 0.0002, i.e, $\alpha = 3.7190$. The related thresholds are indeed the values $\rho_{i,l}^{(m)}$, where i is the index of the substystem, l is the index related to the state of the subsystem, while m is the length of the sliding window (omitted when considering the single residual). These values are compared with the corresponding empirical values $\rho_{EMP,\,i,l}^{(m)}$, which were obtained simulating the system in standard steady-state conditions for 10000 time instants.
While the detection, in case of parameter variation, is very good in all the cases, the results obtained with the first experiment (i.e., persistent additive fault) are of particular interest. Remarkably, from Figure 3.17 we notice that at time step $k = 200$, i.e., 100 time steps after the fault occurrence, only 40% of the systems detected it when a single residual is tested. Even worse performances are obtained, as visible in Figure 3.19, when a window of length $m = 20$ is used: in this case the diagram shows that no fault is detected by the algorithm. The detection capabilities are recovered, instead, when $m = 40$, which is visible from Figure 3.21. This peculiar behaviour will be discussed in Section 3.3.

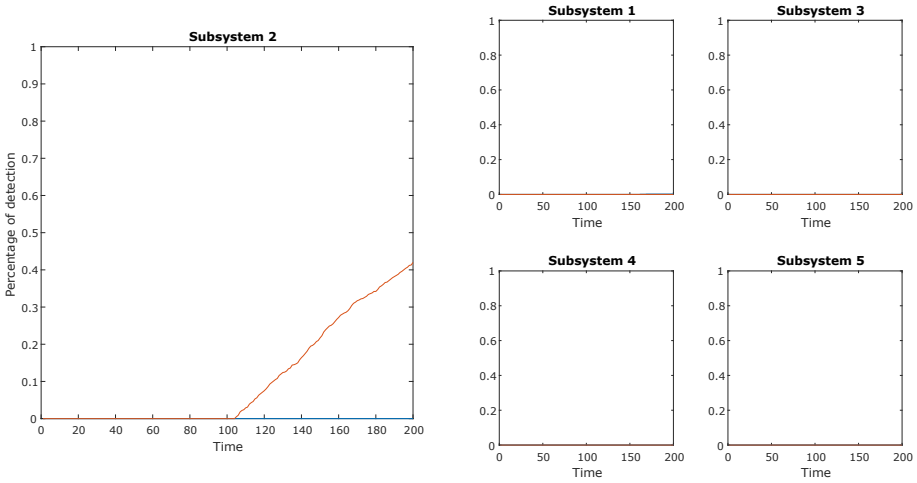

Figure 3.17: Additive fault affecting subsystem 2, single residual testing

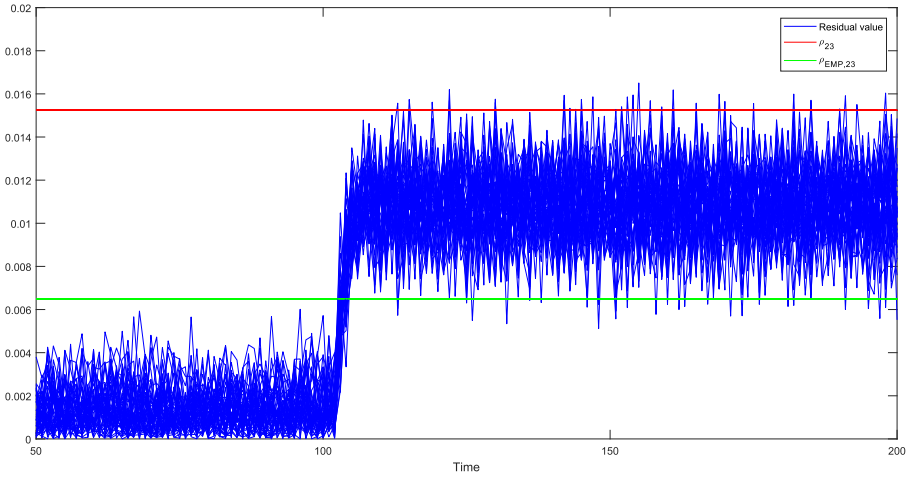

Figure 3.18: Single residual testing, behaviour of the residuals, compared to the related $\rho_{i,l}$ and $\rho_{EMP,i}$

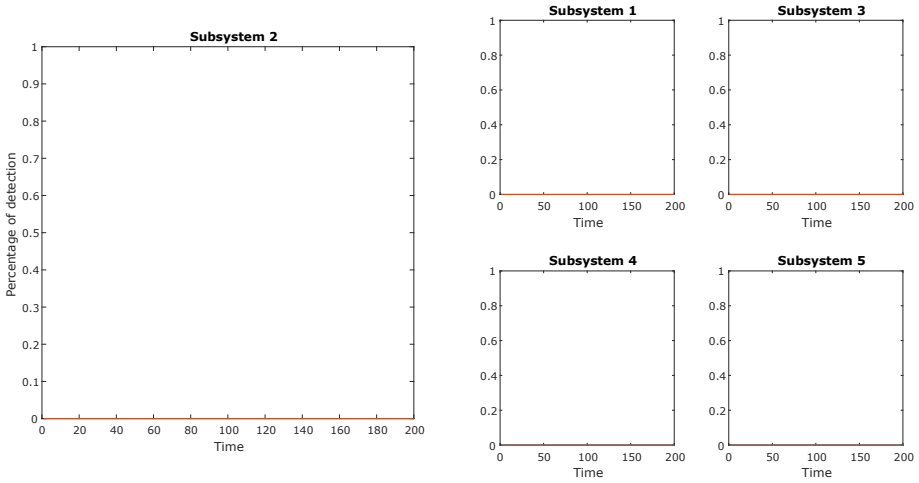

Figure 3.19: Additive fault affecting subsystem 2, moving window average of 20 residuals testing

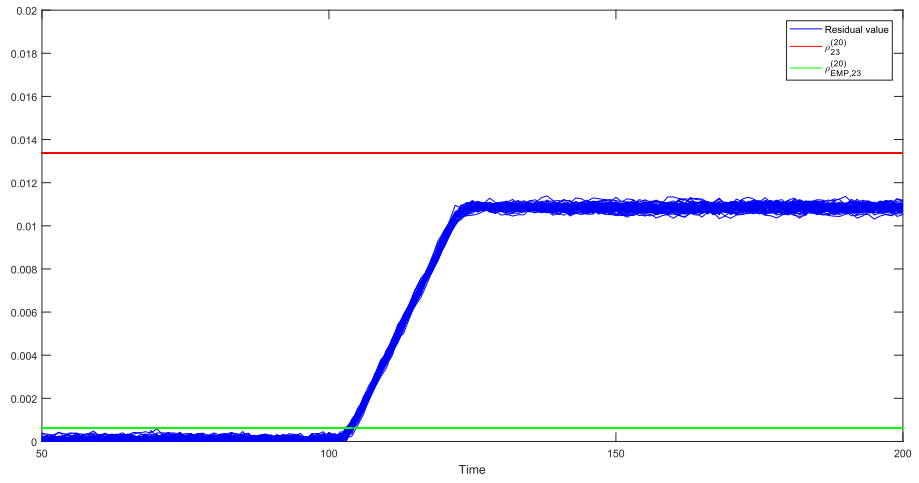

Figure 3.20: Moving window average of 20 residuals testing, behaviour of the mean of the residuals, compared to the related $\rho_{i,l}^{(20)}$ and $\rho_{EMP,i}^{(20)}$

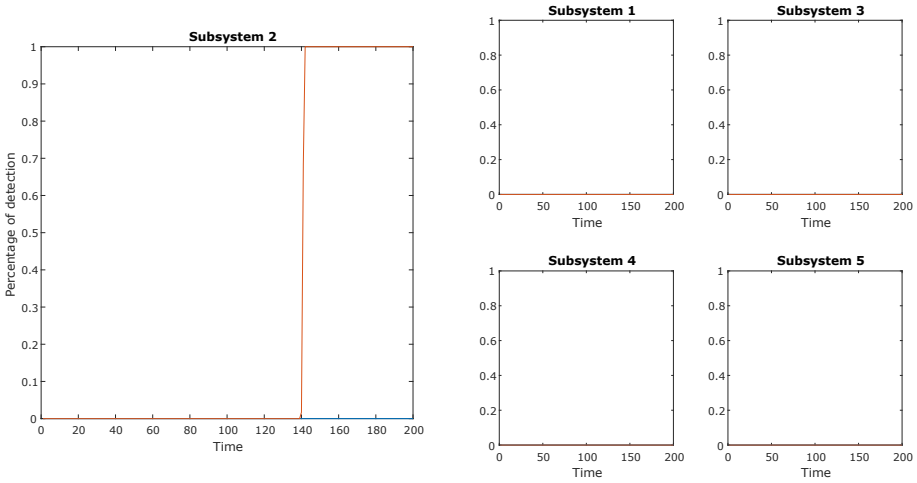

Figure 3.21: Additive fault affecting subsystem 2, moving window average of 40 residuals testing

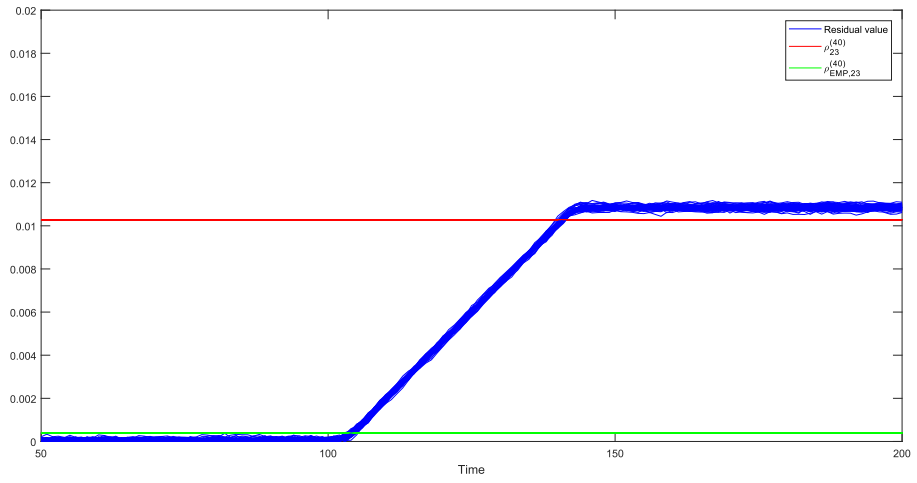

Figure 3.22: Moving window average of 40 residuals testing, behaviour of the mean of the residuals, compared to the related $\rho_{i,l}^{(40)}$ and $\rho_{EMP,i}^{(40)}$

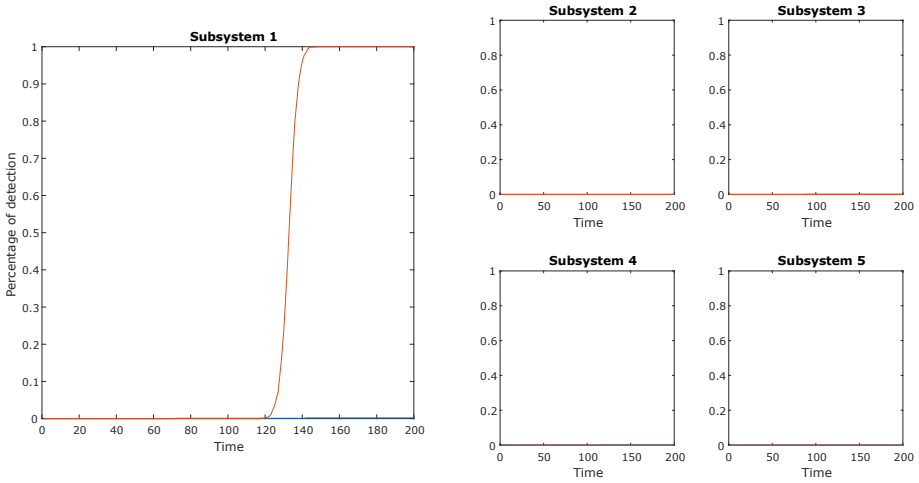

Figure 3.23: Parametric (i.e., multiplicative) fault affecting T_{g_1} (subsystem 1), single residual testing

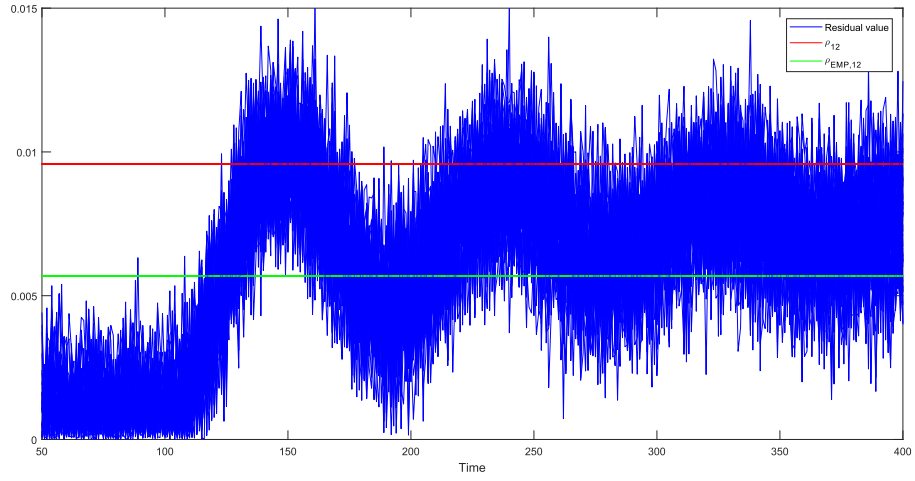

Figure 3.24: Single residual testing, behaviour of the residuals, compared to the related $\rho_{i,l}$ and $\rho_{EMP,i}$

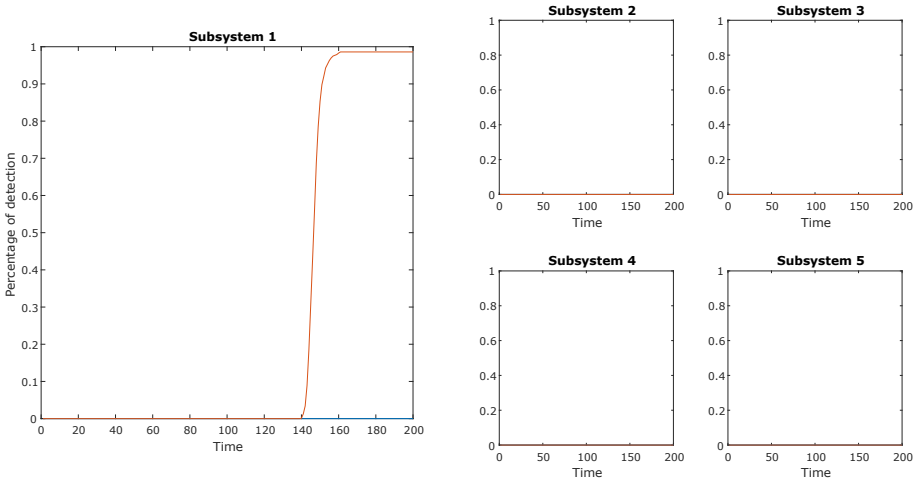

Figure 3.25: Parametric (i.e., multiplicative) fault affecting T_{g_1} (subsystem 1), moving window average of 20 residuals testing

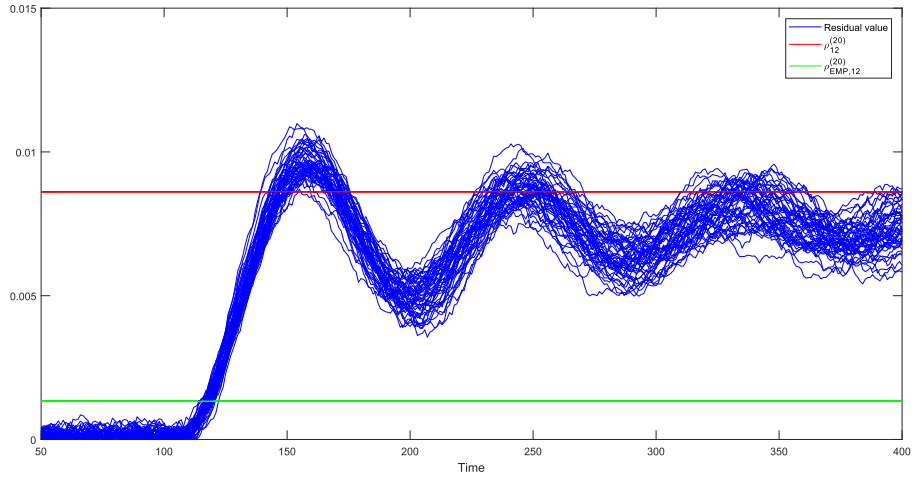

Figure 3.26: Moving window average of 20 residuals testing, behaviour of the mean of the residuals, compared to the related $\rho_{i,l}^{(20)}$ and $\rho_{EMP,i}^{(20)}$

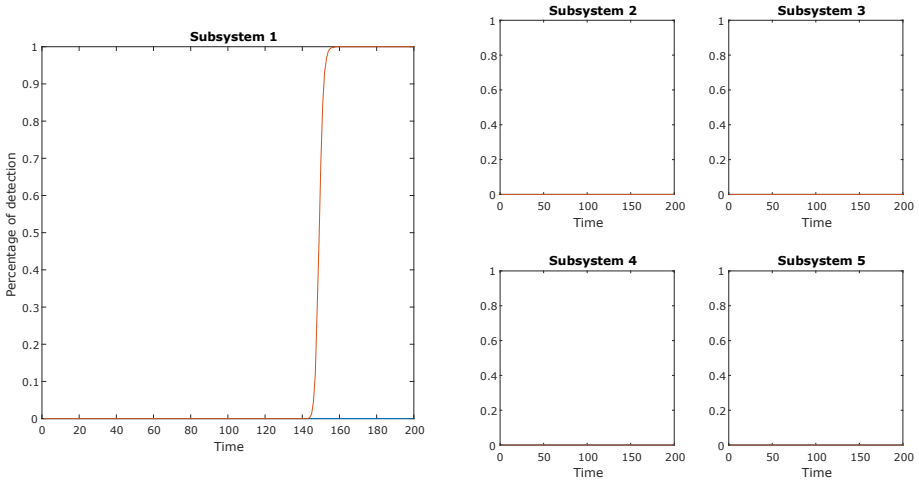

Figure 3.27: Parametric (i.e., multiplicative) fault affecting T_{g_1} (subsystem 1), moving window average of 40 residuals testing

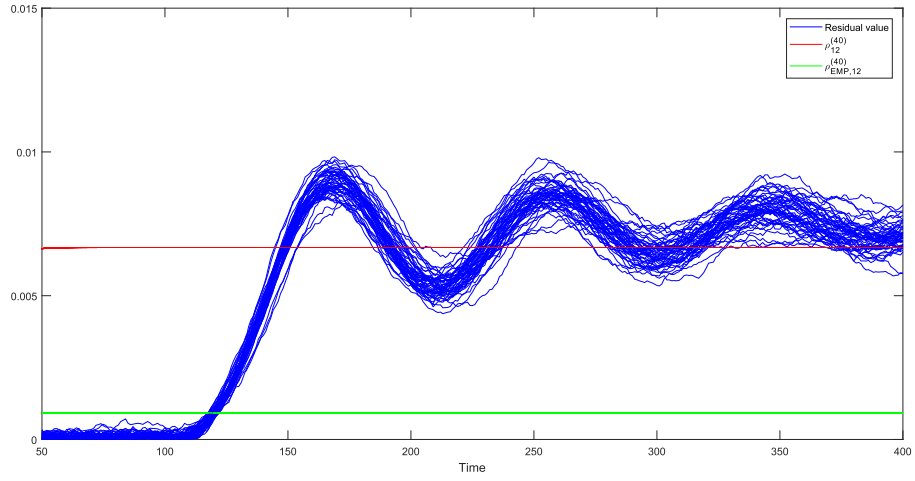

Figure 3.28: Moving window average of 40 residuals testing, behaviour of the mean of the residuals, compared to the related $\rho_{i,l}^{(40)}$ and $\rho_{EMP,i}^{(40)}$

3.3 Insights on the results

In this section we will draw some theoretical considerations about the FD algorithm and its properties. These considerations are motivated by the analysis of the simulation results shown so far, and in particular of the cases when the (additive) fault takes small amplitudes.
In [1] the FD test over a moving sliding window of residuals was meant to be a smart alternative to the single residuals approach, especially facing small amplitude faults. In the paper, the use of the average of m values rather than single values was justified by the fact that $\bar{\sigma}_{i,l}^{(m)}(k) \leq \sigma_{i,l}$, which means that in principle it is possible to obtain the same FA rate with smaller thresholds (which allows to detect faults of smaller amplitude). Furthermore, this gives the possibility to reduce the FA rate without loss in missed detection rate. In [1], an academic example which enforced this choice was proposed.
Implementing the algorithm on more complex large scale systems, we noted that the proposed moving window average method shows some criticalities. It may happen, for example, that the computed threshold $\rho_{i,l}^{(m)}$ results to be too large to detect faults, obstructing in some cases the detection.
For sake of simplicity of presentation but without loss of generality, from now on in this chapter subscripts i and l (i.e., the index of the subsystem and the index of the corresponding output index, respectively) will be omitted.

3.3.1 Statistical behaviour of the residuals

We denote with $r(k)$ the analyzed residual, and with

$$r^{(m)}(k) = \frac{1}{m} \sum_{i=1}^{m} r(k-i+1)$$

the residual mean over a sliding window of length m. In stationary conditions, considering a fault f acting on the system, $r(k)$ can be written as

$$r(k) = \mu_{rf} f + n(k) \tag{3.15}$$

where $n(k)$ is a colored noise, which depends on noises $w(k)$ and $v(k)$, and whose variance is σ^2. Also, μ_{rf} is the gain between the fault and the residual. Thanks to (3.15)

$$r^{(m)}(k) = \mu_{rf} f + n^{(m)}(k) \tag{3.16}$$

where $n^{(m)}(k) = \frac{1}{m} \sum_{i=1}^{m} n(k-i+1)$. Note that $(\sigma^{(m)})^2$, the variance of $n^{(m)}(k)$, verifies $\sigma^2/m < (\sigma^{(m)})^2 < \sigma^2$. Also, it is possible to prove that $\sigma^{(m+1)} < \sigma^{(m)}$ for

all $m \geq 1$, i.e., it is monotonically decreasing with respect to m. Exemplifying plots of the corresponding ideal (i.e., tight) thresholds are given in Figures 3.29 and 3.30, where we compute $\rho = \alpha\sigma$ and $\rho^{(m)} = \alpha\sigma^{(m)}$. When fault is absent (i.e., $f = 0$), the probability to declare fault is the FA rate, i.e., the yellow-colored areas Figures 3.29 and 3.30.

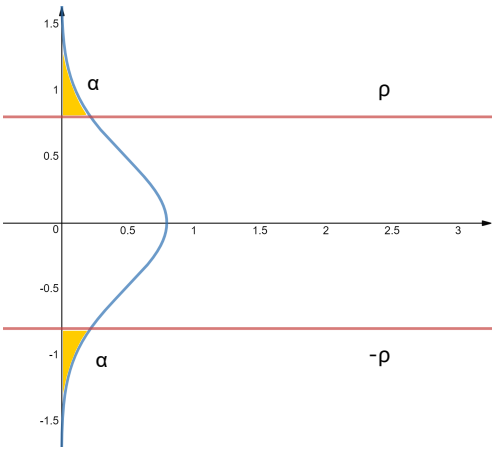

Figure 3.29: Standard distribution of $r(k)$, in blue, and associated threshold ρ, in red.

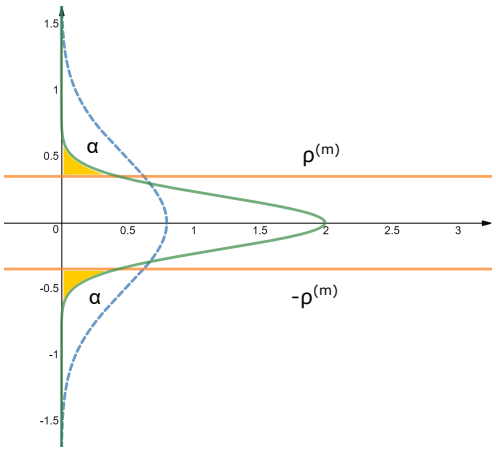

Figure 3.30: Standard distribution of $r^{(m)}(k)$, in green, and associated threshold $\rho^{(m)}$, in orange.

On the other hand, when the fault is present (i.e., $f \neq 0$), the probabilities to detect the fault (when the thresholds are computed in an ideal way) can be computed as the yellow-coloured areas in Figures 3.31 and 3.32.

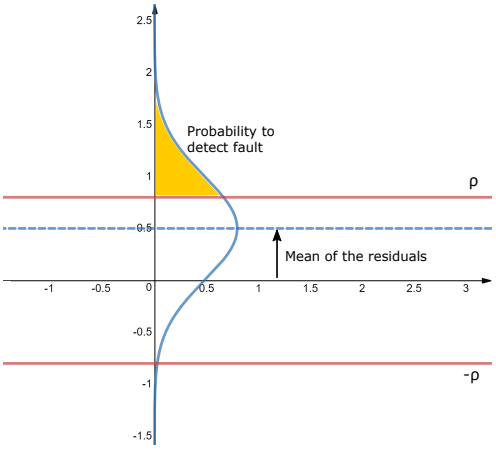

Figure 3.31: Standard distribution of $r(k)$, in blue, and associated threshold ρ, in red.

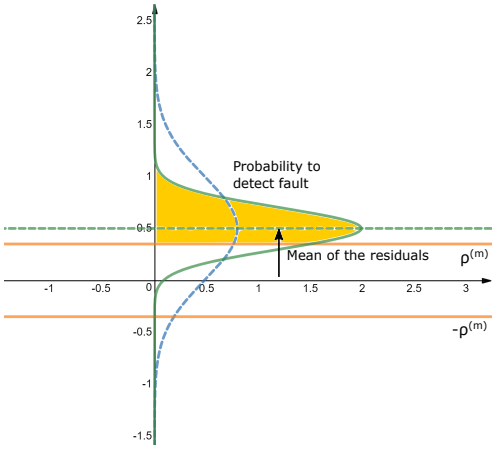

Figure 3.32: Standard distribution of $r^{(m)}(k)$, in green, and associated threshold $\rho^{(m)}$, in orange.

From the inspection of Figures 3.31 and 3.32 it is possible to conclude that the use of $r^{(m)}(k)$ rather than $r(k)$ can be beneficial for a more efficient fault detection. In fact, as expected, the use of $r^{(m)}$ allows to reduce the threshold (maintaining the FA rate invariant) and, as a result, it allows to increase the fault detection rate.

This consideration, however, may not hold when the thresholds are computed in a conservative way, e.g., using the distributed partition-based method discussed in Chapter 2. Denote with $(\sigma^B)^2$ and $(\sigma^{(m),B})^2$ the - conservative - upper bounds of the covariances of $r(k)$ and $r^{(m)}(k)$, respectively, computed distributively. As it will be clarified later, the value of $\sigma^{(m),B}$ may not be strictly decreasing as a function of m for small values of m, making $\sigma^{(m),B}$ (for $m > 1$) more conservative than σ^B, in a relative sense.
In Figure 3.33, the probabilities to detect the fault, when the thresholds are computed using σ^B and $\sigma^{(m),B}$, can be obtained as the areas of the surfaces coloured in grey and yellow, respectively.

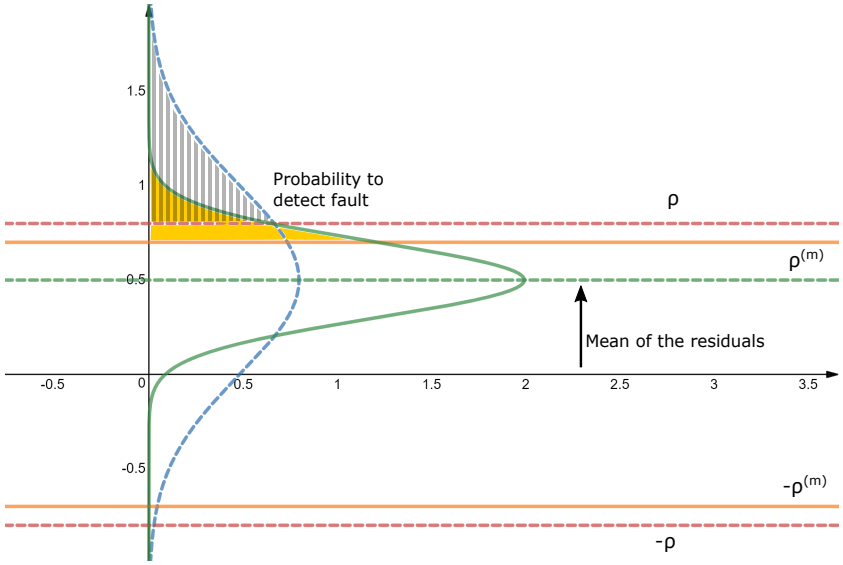

Figure 3.33: Distribution of $r^{(m)}$, in green, and associated threshold $\rho^{(m)}$, in orange, when $(\sigma^{(m)}(k))^2$ slightly decreases. Dashed lines represent the distribution of $r(k)$, in blue, and and associated threshold ρ, in red.

Apparently, in the case depicted in Figure 3.33, the probability to detect the fault using $r^{(m)}(k)$ is smaller than the probability to detect the fault using $r(k)$. The reason lies in the fact that, while the real variance of $r^{(m)}$ has a significant decrease rate as m increases, $\sigma^{(m),B}$ (based on which the threshold is computed) does not display the same behaviour, and is too large to allow an efficient fault detection.

This does not mean that the discussed method, based on the analysis of $r^{(m)}(k)$

is always uneffective. As also apparent in the simulation results shown in the previous sections, when m is sufficiently large, the performances of the methods improve significantly. This happens when

$$\rho^{(m),B} = \alpha\sigma^{(m),B} < \mu_{rf} f \tag{3.17}$$

i.e., when the threshold becomes lower than the mean value of $r^{(m)}$. In the following, the condition on m allowing to verify (3.17) will be studied.

3.3.2 Dependence of $(\sigma^{(m),B}(k))^2$ upon m

The upper bound to the residual variance, computed at time k, is defined as in (2.22), i.e.,

$$(\sigma^{(m),B}(k))^2 = \frac{1}{m^2}\sum_{j,h=0}^{m-1} \bar{\gamma}^B(k-j,k-h) \tag{3.18}$$

where $\bar{\gamma}^B(k-j,k-h)$ is given by

$$\begin{cases} \bar{\gamma}^B(k-j,k-h) = \bar{\gamma}^{B,1}(k-j,k-h) & \text{if } j = h \\ \bar{\gamma}^B(k-j,k-h) = \min\{\bar{\gamma}^{B,1}(k-j,k-h); \bar{\gamma}^{B,2}(k-j,k-h)\} & \text{if } j \neq h \end{cases} \tag{3.19}$$

Terms $\bar{\gamma}^{B,1}(k-j,k-h)$ and $\bar{\gamma}^{B,2}(k-j,k-h)$ are defined as follows

$$\bar{\gamma}^{B,1}(k-j,k-h) = \frac{1}{2}((\sigma^B(k-j))^2 + (\sigma^B(k-h))^2) \tag{3.20}$$

$$\begin{aligned} \bar{\gamma}^{B,2}(k-j,k-h) = \frac{1}{2}\,&\|C\|\,\|diag(\boldsymbol{B}(k-\max(h,j)))\\ &+ CB_i(k-\max(h,j))C^T\mathbb{1}_n\|\mu\lambda^{|h-j|}\\ &+ \|C\|\,\|L_i^c R^c\|\,\mu\lambda^{|h-j|-1} \end{aligned} \tag{3.21}$$

where $(\sigma^B(k))^2$ is the analytically computed upper bound of the variance in the single residual case, while the other ingredients have already been introduced in the Chapter 2.
In view of the stability properties of the state estimator, after a transient, $\sigma^B(k)$ becomes constant (i.e., we set $\sigma^B(k) = \sigma^B$), while $\bar{\gamma}^B(k-j,k-h)$ in (3.19) becomes independent of k and $\bar{\gamma}^B(h,j) = \bar{\gamma}^B(k+h,k+j)$ for all k. Also

$$\bar{\gamma}^B(h,j) = \bar{\gamma}^B(j,h) \tag{3.22}$$

In view of these considerations

1. $\bar{\gamma}^{B,1}(j,h) = (\sigma^B)^2$, constant $\forall j, h = 1, \dots, m$;

2. $\bar{\gamma}^{B,2}(j,h)$ is a exponentially decreasing function with respect to $|h-j|$, $\forall j, h = 1, \dots, m$;.

We can therefore introduce the integer q, defined as the minimum value of $|h-j|$ such that

$$\bar{\gamma}^{B,2}(j,h) < (\sigma^B)^2$$

for all $|h-j| > q$. Therefore, for all $|h-j| > q$, we set

$$\bar{\gamma}^{B}(h,j) = \bar{\gamma}^{B,2}(h,j) \tag{3.23}$$

For clarity, in view of (3.18), we can compute $(\sigma^{(m),B})^2$ as the mean of the m^2 elements of matrix $\Gamma \in \mathbb{R}^{m\times m}$, defined as

$$\Gamma = \begin{bmatrix} \bar{\gamma}^{B1}(1,1) & \cdots & \bar{\gamma}^{B1}(1,q) & \bar{\gamma}^{B2}(1,q+1) & \cdots & \bar{\gamma}^{B2}(1,m) \\ \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\ \bar{\gamma}^{B1}(q,1) & \cdots & \bar{\gamma}^{B1}(q,q) & \cdots & \ddots & \bar{\gamma}^{B2}(q,m) \\ \bar{\gamma}^{B2}(q,1) & \ddots & \cdots & \bar{\gamma}^{B1}(q+1,q+1) & \cdots & \bar{\gamma}^{B1}(q+1,m) \\ \vdots & \ddots & \ddots & \cdots & \ddots & \vdots \\ \bar{\gamma}^{B2}(m,1) & \cdots & \bar{\gamma}^{B2}(m,q) & \bar{\gamma}^{B1}(m,q+1) & \cdots & \bar{\gamma}^{B1}(m,m) \end{bmatrix} \tag{3.24}$$

Colors have been introduced to highlight the structure of Γ. Since $\bar{\gamma}^{B}(j,h) = \bar{\gamma}^{B}(h,j)$ for all j, h, Γ is a symmetric band matrix.

Since the values of $\bar{\gamma}^{B,2}$ exponentially decrease with $|h-j|$, for simplicity we consider them negligible. In view of this and of (3.18), we compute that, if $m > q$

$$(\sigma^{(m),B})^2 \simeq \frac{m^2 - (m-q)(m-q+1)}{m^2}(\sigma^{(m),B})^2 \tag{3.25}$$

since $\bar{\gamma}^{B1}(h,j) = (\sigma^{(m),B})^2$ for all h, j, and appears $m^2 - (m-q)(m-q+1)$ times in matrix Γ. In Figure 3.34 the plot of $\sigma^{(m),B}$ in (3.25) as a function of m is shown. Remark that $\sigma^{(m),B} \to 0$ as $m \to +\infty$.

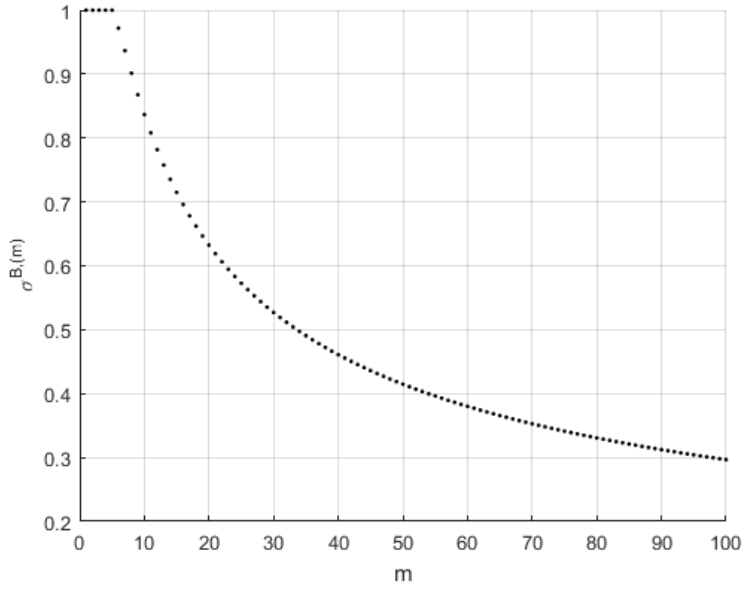

Figure 3.34: Plot of $\sigma^{(m),B}$ in (3.25) as a function of m. In this example $\sigma^{(m),B} = 1$ and $q = 5$.

3.3.3 Choice of the window length

Based on the considerations made in the preceding subsections, a simple method to decide either to opt for a sliding window approach or for a single residual one is here devised. In particular, we want to detect faults with entity $f > \bar{f}_{\min}$ with a maximum detection time delay $\tau_{\max}$. Regarding the detection delay τ, it is proportional to the window length m: for simplicity we set $\tau = m$. Therefore, for a fixed FA rate α, it is possible to verify the stated requirement using a window of length τ_{max}, in view of (3.17) and (3.25), only if

$$\frac{\sqrt{\tau_{\max}^2 - (\tau_{\max} - q)(\tau_{\max} - q + 1)}}{\tau_{\max}} \alpha\sigma^B < \mu_{rf} f_{min} \tag{3.26}$$

From this discussion we can conclude that the moving sliding window approach is definitively the best choice for our FD problem, guaranteeing the detection of small-entity faults of magnitude greater or equal to $\bar{f}_{\min}$ only of (3.26) is verified: in this case we can take m as the minimum value that guarantees that

$$\frac{\sqrt{m^2 - (m - q)(m - q + 1)}}{\tau_{\max}} < \frac{\mu_{rf} f_{min}}{\rho^B}$$

where $\rho^B = \alpha\sigma^B$ is the threshold used in the single residual case. Otherwise, the best choice to take is to analyze the single residual $r(k)$.

Chapter 4

A novel algorithm for centralized and distributed observer-based fault isolation

As already discussed, fault isolation can be described as the step beyond the detection, consisting of the identification of a specific a fault acting on a system.
In this chapter a novel observer-based fault isolation algorithm is proposed and described along its main features. This algorithm exploits the main properties of the FD method in [1], such as the use of a state estimator and the generation of thresholds based on upper bounds to the residual variance. However, differently from the latter, it not only detects the presence of a specific fault, but also it is able to estimate its main characteristics, matching the observed malfunctioning with a specific fault signal which has been modelled.

4.1 System model including modelled faults

In most cases, dealing with the implementation of FDI algorithms for complex engineering systems, the model of typical possible faults which may affect a system, along with its main triggers and consequences, is known (e.g., main faults which need to be detected in PNS are related to a change in the parameters T_g, as in Subsection 3.2.3).
Recalling the fault models introduced in Section 1.3.1, in this section we will refer only to faults whose mathematical behaviour can be characterized by additive terms in state equations.
The linear discrete-time centralized autonomous model of a generic engineering system, including the contribution of a set of selected additive faults, is the fol-

lowing

$$\begin{aligned} x(k+1) &= Ax(k) + B_f f(k) + w(k) \\ y(k) &= Cx(k) + D_f f(k) + v(k) \end{aligned} \tag{4.1}$$

where, $x(k) \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}^p$ is the output vector, $w(k) \in \mathbb{R}^n$ is the noise vector associated with the state, while $v(k) \in \mathbb{R}^p$ is the noise vector associated with the output. $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{n \times p}$ are the state and the output matrices, respectively. Moreover, $f(k) \in \mathbb{R}^{n_f}$ is the fault vector, whose elements are different from zero only in presence of a fault occurrence, while $B_f \in \mathbb{R}^{n \times n_f}$ and $D_f \in \mathbb{R}^{p \times n_f}$ are the matrices relating the behaviour of the fault f to the state and to the output equations, respectively.

4.2 Enlarged system and observer

Given a system model as the one in (4.1), we can redefine its structure as follows. Considering fault vector $f(k)$ as a constant signal under the assumption that the faults are persistent, we can rewrite the state equations in (4.1) as

$$\begin{aligned} x(k+1) &= Ax(k) + B_f f(k) + w(k) \\ f(k+1) &= f(k) \end{aligned} \tag{4.2}$$

Given this particular system form, vector $f(k)$ appears as a variable acting on the state and on the outputs, with its own state equation. We can then define the enlarged model of the system. Its equations in matricial form are the following

$$\begin{cases} \begin{bmatrix} x(k+1) \\ f(k+1) \end{bmatrix} = \begin{bmatrix} A & B_f \\ 0 & I_{nf} \end{bmatrix} \begin{bmatrix} x(k) \\ f(k) \end{bmatrix} + \begin{bmatrix} I_n \\ 0 \end{bmatrix} w(k) \\ y(k) = \begin{bmatrix} C & D_f \end{bmatrix} \begin{bmatrix} x(k) \\ f(k) \end{bmatrix} + v(k) \end{cases} \tag{4.3}$$

where I_{n_f} and I_n are identity matrices of dimension n_f and n, respectively. Noise vector $\tilde{w}(k) = \begin{bmatrix} I_n \\ 0 \end{bmatrix} w(k)$, is justified by the fact that $w(k)$ does not affect directly the fault equations.
We can now introduce the enlarged state vector

$$\boldsymbol{x}^T(k) = \begin{bmatrix} x^T(k) & f^T(k) \end{bmatrix} \tag{4.4}$$

Moreover, we define with $\bar{A} \in \mathbb{R}^{(n+n_f) \times (n+n_f)}$ and with $\bar{C} \in \mathbb{R}^{p \times (n+n_f)}$ the following matrices

$$\bar{A} = \begin{bmatrix} A & B_f \\ 0 & I_{n_f} \end{bmatrix} \qquad \bar{C} = \begin{bmatrix} C & D_f \end{bmatrix} \tag{4.5}$$

The whole enlarged system can be then rewritten in the form

$$\begin{cases} \boldsymbol{x}(k+1) = \bar{A}\boldsymbol{x}(k) + \tilde{w}(k) \\ y(k) = \bar{C}\boldsymbol{x}(k) + v(k) \end{cases} \tag{4.6}$$

Recalling the general equations of the standard centralized Luenberger predictor in (2.3), we can consider to develop a Luenberger observer for the enlarged system introduced in equation (4.6). In this way the collective estimated vector $\hat{\boldsymbol{x}}(k)$ would include both the state vector estimate, namely $\hat{x}(k)$, and an estimate of the fault vector, $\hat{f}(k)$.
Developing the equations of a centralized Luenberger predictor for the enlarged state in equation (4.4), we obtain the following equation:

$$\hat{\boldsymbol{x}}(k+1) = \bar{A}\hat{\boldsymbol{x}}(k) + \bar{L}(y(k) - \bar{C}\hat{\boldsymbol{x}}(k)) \tag{4.7}$$

Gain matrix $\bar{L} \in \mathbb{R}^{p\times(n+n_f)}$ of the enlarged system can be computed by means of the LMI method introduced in Section 2.2.2, which guarantees Schur stability of matrix $\bar{F} = (\bar{A} - \bar{L}\bar{C})$. Note that since we are dealing with the centralized case, it is not necessary to enforce a fixed block structure to $\bar{L}$.

4.3 Observer feasibility conditions

The observability of the enlarged system $(\bar{A}, \bar{C})$ is a necessary condition to properly formulate the estimation problem and reconstruct the whole enlarged state vector $\hat{\boldsymbol{x}}(k)$. Thus, we test the observability of $(\bar{A}, \bar{C})$ by means of the PBH observability test, recalled below

Theorem 3 (PBH observability test)**.** *The generic system (A, C), where matrix $A \in \mathbb{R}^{n\times n}$ and matrix $C \in \mathbb{R}^{n\times p}$, is observable if and only if*

$$rank(\mathcal{O}(\lambda)) = n \qquad \forall\lambda \in \mathbb{C}$$

where $\mathcal{O}(\lambda)$ is defined as

$$\mathcal{O}(\lambda) = \begin{bmatrix} \lambda I - A \\ C \end{bmatrix}$$

and $I \in \mathbb{R}^{n\times n}$ is the identity matrix.

Note that $rank(\begin{bmatrix}\lambda I - A\end{bmatrix}) = n$ except for λ equal to the eigenvalues of A. It is then sufficient to check the rank of matrix $\mathcal{O}(\lambda)$ only for $\forall \lambda \in spec(A)$, where $spec(A)$ is the spectrum of A, i.e., the set of its eigenvalues.

It is straightforward to show that, in view of *Theorem 3*, the enlarged system (4.6) is observable if the following conditions are verified at the same time:

(i) the pair (A, C) is observable;

(ii) the system matrix

$$\mathcal{S}_f = \begin{bmatrix} I - A & B_f \\ C & D_f \end{bmatrix}$$

has full rank $n + n_f$.

This can be proved by writing the matrix $\mathcal{O}(\lambda)$ in case of system (4.6), i.e.

$$\mathcal{O}(\lambda) = \begin{bmatrix} \lambda I - \bar{A} \\ \bar{C} \end{bmatrix} \tag{4.8}$$

our problem reduces to checking the condition

$$rank(\mathcal{O}(\lambda)) = n + n_f \qquad \forall \lambda \in spac(\bar{A}) \tag{4.9}$$

Recalling that $\bar{A}$, as introduced in (4.5), is defined as

$$\bar{A} = \begin{bmatrix} A & B_f \\ 0 & I_{n_f} \end{bmatrix} \tag{4.10}$$

In view of its block-triangular structure

$$spec(\bar{A}) = spec(A) \cup \underbrace{\{1, \ldots, 1\}}_{n_f \text{ times}} \tag{4.11}$$

Moreover, in view of the definition of $\bar{A}$ and $\bar{C}$, we obtain that

$$\mathcal{O}(\lambda) = \begin{bmatrix} \lambda I - \begin{bmatrix} A & B_f \\ 0 & I_{n_f} \end{bmatrix} \\ \\ \begin{bmatrix} C & D_f \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \lambda I - A & -B_f \\ 0 & (\lambda - 1) I_{n_f} \\ C & D_f \end{bmatrix} \tag{4.12}$$

The rank of $\mathcal{O}(\lambda)$ is equal to the one of

$$\tilde{\mathcal{O}}(\lambda) = \begin{bmatrix} \begin{bmatrix} \lambda I - A \\ C \end{bmatrix} & \begin{matrix} -B_f \\ D_f \end{matrix} \\ \begin{matrix} 0 \end{matrix} & \begin{matrix} (\lambda - 1) I_{n_f} \end{matrix} \end{bmatrix} \tag{4.13}$$

- **For $\lambda \neq 1$**
  $rank((\lambda - 1)I_{n_f}) = n_f$ and, in view of the observability of (A, C)

$$rank \begin{bmatrix} \lambda I - A \\ C \end{bmatrix} = n \tag{4.14}$$

  Therefore $rank\tilde{\mathcal{O}}(\lambda) = n + n_f$ for $\lambda \neq 1$.

- **For $\lambda = 1$**

$$\tilde{\mathcal{O}}(1) = \begin{bmatrix} I - A & -B_f \\ C & D_f \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathcal{S}_f \\ 0 \end{bmatrix} \tag{4.15}$$

  clearly, if $rank(\mathcal{S}_f) = n_f + n$, then $rank(\tilde{\mathcal{O}}(1)) = n + n_f$.

Therefore, under the assumptions (i) and (ii), the observer gain $\bar{L}$ exists such that $(\bar{A} - \bar{L}\bar{C})$ is asymptotically stable.

Remark
Condition (ii) requires that no transmission zeroes exist in $z = 1$. It can be interpreted as the condition required (considering System (4.1) with $w(k) = 0$ and $v(k) = 0$ for all $k \geq 0$) for the uniqueness of the pair $\bar{x}, \bar{f}$ guaranteeing that, if $y(t) = \bar{y} \ \ \forall t$, in steady state $f(t) = \bar{f}$ and $x(k) = \bar{x}$.
In fact,in steady state

$$\begin{cases} \bar{x} = A\bar{x} + B_f\bar{f} \\ \bar{y} = C\bar{x} + D_f\bar{f} \end{cases} \tag{4.16}$$

In matrix form,

$$\begin{bmatrix} I - A & -B_f \\ C & D_f \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{f} \end{bmatrix} = \mathcal{S}_f \begin{bmatrix} \bar{x} \\ \bar{f} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{y} \end{bmatrix} \tag{4.17}$$

A unique solution $\bar{x}, \bar{f}$ can be found provided that $\mathcal{S}_f$ has full column rank $n + n_f$. A necessary condition for this to occur is that the number of columns of $\mathcal{S}_f$ smaller or equal than the number of rows, i.e.,

$$n + n_f \leq n + p$$

i.e., that

$$p \geq n_f \tag{4.18}$$

This is a fundamental limitation of any fault detection scheme, requiring that the number of the modelled faults is not larger than the number of the available measures.

4.4 Estimation error of the enlarged observer

The dynamics of the estimation error of the enlarged system $(\bar{A}, \bar{C})$, namely $\bar{e}(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)$, is given by

$$\bar{e}(k+1) = (\bar{A} + \bar{L}\bar{C})\bar{e}(k) - \bar{L}v(k) + \tilde{w}(k) \tag{4.19}$$

The covariance of $\bar{e}(k)$ is given by

$$\begin{aligned} \bar{P}(k) &= \mathbb{E}[(\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k))(\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k))^T] \\ &= \mathbb{E}\left[\begin{bmatrix} x(k) - \hat{x}(k) \\ f(k) - \hat{f}(k) \end{bmatrix} \begin{bmatrix} (x(k) - \hat{x}(k))^T & (f(k) - \hat{f}(k))^T \end{bmatrix}\right] \end{aligned} \tag{4.20}$$

where matrix $\bar{P}(k) \in \mathbb{R}^{(n+n_f)\times(n+n_f)}$, and evolves according to

$$\bar{P}(k+1) = (\bar{A} - \bar{L}\bar{C})\bar{P}(k)(\bar{A} - \bar{L}\bar{C})^T + \tilde{Q} + \bar{L}R\bar{L}^T \tag{4.21}$$

and where, in view of the definition of $\tilde{w}(k)$, $\tilde{Q}$ is defined as

$$\tilde{Q} = \begin{bmatrix} I \\ 0 \end{bmatrix} Q \begin{bmatrix} I & 0 \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \tag{4.22}$$

meaning that the covariance function of the noise affecting the state $x(k)$ directly contributes only in the definition of the covariance of $\hat{x}(k)$ and not of the one of $\hat{f}(k)$.
Note that $\bar{P}$ is the actual variance of the estimation error (i.e., the one introduced in (2.7)) and not an analytical upper bound, since in this section we are considering the fully centralized case.
In particular, referring to equation (4.20), the covariance function estimation error associated to the fault vector, namely $\bar{P}_{ff}(k)$ is expressed by

$$\begin{aligned} \bar{P}_{ff}(k) &= \begin{bmatrix} 0 & I_{n_f} \end{bmatrix} \bar{P}(k) \begin{bmatrix} 0 \\ I_{n_f} \end{bmatrix} \\ &= \mathbb{E}\left[(f(t) - \hat{f}(k))(f(t) - \hat{f}(k))^T\right] \end{aligned} \tag{4.23}$$

Moreover, given that in nominal conditions $f(t) = 0$, equation (4.23) reduces to

$$\begin{aligned} \bar{P}_{ff}(k) &= \mathbb{E}\left[\hat{f}(k)\hat{f}(k)^T\right] \\ &= var(\hat{f}(k)) \end{aligned} \tag{4.24}$$

4.5 Fault isolation test

Once the diagonal entries of matrix $\bar{P}_{ff}(k)$ are known, namely $(\sigma_{\hat{f},i}(k))^2$ (i.e., the variance of the i-th element of vector $\hat{f}(k)$) we have all the ingredients to define the following fault isolation test

$$\begin{cases} |\hat{f}_i(k)| < \rho_i(k), & \text{no fault } f_i \text{ is detected} \\ |\hat{f}_i(k)| \geq \rho_i(k), & \text{fault } f_i \text{ is affecting our system} \end{cases} \tag{4.25}$$

The elements $\rho_i(k)$ are the fault isolation thresholds, define as

$$\rho_i(k) = \alpha\sigma_{\hat{f},i}(k) \qquad \forall i = 1, \ldots, n_f$$

where α is defined based on the required FA rate.

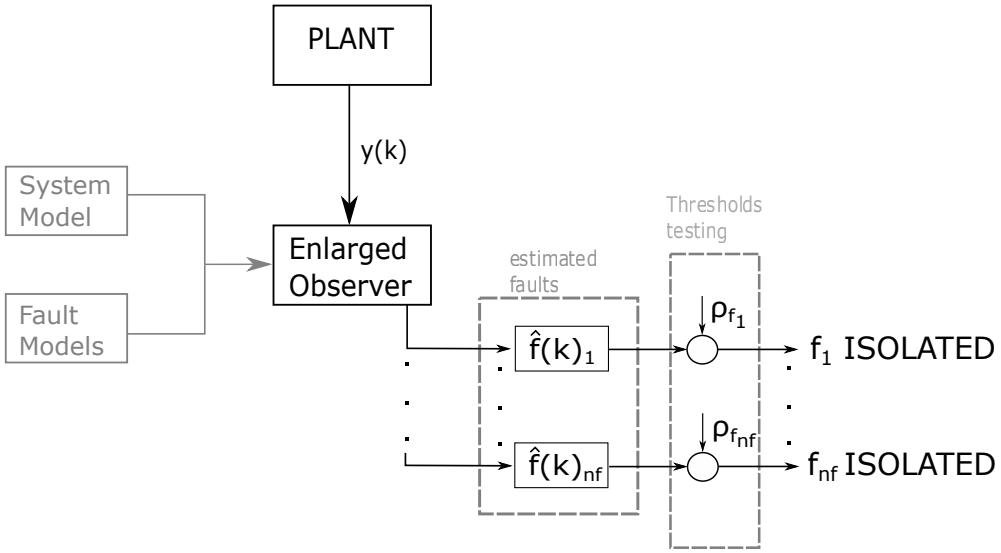

Figure 4.1: General behaviour of the Fault Isolation algorithm

The main principles which allow this FDI algorithm to correctly isolate faults are the following:

- A good mathematical model of a fault generates a perfect matching between the faulty system and and the associated faulty system mathematical model;
- $\hat{f}_i(k)$ in nominal conditions behave as a zero mean variable with variance $(\sigma_{\hat{f},i}(k))^2$;

- When a fault $\bar{f}_i$ affects our system, the estimated variable $\hat{f}_i(k)$, since its structure is fixed and equal to the one of $\bar{f}_i$, will assimilate all the main dynamics introduced by the fault.

When the value of an estimated fault $\hat{f}_i(k)$ overcomes its associated threshold, we can assert, with confidence level of $(1 - FA)$, that exactly a fault f_i is affecting our system.

4.6 Distributed fault isolation algorithm

A distributed version of the algorithm introduced in the previous sections could merge the capabilities of the proposed Fault isolation method, together with the advantages of a partitioned configuration. Since the proposed fault isolation algorithm follows the main ideas of the FD method in [1], which has been used in our FD simulations, it is possible to derive a distributed version following the main steps and considerations already introduced for FD case.

4.6.1 Distributed system model including modelled faults

The general scheme of a linear discrete-time distributed system where faults are absent, characterized by the interconnection of M subsystems, as already introduced in (2.1), is the following

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k+1) + \sum_{j\in\tilde{\mathcal{N}}_i} A_{ij}x_j(k) + w_i(k) \\ y_i(k) &= C_i x_i(k) + v_i(k) \end{aligned} \tag{4.26}$$

Vectors $x_i(k) \in \mathbb{R}^{n_i}$ and $y_i(k) \in \mathbb{R}^{p_i}$ are the state, and the output, of the i-th subsystem for $i = 1, \ldots, M$ respectively. Moreover we recall the definition of set of the predecessors $\mathcal{N}_i = \{j | A_{ij} \neq 0\}$, the set of the strict predecessors $\tilde{\mathcal{N}}_i = \{j | A_{ij} \neq 0, \;\; j \neq i\}$, and set of the successors $\mathcal{S}_i = \{j | i \in \mathcal{N}\}$ of subsystem i. Grouping system equations (4.26) for all $i = 1, \ldots, M$, the following collective model (as the one in (2.2)) is obtained.

$$\begin{aligned} x(k+1) &= Ax(k+1) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{4.27}$$

Collective vectors are defined as

$$x(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_M(k) \end{bmatrix} \in \mathbb{R}^n \qquad \text{where} \quad n = \sum_{i=1}^{M} n_i$$

$$y(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_M(k) \end{bmatrix} \in \mathbb{R}^p \qquad \text{where} \quad p = \sum_{i=1}^{M} p_i$$

Similarly, $w^T(k) = [w_1^T(k) \dots w_M^T(k)]^T \in \mathbb{R}^n$ and $v^T(k) = [v_1^T(k) \dots v_M^T(k)]^T \in \mathbb{R}^p$. Moreover the system matrices have the following block structure:

$$A = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MM} \end{bmatrix} \qquad C = \begin{bmatrix} C_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_M \end{bmatrix}$$

The main idea introduced with the fault isolation algorithm in this section is to include, in the definition of the system equations, the contribution of modelled faults acting on the system.
Considering the collective system in (4.27), we can directly add the contribution of a fault vector $f(k)$ as done for the centralized case in system (4.1).

$$\begin{aligned} x(k+1) &= Ax(k+1) + B_f f(k) + w(k) \\ y(k) &= Cx(k) + D_f f(k) + v(k) \end{aligned} \tag{4.28}$$

To bring us back to a partitioned model, some further considerations on fault vector $f(k)$, and matrix B_f.

- Matrix $B_f \in \mathbb{R}^{n \times n_f}$, entailing the relationships between faults and associated states has the same block structure of matrix A, i.e.,

  $$B_f = \begin{bmatrix} B_{11} & \dots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \dots & B_{MM} \end{bmatrix}$$

  In particular sub-blocks $B_{f_{ij}}$ have the following properties: $B_{f,ij} \in \mathbb{R}^{n_j \times n_{f_i}}$, and $B_{f_{ij}} \neq 0$ only if $A_{ij} \neq 0$, for all $i, j = 1, \dots, M$.

- Similarly matrix $D_f \in \mathbb{R}^{p \times n_f}$, entailing the relationships between modelled faults, and associated outputs, must have the same diagonal block structure of collective matrix C.

  $$D_f = \begin{bmatrix} D_{f_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & D_{f_M} \end{bmatrix}$$

  where $D_{f_i} \in \mathbb{R}^{p_i \times n_{f_i}}$.

All these considerations lead us to the definition of the following distributed system model including the contribution of modelled faults.

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k+1) + B_{f_{ii}}f_i(k) + \sum_{j\in\tilde{\mathcal{N}}_i}(A_{ij}x_j(k) + B_{f_{ij}}f_j(k)) + w_i(k) \\ y_i(k) &= C_i x_i(k) + D_{f_i}f_i(k) + v_i(k) \end{aligned} \tag{4.29}$$

4.6.2 Enlarged distributed system and Luenberger predictor

Following the same considerations made in Section 4.2, we consider fault subvectors $f_i(k)$ as constant variables of our system.

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k+1) + B_{f_{ii}}f_i(k) + \sum_{j\in\tilde{\mathcal{N}}_i}(A_{ij}x_j(k) + B_{f_{ij}}f_j(k)) + w_i(k) \\ f_i(k+1) &= f_i(k) \end{aligned} \tag{4.30}$$

The whole distributed model can be rewritten in the following matricial form including $f(k)$ as a state variable of our system

$$\begin{aligned} \begin{bmatrix} x_i(k+1) \\ f_i(k+1) \end{bmatrix} &= \begin{bmatrix} A_{ii} & B_{f_i} \\ 0 & I_{n_{f_i}} \end{bmatrix} \begin{bmatrix} x_i(k) \\ f_i(k) \end{bmatrix} + \sum_{j\in\tilde{\mathcal{N}}_i} \begin{bmatrix} A_{ij} & B_{f_j} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_j(k) \\ f_j(k) \end{bmatrix} + \begin{bmatrix} I_{n_i} \\ 0 \end{bmatrix} w_i(k) \\ y_i(k) &= \begin{bmatrix} C_i & D_{f_i} \end{bmatrix} \begin{bmatrix} x_i(k) \\ f_i(k) \end{bmatrix} + v_i(k) \end{aligned} \tag{4.31}$$

We denote with $\boldsymbol{x}_i(k)$ the enlarged state related to the i-th subsystem, including both the state and the fault vector

$$\boldsymbol{x}_i(k) = \begin{bmatrix} x_i(k) \\ f_i(k) \end{bmatrix} \tag{4.32}$$

$\bar{A}_{ii} \in \mathbb{R}^{(n_i+n_{f_i})\times(n_i+n_{f_i})}$, for $i = 1, \ldots, M$, is the enlarged sub-state block diagonal matrix

$$\bar{A}_{ii} = \begin{bmatrix} A_{ii} & B_{f_i} \\ 0 & I_{n_{f_i}} \end{bmatrix} \tag{4.33}$$

$\bar{A}_{ij} \in \mathbb{R}^{(n_i+n_{f_i})\times(n_i+n_{f_i})}$, for $i = 1, \ldots, M$ and $j \in \tilde{\mathcal{N}}_i$, is the enlarged sub-state matrix relating subsystem i with subsystem j

$$\bar{A}_{ij} = \begin{bmatrix} A_{ij} & B_{f_i} \\ 0 & 0 \end{bmatrix} \tag{4.34}$$

$\bar{C}_i \in \mathbb{R}^{(n_i+n_{f_i})\times p_i}$, for $i = 1, \ldots, M$, is the enlarged sub-output matrix

$$\bar{C}_i = \begin{bmatrix} C_i & D_{f_i} \end{bmatrix} \tag{4.35}$$

Vector $\tilde{w}_i(k) = \begin{bmatrix} I_{n_i} \\ 0 \end{bmatrix} w_i(k)$ is the enlarged noise vector which acts actively on the sub-state $x_i(k)$ components of $\boldsymbol{x}_i(k)$ only.
The enlarged system model of the generic subsystem i in (4.31) can be rewritten in the form

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \bar{A}_{ii}\boldsymbol{x}_i(k) + \sum_{j\in\tilde{N}_i} \bar{A}_{ij}\boldsymbol{x}_j(k) + \tilde{w}_i(k) \\ y_i(k) &= \bar{C}_i\boldsymbol{x}_i(k) + v_i(k) \end{aligned} \tag{4.36}$$

Implementing the same distributed Luenberger predictor which was introduced in 2.2 for the enlarged distributed system (4.36) we obtain the following equations

$$\begin{aligned} \hat{\boldsymbol{x}}_i(k+1) &= \sum_{j\in\mathcal{N}_i} (\bar{A}_{ij}\hat{\boldsymbol{x}}_j(k) + L_{ij}(y_j(k) - \bar{C}_j\hat{\boldsymbol{x}}_i(k))) \\ \hat{y}_i(k) &= C_i\hat{\boldsymbol{x}}_i(k) \end{aligned} \tag{4.37}$$

Here $\hat{\boldsymbol{x}}_i(k)$ is the estimate of the enlarged state, while $\hat{y}_i$ is the estimated output of subsystem i.
As for the distributed filter introduced in Subsection 2.2, gain L_{ij} must be computed in order to guarantee the convergence of the solution, but also as to maintain the same block structure of the distributed system. Those properties can be both attained computing L_{ij} by means of the LMI based method explained in Section 2.2.2.

4.6.3 Estimation error

The estimation error related to the i-th subsystem is defined as

$$e_i(k) = \begin{bmatrix} \boldsymbol{x}_i(k) - \hat{\boldsymbol{x}}_i(k) \end{bmatrix} = \begin{bmatrix} x_i(k) - \hat{x}_i(k) \\ f_i(k) - \hat{f}_i(k) \end{bmatrix} \tag{4.38}$$

Since is not possible to compute the actual covariance matrix of $e_i(k)$, namely $P_i(k)$, in a distributed way, we rely on the analytical upper bound $B_i(k)$ of $P_i(k)$ in [1] as happened for the FD case in Section 2.
Similarly to (2.8), $B_i(k)$ is updated according to

$$B_i(k) = Q_i + \sum_{j\in\mathcal{N}_i} [(\tilde{\bar{A}}_{ij} - L_{ij}\tilde{\bar{C}}_j)B_i(k)(\tilde{\bar{A}}_{ij} - L_{ij}\tilde{\bar{C}}_j)^T + L_{ij}\tilde{R}_jL_{ij}^T] \tag{4.39}$$

where $\tilde{\bar{A}}_{ij} = \sqrt{\zeta_i}\bar{A}_{ij}$, $\tilde{\bar{C}}_i = \sqrt{\zeta_i}\bar{C}_i$, $\tilde{R}_i = \sqrt{\zeta_i}R_i$ and $\zeta_i = |\mathcal{S}_i|$ for all $i, j = 1, ..., M$. Moreover, Q_i and R_i are the covariance matrices of the noises on the i-th subsystem enlarged state (i.e., there is no contribution on fault vector) and output.
Following the same line of reasoning of Section 4.4, we have that the upper bound to the covariance of the i-th subsystem estimation error, associated to the fault vector only, namely $\bar{B}_{f_i}(k)$, is expressed by

$$B_{f_i}(k) = \begin{bmatrix} 0 & I_{n_{f_i}} \end{bmatrix} \bar{B}(k) \begin{bmatrix} 0 \\ I_{n_{f_i}} \end{bmatrix} \tag{4.40}$$

Moreover, since it is defined as an upper bound to the actual variance value, the following property holds

$$B_{f_i}(k) \geq \mathbb{E}\left[(f_i(t) - \hat{f}_i(k))(f_i(t) - \hat{f}_i(k))^T\right] \tag{4.41}$$

which in nominal conditions, i.e., for $f_i(t) = 0$, reduces to

$$\begin{aligned} B_{f_i}(k) &\geq \mathbb{E}\left[\hat{f}_i(k)\hat{f}_i(k)^T\right] \\ &\geq var(\hat{f}_i(k)) \end{aligned} \tag{4.42}$$

4.6.4 Distributed fault isolation test

Once the diagonal entries of matrix $B_{f_i}(k)$ are known, namely $(\sigma^B_{\hat{f},i,l}(k))^2$ (i.e., the upper bound of the variance of the l-th element of vector $\hat{f}_i(k)$) we have all the ingredients to set the following fault isolation test

$$\begin{cases} |\hat{f}_{i,l}(k)| < \rho_{i,l}(k), & \text{no fault } f_{i,l} \text{ is relevated} \\ |\hat{f}_{i,l}(k)| \geq \rho_i(k), & \text{fault } f_{i,l} \text{ is affecting subsystem } i \end{cases} \tag{4.43}$$

In particular, elements $\rho_{i,l}(k)$ represent our fault isolation thresholds, defined as

$$\rho_{i,l}(k) = \alpha\sigma^B_{\hat{f},i,l}(k) \qquad \forall i = 1, \ldots, M, \quad l = 1, \ldots, n_{f_i}$$

where α is defined based on the required FA rate.

Chapter 5

Application of the novel fault isolation algorithm to selected case studies

In this chapter we show some significant simulation results obtained applying the fault isolation algorithm described in Section 4 on both case studies analysed in Section 3, i.e. the chemical plant and the power network system. Both the centralized and the distributed version of the FI algorithm have been implemented for each system. Moreover, in order to compare the performances of the two algorithms (i.e., the FD and the FI algorithms), the same faults introduced in Section 3 have been tested.

5.1 Chemical plant

5.1.1 Fault model and enlarged observer

In Section 3 a persistent additive fault acting on the 4^{th} state equation of the second subsystem was considered, i.e., the temperature T_2 of the second chemical reactor. Thus, the simplest way to model such fault is to introduce a single fault variable $f(k)$ in the state equation governing T_2.

Centralized case

In the centralized case, the enlarged model of the chemical system is the following

$$\begin{cases} \begin{bmatrix} x(k+1) \\ f(k+1) \end{bmatrix} = \begin{bmatrix} A & B_f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ f(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \bar{u} + \begin{bmatrix} I_n \\ 0 \end{bmatrix} w(k) \\ y(k) = \begin{bmatrix} C & D_f \end{bmatrix} \begin{bmatrix} x(k) \\ f(k) \end{bmatrix} + v(k) \end{cases} \tag{5.1}$$

In particular, we have $n_f = 1$, since we model a single fault. Moreover,

$$B_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.2}$$

selects the 8^{th} row of matrix A.

$$D_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.3}$$

since the fault does not directly affect the output equations.
The centralized Luenberger observer has the structure introduced in (4.7). In particular $\bar{L} \in \mathbb{R}^{13\times 8}$ has been computed by means of the LMI method in Section 2.2.2. Thus, Schur stability of matrix $\bar{F} = (\bar{A} - \bar{L}\bar{C})$ is guaranteed, but no constraints on its structure have been added.

Distributed case

In the distributed case, the only subsystem which is modified by the introduction of the fault model is the second. Its enlarged model is the following

$$\begin{cases} \boldsymbol{x}_2(k+1) = \bar{A}_{22}\boldsymbol{x}_2(k) + \bar{A}_{12}\boldsymbol{x}_1(k) + B_2\bar{u} + \tilde{v}_2(k) \\ y_2(k) = \bar{C}_2\boldsymbol{x}_2(k) + w_2(k) \end{cases} \tag{5.4}$$

In particular, recalling matrices $\bar{A}_{ii}$ and $\bar{A}_{ij}$ structure we have

$$\bar{A}_{22} = \begin{bmatrix} A_{22} & B_{f_2} \\ 0 & 1 \end{bmatrix} \quad \text{with } B_{f_2} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5.5}$$

$$\bar{A}_{21} = \begin{bmatrix} A_{21} & B_{f_1} \\ 0 & 0 \end{bmatrix} \quad \text{with } B_{f_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.6}$$

Moreover

$$\bar{C}_2 = \begin{bmatrix} C_2 & D_{f_2} \end{bmatrix} \quad \text{with } D_{f_2} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{5.7}$$

The associated distributed Luenberger gains L_{ij} have been computed by means of the LMI based method explained in Section 2.2.2. Differently from the centralized

case, L_{ij} have been computed in order to keep the same block structure of the collective system.

Note that centralized and the distributed Luenberger gains, namely L_c and L_d, have been computed solving different LMI. Thus, we expect a different behaviour between the two the estimation processes and no strict correlation between the associated analytical thresholds.

5.1.2 Simulation results

In this section the simulation results obtained with the chemical plant model, both centralized and distributed, are reported.
A persistent additive fault $\bar{f}$ occurs at time step $k = 100$. In particular a fault which corresponds to $\sim 1\%$ of the value taken, in normal working conditions, by the state variable directly affected, in this case T_2, is considered.

$$\bar{f}_2 = 0.01\bar{T}_2 = 3.1478K \tag{5.8}$$

Isolation thresholds are computed based on $\sigma_{\hat{f},i,l}$ and $\sigma^B_{\hat{f},i,l}$ as in Equations (4.25) and 4.6.4. In particular, once the method converges, we have

- Fully centralized case considering L_c

$$(\sigma_{\hat{f},i,l})^2 = 0.0228 \tag{5.9}$$

- Distributed case considering L_d partitioned

$$\begin{aligned} (\sigma_{\hat{f},i,l})^2 &= 0.0174 \quad \text{actual variance} \\ (\sigma^B_{\hat{f},i,l})^2 &= 0.0705 \quad \text{analytical upper bound} \end{aligned} \tag{5.10}$$

Considering FA=0.002, i.e., same FA rate imposed in Section 3, which means a value of $\alpha = 3.7190$, the threshold for the centralized case is

$$\rho_{i,l} = \alpha\sigma_{\hat{f},i,l} = 0.5614 \tag{5.11}$$

while the one in the distributed case is

$$\rho^B_{i,l} = \alpha\sigma^B_{\hat{f},i,l} = 0.9878 \tag{5.12}$$

For each proposed case we display two types of diagrams, recalling the ones in Section 3.1.3.
Figures 5.1 and 5.3 show the values taken by the estimated fault vector $\hat{f}(k)$ at each simulation time instant, compared to the corresponding thresholds and to the actual value of fault $\bar{f}$ considering 30 Montecarlo runs.
Figures 5.2 and 5.4, instead, show the rate of systems in which fault f is isolated (i.e., systems where $\hat{f}$ exceeds the associated threshold) for each time step k. These results are obtained considering 1000 Montecarlo runs.

- **Centralized case**

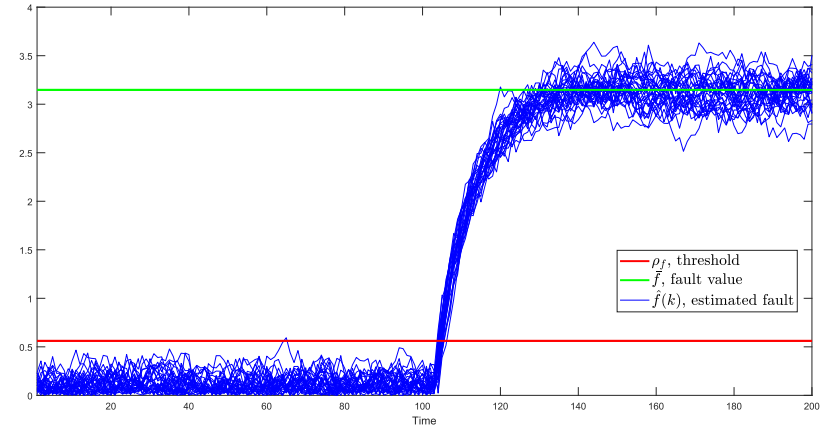

Figure 5.1: Centralized model, behaviour of the estimated vector $\hat{f}$, compared to threshold $\rho_{i,l}$ and the actual value taken by the fault affecting the system $\bar{f}$

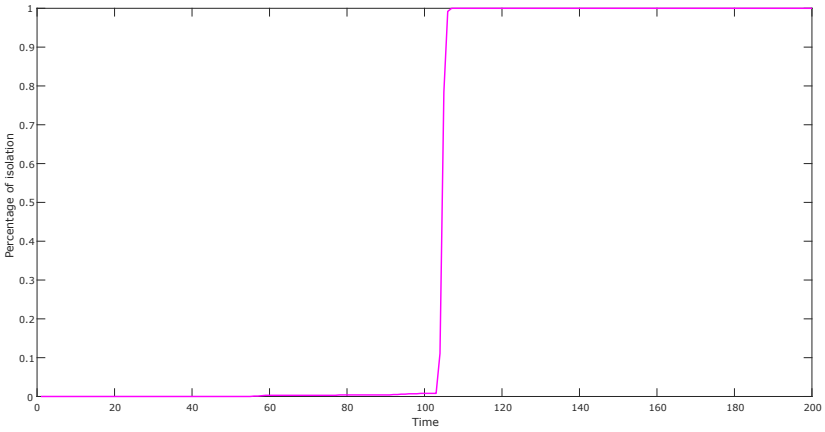

Figure 5.2: Centralized model, percentage of systems in which fault $\bar{f}$ is isolated in each simulation time step

- **Distributed case**

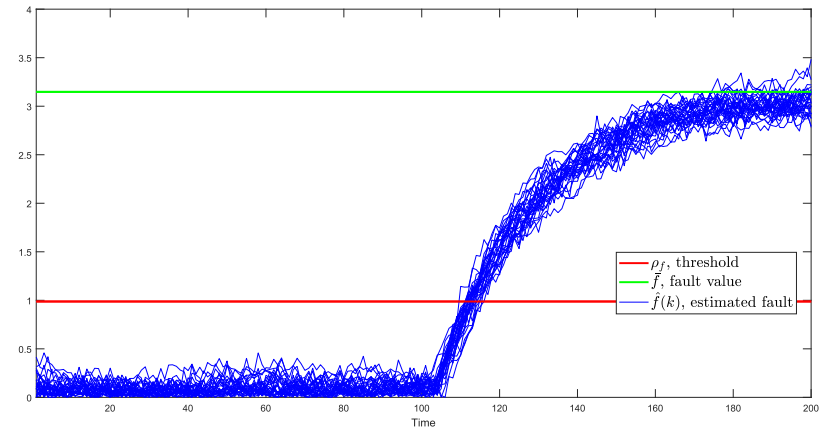

Figure 5.3: Distributed model, behaviour of the estimated vector $\hat{f}$, compared to threshold $\rho_{i,l}$ and the actual value taken by the fault affecting the system $\bar{f}$

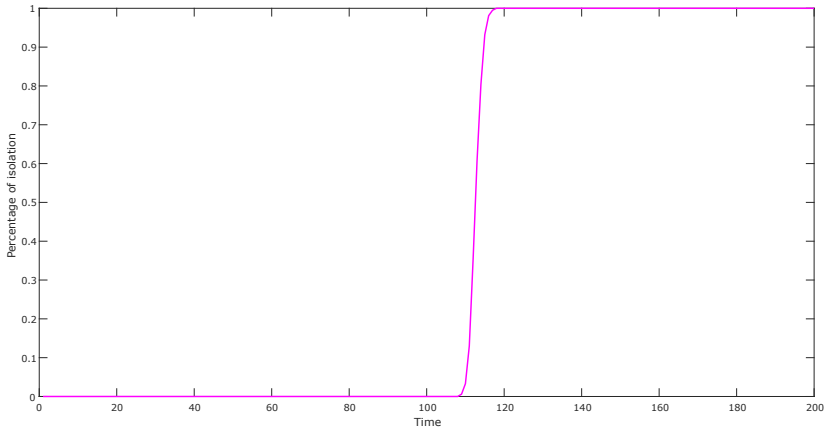

Figure 5.4: DIstributed model, percentage of systems in which fault $\bar{f}$ is isolated in each simulation time step

5.1.3 Comparison between the performances of the FD and FI algorithms

In this section the performances of the distributed FD algorithm and of the distributed FI algorithm are compared.

	Systems detecting Fault in	10 k-steps	50 k-steps	150 k-steps
FD	m=1	93.0%	100%	100%
	m=20	0%	88.6%	100%
	m=40	0%	100%	100%
FI	Centralized	99.6%	100%	100%
	Distributed	15.8%	100%	100%

Table 5.1: Percentage of systems in which is declared fault, after 10, 50 and 150 time steps after the actual fault occurrence

From Table 5.1 we clearly see that the novel FI algorithm detects the presence of a modelled fault already after 10 steps. The main difference between the performances of the distributed and of the centralized versions are due to the choice of the observer gains. Moreover, as we can see in Figures 5.1 and 5.3, a few time steps after the fault is isolated, the value of the estimated fault $\hat{f}$ converges to the actual value of $\bar{f}$.

5.2 Power network system

5.2.1 Fault model and enlarged observer

In Section 3, dealing with FD for a power network system, was considered a persistent additive fault occurring on the 3^{rd} state equation of the second subsystem.

Centralized case

In the standard centralized case, the enlarged model of the power network system is the following

$$\begin{cases} \begin{bmatrix} x(k+1) \\ f(k+1) \end{bmatrix} = \begin{bmatrix} A_d & B_f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ f(k) \end{bmatrix} + \begin{bmatrix} \tilde{B}_d \\ 0 \end{bmatrix} \bar{u} + \begin{bmatrix} I_n \\ 0 \end{bmatrix} w(k) \\ y(k) = \begin{bmatrix} C_d & D_f \end{bmatrix} \begin{bmatrix} x(k) \\ f(k) \end{bmatrix} + v(k) \end{cases} \tag{5.13}$$

In particular, we have $n_f = 1$, since we model a single fault. Moreover,

$$B_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.14}$$

selects only the 3^{rd} state equation of the second subsystem, i.e., the 7^{th} row of matrix A.

$$D_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.15}$$

since our fault does not directly affect the output equations.
The centralized Luenberger observer have the structure introduced in (4.7). In particular $\bar{L} \in \mathbb{R}^{21 \times 10}$ has been computed by means of the LMI method in subsection 2.2.2. Thus, Schur stability of matrix $\bar{F} = (\bar{A} - \bar{L}\bar{C})$ is guaranteed, but no constraints on its structure have been added.

Distributed case

In the distributed case, the only subsystem which is modified by the introduction of the fault model is the second. Its enlarged model is the following

$$\begin{cases} \boldsymbol{x}_2(k+1) = \bar{A}_{d,22}\boldsymbol{x}_2(k) + \bar{A}_{d,21}\boldsymbol{x}_1(k) + \bar{A}_{d,23}\boldsymbol{x}_3(k) + \bar{A}_{d,25}\boldsymbol{x}_5(k) + \tilde{B}_{d,2}\bar{u}_2 + \hat{w}_2(k) \\ y_2(k) = \bar{C}_{d,2}\boldsymbol{x}_2(k) + v_2(k) \end{cases} \tag{5.16}$$

In particular, recalling matrices $\bar{A}_{ii}$ and $\bar{A}_{ij}$ structure we have

$$\bar{A}_{22} = \begin{bmatrix} A_{d,22} & B_{f_2} \\ 0 & 1 \end{bmatrix} \quad \text{with } B_{f_2} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{5.17}$$

and for $j = 1, 3, 5$

$$\bar{A}_{d,2j} = \begin{bmatrix} A_{2j} & B_{f_j} \\ 0 & 0 \end{bmatrix} \quad \text{with } B_{f_j} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.18}$$

Moreover

$$\bar{C}_{d,2} = \begin{bmatrix} C_{d,2} & D_{f_2} \end{bmatrix} \quad \text{with } D_{f_2} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{5.19}$$

The associated distributed Luenberger gains L_{ij} have been computed by means of the LMI based method explained in Section 2.2.2. Differently from the purely centralized case, L_{ij} have been computed in order to keep the same block structure of the collective system.

Also in this case, the centralized and the distributed Luenberger gains, namely L_c and L_d, have been computed solving different LMI. Thus, we expect a different behaviour between the two estimation processes and no strictly correlation between the associated analytical thresholds.

5.2.2 Simulation results

In this section the simulation results obtained with the power network system model, both centralized and distributed, are reported.
A persistent additive fault $\bar{f}$, of amplitude 1, occurs at time step $k = 100$.
Isolation thresholds are computed based on $\sigma_{\hat{f},i,l}$ and $\sigma^B_{\hat{f},i,l}$ as in equations 4.25 and 4.6.4. In particular, once the method converges, we have

- Fully centralized case considering L_c

$$(\sigma_{\hat{f},i,l})^2 = 3.2987 \times 10^{-5} \tag{5.20}$$

- Distributed case considering L_d partitioned

$$\begin{aligned} (\sigma_{\hat{f},i,l})^2 &= 1.9315 \times 10^{-6} \quad \text{actual variance} \\ (\sigma^B_{\hat{f},i,l})^2 &= 1.8563 \times 10^{-5} \quad \text{analytical upperbound} \end{aligned} \tag{5.21}$$

Considering FA=0.002, i.e., same FA rate imposed in Section 3, which means a value of $\alpha = 3.7190$, the threshold for the centralized case is

$$\rho_{i,l} = \alpha \sigma_{\hat{f},i,l} = 0.0213 \tag{5.22}$$

while the one in the distributed case is

$$\rho^B_{i,l} = \alpha \sigma^B_{\hat{f},i,l} = 0.0160 \tag{5.23}$$

For each proposed case we display two types of diagrams, recalling the ones in Section 3.1.3.
Figures 5.5 and 5.7 show the values taken by the estimated fault vector $\hat{f}(k)$ at each simulation time instant, compared to the corresponding thresholds and to the actual value of fault $\bar{f}$ considering 30 Montecarlo runs.
Figures 5.6 and 5.8, instead, show the rate of systems in which fault f is isolated (i.e., systems where $\hat{f}$ exceeds the associated threshold) for each time step k. These results are obtained considering 1000 Montecarlo runs.

- **Centralized case**

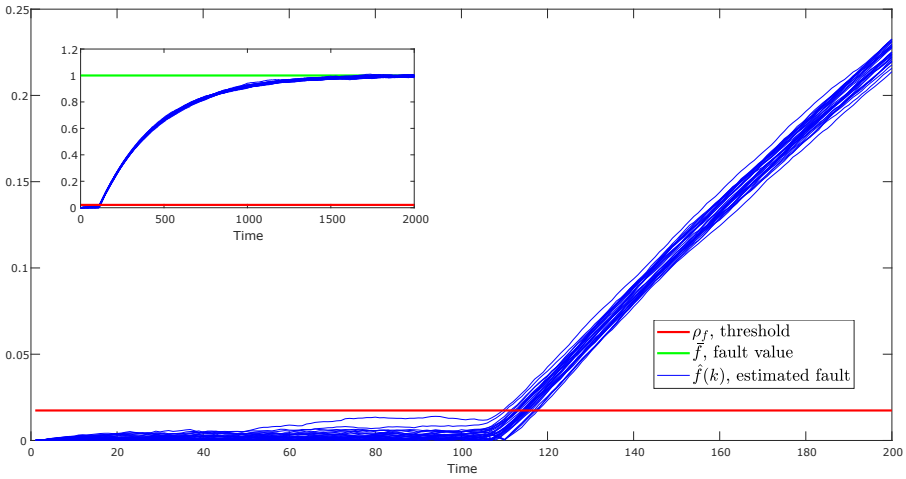

Figure 5.5: Centralized model, behaviour of the estimated vector $\hat{f}$, compared to threshold $\rho_{i,l}$ and the actual value taken by the fault affecting the system $\bar{f}$

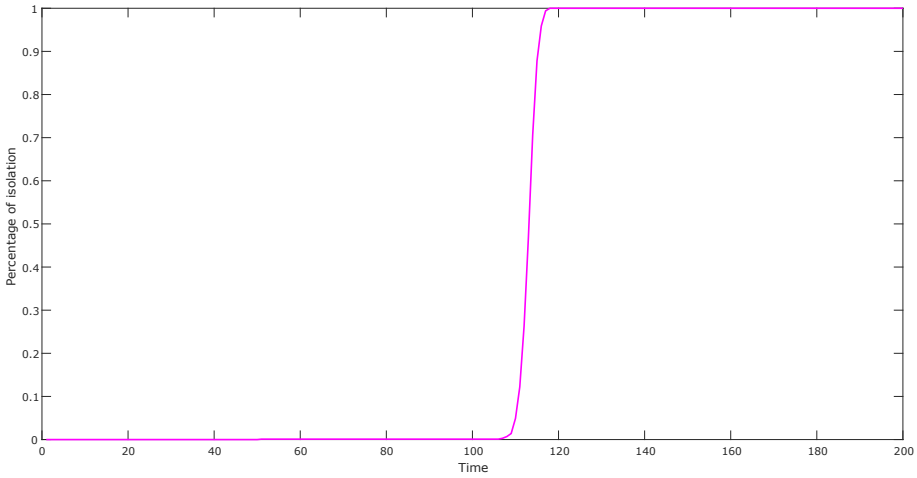

Figure 5.6: Centralized model, percentage of systems in which fault $\bar{f}$ is isolated in each simulation time step

- **Distributed case**

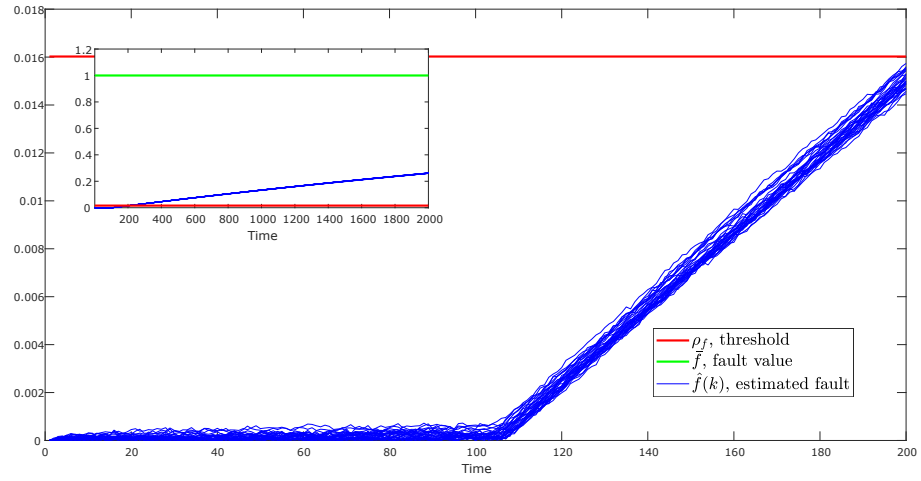

Figure 5.7: Distributed model, behaviour of the estimated vector $\hat{f}$, compared to threshold $\rho_{i,l}$ and the actual value taken by the fault affecting the system $\bar{f}$

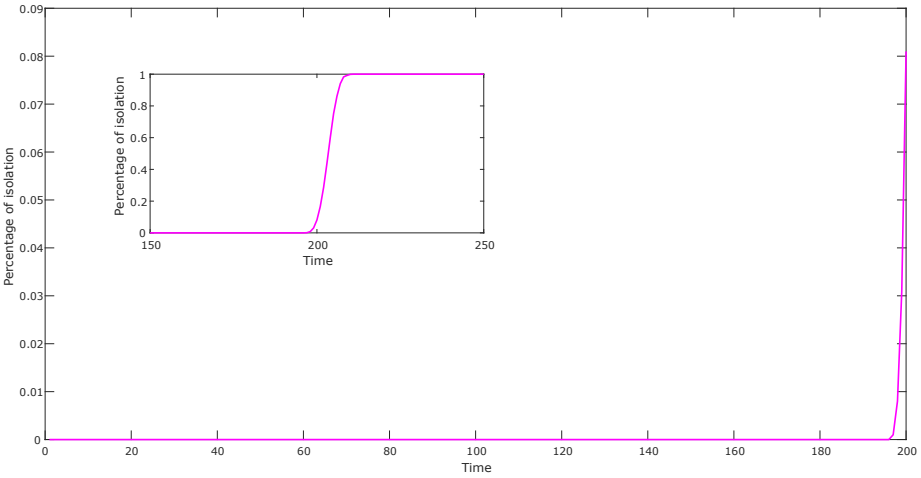

Figure 5.8: DIstributed model, percentage of systems in which fault $\bar{f}$ is isolated in each simulation time step

5.2.3 Comparison between FD and FI performances

In this section the performances of the distributed FD algorithm and of the distributed FI algorithm are compared.

	Systems detecting Fault in	10 k-steps	50 k-steps	150 k-steps
FD	m=1	3.2%	24.8%	62.6%
	m=20	0%	0%	0%
	m=40	0%	100%	100%
FI	Centralized	4.8%	100%	100%
	Distributed	0%	0%	100%

Table 5.2: Percentace of systems in which is declared fault 10, 50 and 150 time steps the actual fault occurrence

Considering table 5.2 we see that, in general, the FI method seem to be more reliable than the FD one. In particular, referring to the distributed case, the large time delay in the detection is mostly related to the choice of the L matrix, which can be improved by imposing a dedicated cost function in the LMI problem.

Conclusions and future perspectives

In this Thesis a recently-proposed scheme for model-based observer-based fault detection has been tested. Also, a new observer-based scheme for fault isolation, based on fault models, has been introduced, described and tested.
Both algorithms have been designed to be implemented in a distributed fashion based on partitioned system models, in order to be used in a large scale system scenario.
After we implemented the algorithm in [1] for two selected case studies (the chemical plant in [11] and the power network system in [3]), it has been possible to analyse the capabilities of the FD method. In particular, the simulations results have shown how the intrinsic properties of the system, affecting thresholds computation, influence the performance of the different testing methods, i.e., single residual approach and moving window average of the residuals. In addition, a possible approximated test guaranteeing the effectiveness of a moving window approach, which is directly based on system parameters, has been proposed.
In the second part of the thesis, a novel model based algorithm for fault isolation has been developed. We started with the development of the algorithm in the centralized case, with special focus on the structural properties ensuring the feasibility of the enlarged state observer. Then, on the basis of the studies of distributed FD in [1], we were able to propose also a distributed version of the algorithm. The novel FI method has also been tested on the chemical plant in [11] and the power network system in [3], offering the opportunity to compare the results with the ones obtained with the FD method.
Possible future developments of the work carried out in this Thesis rely on the possibility to combine both the FD and the FI methods in order to improve the whole detection process, and on the development of more detailed mathematical models of the faults, including the possibility to isolate also non constant fault signals.

Bibliography

[1] S. Attuati, M. Farina, F. Boem, and T. Parisini. "Reducing False alarm rates in observer based distributed fault detection schemes by analyzing moving averages". In: *IFAC PapersOnLine* 51.24 (2018), 473–479.

[2] Michèle Bassevile. "On Fault Detectability and Isolability". In: *European Journal of Control* 7 (2001), 625–637.

[3] F. Boem, R. Carli, M. Farina, G. Ferrari-Trecate, and T. Parisini. "Scalable Monitoring of Interconnected Stochastic Systems". In: *2016 IEEE 55th Conference on Decision and Control* (2016).

[4] M. Farina and R. Carli. "Partition-based Distributed Kalman Filter with plug and play features". In: *IEEE Transactions on control of network systems* 5.1 (2018), pp. 560–570.

[5] M. Farina, P. Colaneri, and R. Scattolini. "Block-wise discretization accounting for structural constraints". In: *Automatica* 49 (2013), pp. 3411–3417.

[6] Janos J. Gertler. *Fault detection and diagnosis in engineering systems*. 1998.

[7] Janos J. Gertler. "Survey of model-based failure detection and isolation in complex plants". In: *IEEE Control Systems Magazine* 8.6 (1988), pp. 3–11.

[8] M.Lauricella, M.Farina, R.Schneider, and R.Scattolini. "A distributed fault detection and isolation algorithm based on moving horizon estimation". In: *IFAC PapersOnLine* 50.1 (2017), 15259–15264.

[9] H. Saadat. *Power System Analysis*. 2nd. McGraw-Hill Series in Electrical and Computer Engineering, 2002.

[10] *Smart Grid*. CLP Group. URL: `https://www.clp.com.hk/en/about-clp/power-transmission-and-distribution/smart-grid`.

[11] B. T. Steward, A. N. Venkat, J. Rawlings, S. J. Wright, and G. Pannocchia. "Cooperative distributed model predictive control". In: *Systems and Control Letters* (2010).

[12] Arne Wahrburg, Dominik Haumann, and Volker Willert. "Minimum-Variance Fault Isolation Observers for Discrete-Time linear systems". In: *52nd IEEE Conference on Decision and Control* (2013).

[13] Feng Xu, Vicenç Puig, Carlos Ocampos-Martinez, Florin Stoican, and Sorin Olaru. "Improved Fault Detection and Isolation Strategy using Bank of Interval Observers". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 8024–8029.