FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multi-Agent Systems with Fault and Security Constraints

André Miguel Herdeiro Teixeira

Dissertation submitted to obtain the degree
Master of Science in Electrical and Computer Engineering
Major in Automation

Faculdade de Engenharia da Universidade do Porto

Supervisor: João Tasso de Figueiredo Borges de Sousa (Assistente Convidado)

KTH-Royal Institute of Technology

Supervisor: Karl Henrik Johansson (Associate Professor)

Co-supervisor: Henrik Sandberg (Research Associate)

June 2009

Resumo

Ao longo dos últimos anos tem ocorrido uma mudanca de paradigma em Sistemas de Controlo. Devido em parte ao aumento do poder computacional, existe agora a possibilidade de utilizar sensores e actuadores inteligentes, os quais podem tomar acções de forma distribuída. Deste modo deixa de haver necessidade de um controlador centralizado ou de uma estação de fusão de dados, o que é apropriado para Sistemas de Controlo por Rede. Este facto levou ao desenvolvimento de teorias para controlo distribuído e fusão de sensores, havendo ainda bastante investigação a ser realizada neste campo.

Como não existe mais uma estação central contendo toda a informação, mas ao invés temos diversos agentes a controlar ou monitorizar o sistema, a comunicação entre estes agentes torna-se essencial. Assim sendo, um aspecto importante deste tipo de sistemas é a segurança, não só relativamente às comunicações entre os agentes, mas também quanto à possibilidade de um mal funcionamento de um ou vários agentes.

O âmbito desta tese envolve o estudo de métodos de detecção de possíveis falhas ou quebras de segurança nas comunicações entre agentes, de um ponto de vista de Teoria de Controlo. Neste sentido, um controlador distribuído específico - o protocolo de consenso - será analisado sob o efeito de falhas e ataques nas comunicações. Será proposto um método de Detecção e Isolamento de Falhas (FDI) para detectar estes eventos, utilizando apenas a informação local disponível. Será também estudado um exemplo relacionado com Sistemas Eléctricos, onde um estimador descentralizado é utilizado para detectar falhas dentro de uma área do sistema, utilizando apenas informação local.

Abstract

Over the last few years, a change in one of the classical paradigms in Control Systems has been happening. Partially due to the increase of the computational power, there exists now the possibility of having intelligent sensors and actuators which may take actions in a distributed fashion. This way, there is no longer the need for a centralized controller or data fusion station, which is really well suited for Networked Control Systems. This led the way to the development of distributed control and sensor fusion frameworks and there is still a a great amount of research to be done in this field.

Since there is no longer a central station containing all the information but instead there are several distinct agents controlling or monitoring the system, communication between these agents is now necessary. Thus one important aspect in these kind of systems is security, regarding not only the communications between agents but also a possible malfunction of one or several agents.

The scope of this thesis is to study methods to detect possible misbehaviors and security breaches in the communications between agents from a control theoretical perspective. In this sense, a specific kind of distributed controller - the consensus problem - will be analyzed under the effect of faults and communication attacks. A Fault Detection and Isolation method will be used in order to detect these events using only the local information available. An example regarding power systems will also be given, where a decentralized state estimator is used in order to detect faults within an area, requiring only local information.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

CPS	Cyber-Physical System
DMS	Distribution Management System
DOS	Dedicated Observer Scheme
EMS	Energy Management System
FDI	Fault Detection and Isolation
GOS	Generalized Observer Scheme
LTI	Linear Time-Invariant
MAS	Multi-Agent System
NCS	Networked Control System
NMAS	Networked Multi-Agent System
UIO	Unknown Input Observer
SCADA	Supervisory Control And Data Acquisition

Chapter 1

Introduction

In this chapter, some motivation for the work done in this thesis is presented, beginning by stating several applications of Networked Control Systems and the advantages of having distributed control/estimation. This is followed by some considerations on what kind of security or fault constraints these systems are subject to. Finally we summarize the outline of the following chapters.

1.1 Networked Multi-Agent Systems

A classical paradigm within the design of monitoring and control systems is to use centralized solutions, even when the system itself is somehow distributed. This is the case of Networked Control Systems (NCS) and Multi-Agent Systems (MAS) - such as vehicle coordination, for instance, which we will consider to be different kinds of Networked Multi-Agent Systems (NMAS).

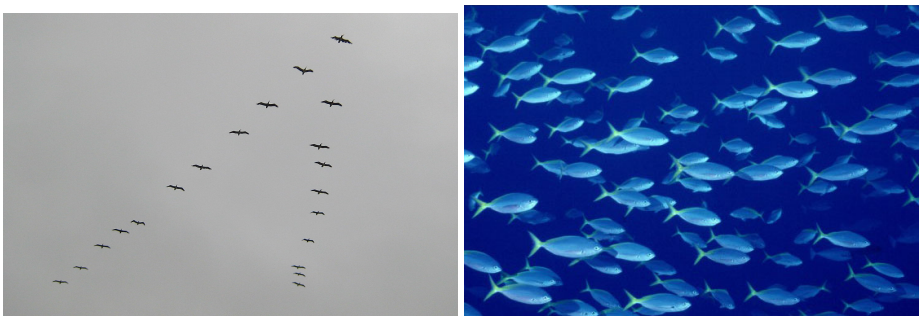
Figure 1.1: Central station of a SCADA system.

In fact, most of the industrial plants nowadays have their decision making processes carried out at a central station, which receives all the relevant data, takes a decision and finally transmits the outcome of such decision to the rest of the system. This is the case of most Supervisory Control And Data Acquisition (SCADA) systems used in industrial applications and as a particular case we have the monitoring and control of an electrical power grid - known as SCADA/EMS and SCADA/DMS. Although it is a very complex and geographically distributed system, almost all the monitoring and supervisory control are made at a central station. The functions of such central station include receiving several measurements from the distribution network, evaluating the state of the system and, according to the actual state, sending an appropriate control signal - which may be the opening of a breaker or a change in the power production setpoint, among others.

This paradigm is also present in pure monitoring applications, such as the monitoring of the vibrations on buildings, where the raw data is gathered by a large set of distributed sensors and sent to a central station which the processes and analyzes it [2].

However, one can see that these centralized solutions impose quite a heavy burden on both the communications and the computations, since all the data has to be sent to one station and this station must perform all the computations by itself. Some other important issues are the robustness and flexibility of such systems - if the central station shuts down for some reason, the whole system collapses and if some new component has to be introduced in the system, the behavior of such central station may have to be modified to accommodate this change in the system.

Partially due to the previous reasons, recently there has been a substantial amount of work done in the sense of designing distributed solutions, therefore increasing the flexibility and robustness of the system and decreasing the overall computational burden on the several components of the system. Some of these research problems include sensor data fusion [3, 4], distributed control of multi-agent systems [5, 6] and several others.



(a) Formation of birds (b) Fish school

Figure 1.2: Examples of multi-agent systems in nature.

A nice inspiration for the research in the field of distributed control comes from several biological examples, some of which are depicted in Figure 1.2. In these examples, each animal has a limited amount of information, as it only senses what its neighbors are doing, and acts according to this information. Although this sensing limitation, some quite complex behaviors are

observable in these groups of animals, such as the movement of birds within a formation, shown in Figure 1.2(a). Part of this thesis will study one kind of distributed control system closely related to these examples - the consensus problem.

Security Issues in Networked Multi-Agent Systems

It has been shown that the interactions between the several agents of a NMAS play an important role on the stability and performance of the overall system [7] and that cooperation from each agent is indeed a necessary requirement in order to achieve a common goal. Thus, if a single agent misbehaves or if the interactions are somehow disturbed, the performance and stability of the whole system may be compromised - as it can be seen in the nature inspired example in Figure 1.3.

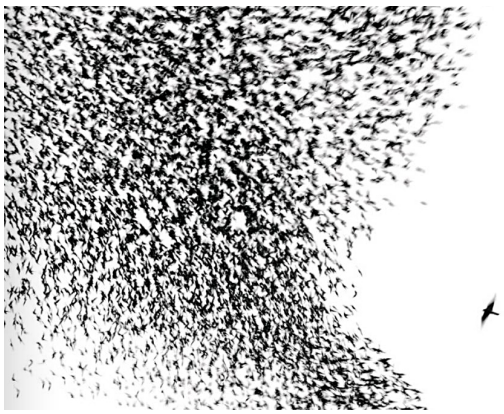


Figure 1.3: A flock of birds being attacked by a hawk.

There has been some work done in this field concerning security in sensor networks[8], secure computations of functions in networks[9, 10, 11], secure data aggregation[12], robust consensus algorithms [13, 14] and, more recently, security in Cyber-Physical Systems (CPS), such as the SCADA system[15, 16], which is now receiving a large amount of attention due to its vulnerability to cyber attacks from hackers. However, there are still many contributions to be made in this field, as it is a very broad one, covering several different areas [16].

The main scope of this thesis is to study these distributed control systems under the effect of physical faults and communication attacks, represented in Figure 1.4 by A5 and A1 to A4 respectively, formulating the effect of such events and providing methods to detect and isolate them in a distributed fashion.

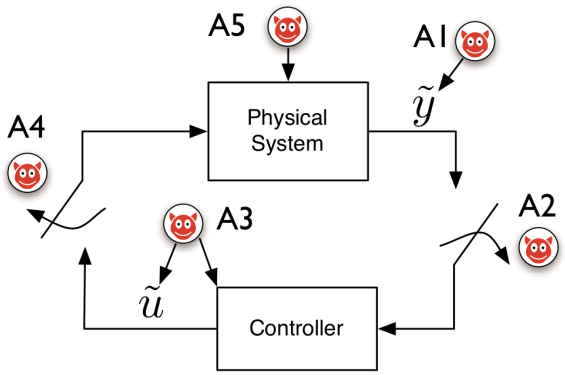
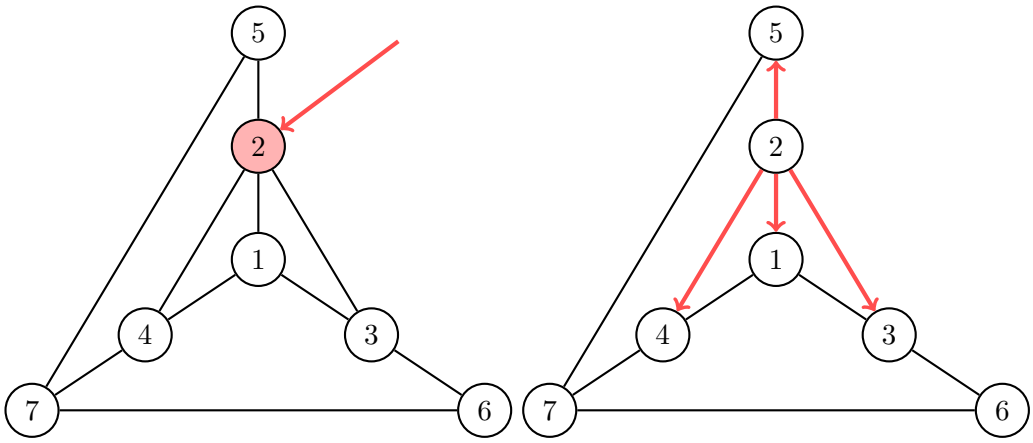


Figure 1.4: From [1]: Attacks on a control system: A1 and A3 indicate integrity attacks, A2 and A4 indicate DoS attacks, and A5 indicate direct physical attacks to the process.



(a) Fault in a node

(b) Attack on the out-going communications of a node

Figure 1.5: Some fault and security issues in networked systems

In particular, we will first focus on the consensus problem in two different scenarios, a physical fault in one node - as represented in Figure 1.5(a) - and a healthy node suffering an attack on its out-going communications - shown in Figure 1.5(b). In order to address these events, we propose some Fault Detection and Isolation (FDI) methods based on the Unknown Input Observer (UIO) theory to detect and isolate the misbehaving node in a distributed fashion, where each node monitors all its neighbors. Some sufficient conditions for the existence of such FDI scheme will also be given, based on the set of available measurements and the topological properties of the network. Furthermore, it will be shown that it is impossible for the "healthy" part of the network to distinguish between a fault or an attack, but this is not the case for the node itself, given that it has a reliable local measurement of its own state. The reduction of the number of observer nodes will also be discussed and two different solutions will be given.

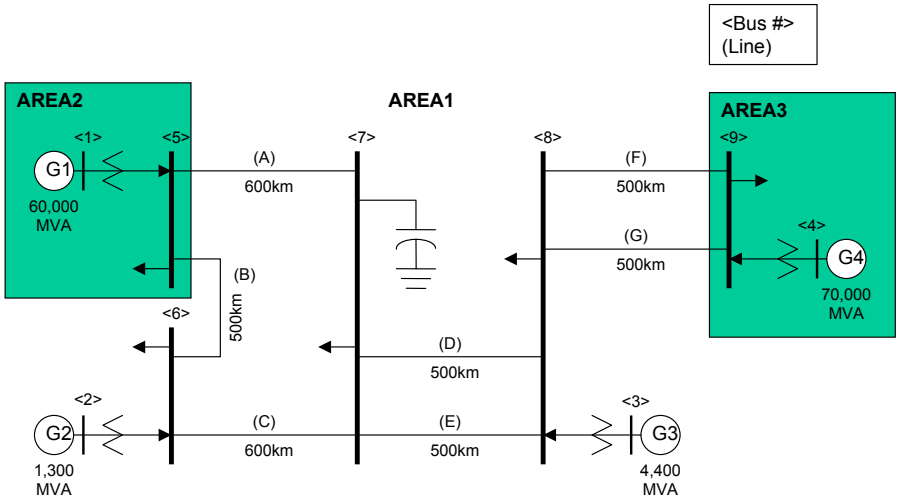


Figure 1.6: Example of a power system with three inter-connected areas

Then we present an example of a decentralized state estimation on a power system such as the one in Figure 1.6, followed by a decentralized FDI when such system is subject to faults - which are represented by A5 in Figure 1.4. Both schemes are based on the UIO theory and we will provide some constraints on the set of available measurements so that one of the two conditions required for the existence of such schemes is met. As for the other condition, although we do not have any consistent theoretical results, the simulation results give a good insight on what additional measurements are required for such FDI scheme to exist.

1.2 Outline

Due to the broad scope of this problem, we reserved Chapter 2 to introduce some background concepts which will constitute our framework for the following chapters. In that chapter, some concepts of Graph Theory, Unknown Input Observers and Fault Detection and Isolation will be described.

In Chapter 3 we will study the consensus problem , beginning by formulating the standard problem and then introducing faults on the nodes and attacks on the communications. Some methods in order to reduce the complexity of the proposed scheme will be discussed and some other further aspects will also be mentioned.

We will analyze the state estimation problem on a power system in Chapter 4, showing how the power system can be seen as a MAS, then using the UIO theory to design a decentralized state estimator and finally we will use this decentralized estimator to detect faults within each area.

The work done in this thesis will then summarized in Chapter 5, where we will also mention some future aspects to consider in this field and some possible improvements to the proposed methods.

Chapter 2

Background Overview

As it can be seen in the previous chapter, this thesis focuses on security issues in NCS and so several distinct concepts must be understood in order to have a suitable framework to tackle the problem. During this chapter we will present some background concepts that were used, which will be mentioned throughout this report.

2.1 Graph Theory

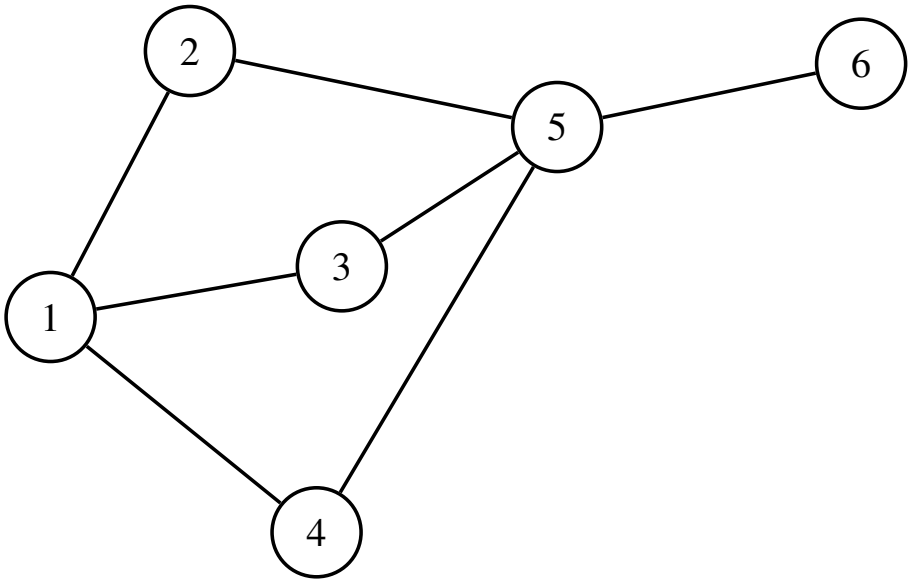


Figure 2.1: Example of a graph of a network

Since the interactions within a NCS or a MAS can be represented by means of a graph with the agents being the nodes and the interactions between agents being the edges, as seen in Figure 2.1,

one useful tool to analyze these systems is the well-know Graph theory [17] and we will now present some important concepts in this framework, focusing on undirected graphs (*i.e.* graphs which do not have uni-directional links).

The graph of a network with N nodes can be represented as $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V} = \{1, 2, \ldots, N\}$ is the set of vertices or nodes and $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ is the set of edges. Since the network is undirected, we say that the vertices i and j are *adjacent* or neighbors if $(i, j) \in \mathscr{E}$, denoting the neighbor set of i by $N_i = \{j \in \mathscr{V} : (i, j) \in \mathscr{E}\}$. The *out-degree* of node i is given by $d_i = |N_i|$, which corresponds to the number of neighbors of node i.

A *path* between vertices i and j is a sequence of distinct vertices starting in i and ending in j, such that each consecutive vertices in the sequence are adjacent, were the length of the path is the number of edges it contains. We say the graph $\mathscr{G}$ is connected if there exists a path between any two distinct vertices.

There are some matrices which characterize a few properties of a graph, such as the *adjacency*, the *incidence*, the *degree* and the *Laplacian* matrices, which will now be discussed.

The adjacency matrix of a weighted graph $\mathscr{G}$ is denoted by $\mathscr{A}(\mathscr{G}) \in \mathbb{R}^{N \times N}$ and its elements are given by:

$$[\mathscr{A}(\mathscr{G})]_{ij} = \begin{cases} w_{ij} & \text{, if } (i, j) \in \mathscr{E}(\mathscr{G}) \\ 0 & \text{, otherwise} \end{cases}, \tag{2.1}$$

where $w_{ij} > 0$ is the weight of the edge between node i and j. The degree matrix of such graph is given by $\Delta(\mathscr{G}) \in \mathbb{R}^{N \times N}$, with

$$[\Delta(\mathscr{G})]_{ij} = \begin{cases} \sum_{k \in N_i} w_{ik} & \text{, if } i = j \\ 0 & \text{, otherwise} \end{cases}. \tag{2.2}$$

The incidence matrix $\mathscr{B} \in \mathbb{R}^{N \times M}$, when derived for an undirected graph $\mathscr{G}$ with M edges, has its elements in the set $\{0, 1\}$ and is given by:

$$[\mathscr{B}(\mathscr{G})]_{kl} = \begin{cases} 1 & \text{, if node } k \text{ is in the edge } l \\ 0 & \text{, otherwise} \end{cases}. \tag{2.3}$$

However, due to a particular property of this matrix, which is evident in (2.5), it is convenient to derive the incidence matrix of a graph $\mathscr{G}$ by considering an oriented version of it, $\mathscr{G}^{\sigma}$, with an arbitrary orientation σ. An orientation of a graph $\mathscr{G}$ consists of the assignment of a direction to each of its edges, defining which node is the head, which is the tail and considering the edge to be oriented from its tail to its head. An example of an orientation of the graph presented in Figure 2.1 can be seen in Figure 2.2.

In this case, the incidence matrix of the oriented graph $\mathscr{G}^{\sigma}$ has its elements belonging to the

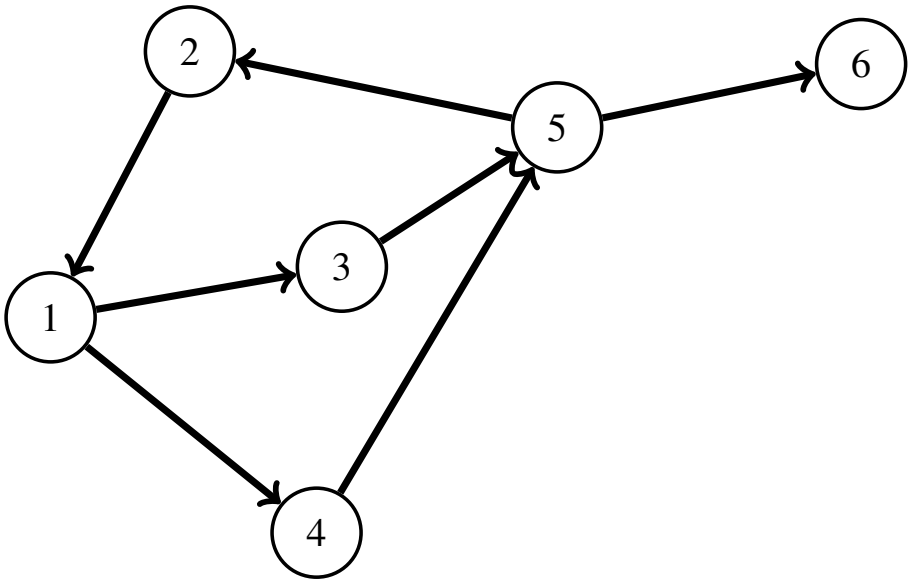


Figure 2.2: Example of an oriented graph

set $\{-1,0,1\}$, which are given by:

$$[\mathscr{B}(\mathscr{G}^{\sigma})]_{kl} = \begin{cases} 1 & \text{, if node } k \text{ is the head of edge } l \\ -1 & \text{, if node } k \text{ is the tail of edge } l \\ 0 & \text{, otherwise} \end{cases} . \tag{2.4}$$

The Laplacian matrix of a weighted graph $\mathscr{G}$ can be defined from the previous matrices using the following equations:

$$\mathscr{L}(\mathscr{G}) = \Delta(\mathscr{G}) - \mathscr{A}(\mathscr{G}) = \mathscr{B}(\mathscr{G}^{\sigma}) W \mathscr{B}(\mathscr{G}^{\sigma})^{T}, \tag{2.5}$$

with $W \in \mathbb{R}^{M \times M}$ being a diagonal matrix containing the weights of the edges.

The adjacency and Laplacian matrices contain some interesting properties of the graph they refer to and an interested reader is advised to see these in detail by consulting [17].

Throughout the rest of this thesis, we will omit the argument specifying the graph for the previous matrices, when such graph can easily be identified from the context.

We will now compute the previous matrices for the graph in Figure 2.1 and its oriented version in Figure 2.2 assuming constant unitary weights for the edges and, since the graph these matrices refer to is well-defined in this context, we will omit it from the notation.

The adjacency, degree and incidence matrices are then given by:

$$\mathscr{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \Delta = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathscr{B} = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The Laplacian matrix can then be computed using these matrices:

$$\mathscr{L} = \Delta - \mathscr{A} = \mathscr{B}\mathscr{B}^T = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 \\ -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & -1 & -1 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

.

2.2 Unknown Input Observer

In this section, we will present some of the techniques mentioned in [18] to design an observer for a linear time-invariant system affected by an unknown disturbance, described by the following state space equations:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ed(t) \\ y(t) = Cx(t) \end{cases}, \tag{2.6}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^r$ is the known input vector, $d(t) \in \mathbb{R}^q$ is the unknown input (disturbance) vector, $y(t) \in \mathbb{R}^p$ is the output vector and A, B, E, C have appropriate dimensions, with the assumption that the matrix E has full column rank.

Definition 2.2.1 ([18]) *A state observer is an unknown input observer (UIO) if the state estimation error $e(t)$ approaches zero asymptotically, regardless of the presence of an unknown input $d(t)$.*

A full-order observer for the dynamical system in (2.6) is described by:

$$\begin{cases} \dot{z}(t) = Fz(t) + TBu(t) + Ky(t) \\ \hat{x}(t) = z(t) + Hy(t) \end{cases}, \tag{2.7}$$

where $\hat{x}(t) \in \mathbb{R}^n$ is the estimated state and $z(t) \in \mathbb{R}^n$ is the observer's state. Note that if we choose $F = A$, $T = I$ and $H = 0$ we have a full-order Luenberger observer.

The matrices in the observer's equations must be designed in order to achieve the decoupling from the unknown input and meet some other requirements, such as the stability and convergence rate of the observer.

Using equations (2.6) and (2.7), we can derive the dynamics of the estimation error $e(t) = x(t) - \hat{x}(t)$:

$$\begin{aligned} \dot{e}(t) &= (A - HCA - K_1C)\,e(t) + [F - (A - HCA - K_1C)]\,z(t) \\ &+ [K_2 - (A - HCA - K_1C)\,H]\,y(t) + [T - (I - HC)]\,Bu(t) \\ &+ (HC - I)\,Ed(t) \end{aligned} \quad , \tag{2.8}$$

with $K_1 + K_2 = K$.

Choosing the matrices F, T, K, H to satisfy the following conditions:

$$\begin{aligned} F &= (A - HCA - K_1C) \\ T &= (I - HC) \\ (HC - I)\,E &= 0 \\ K_2 &= FH \end{aligned} \quad , \tag{2.9}$$

we have the estimation error's dynamics as:

$$\dot{e}(t) = Fe(t)\,. \tag{2.10}$$

We conclude that if the conditions in (2.9) are fulfilled and F is stable, then the observer in (2.7) is an UIO since $\lim_{t \to +\infty} e(t) = 0$, regardless the value of the unknown signal $d(t)$.

Theorem 2.2.1 ([18]) *The necessary and sufficient conditions for the observer described by (2.7) to be an UIO for the system in (2.6) are:*

i) $\operatorname{rank}(CE) = \operatorname{rank}(E)$

ii) (C, A_1) *is a detectable pair, where*

$$A_1 = A - HCA.$$

Intuitively, the first condition means that for an UIO given by (2.7) to exist, the system should have at least as many independent measurements as disturbances to be decoupled. Moreover, the disturbances should directly affect the measured states, so that $CE_i \neq 0$, where E_i is any column of E. Consider, as an example, a system with $C = [1\ 0]$ and $E = [0\ 1]^T$; even if we have the same number of measurements and disturbances, the first condition in Theorem 2.2.1 is not fulfilled. This condition is required for the existence of matrix H, but this matrix is usually not unique, having as a possible solution the product of E times the pseudoinverse of CE:

$$H = E\left((CE)^T CE\right)^{-1} (CE)^T. \tag{2.11}$$

The second condition ensures the stability and convergence of the observer by requiring F to be stabilizable. Note that $F = A_1 - K_1 C$ and so if the pair (C, A_1) is detectable, then due to the duality between detectability and stabilizability, the pair $\left(A_1, C^T\right)$ is stabilizable and K_1 can be computed so that F is stable.

2.3 Process Fault Detection and Isolation using Unknown Input Observers

A linear system in 2.6 subject to process faults $f_a(t) \in \mathbb{R}^m$ and sensor faults $f_s(t) \in \mathbb{R}^p$ can be described as:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ed(t) + B_f f_a(t) \\ y(t) = Cx(t) + f_s(t) \end{cases}, \tag{2.12}$$

where $B_f \in \mathbb{R}^{n \times m}$ is assumed to have full-column rank, which is not restrictive assumption since any singular matrix $D \in \mathbb{R}^{n \times l}$ can be decomposed in $D = D_1 D_2$, with D_1 being full column rank. A suitable method would be to use the Singular-Value Decomposition (SVD) $D = U \Sigma V^T$ and choose $B_f = D_1 = [U_1 \cdots U_r]$, where $U_i \in \mathbb{R}^n$ is the i^{th} column of U and $r = \text{rank}(D)$.

The matrix B_f is often called a fault distribution matrix, since it contains informations on how a vector of process fault signals affect the states of the dynamical system. The reason to consider this matrix to have a full column rank should be clearer after the following example:

Let two different faults $f_1(t), f_2(t) \in \mathbb{R}$ have the same distribution vector $B_{f_1} = B_{f_2}$, $B_{f_1}, B_{f_2} \in \mathbb{R}^n$ such that $B_f = [B_{f_1}\ B_{f_2}]$, $f_a = [f_1\ f_2]^T$. Since we have $B_{f_1} = B_{f_2}$, we can rewrite $B_f f_a = B_{f_1} f_1 + B_{f_2} f_2$ as $B_f f_a = B_{f_1}(f_1 + f_2)$. A similar procedure can be done for a set of linear dependent distribution vectors $\{B_{f_i}\}$, obtaining a set of linear independent distribution vectors which can be concatenated into a full column ranked matrix, both providing the same information on how the process faults are distributed in the dynamical system.

Consider now the system in (2.12) with no unknown inputs and no sensor faults, *i.e.* $E = 0$ and $f_s = 0$. As suggested in [18], a possible method of detecting and isolating the faults present in the process is to use the so-called Generalized Observer Scheme (GOS), where using the concept of UIO we construct a bank of observers generating a structured set of residuals such that each residual is decoupled from one and only one fault, being sensitive to all others. By "*isolate*" we mean to *locate* the fault within the dynamical system, to know which fault signal is active. This is a standard term in the FDI literature and will also be used in this thesis.

Definition 2.3.1 *A residual $r(t)$ is a fault indicator function, based on deviations between measurements and model-equation-based computations, which should satisfy the following condition:*

$$r(t) = 0 \iff f_a(t) = 0.$$

With this set of residuals, the detection and isolation of a fault in the i^{th} component uses the following logic:

$$\begin{cases} \|r_i(t)\| < T_{f_i} \\ \|r_k(t)\| \geq T_{f_k} \ , \forall k \neq i \end{cases} , \tag{2.13}$$

where $r_j(t)$ is the residual insensitive to a fault in the j^{th} component and T_{f_j} is the respective isolation threshold, which can be static or dynamic (*i.e.* an upper bound on the convergence of the estimation error).

Note, however, that this method is feasible only if a single process fault is present at a time. If more faults are present, they can be detected using this method, but not isolated. In order to isolate simultaneous faults, a different observer scheme should be used.

Under the assumption that there is a single process fault, let $f_{a_i} \neq 0$ be the active fault. In order to render an observer insensitive to f_{a_i}, this fault should be regarded as an unknown input and the observer could then be computed using the UIO theory. The system in (2.12) (with $E = 0$ and $f_s = 0$) can be rewritten as:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + B_f^i f_a^i(t) + b_{f_i} f_{a_i}(t) \\ y(t) = Cx(t) \end{cases} , \tag{2.14}$$

where b_{f_i} is the i^{th} column of B_f, f_{a_i} is the i^{th} component of f_a, B_f^i is B_f with the i^{th} column deleted and f_a^i the fault vector f_a with its i^{th} component removed. The UIO decoupled from b_{f_i} has the same structure as in (2.7) and is described by:

$$\begin{cases} \dot{z}_i(t) = F_i z_i(t) + T_i B u(t) + K_i y(t) \\ \hat{x}_i(t) = z_i(t) + H_i y(t) \end{cases} . \tag{2.15}$$

The estimation error dynamics using the UIO in (2.15) are given by:

$$\begin{aligned} \dot{e}_i(t) &= (A - H_i CA - K_{1_i} C) e_i(t) + [F_i - (A - H_i CA - K_{1_i} C)] z_i(t) \\ &+ [K_{2_i} - (A - H_i CA - K_{1_i} C) H_i] y(t) + [T_i - (I - H_i C)] B u(t) \\ &+ (H_i C - I) E_i d_i(t) - (I - H_i C) B_f^i f_a^i \end{aligned} , \tag{2.16}$$

with $E_i = b_{f_i}$. By choosing the UIO matrices similarly as in (2.9) and by using as a residual the signal $r_i(t) = y(t) - C\hat{x}_i$, we arrive to the following error and residual dynamics:

$$\begin{cases} \dot{e}_i(t) = F_i e_i(t) - T_i B_f^i f_a^i \\ r_i(t) = C e_i(t) \end{cases} . \tag{2.17}$$

Note that the residual dynamics are driven by the k^{th} fault if and only if $T_i b_{f_k} \neq 0, \forall k \neq i$. If we ensure this happens for all $k \neq i$, we can compute similar observers for all the other faults and then use the threshold logic in (2.13) to isolate the active plant fault.

In the case where the system is also under the influence of unknown inputs ($E \neq 0$), the previous observer can be made insensitive to the unknown inputs by augmenting the matrices $\{E_i\}$

to include the disturbance distribution matrix E. Note, however, that such decoupling is possible only if all the faults and disturbances have different distribution vectors, *i.e.* B_f and E are linearly independent.

The FDI method described in this section is an observer-based method, but there are several other techniques to perform FDI in dynamical systems, such as parity equations, parameter estimation and signal models. An overview of such methods is given in [19].

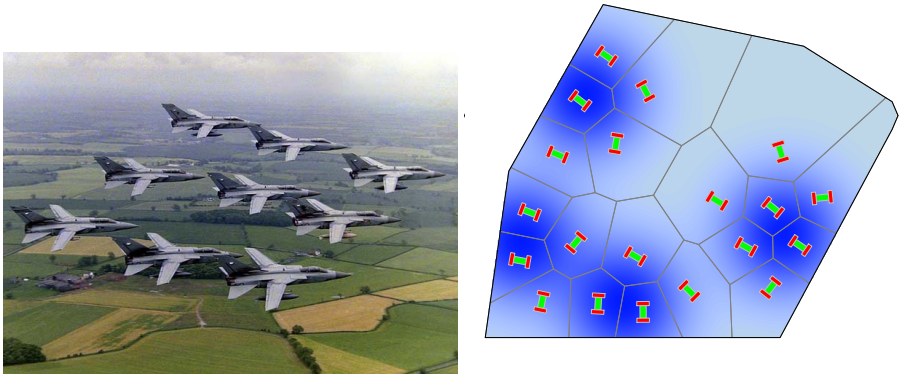
Concerning observer-based methods, there are more efficient methods to detect and isolate faults, such as the Beard-Jones Detection Filter (BJDF), which can detect and isolate several different faults with a single observer - although not simultaneously. Some other linear observers have integrator states as well, in order to estimate the fault signal [20]. Other techniques include also nonlinear methods like the sliding-mode observer [21].

Chapter 3

Security of Consensus in Networked Multi-Agent Systems

Using the framework introduced in Chapter 2, we will analyze a very common example of a NMAS present in the literature [7, 5, 14, 22], the consensus problem. We begin by describing this problem in a general way and then focus on the security issues that may arise due to the networked nature of the system. The effect of both physical faults and malicious attacks will be formulated, analyzed and compared and a possible method to distinguish both cases and isolated the misbehaving agent is proposed.

3.1 The Consensus Problem



(a) Formation of aircrafts (b) Robot deployment [23]

Figure 3.1: Example of applications

The consensus problem is a typical example of a cooperative distributed algorithm, having been studied extensively by computer scientists in the field of distributed computing [14, 24, 25] and consists on reaching an agreement on a certain quantity among all the nodes in a network, which usually is a function of the initial state of each node.

More recently, this problem has received an increasing amount of attention from the control community due to its application in the distributed control of networked dynamic systems, being related to problems regarding sensor fusion [3], decentralized estimation [4] and control of mobile agents such as flocking, rendezvous, deployment, containment and formation [23]. From this point of view, several studies have been done in order to understand how the topological properties of the network affect the performance of a group of dynamic agents using the consensus algorithm as a distributed control law, analyzing convergence, stability [7, 26, 27], controllability [28, 6] and observability [29, 30, 31] properties, among others.

We will now consider the consensus protocol running in a group of N dynamic agents with single integrator dynamics:

$$\begin{cases} \dot{x}_i &= u_i \quad , x_i(0) = x_{i_0} \in \mathbb{R} \\ y_i &= x_i \end{cases} . \tag{3.1}$$

The several agents are connected through a communication network, which is represented by a graph $\mathscr{G}$ such as the one in Figure 3.2, and the agents are using the following control law:

$$u_i = -\sum_{j \in N_i} (y_i - y_j), \tag{3.2}$$

where the N_i is the set of neighbors of the i^{th} agent, as described in Section 2.1.

Under this setting, the dynamics of the entire network can be written by

$$\dot{\mathbf{x}} = -\mathscr{L}\mathbf{x}, \tag{3.3}$$

with $\mathbf{x} = \left[x_1^T \cdots x_N^T\right]^T$ being the collection of all the states and $\mathscr{L}$ the Laplacian of the graph $\mathscr{G}$.

It has been proved [7] that, for connected graphs, the system in (3.3) has an unique stable equilibrium point $\mathbf{x}^* = [\alpha \cdots \alpha]^T \in \mathbb{R}^N$, with α being a function of the initial states, and thus a consensus state is achieved. Furthermore, for connected graphs the convergence rate of the system in (3.3) has a lower bound equal to λ_2, the second smallest eigenvalue of $\mathscr{L}$.

The scope of this thesis does not focus on studying these properties in detail and so an interested reader is advised to see the references mentioned in this section for further details. During the following sections in this chapter we will focus on the consensus problem here described subject to faults in the nodes and also to deception attacks from outside agents.

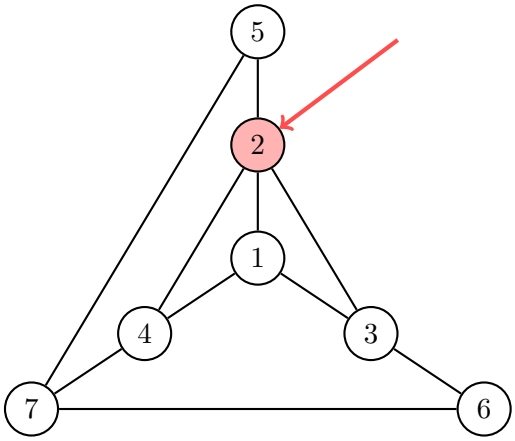


Figure 3.2: Network with node 2 having a fault

3.2 Consensus in Networked Multi-Agent Systems subject to Faults

Given the general setting of the consensus problem in the previous section, we will now study the case where the agents may be affected by unknown disturbances, which will be treated as faults in the respective nodes.

A very similar approach is presented in [9], in which the consensus problem is analyzed in its discrete form. The proposed solution is to design a particular UIO in each node decoupled from fault signals in all its neighbors and then to use the iteration error of the observer as a residual.

A different approach can be found in [10, 11], where the authors analyze the general problem of distributed function calculation via linear iterations with misbehaving nodes, which includes the consensus problem in its discrete form. The proposed method uses observability techniques in order to isolate the misbehaving nodes and recover the initial conditions of all the nodes in the network, which are then used by each node to compute the desired function. The great advantage in this work is that it can overcome the effect of multiple faults occurring at the same time.

Our approach differs from the previous in the sense that we will consider the continuous time problem, the FDI scheme has a different structure, although based on the UIO theory, and we do not require a particular observer. Furthermore, we also propose a method to reduce the number of observer nodes based on the graph of the network.

In Section 3.3 we show how an attack on the out-going communications of a node can be detected and distinguished from a fault scenario using the proposed FDI scheme, which was not part of the scope of the previous solutions.

The analysis of the consensus problem subject to faults is done under the following assumption:

Assumption 3.2.1 *At any time instant, there is only one node in the network having a fault.*

This assumption is reasonable in the sense that the probability of having two different faults at the same time is small and it simplifies the task of detecting and isolating the fault, since we do not need to deal with simultaneous faults.

The behavior of the node k running the consensus algorithm while affected by an unknown fault signal f_k, as shown in Figure 3.2 can be described by

$$\begin{cases} \dot{x}_k &= -\sum_{j \in N_k} (y_k - y_j) + f_k \\ y_k &= x_k \end{cases}, \tag{3.4}$$

and so the global dynamics of the network can be written as

$$\dot{\mathbf{x}} = -\mathscr{L}\mathbf{x} + b_f^k f_k, \tag{3.5}$$

with $b_f^k \in \mathbb{R}^N$ being a vector with the k^{th} component set to 1 and all the others to 0.

3.2.1 Detection and Identification of the Faulty Node

In order to detect and isolate the misbehaving node, we propose a similar approach to that in [9], a distributed model-based FDI scheme using the Unknown Input Observers described in Section 2.2. As it is a model-based approach, we assume the global model of the system is known by all the nodes in the network and so the following assumption is made:

Assumption 3.2.2 *The graph of the network is know by all nodes and it remains constant.*

Comparing the equations (3.5) and (2.15) we realize that a faulty node can be seen as a process fault, regarding the global dynamics of the network, thus the FDI scheme described in Section 2.3 can be used to detect and isolate the faulty node.

As seen in Section 2.3 such FDI scheme needs the model of the process to be monitored. In this case, the dynamics of the process with a fault present are described by (3.5), where b_k is the fault distribution vector, but we still lack a set of measurements.

Since we have a networked system, the FDI system should be implemented in a distributed or at least decentralized way. In that sense, we will design the FDI system so that each node is observing its neighbors in order to discover if they are misbehaving or not. This way, the set of information available to an observing node i would be:

$$\mathbf{y}_i = \begin{bmatrix} y_i \\ y_{i_1} \\ \vdots \\ y_{i_{|N_i|}} \end{bmatrix} = \begin{bmatrix} x_i \\ x_{i_1} \\ \vdots \\ x_{i_{|N_i|}} \end{bmatrix} = C_i \mathbf{x}, \tag{3.6}$$

where $\{i_1 \cdots i_{|N_i|}\} \in N_i$ are the indexes corresponding the i^{th} node's neighbors, thus we conclude that each observer uses the locally available information and knowledge about the network's dynamics. Using this formulation we are able to apply the previously described UIO theory and implement a FDI system in each node able to detect faults in its neighborhood.

Consider the following system to be monitored, which includes all the possible faults in the network:

$$\begin{cases} \dot{\mathbf{x}} &= -\mathscr{L}\mathbf{x} + B_f f \\ \mathbf{y}_i &= C_i \mathbf{x} \end{cases}, \tag{3.7}$$

with $B_f \in \mathbb{R}^{N\times N} = \left[b_f^1 \cdots b_f^N\right]$, where b_f^k is the distribution vector of a fault in the k^{th} node.

Using the GOS scheme described in Section 2.3, we will build a FDI system in node i using a bank of UIOs so that each UIO is sensitive to all the faults in the network except one, corresponding to a single neighbor of the i^{th} node. It is clear that each node will then have as many observers as the size of its neighborhood.

Assuming that node k is a neighbor of node i, (3.7) can be rewritten so that the effect of a fault in the k^{th} node is evident:

$$\begin{cases} \dot{\mathbf{x}} &= -\mathscr{L}\mathbf{x} + b_f^k f_k + B_f^{\bar{k}} f_{\bar{k}} \\ \mathbf{y}_i &= C_i \mathbf{x} \end{cases}. \tag{3.8}$$

The similarities between (3.7) and (2.14) are clear, thus we now proceed by computing the UIO described by (2.15), which we rewrite according to the notation followed in this section:

$$\begin{cases} \dot{z}_i^k &= F_i^k z_i^k + T_i^k B\mathbf{u} + K_i^k \mathbf{y}_i \\ \hat{\mathbf{x}}_i^k &= z_i^k + H_i^k \mathbf{y}_i \end{cases}, \tag{3.9}$$

with $\hat{\mathbf{x}}_i^k \in \mathbb{R}^N$ being the estimate of the network's state insensitive to a fault in node k, which is computed by node i. Note that since the network is only running the consensus algorithm, we assume that there is no known input $\mathbf{u}$ and we consider $B = 0$.

From Theorem 2.2.1, it is known that for such observer to exist two conditions must be satisfied:

i) $\text{rank}\left(C_i b_f^k\right) = \text{rank}\left(b_f^k\right) = 1$

ii) $\left(C_i, -\mathscr{L} + H_i^k C_i \mathscr{L}\right)$ is detectable

where H_i^k satisfies the equation $I - H_i^k C_i = b_f^k$, having as a possible solution [18]:

$$H_i^k = b_f^k \left(\left(C_i b_f^k\right)^T C_i b_f^k\right)^{-1} (C_i b_f^k)^T. \tag{3.10}$$

Concerning the proposed FDI scheme, it is possible to verify that both conditions are always met:

Theorem 3.2.3 *There exists a UIO for the system* $\left(-\mathscr{L}(\mathscr{G}), b_f^k, C_i, 0\right)$ *if the graph* $\mathscr{G}$ *is connected and* $k \in N_i$.

Before proving the previous theorem, we will introduce the following lemma, which is proved in Appendix A:

Lemma 3.2.4 *If an undirected graph $\mathcal{G}$ is connected, then any partition of its Laplacian matrix $\mathscr{L}$, induced by a strict subset of nodes $\bar{F} \subset \mathcal{V}$, is invertible.*

Proof of Theorem 3.2.3. It can be easily seen that if the faulty node k is a neighbor of the observer node i, then the first condition of Theorem 2.2.1 is satisfied: since C_i has full row rank and the row corresponding to node k is $C_i^k = b_f^{kT}$, then it follows that $\text{rank}\left(C_i b_f^k\right) = \text{rank}\left(C_i^k b_f^k\right) = \text{rank}\left(b_f^{kT} b_f^k\right) = \text{rank}\left(b_f^k\right) = 1$.

As for the second condition in Theorem 2.2.1, this condition is equivalent to say that the transmission zeros of the system $\left(-\mathscr{L}, b_f^k, C_i, 0\right)$ must be stable [18], *i.e.*

$$\begin{bmatrix} sI_N + \mathscr{L} & b_f^k \\ C_i & 0 \end{bmatrix},$$

is of full column rank for all s such that $\Re(s) \geq 0$.

Suppose now that we apply a transformation P to the system $\left(-\mathscr{L}, b_f^k, C_i, 0\right)$ so that $\bar{\mathbf{x}} = P\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\tilde{N}_i}^T & \mathbf{x}_{\bar{N}_i}^T \end{bmatrix}^T$ and $\bar{C}_i = \begin{bmatrix} I_{|\tilde{N}_i|} & 0_{|\tilde{N}_i| \times |\bar{N}_i|} \end{bmatrix}$, where $\tilde{N}_i = N_i \cup i$ and $\bar{N}_i = \mathcal{V} \setminus \{\tilde{N}_i\}$, which consists on a simple permutation operation. After this operation we can write the Laplacian as

$$\bar{\mathscr{L}} = P^{-1}\mathscr{L}P = \begin{bmatrix} \mathscr{L}_{\tilde{N}_i} & l_{\tilde{N}_i\bar{N}_i} \\ l_{\bar{N}_i\tilde{N}_i} & \mathscr{L}_{\bar{N}_i} \end{bmatrix},$$

and hence we have:

$$\begin{bmatrix} sI_N + \mathscr{L} & b_f^k \\ C_i & 0 \end{bmatrix} = \begin{bmatrix} sI_{|\tilde{N}_i|} + \mathscr{L}_{\tilde{N}_i} & l_{\tilde{N}_i\bar{N}_i} & \bar{b}_f^k \\ l_{\bar{N}_i\tilde{N}_i} & sI_{|\bar{N}_i|} + \mathscr{L}_{\bar{N}_i} & 0_{\bar{N}_i \times 1} \\ I_{|\tilde{N}_i|} & 0_{|\tilde{N}_i| \times |\bar{N}_i|} & 0_{\tilde{N}_i \times 1} \end{bmatrix},$$

with $\bar{b}_f^k = P^{-1} b_f^k$.

Note that due to the last row of the previous matrix, the first column is independent of the others and furthermore it is of full column rank, thus:

$$\text{rank}\begin{bmatrix} sI_N + \mathscr{L} & b_f^k \\ C_i & 0 \end{bmatrix} = |\tilde{N}_i| + \text{rank}\begin{bmatrix} l_{\tilde{N}_i\bar{N}_i} & \bar{b}_f^k \\ sI_{|\bar{N}_i|} + \mathscr{L}_{\bar{N}_i} & 0_{|\bar{N}_i| \times 1} \end{bmatrix}.$$

From Lemma 3.2.4 we know that any square partition of the Laplacian is invertible if the respective graph is connected, thus $\mathscr{L}_{\bar{N}_i} \succ 0$ and since $\Re(s) \geq 0$, $sI_{|\bar{N}_i|} + \mathscr{L}_{\bar{N}_i}$ is invertible as well and has full rank, following that:

$$\text{rank}\begin{bmatrix} sI_N + \mathscr{L} & b_f^k \\ C_i & 0 \end{bmatrix} = |\tilde{N}_i| + |\bar{N}_i| + 1 = N + 1,$$

which proves that the transmission zeros are all stable and that a UIO exists. ■

Once verified that the UIO is insensitive to the fault in the k^{th} node exists, the observer matrices can be computed by:

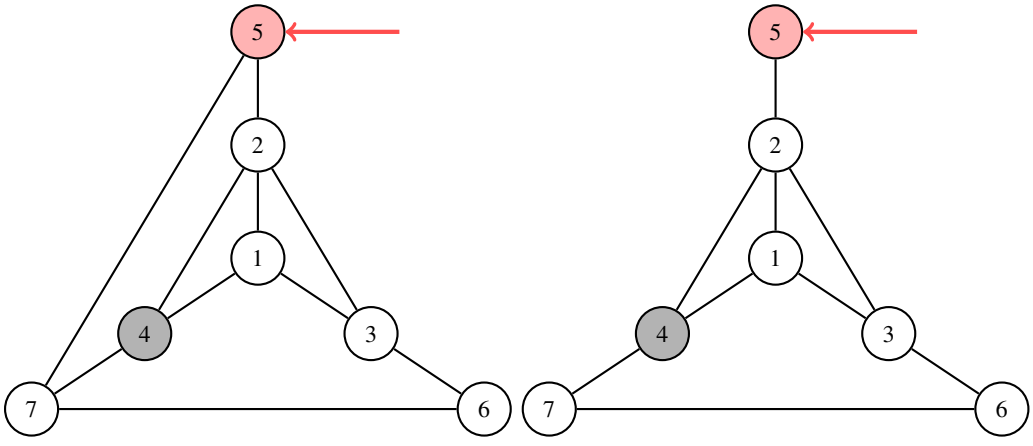
$$\begin{array}{rcl} F_i^k & = & \left(-\mathscr{L} + H_i^k C_i \mathscr{L} - K_{i_1}^k C_i\right) \\ T_i^k & = & \left(I - H_i^k C_i\right) \\ K_{i_2}^k & = & F_i^k H_i^k \\ K_i^k & = & K_{i_1}^k + K_{i_2}^k \end{array} , \tag{3.11}$$

using H_i^k as suggested previously in (3.10). Note that the only requirement for $K_{i_1}^k$ is that it should stabilize the observer, thus the designer has some degrees of freedom available which can be used to meet some additional requirements, such as the convergence rate of the filter, directional properties of each residual or, in case of noisy measurements, a minimum estimation error variance.

From the system and observer dynamics in (3.8) and (3.9) respectively, we can derive the error dynamics and the residual, similarly to (2.17):

$$\left\{ \begin{array}{rcl} \dot{e}_i^k & = & F_i^k e_i^k - T_i^k B_f^{\tilde{k}} f_{\tilde{k}} \\ r_i^k & = & C_i e_i^k \end{array} \right. , \tag{3.12}$$

where $f_{\tilde{k}}$ is obtained by removing the fault elements of $f_{\tilde{k}}$ which have the same distribution vector as f_k, since in this case the observer will also be insensitive to these faults. This situation may occur only for nodes outside the neighborhood of the observer node, since a fault in another neighbor necessarily has a distinct distribution vector. Furthermore, it happens only when all the possible paths connecting the faulty node $j \notin N_i$ to node i contain only node $k \in N_i$ - which is possible only in 1-connected graphs. To clarify this statement, consider the following situations presented in Figure 3.3(a) and Figure 3.3(b), where node 5 is the faulty one.



(a) Example of a 2-connected graph (b) Example of a 1-connected graph

Figure 3.3: Example of graphs with different connectivity properties

In Figure 3.3(a), the faulty node 5 affects two neighbors of node 4, hence the FDI system in the 4^{th} node will not be able to isolate the fault, as all the residuals will react to it and be non-zero.

Thus the FDI system behaves correctly in this scenario.

However, in the scenario presented in Figure 3.3(b), only the neighbor node 2 will be affected by the faulty node 5 and so the FDI system implemented in node 4 will detect the fault and isolate 2 as being the faulty node, thus producing a wrong isolation of the fault.

Note that in the first scenario there were at least two paths from node 5 to node 4 which contained different neighbors of node 4, nodes 2 and 7. On the other hand, in the second scenario the only path from node 5 to node 4 contains node 2 only, thus a fault in node 5 has the same distribution vector as a fault in node 2, seen from the observer node 4.

Due to the distributed nature of the proposed FDI scheme and to the fact that the effect of a fault is attenuated depending on its distance to the monitoring node, we will assume that the faulty node 5 will be detected by one of its own neighbors before triggering any alarm in node 4, thus we will consider from now on that $f_{\tilde{k}} = f_{\bar{k}}$ in order not to increase the complexity of the notation.

Table 3.1: Sensitivity of the residual bank to each fault

	f_{i_1}	$\cdots$	f_{i_k}	$\cdots$	$f_{i_{\lvert N_i \rvert}}$	$f_{\bar{N}_i}$
r_i^1	0	1	1	1	1	1
$\vdots$	1	$\ddots$	1	1	1	1
r_i^k	1	1	0	1	1	1
$\vdots$	1	1	1	$\ddots$	1	1
$r_i^{\lvert N_i \rvert}$	1	1	1	1	0	1

The previous method to compute the UIO should then be applied to the rest of the i^{th} node's neighbors, after which node i is able to detect and isolate a fault in one of its neighbors using the residual in (3.12) and a threshold logic similar to the one described in (2.13). The Table 3.1 is a truth table summarizing the sensitivity of each residual to each fault, with $f_{\bar{N}_i}$ being all the faults in nodes outside the neighborhood of node i and so we have that three possible situations may occur:

- $\left\|r_i^k\right\| < T_{f_k}$, $\forall k \in N_i$
- $\begin{cases} \left\|r_i^j\right\| & < \quad T_{f_j} \\ \left\|r_i^k\right\| & \geq \quad T_{f_k} \end{cases}$, $\forall k \neq j \in N_i$
- $\left\|r_i^k\right\| \geq T_{f_k}$, $\forall k \in N_i$

The first case happens when there is no fault in the system, while the second occurs when node k in the neighborhood of node i is the faulty node. The last situation is verified when there is a faulty node in the network, but it is not one of node i's neighbor, thus the FDI system is able to detect but not isolate the misbehaving node.

3.2.2 Reduction of the number of observer nodes

The FDI scheme proposed in the previous section requires that each node in the network has a bank of observers to monitor each one of its neighbors, resulting in a distributed but computationally

heavy FDI scheme. Since each node monitors all its neighbors, it is clear that there exists a certain amount of redundancy as the same node is monitored by all its neighbors. Hence, one possible improvement to be made is to reduce the number of monitoring nodes to a minimum number, thus reducing the overall computational burden of the network. We will propose two different methods to reduce the number of observer nodes, one assuming that the node only monitors its neighbors, as what happens in the previous section, and another one where each node monitors both the neighborhood and itself. Although it may seem the last option is the better one, since it is intuitive that the nodes should monitor themselves as well, we will provide arguments that also make the first case reasonable.

Observers monitoring only the neighborhood

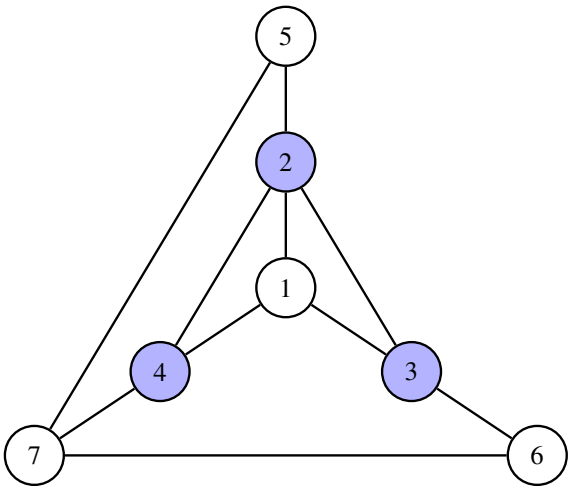


Figure 3.4: Set of nodes covered by an observer in 1 with N_1

Assuming that each node monitors only its neighbors, we can say that a FDI system in node i covers the set of nodes $j \in N_i$. Therefore, the objective is to select a minimum number of observer nodes so that they cover all the nodes in the network, *i.e.*

$$\begin{aligned} \min_{S_o \subseteq \mathcal{V}} \quad & |S_o| \\ s.t. \quad & \bigcup_{i \in S_o} N_i = \mathcal{V} \end{aligned} ,$$

where S_o is the set of observer nodes.

As it can be seen, this is actually a set cover problem where we wish to determine the minimum dominating set - the minimum number of sets which cover the union of all the sets. This is a well studied problem, having been classified as a *NP*-hard problem and we can find two proposed algorithms in [32] that determine the minimum dominating set, a recursive one and another based on dynamic programming.

Both algorithms receive the collection of all the sets $\{N_i\}$ to be taken into account and have as the output the cardinality of the cover set, therefore some modifications are needed in order to also return the indices of the selected nodes, which is not a complex task.

Although the number of observers obtained by using N_i as the set of nodes covered by node i is not minimum, this method has one interesting property: all the observer nodes are monitored by at least one neighbor, which also is an observer. This means that even if an observer node has a fault, at least one other observer node in the network can detect and isolate it, which lowers the vulnerability of such scheme to faults in the monitoring nodes.

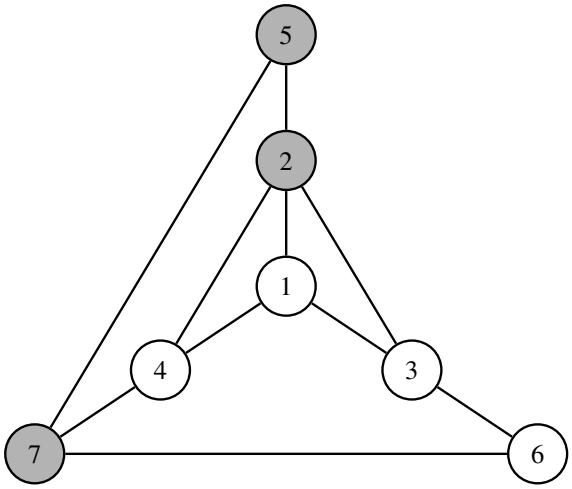


Figure 3.5: The chosen observer nodes considering the cover set N_i for each node i

Observers monitoring themselves and the neighborhood

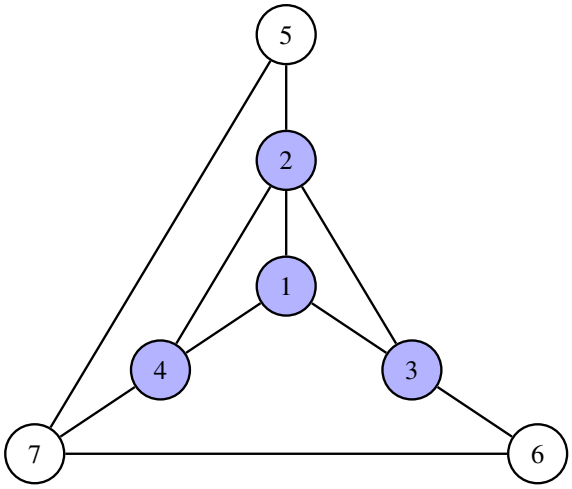


Figure 3.6: Set of nodes covered by an observer in 1 with $\tilde{N}_1$

We now assume that each observer node monitors itself and its neighborhood, thus an observer node i covers the set $\tilde{N}_i = N_i \cup i$. With this formulation, the minimum cover set will actually result in the minimum dominating set, which by definition is the set composed by the minimum number of nodes such that every other node is adjacent with at least one node from that set, thus an optimal number of observers is achieved.

Using the set $\tilde{N}_i$ as the set covered by node i, the algorithms in [32] can be used in order to derive the minimum dominating set. It is easy to verify that the number of nodes obtained this way is always less or equal to that obtained by the previous method, but we can no longer guarantee that the observer nodes are also being monitored by one of their neighbors, as it can be seen by comparing Figure 3.5 and Figure 3.7

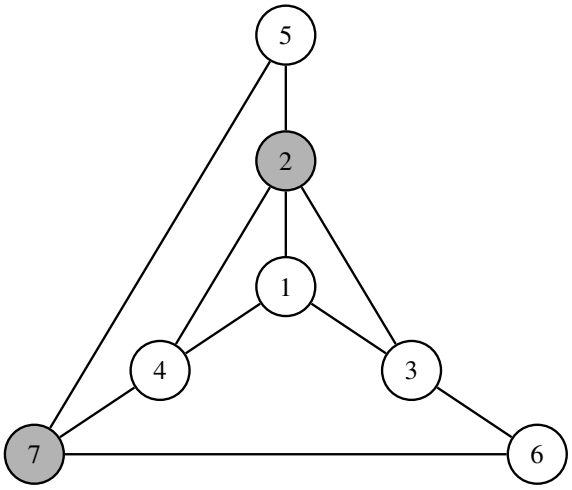


Figure 3.7: The chosen observer nodes considering the cover set $\tilde{N}_i$ for each node i

Two different methods of reducing the number of observer nodes were presented, each with different characteristics, but these are not the only options. For instance, these methods do not guarantee that the induced subgraph is connected, which could be useful in situations where the set of monitoring nodes are also required to decide which action to take in a distributed fashion, using the already existing communication graph.

Furthermore, there are also other alternatives to reduce the complexity of the FDI scheme, such as using another type of filter, computing the minimal realization of the models used to derive the UIOs, design of a reduced-order UIO instead of a full-order one, among others.

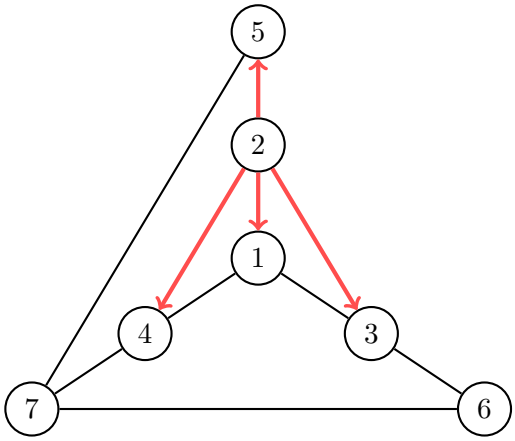


Figure 3.8: Network with node 2 being attacked

3.3 Consensus in Networked Multi-Agent Systems subject to Communication Attacks

In this section we will once again consider the consensus problem but now assuming that, while there is no process fault, the information exchanged between nodes may be modified by an external hostile agent - the attacker. As in Section 3.2, we assume there is only one compromised node and thus the following assumption is made:

Assumption 3.3.1 *Only one node may be compromised at a given time instant and only the outgoing data of such agent may be modified by the attacker.*

The dynamics of a normal agent, which is a single integrator, are described by (3.1) and the control law is similar to the one in (3.2):

$$u_i = -\sum_{j \in N_i} (w_i - y_j), \tag{3.13}$$

where $w_i = x_i$ is an internal measurement of node i and y_j is the measurement received from the neighbor node j. The notion of "*internal measurement*" is introduced in this scenario because, unlike in Section 3.2, where there is the possibility of having disturbed measurements being transmitted between neighbors due to the effect of an external hostile agent, but the internal measurement is assumed to be secure and reliable, since it is not transmitted. Note that the communicated measurement is assumed to be unknown to the node sending it, which means that the compromised node believes it is sending the correct information.

The global dynamics of the network with node k being compromised can be written as:

$$\begin{cases} \dot{\mathbf{x}} &= -\mathscr{L}\mathbf{x} + I_{\bar{k}} l^k f_{s_k} \\ \mathbf{y} &= \mathbf{x} + b_f^k f_{s_k} \\ \mathbf{w} &= \mathbf{x} \end{cases}, \tag{3.14}$$

with $\mathbf{w} \in \mathbb{R}^N$ as the vector of all the internal measurements, $\mathbf{y} \in \mathbb{R}^N$ the set of communicated measurements, $I_{\bar{k}} \in \mathbb{R}^{N\times N}$ is the identity matrix with the k^{th} diagonal entry set to zero, $l^k \in \mathbb{R}^N$ is the k^{th} column of the Laplacian matrix and $b_f^k \in \mathbb{R}^N$ is the k^{th} column of the identity matrix.

By comparing (3.5) and (3.14), we can see that these two scenarios are quite different at a structural level, since in the case of the attack we have an unknown signal affecting both the process and the communicated measurements. In this sense, it seems that the FDI scheme proposed in Section 2.3 cannot be used to successfully detect and isolate a compromised node, at least as it is.

3.3.1 Detection and Identification of the Compromised Node

Although it may seem that both scenarios considered in this chapter are very different, surprisingly enough, they are equivalent if seen through the healthy/uncompromised nodes' point of view. The argument at this stage is that the scenario where only node k is compromised and its out-going data is tampered can be seen by the rest of the network as a misbehave or fault on such node, which is evident in the following equation that describes the behavior of the healthy part of the network having the information from the compromised node as an input:

$$\begin{cases} \dot{\mathbf{x}}_{\bar{k}} &= -\mathscr{L}_{\bar{k}}x_{\bar{k}} - l_{\bar{k}k}y_k \\ \mathbf{y}_{\bar{k}} &= \mathbf{x}_{\bar{k}} \\ \dot{\mathbf{x}}_k &= -\mathscr{L}_k x_k - l_{k\bar{k}}y_{\bar{k}} \\ \mathbf{y}_k &= \mathbf{x}_k + f_{s_k} \end{cases}, \tag{3.15}$$

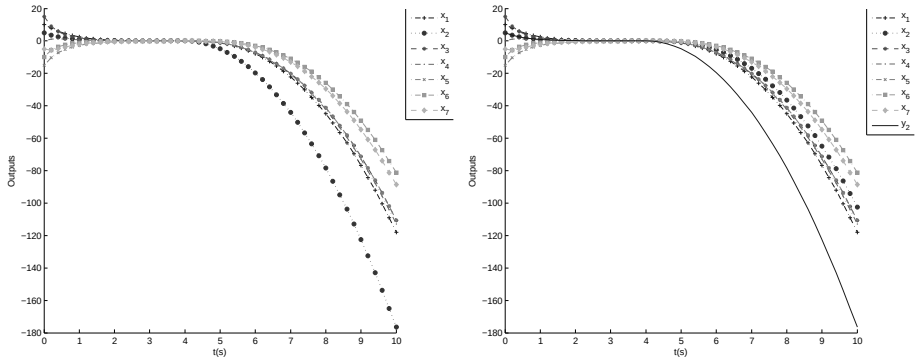
where $l_{k\bar{k}}^T = l_{\bar{k}k} \in \mathbb{R}^{N-1}$ are obtained from a permutation of the Laplacian matrix, P_k, so that node k is the last node:

$$P_k^{-1}\mathscr{L}P_k = \begin{bmatrix} \mathscr{L}_{\bar{k}} & l_{\bar{k}k} \\ l_{k\bar{k}} & \mathscr{L}_k \end{bmatrix}. \tag{3.16}$$

If we apply a similar procedure to the fault scenario described in (3.5) we can see how both cases are similar:

$$\begin{cases} \dot{\mathbf{x}}_{\bar{k}} &= -\mathscr{L}_{\bar{k}}x_{\bar{k}} - l_{\bar{k}k}y_k \\ \mathbf{y}_{\bar{k}} &= \mathbf{x}_{\bar{k}} \\ \dot{\mathbf{x}}_k &= -\mathscr{L}_k x_k - l_{k\bar{k}}y_{\bar{k}} + f_k \\ \mathbf{y}_k &= \mathbf{x}_k \end{cases}. \tag{3.17}$$

In fact, the interaction between node k and the rest of the network is made through its out-going data y_k, thus if the malicious data f_{s_k} is chosen in a way such that y_k follows the trajectory due to the effect of f_k, the healthy part of the network will see the same behavior from node k in both the fault and communication attack scenarios. Such similarity between the different scenarios can be seen in Figure 3.15(a) and Figure 3.23(a) in the Section 3.4, which are reproduced in Figure 3.9.



(a) Fault (b) Communication Attack

Figure 3.9: Fault and communication attack in node 2

By observing both figures, it can be easily seen that all the nodes except node 2 have the same trajectory and so behaved the same way in both cases. As for the faulty node 2, it reacted differently, since on the fault scenario it had a disturbance in its own process dynamics and in the communication attack scenario, it followed the rest of the network.

This last observation is therefore the key to design a method capable of detecting and isolating an attack on the communications of a single node. Due to the similarity of both the fault and the attack as seen from the rest of the network, a possible method to detect and isolate the misbehaving node is the one proposed in Section 3.2, where each node monitors its own neighbors. By applying this method, the network is able to acknowledge that there is a problem in a specific node and know its identity, but it still cannot distinguish if it is a physical fault or a communication attack and therefore some more information is needed.

One way of isolating the compromised node takes into account the fact that this node will react to the "strange" behavior of the rest of the network, as seen in Figure 3.9(b). Although the rest of the network is not able to distinguish between a fault or an attack with the available information from the communications, the node itself can, using its own internal measurement.

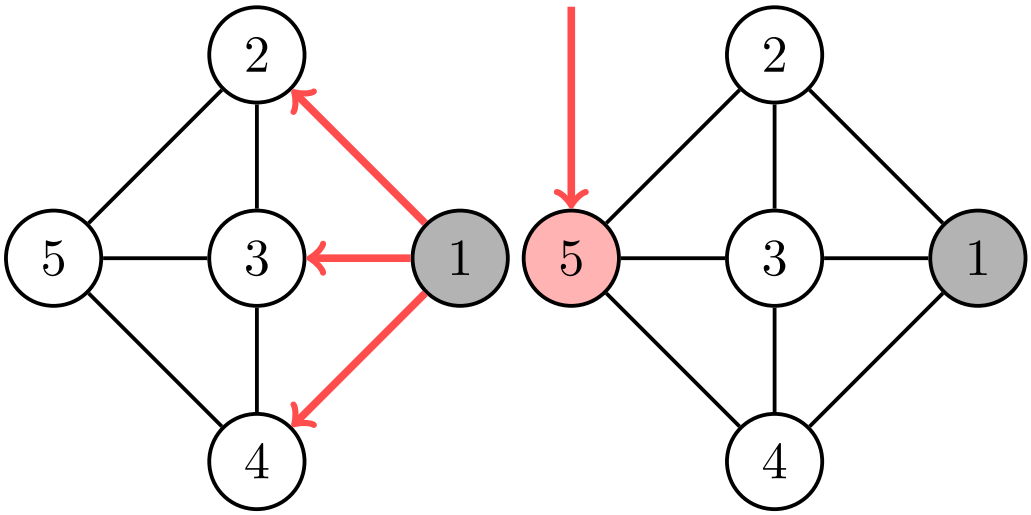
The proposed solution is to add one more observer to the FDI scheme proposed in Section 3.2, which should then monitor the system in (3.14) for a fault signal with the distribution vector given by $I_k l^k$. Note that in this section we assume that each node does not have access to its own communicated measurement, but only to its internal one. Therefore, in order to implement the proposed solution, we will describe the network with only the k^{th} node being compromised and observed from a neighboring node i as:

$$\begin{cases} \dot{\mathbf{x}} &= -\mathscr{L}\mathbf{x} + b_f^k f_k \\ \mathbf{y}_i &= C_i \mathbf{x} \end{cases}, \tag{3.18}$$

with $b_f^k \in \mathbb{R}^N$ being the distribution vector of f_k and $\mathbf{y}_i = \begin{bmatrix} w_i \, y_{i_1} \, \cdots \, y_{i_{|N_i|}} \end{bmatrix}^T$ being the available information at node i.

The main concept of this approach is that each node should be able to check if it is behaving correctly by using the communicated measurements from its neighbors and its own internal measurement. As the compromised node is not aware of its own transmitted data and believes it is sending the right information, it will seem to it that all the other nodes in the network are misbehaving, which is modeled by the term $b_f^k = I_{\bar{k}} l^k$.

Note that the Theorem 3.2.3 still holds for this particular distribution vector, as rank $\left(C_i I_{\bar{k}} l^k\right) =$ rank $\left(C_i l^k\right) = 1$ since the network is connected, hence such FDI scheme is feasible.



(a) Attack in node 1 (b) Fault in node 5

Figure 3.10: False alarm when monitoring for attacks in node 1

However, there is at least one scenario different from an attack where a fault may have the distribution vector given by $b_f^k = I_k l^k$, which is presented in Figure 3.10.

From Figure 3.10(b), we see that a fault in node 5 actually affects all of the 1^{st} node's neighbors in the same way. For the FDI system implemented in node 1, this behavior will be interpreted as if there was an attack on the communications, as shown in Figure 3.10(a), and so node 1 would redefine its security keys or perform another corrective procedure. Although a false alarm may happen, this will have little significance, both because a redefinition of the security keys will not have great impact of the network, but also because node 1's neighborhood should be able to detect the fault in node 5 and disconnect it from the network before the false alarm of the FDI system in node 1.

3.4 Simulation Examples

Some simulation results will now be presented, in order to verify the methods proposed in the previous sections and to analyze the differences between an attack and a fault and their impact on the network.

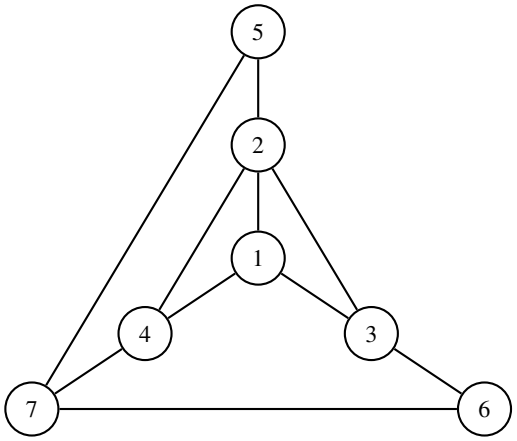


Figure 3.11: Example of a healthy network

3.4.1 Physical Fault in a node

We will now apply the FDI scheme described in Section 3.2 and implement a bank of observers in each node that is able to detect and isolate faults in its neighborhood. For simplicity's sake, we will only show the FDI scheme implemented in one node, subject to different types of faults inside and outside of its neighborhood.

Consider the network presented in Figure 3.11, with the respective graph's Laplacian being:

$$\mathscr{L} = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & 3 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 & 3 \end{bmatrix}. \tag{3.19}$$

The node running the FDI component will be node 1 and so the set of independent measurements will be formed by the state of nodes 1 to 4. As for the faults, since node 1 is observing its neighbors, the set of faults we are interested in are those affecting directly these neighbors and so, according to the concepts presented in Section 3.2, node 1 will have a bank of $|N_1| = 3$ observers, each one insensitive to a fault in a distinct neighbor.

An UIO with dynamics described by (2.15) applied to the system in (3.8) for each neighbor $k \in N_1$ can now be computed using the method described in Section 2.3, resulting in a bank of UIOs to monitor the neighbors of node 1. In the examples presented during this section, no requirements other than stability have been made to the observers, therefore the observer gains are design arbitrarily so that each observer is stable.

We will demonstrate how this FDI scheme works with different types of faults driving not only the set of neighbors being monitored, but also other nodes in the network.

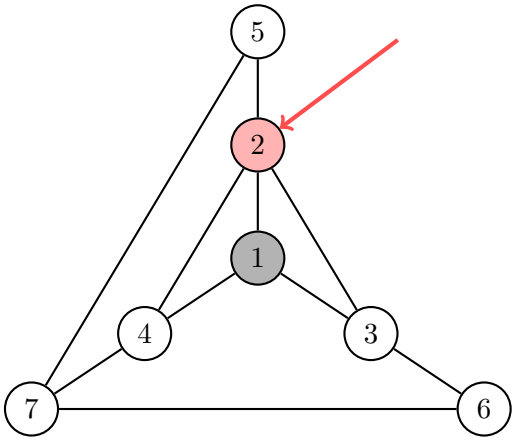


Figure 3.12: Network with faulty node 2 and observer node 1

Fault in node 2. In this example, node 2 is the one having a fault f_2 at the time instant $t = 4s$, while all other faults are not active - as shown in Figure 3.12. The simulation results presented in Figures 3.13 - 3.15 show the network's response and the response of the UIO bank for three different types of faults:

i) $f_2 = 10$

ii) $f_2 = 10\sin(t)$

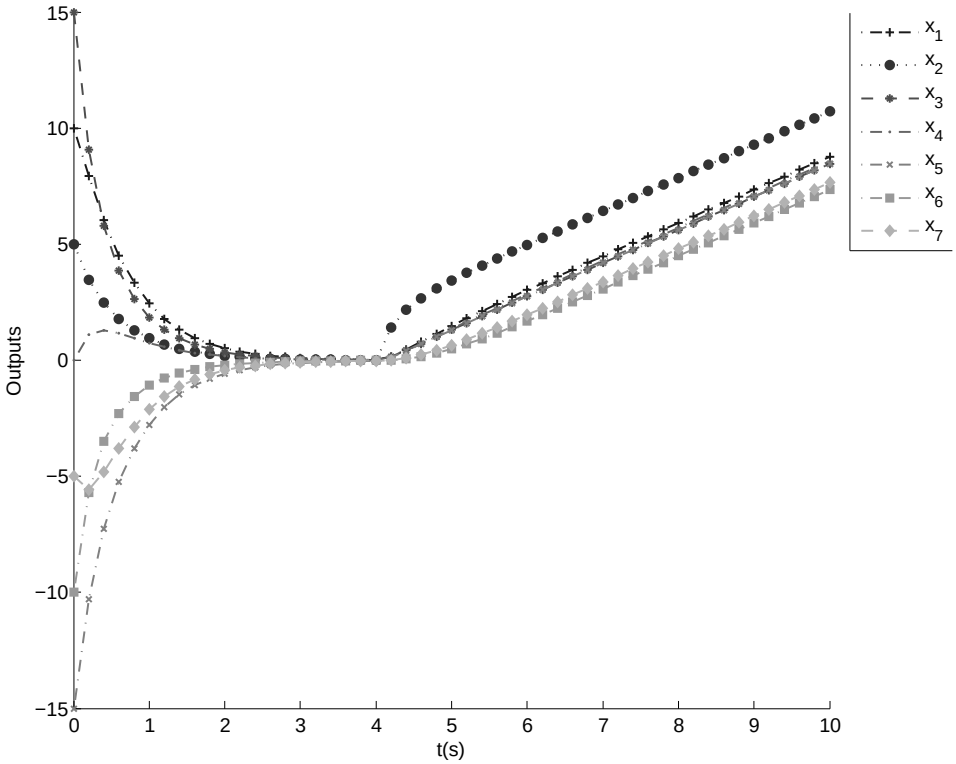
iii) $f_2 = -u_2 + \dot{x}_2(4) - 9.8(t-4)$

The first two faults should demonstrate how the FDI system responds to constant and sinusoidal fault signals, while the last one models a specific situation taken from [22] where the nodes are in fact Unmanned Air Vehicles (UAVs) performing a consensus algorithm on their altitude when suddenly one of them begins a free fall.

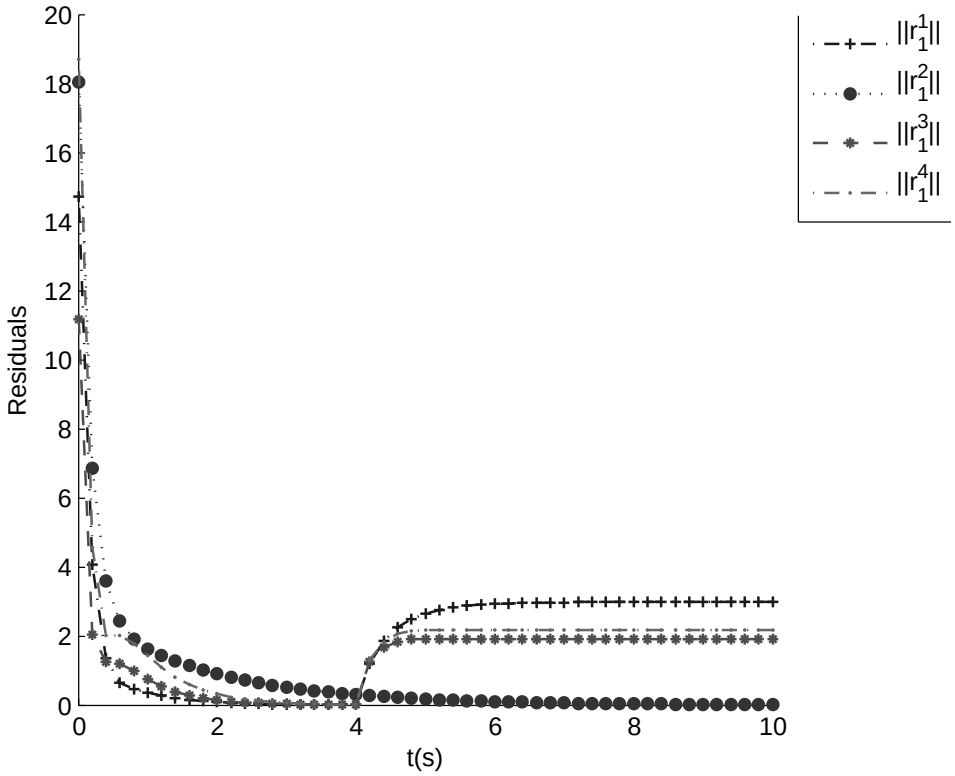
As it can be seen in Figures 3.13 - 3.15, in all cases the state of the network suffers a drastic change, which can be less or more significant depending on the application. Assuming the network is trying to achieve convergence on a function of the initial values, in the presence of a fault the consensus may not even be achieved, or it may achieve a wrong value. The example concerning the UAVs shows that the safety of the agents is also affected, as they will eventually crash.

Analyzing the residuals, we observe that r_1^2 stays in zero, while the residuals r_1^3 and r_1^4 react to the presence of the fault, thus the fault could be detected and isolated according to the threshold logic in (2.13).

From the results we can also conclude that, in order to allow the FDI system to detect and isolate faults within the transient response, the thresholds used by the FDI system should be dynamic. One possible function for the thresholds could be an upper bound on the convergence rate of each observer, controlled by the choosing an appropriate matrix K_1 in (2.9). Once the residuals reach the steady state, the threshold value may be constant.

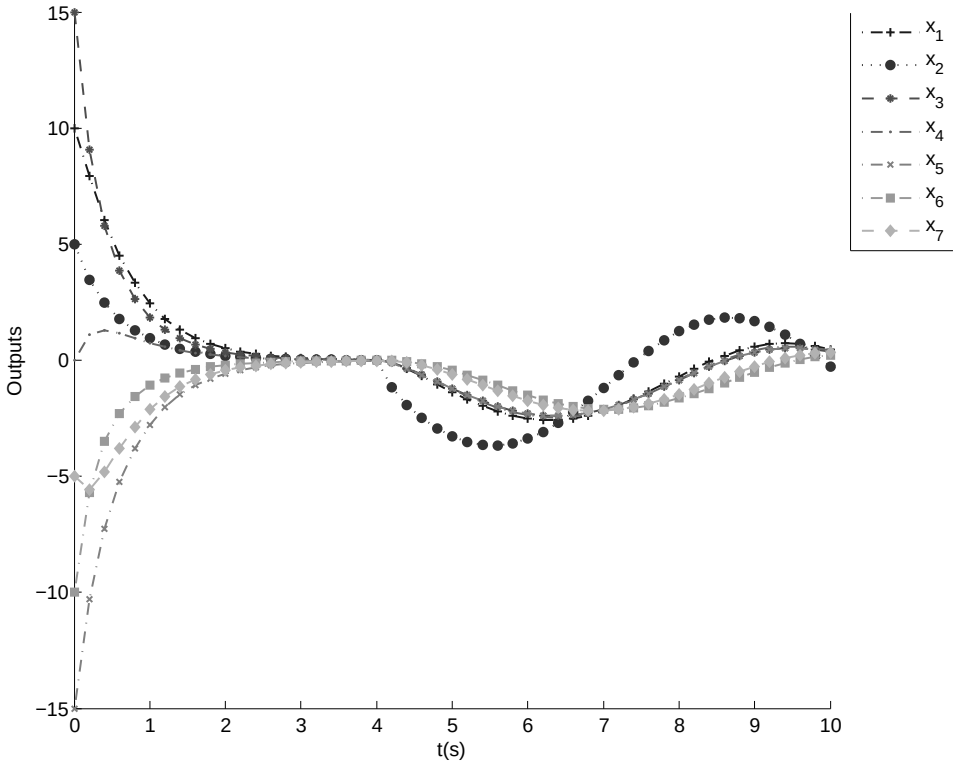


(a) Outputs

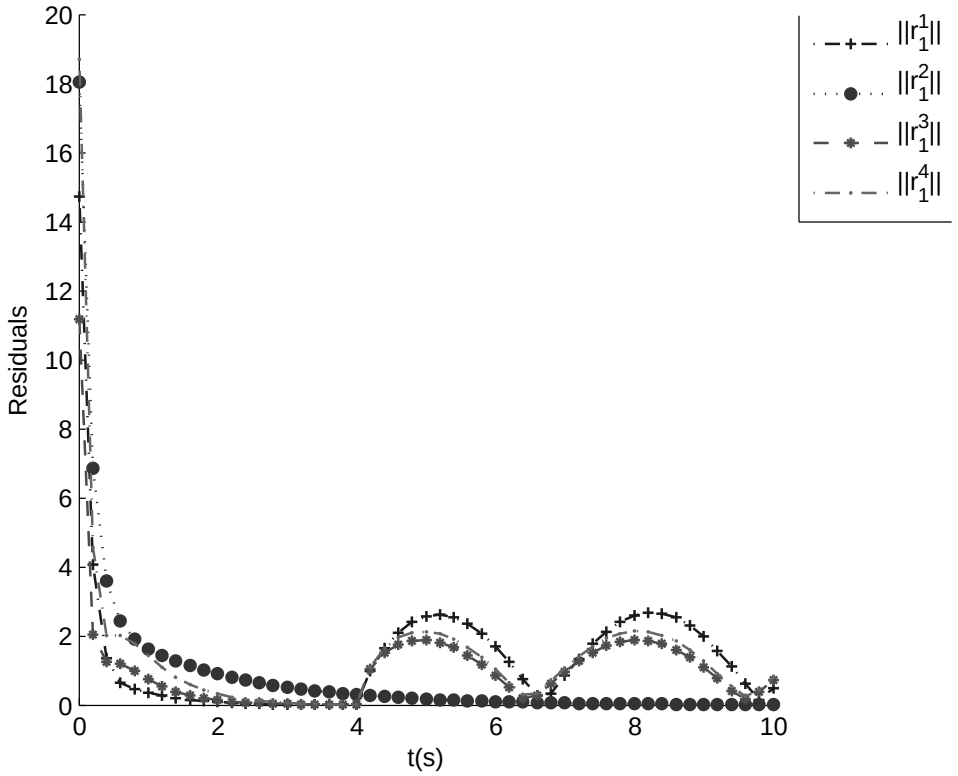


(b) Residuals

Figure 3.13: Fault *i)* in node 2

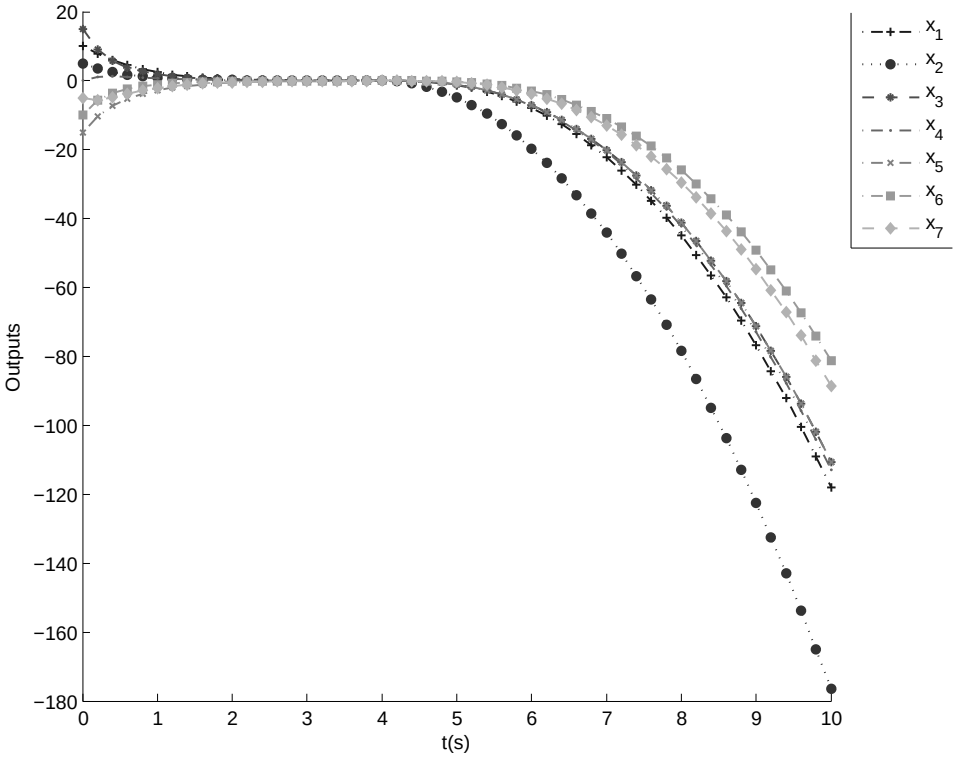


(a) Outputs

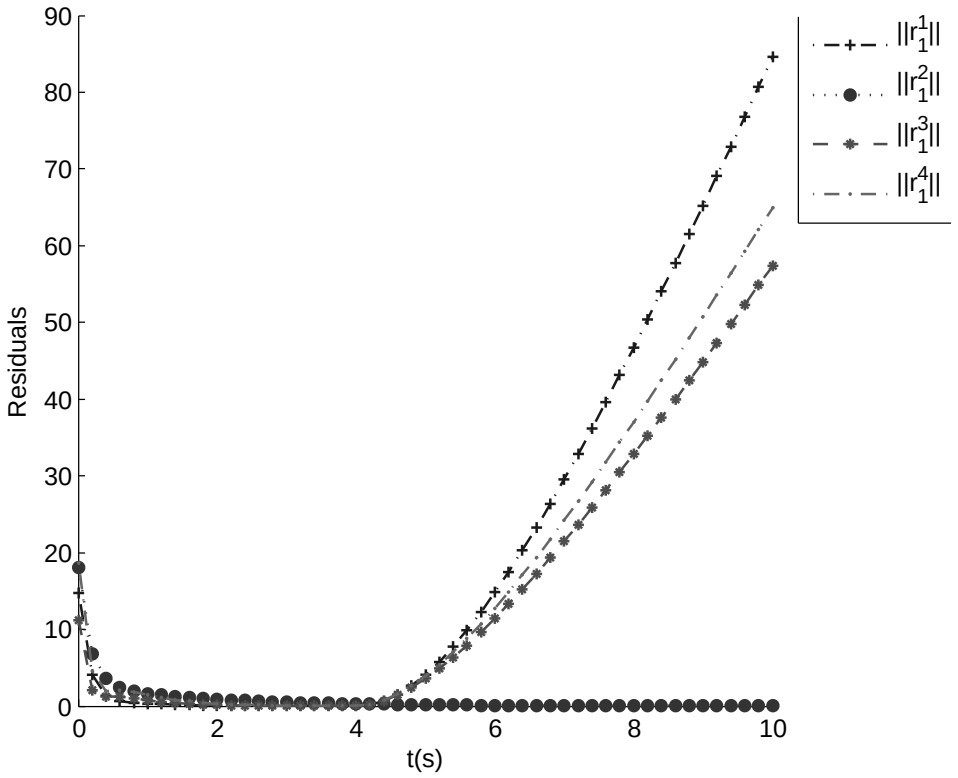


(b) Residuals

Figure 3.14: Fault *ii)* in node 2



(a) Outputs



(b) Residuals

Figure 3.15: Fault *iii)* in node 2

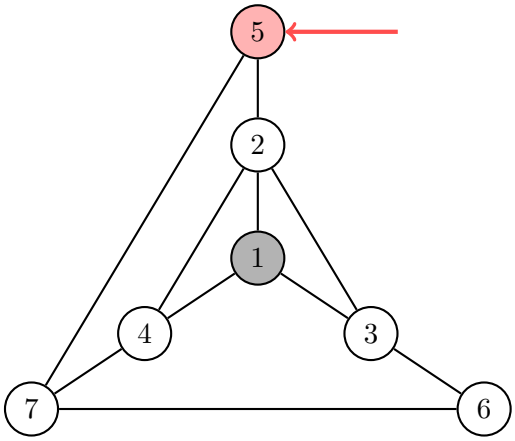


Figure 3.16: Network with faulty node 5 and observer node 1

Fault in node 5. We now consider the case where node 5 is the misbehaving one, as represented in Figure 3.16, once again at the time instant $t = 4s$. Note, however, that the UIO bank only monitors the neighbors of node 1, *i.e.* all the observers are sensitive to the fault in node 5. The responses of the observers and the network are presented in Figures 3.17 - 3.19.

From the results we can see that, as expected, the residual r_1^2 is no longer zero, although its value is small close to the time instant when the fault occurs, compared to the other residuals. The fact that r_1^2 is the smallest residual at the time interval close to $t = 4s$ has to do with the fact that node 5 is in the neighbor set of node 2, which does not happen with nodes 3 and 4. Since r_1^2 is insensitive to faults in node 2, the sensitivity to faults in the neighbor set of node 2 is also reduced.

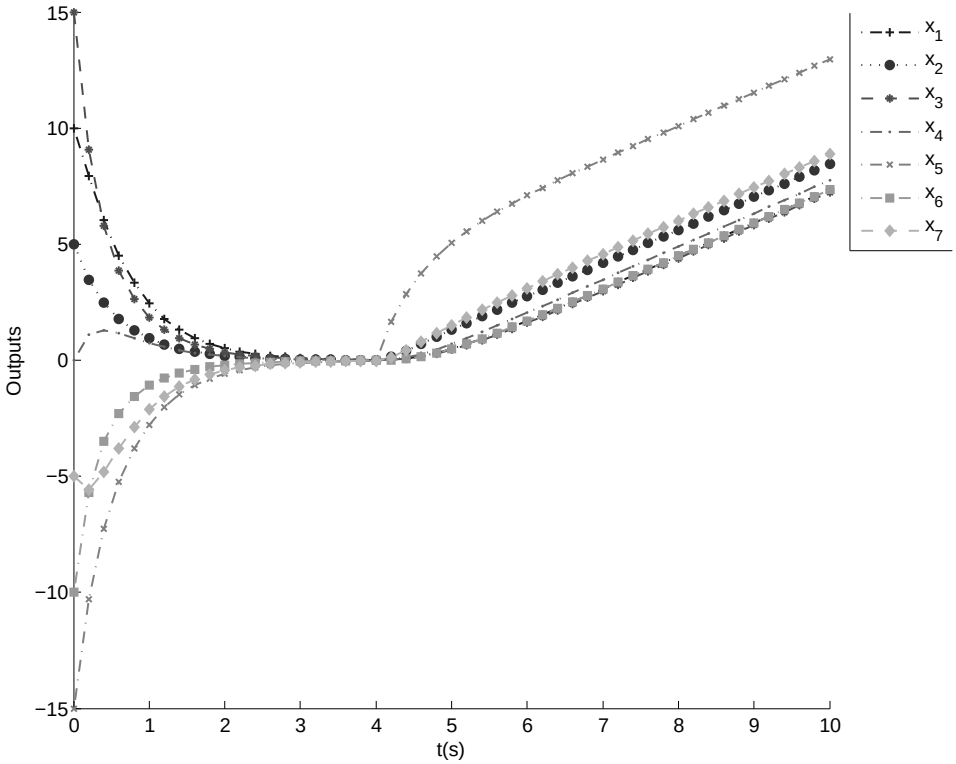
In fact, if node 5 was connected only to node 2, the residual r_1^2 would be zero for faults in both nodes. This suggests that the network's connectivity has a great influence in the performance of the FDI component. Furthermore, if both node 1 and 2 are running a FDI component and node 5, whose only neighbor is 2, is having a fault, there should be some coordination or priority concerning which node should act on the presence of such fault.

However, due to the fact that r_1^2 is small, care should be taken when choosing the threshold, as it may cause the removal of the link between nodes 1 and 2. This may not be such a strict constraint when choosing the thresholds, since node 2 should react to the fault before 1, assuming the FDI systems in both nodes have similar behaviors.

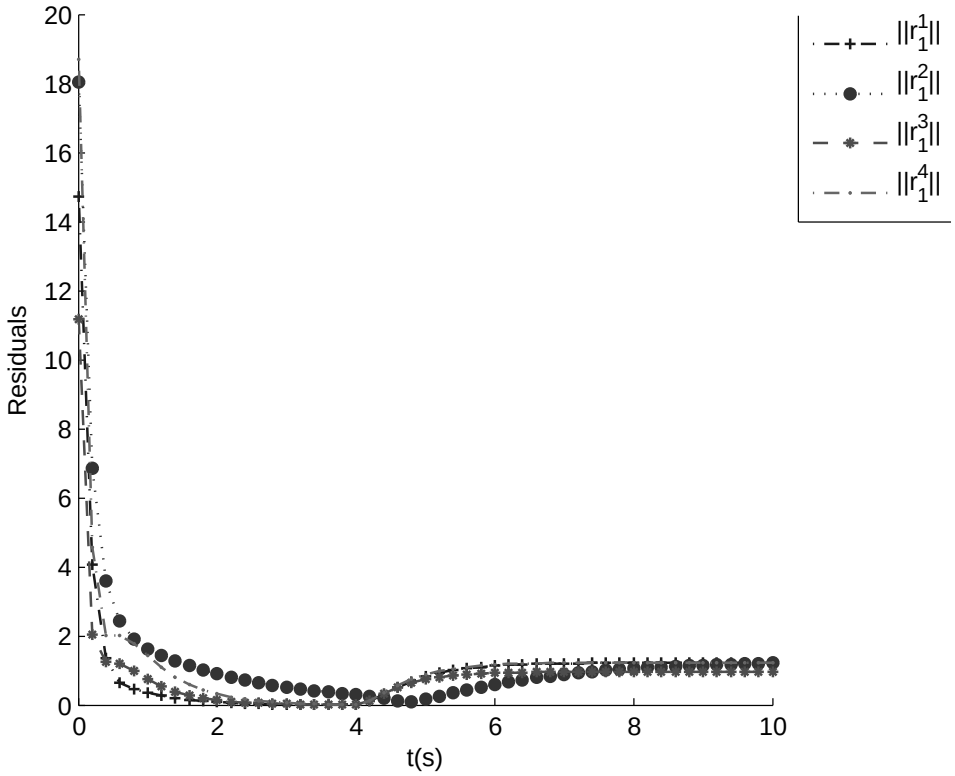
3.4.2 Deception Attack in a node

In this type of attack, the attacker has gained access to security passwords and protocol for the communication and is able to introduce false data in the communication links. The results we will now present assume the attacker was able to compromise node k in the network and is now changing the outgoing data from that agent. Such system can be described by (3.14).

The following results focus on the response of the FDI scheme proposed in Section 3.3 to these type of communication attacks.

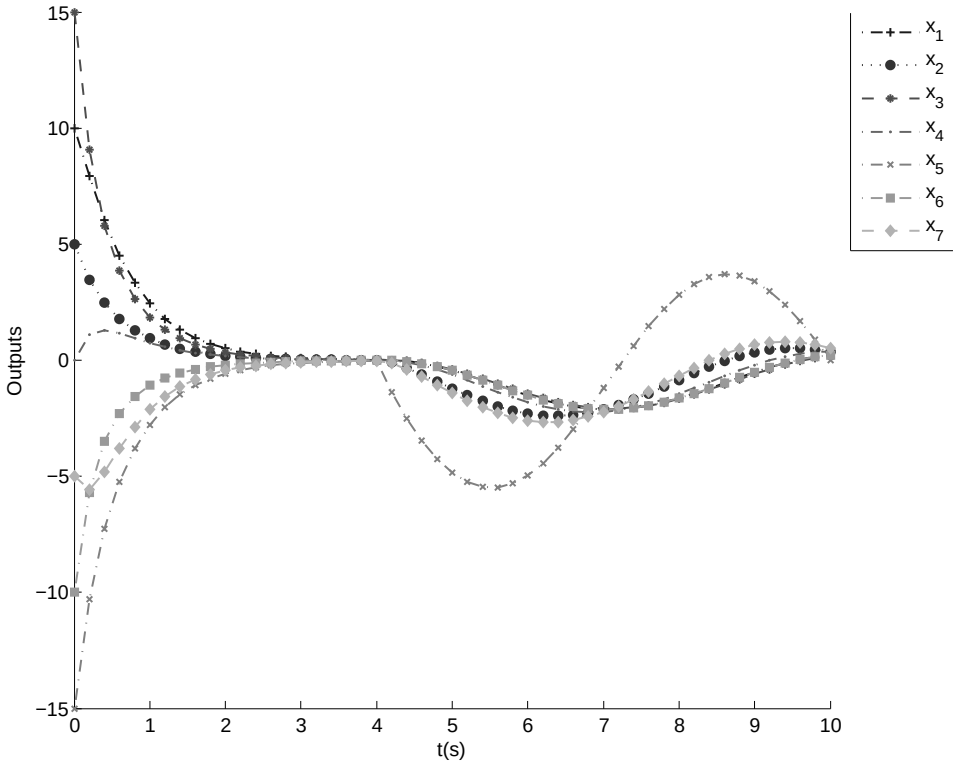


(a) Outputs

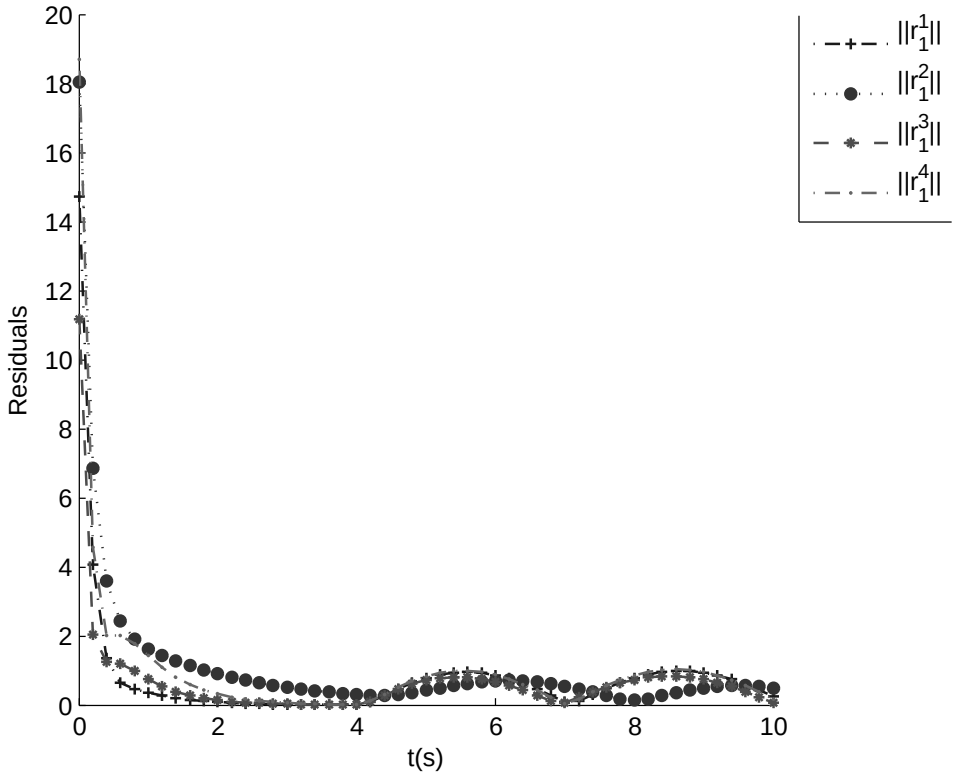


(b) Residuals

Figure 3.17: Fault *i)* in node 5

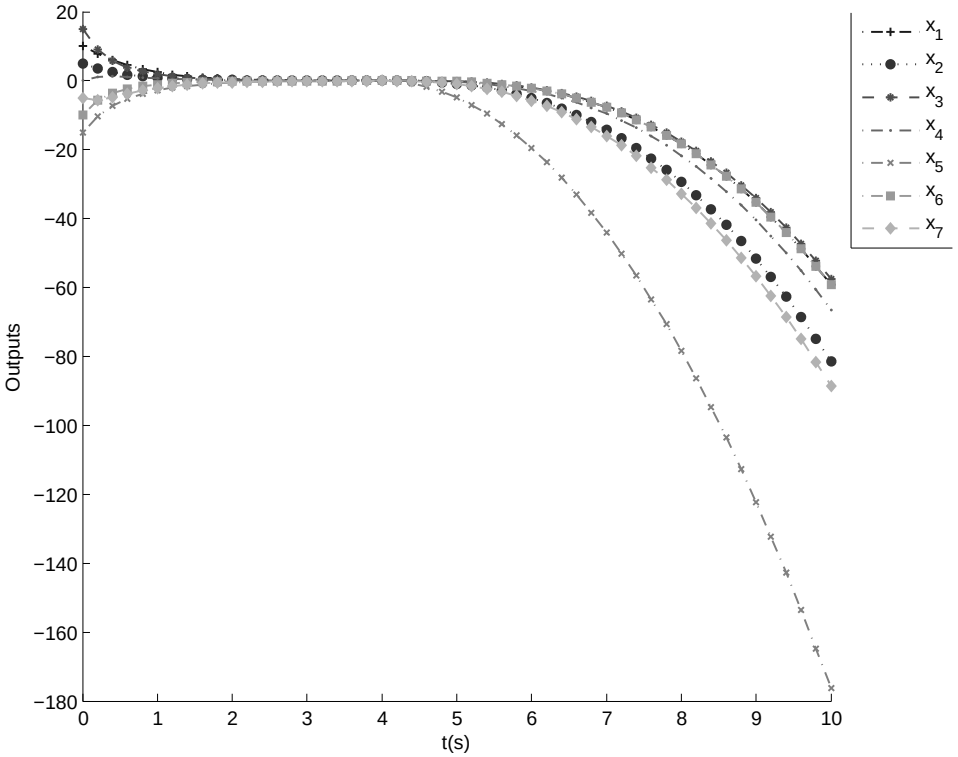


(a) Outputs

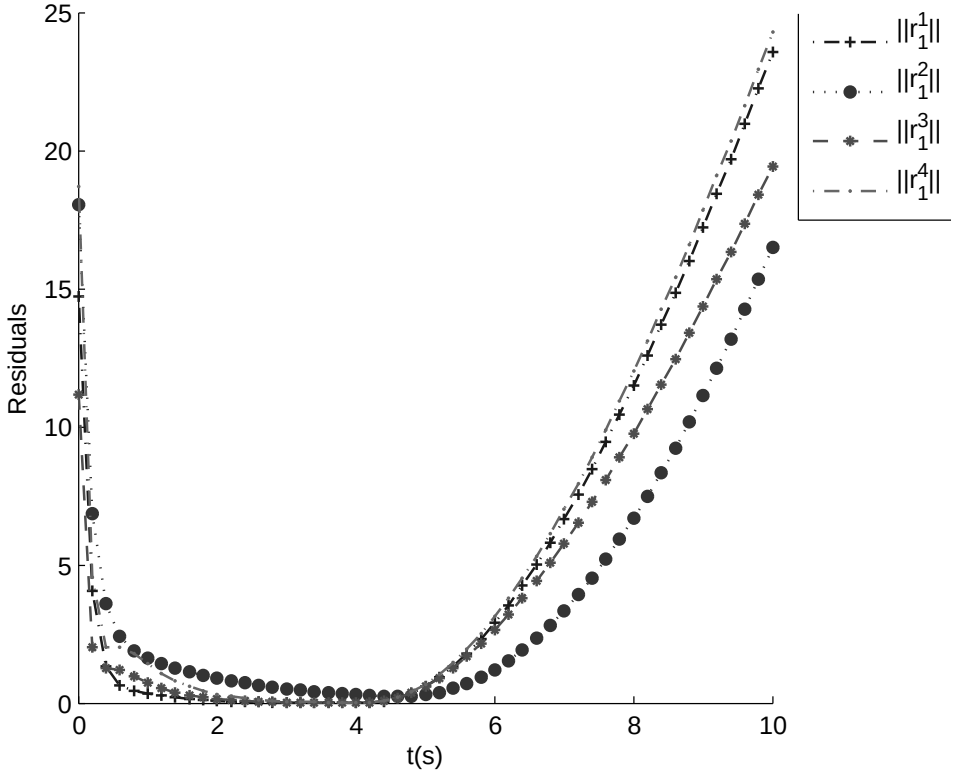


(b) Residuals

Figure 3.18: Fault *ii)* in node 5



(a) Outputs



(b) Residuals

Figure 3.19: Fault *iii)* in node 5

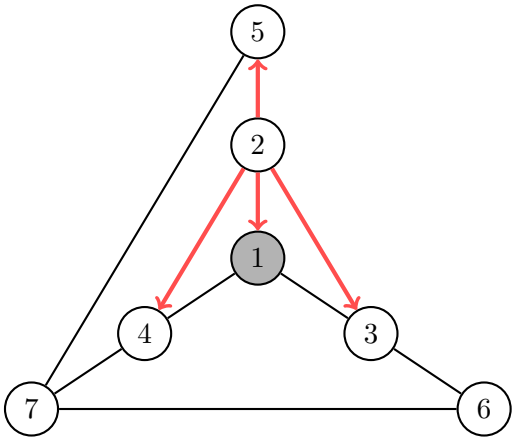


Figure 3.20: Network with an attack in node 2 and observer node 1

Deception attack in node 2. As when analyzing the FDI system's response to a fault, we will present the response of the the same FDI system to three different types of malicious data inserted into the network from node 2, as shown in Figure 3.20:

i) $f_{s_2} = 10$

ii) $f_{s_2} = 10\sin(t)$

iii) $f_{s_2} = -x_2 + \dot{x}_2(4)(t-4) - \frac{9.8}{2}(t-4)^2$

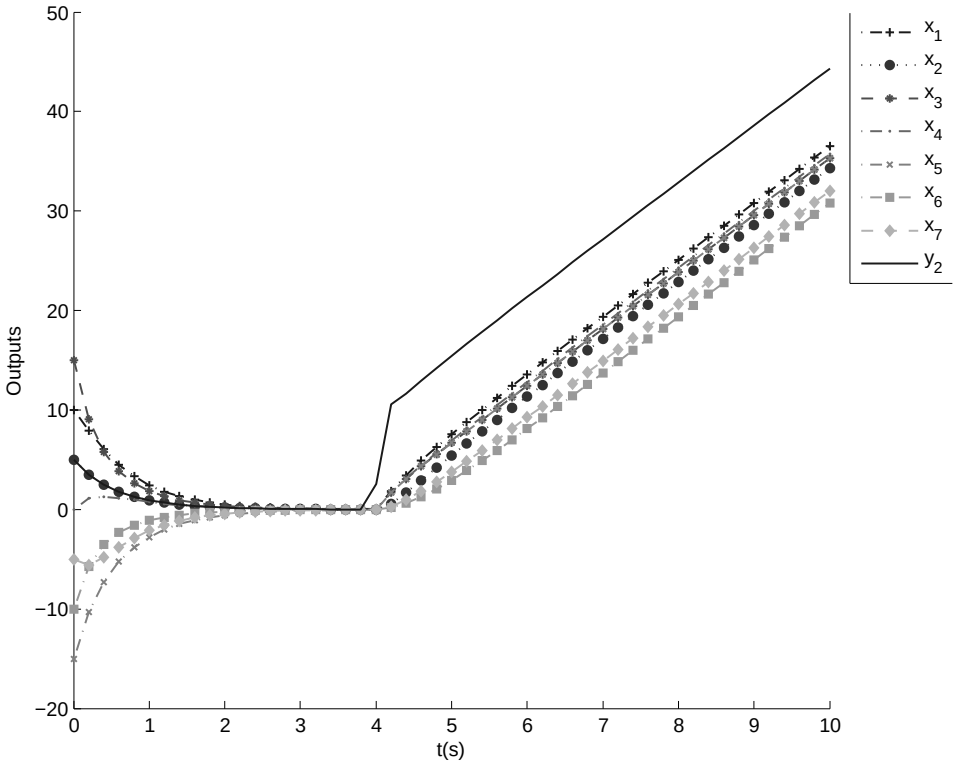
Once again, considering this to be a network of UAVs, we can see that the third case corresponds to malicious data simulating a free fall, but all UAVs are in fact fault free.

Observing the results in Figures 3.21 - 3.23, we conclude that the same FDI component in node 1 is also able to detect the misbehavior caused by the insertion of malicious data in one of its neighbors, according to the model in (3.14).

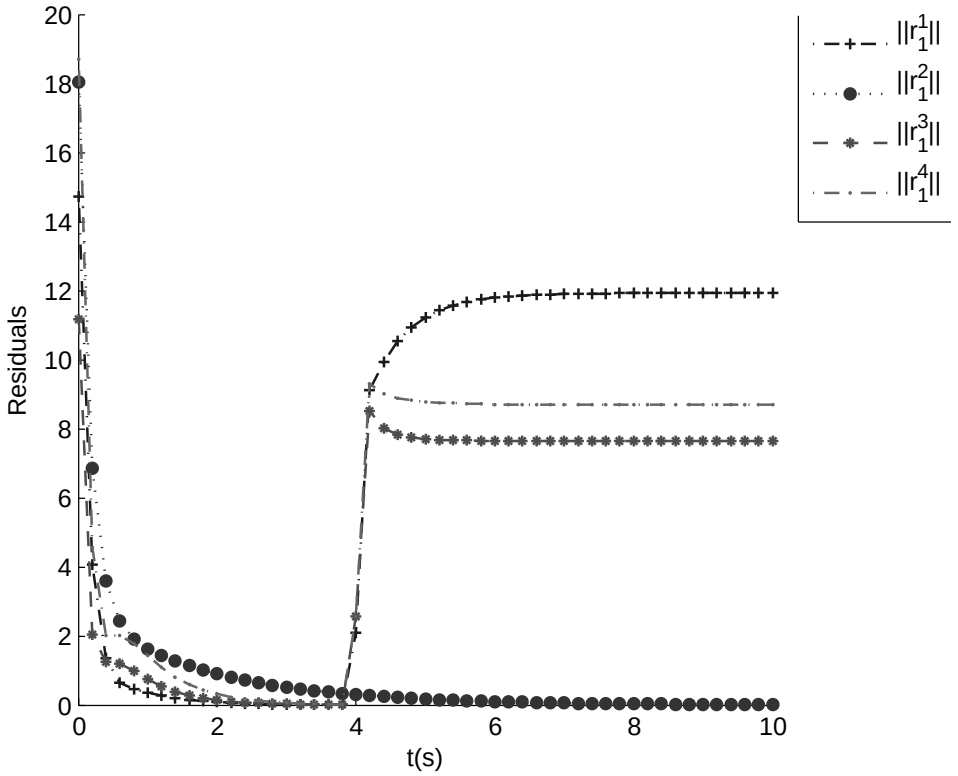
However, comparing Figure 3.13 and Figure 3.21, we see that the response of each node in the network is the same in both cases, with node 2 as the exception. This indicates that, although both abnormal situations have different causes, neighboring nodes monitoring node 2 with this FDI scheme are able to detect the misbehavior, but not to determine its cause, which indicates the necessity of a different scheme.

Note that if this FDI scheme is implemented in node 2 as well so that it is monitoring its own interactions with the network by evaluating r_2^2, as proposed in Section 3.3, thus it is capable of detecting the abnormal behavior of all its neighbors caused by an attack on its own communications. At this stage, node 2 can redefine the security keys in order to correct the situation and prevent itself from being disconnected from the network by its neighbors. Such situation is analyzed later for an attack on node 1.

Another interesting observation is that node 2, although computing its own control law without the direct influence of the malicious data, follows the rest of the network as expected, since he is still running the consensus algorithm.

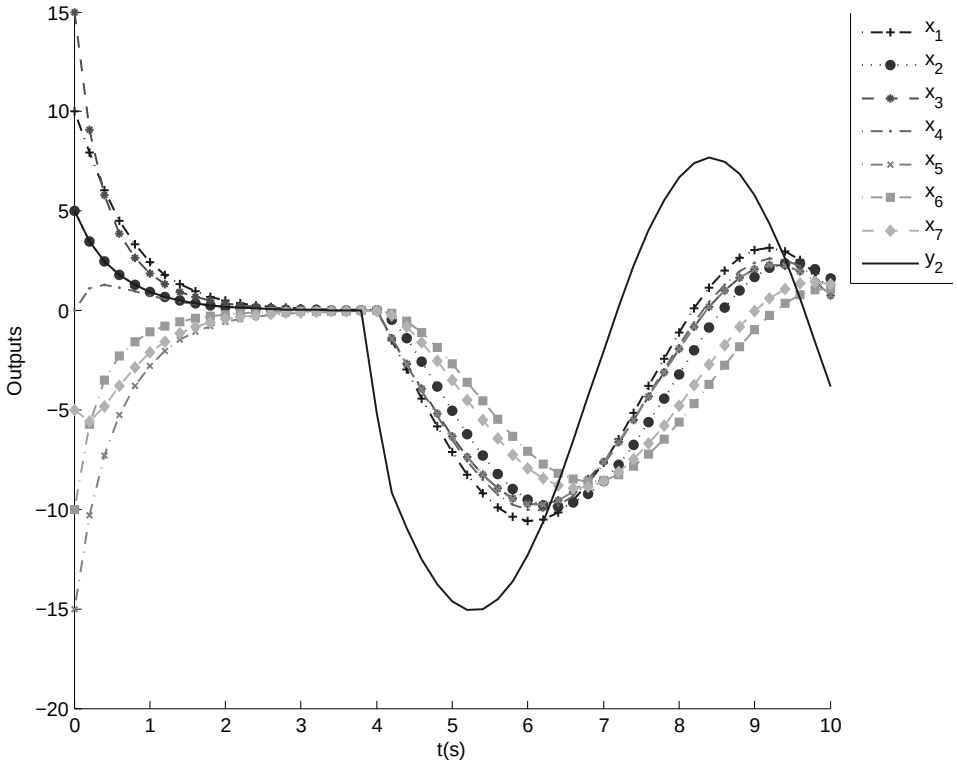


(a) Outputs

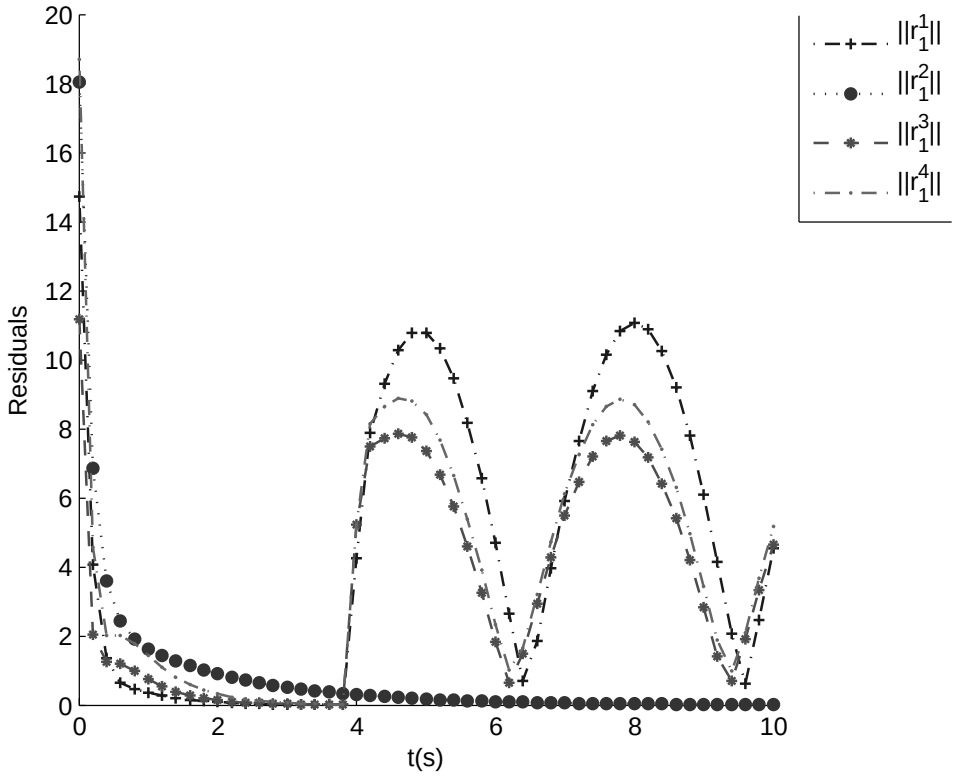


(b) Residuals

Figure 3.21: Deception attack *i)* in node 2

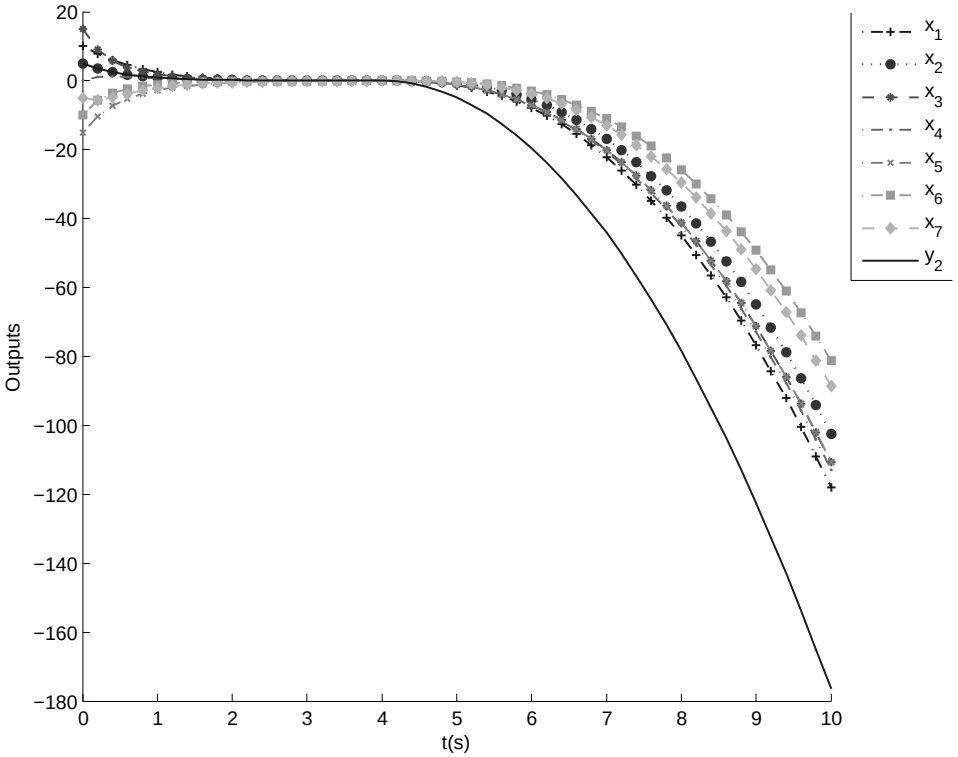


(a) Outputs

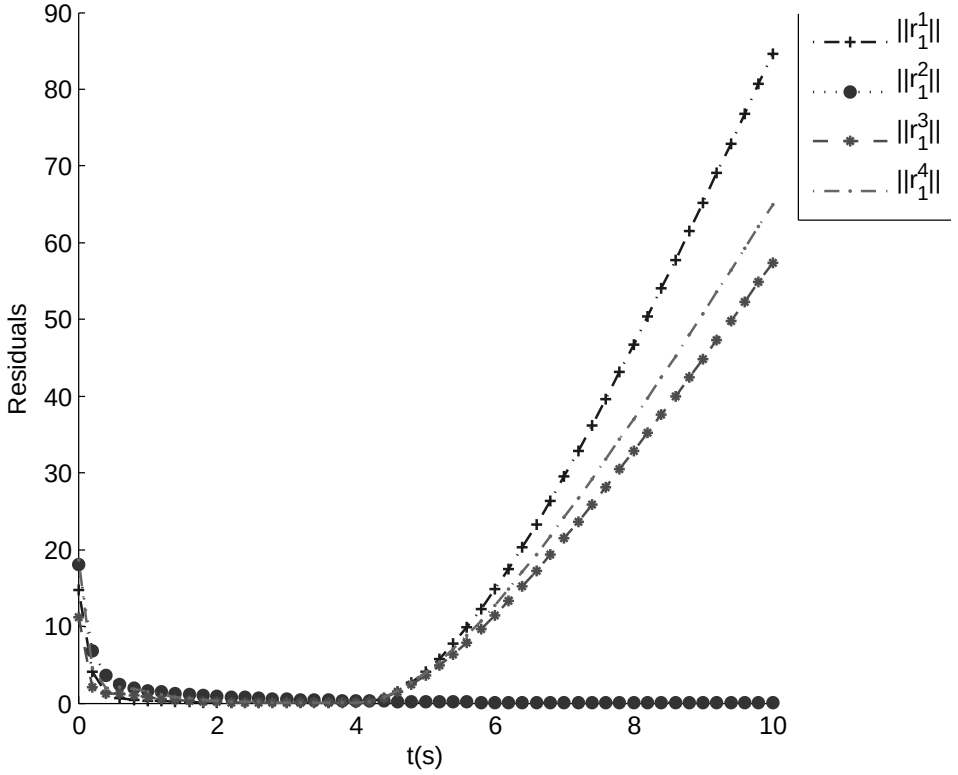


(b) Residuals

Figure 3.22: Deception attack *ii)* in node 2



(a) Outputs



(b) Residuals

Figure 3.23: Deception attack *iii)* in node 2

These examples allow us to demonstrate the drastic effect of erroneous data in a Networked Control System (NCS) with distributed controllers, affecting not only the convergence and performance, but also the safety of the NCS.

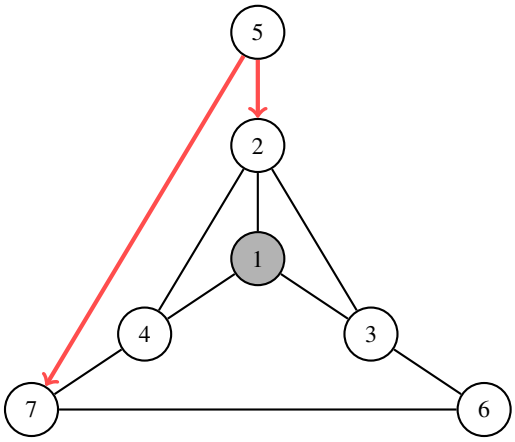


Figure 3.24: Network with an attack in node 5 and observer node 1

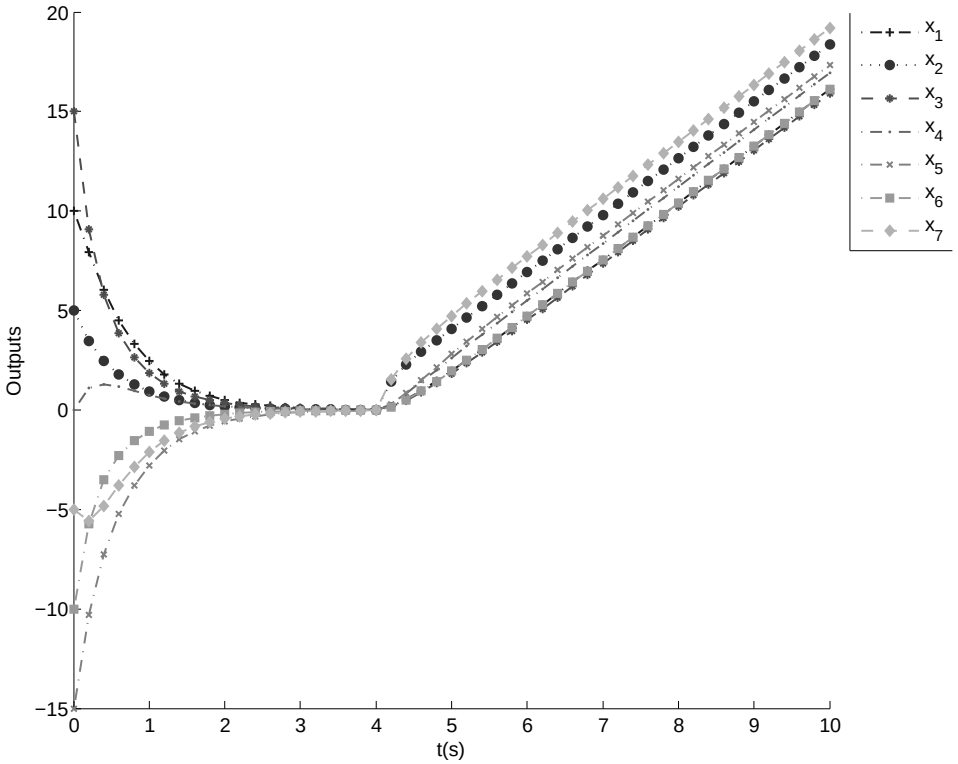
Deception attack in node 5. Similarly to the analysis of the FDI component's response to a physical fault, we will now present some results in a case where a node not belonging to the neighbor set node 1 is the compromised one. Again, the compromised node is agent 5, as depicted in Figure 3.24, and the types of malicious data are the same as when node 2 was compromised.

The results for such situations are presented in Figures 3.25 - 3.27.

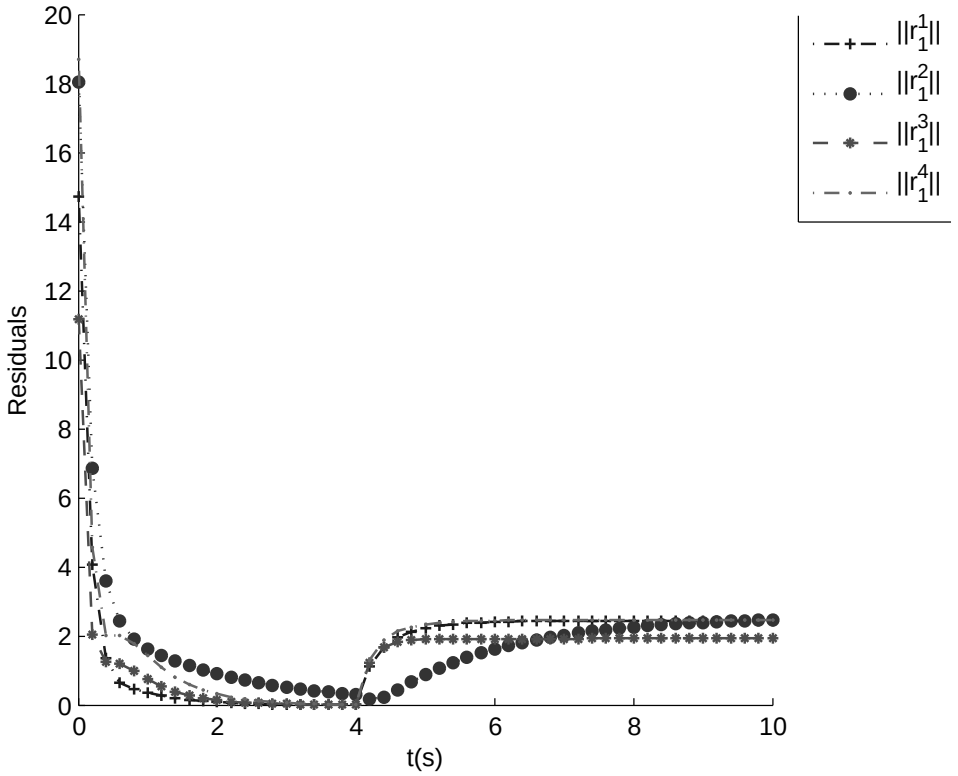
The responses of the FDI system in these cases are similar to the ones for a physical fault in the same node and so the same reasoning can be applied here.

Deception attack in node 1. An example where the deception attack is detected by the attacked node is now presented. As before, we focus on the FDI system in node 1 and how it reacts to an attack on the communications of the same node - represented in Figure 3.28 - which can be seen in Figure 3.29.

Note that the whole network is affected by the signal y_1, which is unknown to node 1 and is different from x_1, similarly to the scenario where node 2 was the compromised one. Furthermore, we see that the FDI system is able to successfully detect and isolate the "fault" to which r_1^1 is insensitive to, which in this case corresponds to an attack to the communications. As suggested before, at this stage node 1 can redefine its own security keys to prevent itself from being disconnected by its neighbors.

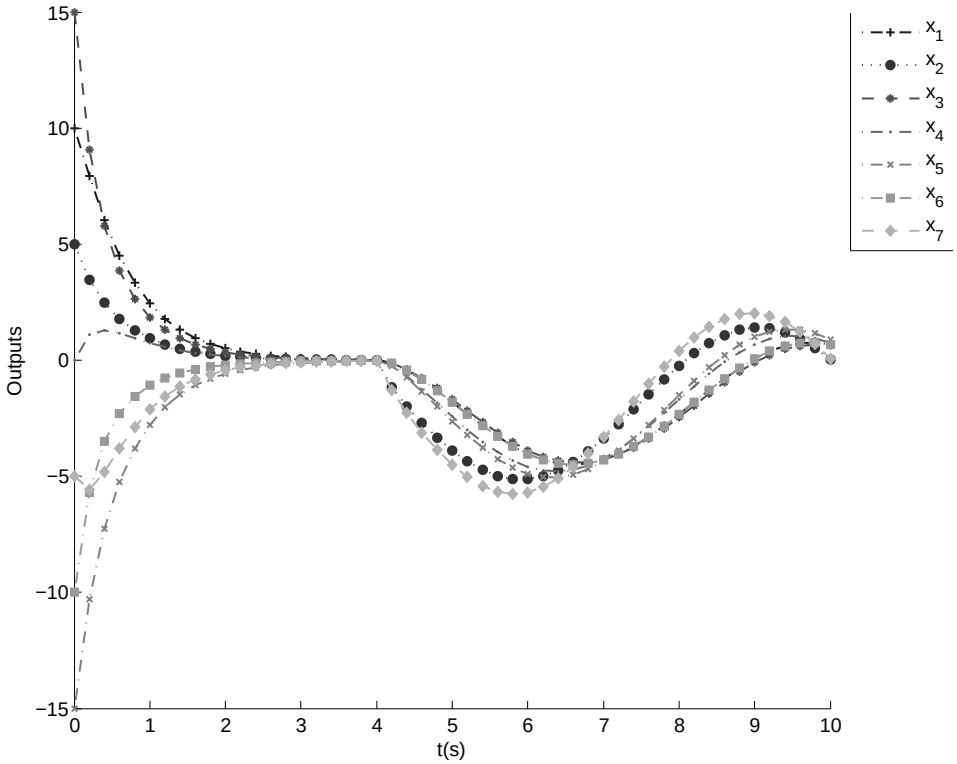


(a) Outputs

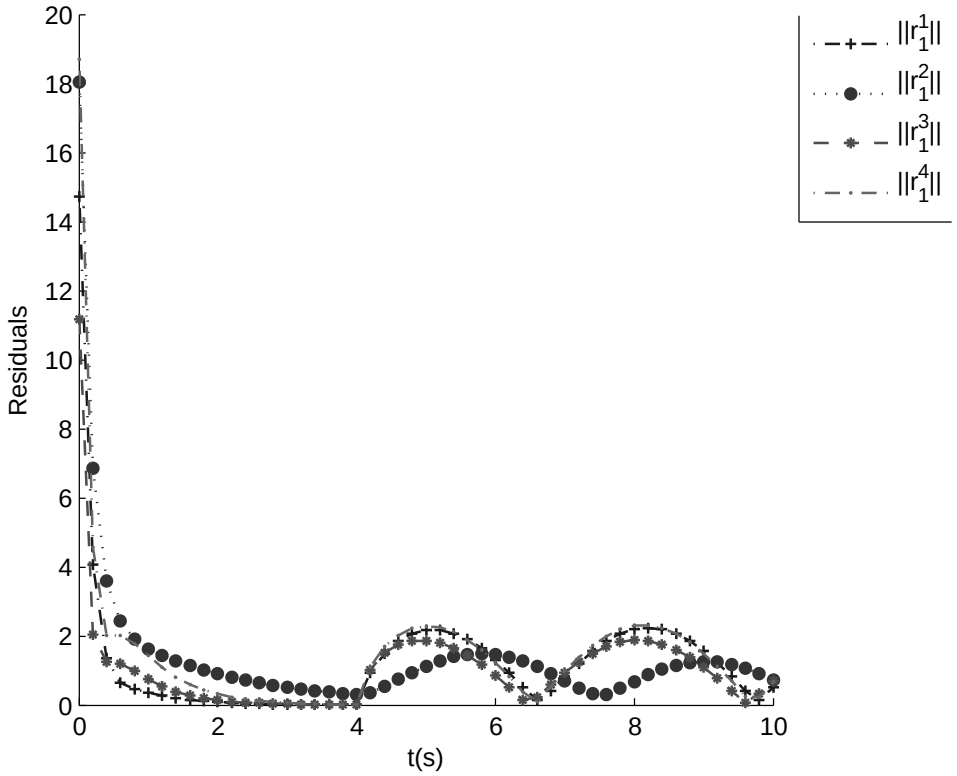


(b) Residuals

Figure 3.25: Deception attack *i)* in node 5

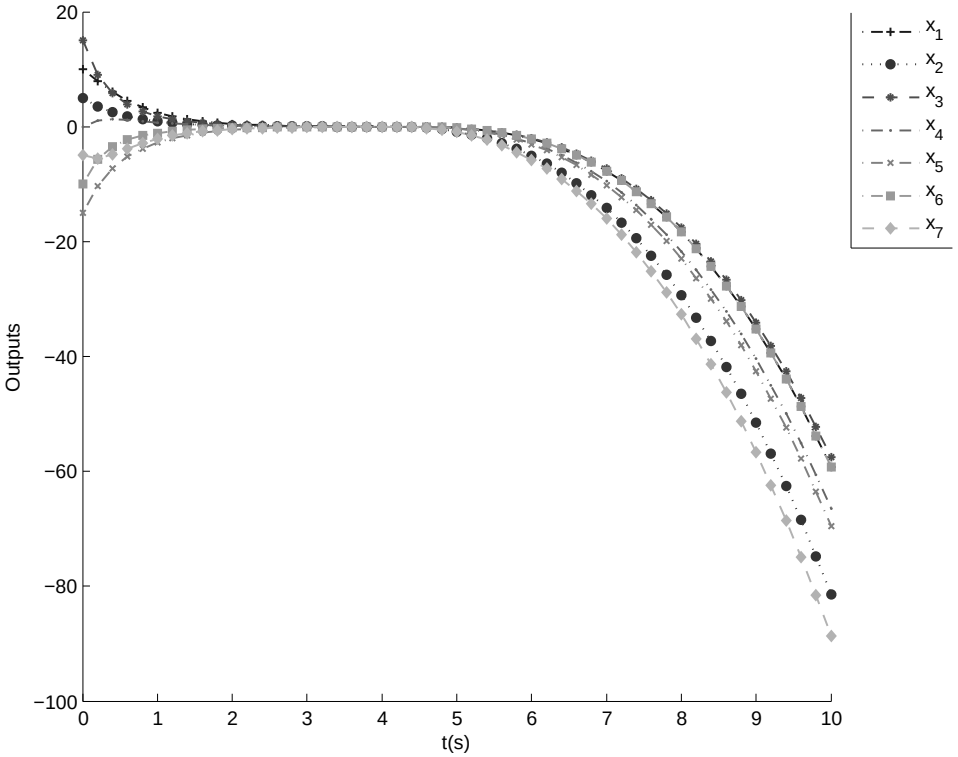


(a) Outputs

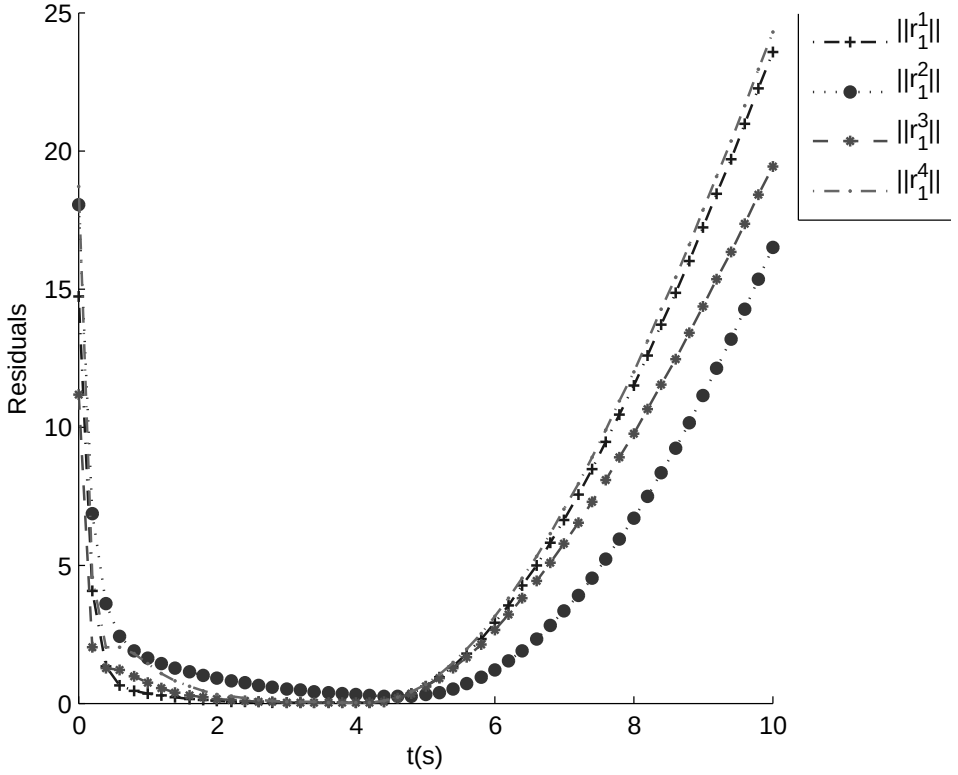


(b) Residuals

Figure 3.26: Deception attack *ii)* in node 5



(a) Outputs



(b) Residuals

Figure 3.27: Deception attack *iii)* in node 5

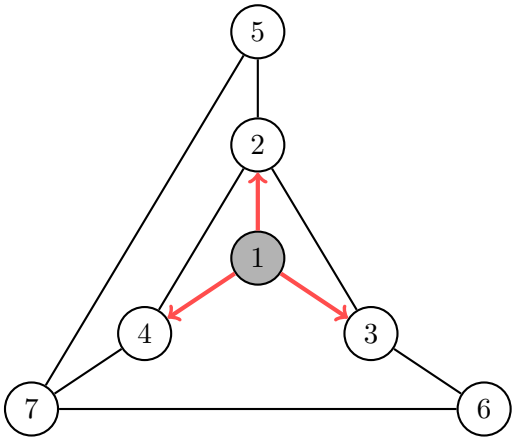


Figure 3.28: Network with an attack in node 1 and observer node 1

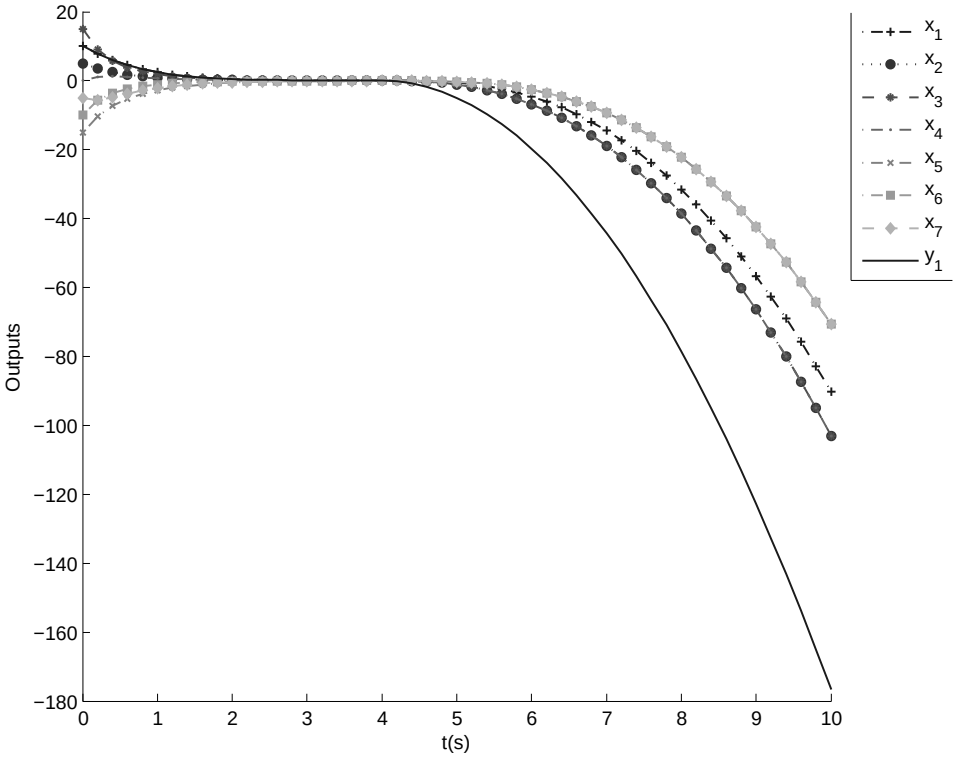
3.5 Summary

In this chapter we analyzed the consensus problem in NMAS from a security perspective, having formally described the effect of both faults and communication attacks on the network as a dynamical system. In order to detect and isolate both the faults and the attacks, model-based FDI methods suitable for a distributed implementation were proposed and the results of such FDI schemes were presented and discussed.

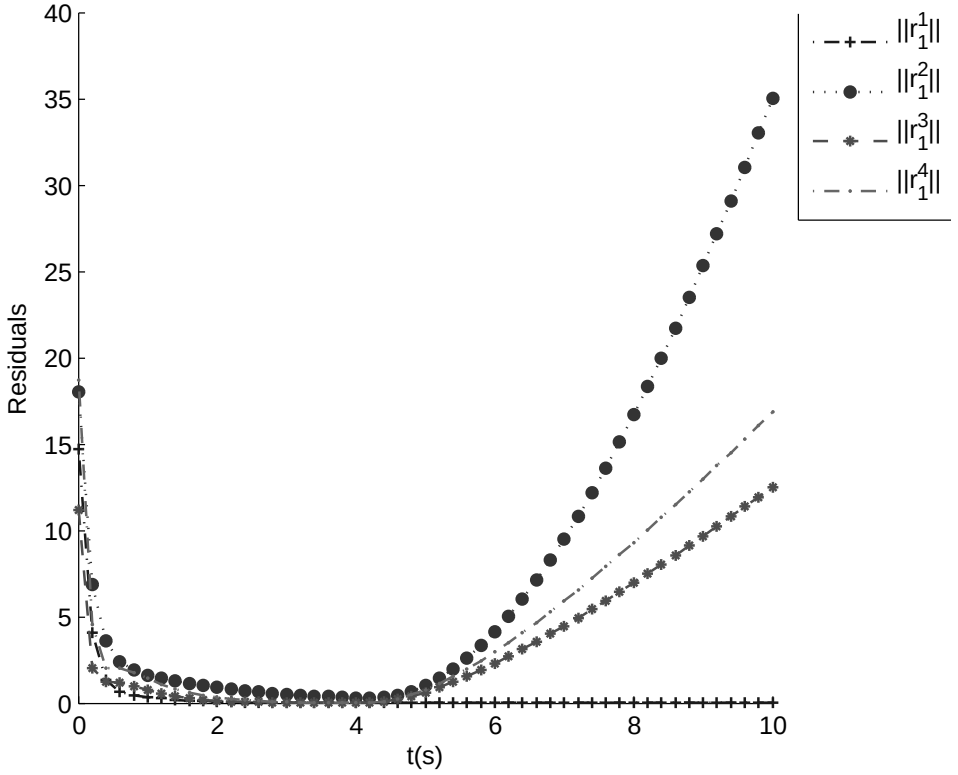
It was shown that a fault and a deception attack can have the exact same impact on the healthy part of the network, thus showing that the healthy network cannot distinguish between the two scenarios only with local information.

For the particular case of detecting faults, two possible methods of reducing the number of observer nodes within the network were proposed and discussed, while some additional others with different properties were also mentioned.

Finally, some simulation results were presented in order to verify the theoretical results derived throughout the chapter and the feasibility of such distributed FDI scheme.



(a) Outputs



(b) Residuals

Figure 3.29: Deception attack *iii)* in node 1

Chapter 4

Power Systems

As mentioned in Chapter 1, the power system is an example of a very complex system in which its several elements, such as generators and loads, are dynamically interconnected - thus it can be seen as a networked system. Furthermore, the power system is usually divided in smaller subsystems, denominated by *areas*, which may be due to topological, geographical or market properties. In this sense the power system can be seen as a NMAS, where each agent represents an area. It is easy to see that the agents will interact with each other, since they are dynamically coupled due to the networked nature of the system. This formulation will be described in Section 4.1.

This chapter will focus on the power system's state estimation problem, which still has many open issues [33]. In particular, the methods implemented nowadays assume that the power system is at steady-state and do not take into account the actual dynamics of the system. This is partly due to technological constraints, since the measurements available in the SCADA systems have a low time-resolution, thus they can only capture the steady-state behavior of the system.

However, a new measuring technology has been under development and is starting to be implemented in order to enhance the actual state estimator. This technology is called *"Phasor Measurement Unit"* and it consists on a measuring device capable of measuring the phase angle, the frequency and other variables, using the GPS system to provide accurate timestamps. Furthermore, this unit has a high sampling rate, being capable of delivering up to 50 samples per second [34]. Since the power oscillations within an area have frequencies between 0.7 to 2.0Hz and the inter-area oscillations between 0.1 to 0.7Hz, it seems reasonable to use the PMU's measurements to monitor these oscillations.

There are several observers proposed in the literature that already take into account the dynamics of the power system, some based on the "swing dynamics" - as in our approach - while others use some more complex dynamics. However, most of the proposed schemes have either a centralized FDI system [20, 35, 36] or a decentralized state estimation with no FDI component at all [37, 38, 39].

In the following sections we will use the previously discussed UIO theory to implement a decentralized state estimation, where each area estimates its own state by decoupling the observer from the neighbor areas' state. We will then show how to detect faults within each area by designing a FDI scheme using a bank of such UIOs, as described in Section 2.3.

4.1 Modeling of Power Systems

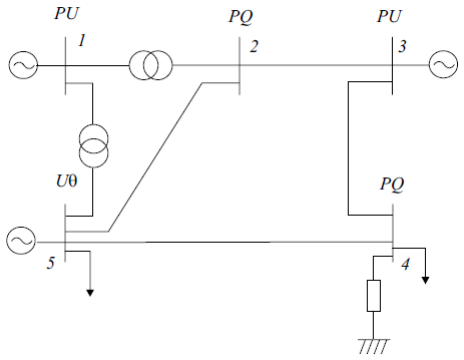


Figure 4.1: Example of a power grid

In this chapter we only consider the active power flow on a power grid as presented in Figure 4.1, which depends on the dynamical properties of the buses supplying and consuming power and on the topology and parameters of the transmission lines. We will now show how to model this dynamical system as a Linear Time-Invariant (LTI) system, under some simplifications, and we refer the reader to [40, 41] for a more detailed modeling of Power Systems.

In a realistic power grid, there may exist different types of power distribution systems with different nominal voltages with transformers interconnecting these systems. Due to this fact, the analysis of such systems using SI units is complex and tedious. To overcome this problem, Power Systems analysis is usually made in the *Per-Unit* system [40], where all different power distribution systems are scaled to the same unit (*p.u.*) and the transformers are then modeled as simple loads on the transmission line.

The transmission lines are described by the admittance matrix Y, where the entry $Y_{ij} = -y_{ij}$, with $y_{ij} = g_{ij} + jb_{ij}$ being the admittance (which is the inverse of the impedance) of the line between bus i and bus j. In the case of $i = j$, assuming there is no connection from bus i to the ground, the diagonal entry Y_{ii} corresponds to the i^{th} bus self-admittance, being given by:

$$Y_{ii} = \sum_{j=1}^{N} y_{ij} \quad , i \neq j \ , \tag{4.1}$$

with N being the number of buses in the power grid.

Note that if there is no direct connection between bus i and j then $y_{ij} = 0$, since it corresponds to an infinite impedance. This shows that the admittance matrix captures both the transmission line

parameters and the network topology - in fact, this matrix can be seen as a *weighted Laplacian matrix* of the power grid's graph, describing how the power flow (interaction) between buses occurs. We now simplify the admittance matrix and thus the overall system by assuming the following:

Assumption 4.1.1 *The transmission lines' resistance is zero and so we have* $y_{ij} = jb_{ij}$, $\forall i \neq j$.

This is a reasonable assumption to make, since usually the resistances of the lines (R) are neglectable compared to its reactances (X) and so, as $g_{ij} = \frac{R_{ij}}{R_{ij}^2 + X_{ij}^2} \approx 0$, we have $y_{ij} \approx jb_{ij}$.

Considering $\mathscr{V}_i = |\mathscr{V}_i| e^{j\delta_i}$ and δ_i to be, respectively, the complex voltage and the phase angle of bus i, the active power flow between bus i and bus j, P_{ij}, is given by:

$$P_{ij} = |\mathscr{V}_i| \left|\mathscr{V}_j\right| b_{ij} \sin(\delta_i - \delta_j). \tag{4.2}$$

The previous expression for the active power flow is indeed in a simplified form due to Assumption 4.1.1 and the reader is referred to [40] for a more exact expression.

Regarding the buses, the behavior of a bus i can be described by the so-called "swing equation":

$$M_i \ddot{\delta}_i + D_i \dot{\delta}_i - P_{mi} = -\sum_{j \in N_i} P_{ij}, \tag{4.3}$$

where N_i denotes the set of neighbors of bus i, M_i and D_i are the inertia and damping coefficients, respectively and P_{mi} is the mechanical input power.

In a power grid, two different types of buses may exist, the Generator bus, which provides energy to the network and the Load bus, which consumes power. Both type of buses can be modeled by (4.3), with the proper parameters. As an example, a Load bus can be modeled as a (synchronous) motor, with some specific parameters $P_{mi} \leq 0$ and $M_i, D_i \geq 0$ or as a constant sink bus with $M_i = D_i = 0$ and $P_{mi} \leq 0$, while (synchronous) a generator bus would be modeled by some $P_{mi} \geq 0$ and $M_i, D_i \geq 0$.

Using (4.2) and (4.3), the behavior of bus i can be written as

$$M_i \ddot{\delta}_i + D_i \dot{\delta}_i - P_{mi} = -\sum_{j \in N_i} |\mathscr{V}_i| \left|\mathscr{V}_j\right| b_{ij} \sin(\delta_i - \delta_j), \tag{4.4}$$

which can be seen as a nonlinear cooperative protocol between distinct nodes of a network.

Small-signal model. A small-signal model of the previous system can be computed around an equilibrium point $\delta^* = \begin{bmatrix} \delta_1^* & \cdots & \delta_N^* \end{bmatrix}^T$ considering a small perturbation $\tilde{\delta}$ caused by a variation in the input power $\tilde{P}_m$ such that $\delta = \delta^* + \tilde{\delta}$ and $P_m = P_m^* + \tilde{P}_m$, resulting on the following dynamics for bus i:

$$M_i \ddot{\tilde{\delta}}_i + D_i \dot{\tilde{\delta}}_i = -\sum_{j \in N_i} k_{ij} \left(\tilde{\delta}_i - \tilde{\delta}_j\right) + \tilde{P}_{mi}, \tag{4.5}$$

$$k_{ij} = |\mathscr{V}_i|\,|\mathscr{V}_j| \cos\left(\delta_i^* - \delta_j^*\right) b_{ij}. \tag{4.6}$$

In the previous equations node i is performing a *weighted consensus algorithm*, where k_{ij} is the entry of the weighted adjacency matrix of the power system's graph. Rewriting (4.5) and (4.6) in a state-space form, we have

$$\begin{cases} \dot{x}_i = \begin{bmatrix} \ddot{\tilde{\delta}}_i \\ \dot{\tilde{\delta}}_i \end{bmatrix} = \begin{bmatrix} -\frac{D_i}{M_i} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{\delta}}_i \\ \tilde{\delta}_i \end{bmatrix} + \begin{bmatrix} \frac{1}{M_i} \\ 0 \end{bmatrix} u_i \\ y_i = \begin{bmatrix} 0 & 1 \end{bmatrix} x_i \end{cases}, \tag{4.7}$$

$$u_i = -\sum_{j \in N_i} k_{ij} \left(\tilde{\delta}_i - \tilde{\delta}_j\right) + \tilde{P}_{mi}. \tag{4.8}$$

Now looking at the entire network, considering the state as $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$, the output to be $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = C\mathbf{x}$ and $\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = -\mathscr{L}_w \mathbf{y} + \tilde{P}_m$, the network's dynamics in a normal behavior case are given by

$$\dot{\mathbf{x}} = (A - B\mathscr{L}_w C)\,\mathbf{x} + B\tilde{P}_m, \tag{4.9}$$

where

$$A = \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & A_N \end{bmatrix}, B = \begin{bmatrix} B_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & B_N \end{bmatrix}, C = \begin{bmatrix} C_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & C_N \end{bmatrix}$$

and $\mathscr{L}_w$ is the weighted Laplacian matrix of the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ which represents the power network, having each bus as a vertex and the transmission lines as weighted edges. The Laplacian is then defined by $\mathscr{L}_w = \Delta_w - \mathscr{A}_w$ with

$$\begin{aligned} [\mathscr{A}_w]_{ij} &= \begin{cases} -k_{ij} & , j \in N_i \\ 0 & otherwise \end{cases} \\ [\Delta_w]_{ii} &= -\sum_{j=1}^{N} [\mathscr{A}_w]_{ij}. \end{aligned}$$

Linearized model. Another method to linearize the system in (4.4) would be to linearize just the right hand term, which is possible if $\delta_{ij} = \delta_i - \delta_j$ is small. In this case we have the following dynamics for each node:

$$M_i \ddot{\delta}_i + D_i \dot{\delta}_i = -\sum_{j \in N_i} k_{ij} (\delta_i - \delta_j) + P_{mi}, \tag{4.10}$$

where we now have

$$k_{ij} = |\mathscr{V}_i| \left|\mathscr{V}_j\right| b_{ij}. \tag{4.11}$$

By defining $x_i = \left[\ddot{\delta}_i \ \dot{\delta}_i\right]^T$ and computing $\mathscr{L}_w$ using the new weights defined in (4.11), we have the following dynamics for the entire network:

$$\dot{\mathbf{x}} = (A - B\mathscr{L}_w C)\,\mathbf{x} + BP_m. \tag{4.12}$$

The similarity between (4.9) and (4.12) is obvious, as the only differences are the values of $\mathscr{L}_w$ and the input.

4.2 Decentralized State Estimation

In this section, we will describe how a decentralized state estimation could be performed by using the UIO theory described in Section 2.2.

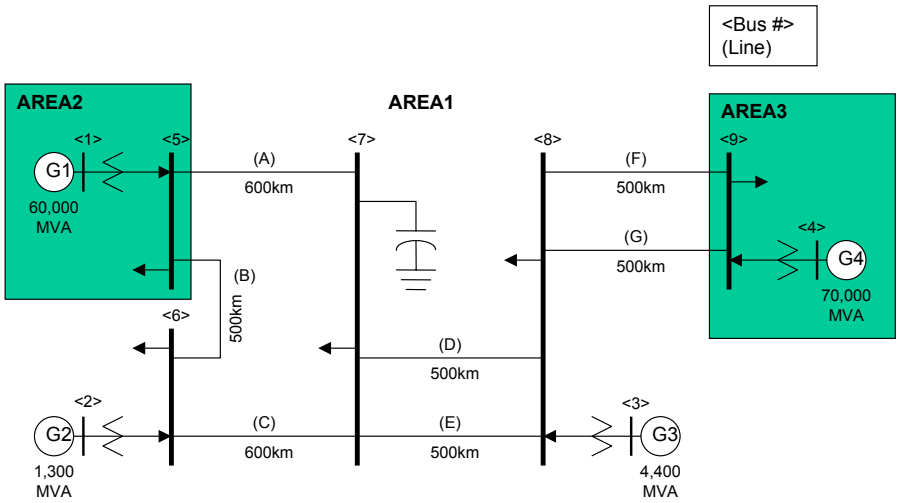


Figure 4.2: Example of a power system with three inter-connected areas

In order to better explain the concept, we will use the power system in Figure 4.2 as an example. Let the transmission network be described by the weighted graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. As it can be seen this power system is divided in three distinct areas, each containing a subset of the buses $\mathscr{V}_i \subset \mathscr{V}$ such that $\mathscr{V}_i \cap \mathscr{V}_j = \emptyset$, $i \neq j$. Another important property is that each subgraph $\mathscr{G}_i$ induced by $\mathscr{V}_i$ is connected. Given the formulation used in Section 4.1, the dynamics of each area are described by

$$\dot{\mathbf{x}}_i = \left(A_i - B_i\bar{\mathscr{L}}_{w_{ii}}C_i\right)\mathbf{x}_i - \sum\left(B_i\bar{\mathscr{L}}_{w_{ij}}C_j\mathbf{x}_j\right) + B_iP_{m_i}, \tag{4.13}$$

where $\mathbf{x}_i \in \mathbb{R}^{n|\mathscr{V}_i|}$ is a vector containing the states of all the buses in area i, $n = 2$ is the order of the

model for each bus. In the previous equation, $\bar{\mathscr{L}}_w$ is a permutation of $\mathscr{L}_w$ such that

$$\bar{\mathscr{L}}_w = \begin{bmatrix} \bar{\mathscr{L}}_{w_{11}} & \cdots & \bar{\mathscr{L}}_{w_{1N}} \\ \vdots & \ddots & \vdots \\ \bar{\mathscr{L}}_{w_{N1}} & \cdots & \bar{\mathscr{L}}_{w_{NN}} \end{bmatrix}, \tag{4.14}$$

with $\bar{\mathscr{L}}_{w_{ii}}$ describing the interactions between buses within the area i and $\bar{\mathscr{L}}_{w_{ij}}$ the interactions between nodes from areas j and i. Note that if areas i and j are not interconnected they are not considered as neighbors, since $\bar{\mathscr{L}}_{w_{ij}} = 0$.

From (4.13) we see that an observer for area i will have to somehow deal with the influence from neighboring areas. This could be done by allowing each area to communicate its own state estimate with its neighbors and so we would replace $\mathbf{x}_j$ by $\hat{\mathbf{x}}_j$, treating it as a known input. Since each area needs to transmit its own state estimate to its neighbors, the burden on the communications would increase but the computational burden would still be smaller when compared to a centralized state estimator. Some approaches following a scheme where each area communicates with its neighbors can be found in [42].

One other option to have a decentralized state estimator would be to design an observer which is decoupled from the other areas. This can be achieved by designing an UIO for area i insensitive to $\mathbf{x}_j, \forall j \in N_i$. An identical approach was proposed in [37] and it is clear that in this setting no communications between different areas is necessary.

As stated in the beginning of this section, the focus will be on the decentralized state estimation based on the UIO theory. Some considerations will be made as to which measurements are required in order for such UIO to exist, given the necessary and sufficient conditions from Theorem 2.2.1.

First, we should state that both observers need to have access to the power consumed or produced at each bus of interest, *i.e.* P_{m_i} must be known and we assume this is indeed the case. Such knowledge may come from either known setpoints, concerning the generator buses, or from local measurements.

Let the set of measurements available from area i be defined as $\mathbf{w}_i = J_i \mathbf{x}_i$, where $J_i \in \mathbb{R}^{p \times n|\mathscr{V}_i|}$ is assumed to be a free parameter on which we will impose some constraints so that there exists a UIO for area i.

Having in mind the previous reasoning and considering the states of neighbor areas as unknown inputs, we can rewrite the dynamics of the system similarly to (2.6):

$$\begin{cases} \dot{\mathbf{x}}_i &= \bar{A}_i \mathbf{x}_i - E_i \mathbf{d}_i + B_i P_{m_i} \\ \mathbf{w}_i &= J_i \mathbf{x}_i, \end{cases}, \tag{4.15}$$

where the unknown input is described by $\mathbf{d}_i = \begin{bmatrix} d_{i_1}^T \cdots d_{i_{N_i}}^T \end{bmatrix}^T$ with $d_{i_j} = C_{i_j} \mathbf{x}_{i_j}$, its distribution matrix $E_i \in \mathbb{R}^{n|\mathscr{V}_i| \times |N_i|}$ is a block column matrix with each column block given by $[E_i]_j = B_i \bar{\mathscr{L}}_{w_{ij}},\ j \in N_i$ and $\bar{A}_i = A_i - B_i \bar{\mathscr{L}}_{w_{ii}} C_i$.

As discussed in Section 2.2, the UIO for the system in (4.15) is given by:

$$\begin{cases} \dot{z}_i &= F_i z_i + T_i B_i \mathbf{u}_i + K_i \mathbf{w}_i \\ \hat{\mathbf{x}}_i &= z_i + H_i \mathbf{w}_i \end{cases}, \tag{4.16}$$

and from Theorem 2.2.1 we know that, for such UIO to exist, the following conditions must be satisfied:

i) $\text{rank}\,(J_i E_i) = \text{rank}\,(E_i)$;

ii) $\left(J_i, \bar{A}_i - H_i J_i \bar{A}_i\right)$ is a detectable pair.

Condition i) is related only to the available measurements and unknown inputs, while Condition ii) depends also on the structure of the dynamical system, as seen in Section 3.2 when proving Theorem 3.2.3. Therefore there are some straightforward conclusions we can draw from the first condition, concerning the set of measurements.

From (4.15) we have that $E_i = \left[B_i \bar{\mathscr{L}}_{w_{ii_1}} \cdots B_i \bar{\mathscr{L}}_{w_{ii_{N_i}}}\right]$. As seen in Section 4.1, B_i han non-zero elements only on the entries corresponding to the frequency offset $\dot{\delta}_i$, which means that the measurement set $\mathbf{w}_i$ must include the frequency measurements. As an example, let $J_i = C_i$, which means that $\mathbf{w}_i$ measures only the phase-angles of all the buses in area i. Recalling the structure of B_i and C_i from (4.7) and (4.9), we have

$$B_i = \begin{bmatrix} B_{i_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{i_{N_i}} \end{bmatrix}, \; C_i = \begin{bmatrix} C_{i_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_{i_{N_i}} \end{bmatrix},$$

where $B_{i_j} = \begin{bmatrix} \frac{1}{M_{i_j}} & 0 \end{bmatrix}^T$ and $C_{i_j} = [0\ 1]$. By checking the rank condition under this setting, we see that $J_i B_i = C_i B_i = 0$ and thus the condition is not fulfilled.

There are some more conclusions we can draw from Condition i) concerning J_i. Previously, we stated that J_i should include frequency measurements, but we did not specified from which buses. By observing Figure 4.2, we see that only a few buses from area i are under the direct influence of neighboring areas. These buses are called as "*boundary buses*" and other approaches to multi-area state estimation have shown how important the measurements from these buses are[42]. In fact, having frequency measurements from all the boundary buses is sufficient to satisfy the rank condition in our approach.

Proposition 4.2.1 *A measurement set defined by J_i satisfies Condition i) of the Theorem 2.2.1 for the system in (4.15) if it contains the frequency measurements from all the boundary buses of area i.*

Proof. This intuitive conclusion can be formally explained through the Laplacian of the power system, more specifically from the interconnection between area i and its neighbors. The way this interconnection affects the buses within area i is represented by the partition of the Laplacian

matrix of the power grid $\bar{\mathscr{L}}_{w_{iN_i}} = \left[\bar{\mathscr{L}}_{w_{ii_1}} \cdots \bar{\mathscr{L}}_{w_{ii_{N_i}}}\right]$, which is also part of the disturbance distribution matrix E_i. Since only the boundary buses are connected to buses from the neighboring areas, the entries of $\bar{\mathscr{L}}_{w_{iN_i}}$ corresponding to the inner buses are zero. Thus, by reordering the buses within area i through a suitable permutation operation P_1, so that the boundary buses are at the top of the state vector $\mathbf{x}_i$, we have:

$$\tilde{\mathscr{L}}_{w_{iN_i}} = \begin{bmatrix} \tilde{\mathscr{L}}_{w_{ii_1}} & \cdots & \tilde{\mathscr{L}}_{w_{ii_{N_i}}} \\ 0 & \cdots & 0 \end{bmatrix}. \tag{4.17}$$

In order to simplify our proof, another permutation P_2 will be applied, so that the state vector becomes $\mathbf{x}_i = \left[\dot{\delta}_i^T \, \delta_i^T\right]^T$, where $\dot{\delta}_i \in \mathbb{R}^{|\mathscr{V}_i|}$ contains all the states of area i corresponding to the frequency. Note that P_2 does not influence the ordering of the buses, thus there is no change on the Laplacian matrix. With this permutation, the input matrix becomes:

$$\tilde{B}_i = \begin{bmatrix} \tilde{B}_{i_b} & 0 \\ 0 & \tilde{B}_{i_{\bar{b}}} \\ 0 & 0 \end{bmatrix}, \tag{4.18}$$

with $\tilde{B}_{i_b} \in \mathbb{R}^{|\mathscr{V}_{i_b}| \times |\mathscr{V}_{i_b}|}$ corresponds to the input matrix of the set of boundary buses $\mathscr{V}_{i_b} \subseteq \mathscr{V}_i$ and $\tilde{B}_{i_{\bar{b}}} \in \mathbb{R}^{|\mathscr{V}_{i_{\bar{b}}}| \times |\mathscr{V}_{i_{\bar{b}}}|}$ corresponds to the inner buses of area i $\mathscr{V}_{i_{\bar{b}}} \subseteq \mathscr{V}_i$, both matrices being diagonal and invertible.

This way, the distribution matrix can be rewritten as

$$\tilde{E}_i = \begin{bmatrix} \tilde{B}_{i_b}\tilde{\mathscr{L}}_{w_{ii_1}} & \cdots & \tilde{B}_{i_b}\tilde{\mathscr{L}}_{w_{ii_{N_i}}} \\ 0 & \cdots & 0 \end{bmatrix}. \tag{4.19}$$

Without loss of generality, consider $\tilde{J}_i \in \mathbb{R}^{|\mathscr{V}_{i_b}| \times n \| \mathscr{V}_i|}$ to include only frequency measurements from the boundary buses, such that $\tilde{J}_i\tilde{B}_i = \left[\tilde{B}_{i_b} \; 0\right]$. Then we have

$$\tilde{J}_i\tilde{E}_i = \tilde{J}_i \begin{bmatrix} \tilde{B}_i\bar{\mathscr{L}}_{w_{ii_1}} & \cdots & \tilde{B}_i\bar{\mathscr{L}}_{w_{ii_{N_i}}} \end{bmatrix} = \begin{bmatrix} \tilde{J}_i\tilde{B}_i\bar{\mathscr{L}}_{w_{ii_1}} & \cdots & \tilde{J}_i\tilde{B}_i\bar{\mathscr{L}}_{w_{ii_{N_i}}} \end{bmatrix} = \begin{bmatrix} \tilde{B}_{i_b}\tilde{\mathscr{L}}_{w_{ii_1}} & \cdots & \tilde{B}_{i_b}\tilde{\mathscr{L}}_{w_{ii_{N_i}}} \end{bmatrix}.$$

Analyzing the rank of $\tilde{E}_i$ we observe that $\text{rank}\left(\tilde{E}_i\right) = \text{rank}\left(\begin{bmatrix} \tilde{B}_{i_b}\tilde{\mathscr{L}}_{w_{ii_1}} & \cdots & \tilde{B}_{i_b}\tilde{\mathscr{L}}_{w_{ii_{N_i}}} \end{bmatrix}\right) = \text{rank}\left(\tilde{J}_i\tilde{E}_i\right)$, thus the matching Condition i) is satisfied. ■

However, note that such set of measurements may not be minimum. In fact, taking the power grid in Figure 4.2 as an example, we see that the boundary buses are the buses 6, 7 and 8, but only two measurements are necessary to satisfy Condition i), a measurement from bus 8 and another one from bus 6 or 7. The reason has to do, once again, with the Laplacian matrix and it can be explained from the topological properties of the power grid. Area 3 is interconnected to area 1 through node 8 only, thus this is a critical bus. As for area 2, it is connected to area 1 by nodes 6 and 7. However, only a single bus from area 2 is connected to area 1, bus 5. This means that node 6 and 7 are influenced by area 2 in the same way, thus measurements from those nodes give the same information regarding the influence of area 2, meaning that we only need one measurement.

Formally, we have that the minimum number of measurements is given by

$$\text{rank}\left(\begin{bmatrix} \bar{\mathscr{L}}_{w_{ii_1}} & \cdots & \bar{\mathscr{L}}_{w_{ii_{N_i}}} \end{bmatrix}\right) \leq |\mathscr{V}_{i_b}|, \tag{4.20}$$

and, as discussed previously, in this case the strict inequality holds.

The second condition depends on the structural properties of the system in (4.15), as it is equivalent to say that the matrix

$$\begin{bmatrix} sI - \bar{A}_i & E_i \\ J_i & 0 \end{bmatrix}$$

is of full column rank for all s such that $\Re(s) \geq 0$. Given the complexity of this kind of interconnected system, such condition is harder to evaluate analytically on this particular system than on the system studied in Chapter 3.

There are, however, some comments to be made on this matter. First notice that one necessary condition is that the system $(J_i, \bar{A}_i)$ is detectable, since otherwise the matrix

$$\begin{bmatrix} sI - \bar{A}_i \\ J_i \end{bmatrix}$$

will not have full column rank for some s such that $\Re(s) \geq 0$, which is a standard test for detectability.

Since we assumed the measurement matrix J_i to be a design parameter, there is always a matrix J_i such that an UIO for the system in (4.15) exists. The obvious option would be to have all the measurements from all the buses, which means that J_i would have rank $n|\mathscr{V}_i|$ and thus the Condition ii) is satisfied.

An example for such UIO observer for the system in Figure 4.2 will be presented in Section 4.4 and it will be shown that the observer converges to the correct state even under the presence of a disturbance in the neighboring areas.

4.3 Decentralized Fault Detection and Isolation

Given the decentralized state estimator based on the UIO theory, one possibility is to perform fault detection within the area itself, as the estimator is insensitive to faults from neighboring areas.

In order to perform this, a FDI system similar to the one discussed in Section 2.3 can be designed, we the difference that all the observers in the bank will also be decoupled from the effect of neighboring areas. This FDI system is based on the GOS scheme, where each observer is insensitive to only one fault.

Let j be the index of the bus within area i that is subject to an unknown disturbance, which we consider to be a fault. The dynamics of area i under the effect of such fault are described by:

$$\begin{cases} \dot{\mathbf{x}}_i &= \bar{A}_i \mathbf{x}_i - E_i \mathbf{d}_i + B_i^{\bar{j}} P_{m_i}^{\bar{j}} + b_i^j f_j \\ \mathbf{w}_i &= J_i \mathbf{x}_i, \end{cases}, \tag{4.21}$$

where $B_i^{\bar{j}}$ is obtained from matrix B_i by removing its j^{th} column b_i^j. Note that such column has only one non-zero entry, which corresponds to the frequency state of bus j.

Since we desire an UIO decoupled from both E_i and b_i^j, we will design an UIO insensitive to $E_i^j = \left[E_i \; b_i^j\right]$, which is given by:

$$\begin{cases} \dot{z}_i^j = F_i^j z_i^j + T_i^j B_i^{\bar{j}} P_{m_i}^{\bar{j}} + K_i^j \mathbf{w}_i^j \\ \hat{\mathbf{x}}_i^j = z_i^j + H_i^j \mathbf{w}_i^j \end{cases}. \tag{4.22}$$

The conditions for such observer to exist are given in Theorem 2.2.1 and were already introduced in the previous section. From the Condition i) and the reasoning in the previous section, we see that the set of measurements must include not only the frequency measurements from the boundary buses, but also the frequency measurement from the j^{th} bus. Such set of measurements is characterized by $J_i^j = [J_i^T J_{i_j}^T]^T$, where $J_{i_j} \in \mathbb{R}^{1\times n|\gamma_i|}$ with the value 1 on the column corresponding to the frequency state of the j^{th} bus and zero elsewhere. Therefore we have that $\text{rank}\left(J_{i_j} b_i^j\right) = \text{rank}\left(b_i^j\right) = 1$ and, as we already proved in the previous section that J_i satisfies this condition for E_i, we conclude that the rank condition is indeed satisfied.

One should note, however, that the UIO insensitive to the neighbor areas is already insensitive to faults on the boundary buses that compose the measurement set J_i, thus it is not possible to detect faults on these buses using the proposed method.

As for the Condition ii), the same reasoning as in the previous section applies.

In the next section we will present results showing that such FDI system is possible to implement and that it can successfully detect and isolate faults within the inner buses, remaining insensitive to disturbances from the neighboring areas.

4.4 Simulation Examples

The power grid represented in Figure 4.2, modeled by (4.12), will be used to present examples of the methods discussed in the previous sections. More specifically, the focus will be on area 1 and the proposed decentralized state estimation and fault detection methods will be applied to this area, under different conditions.

4.4.1 Decentralized State Estimation

An example of a decentralized state estimation of area 1 will be presented in this section, considering both a normal and a perturbed behavior of the global system.

The normal behavior will be analyzed with different sets of measurements, one including only the frequency measurements from the boundary buses and the other including also the phase angles from those buses.

As for the perturbed system, the decentralized state estimator with the previous set of measurements will be analyzed when a disturbance occurs in a neighbor area.

Normal behavior. Here we consider the unperturbed system at steady-state, having the initial conditions of the state observer set to a value close to the steady-state values.

The phase angles of the entire system are presented in Figure 4.3 and it can be seen that the system is indeed in steady-state.

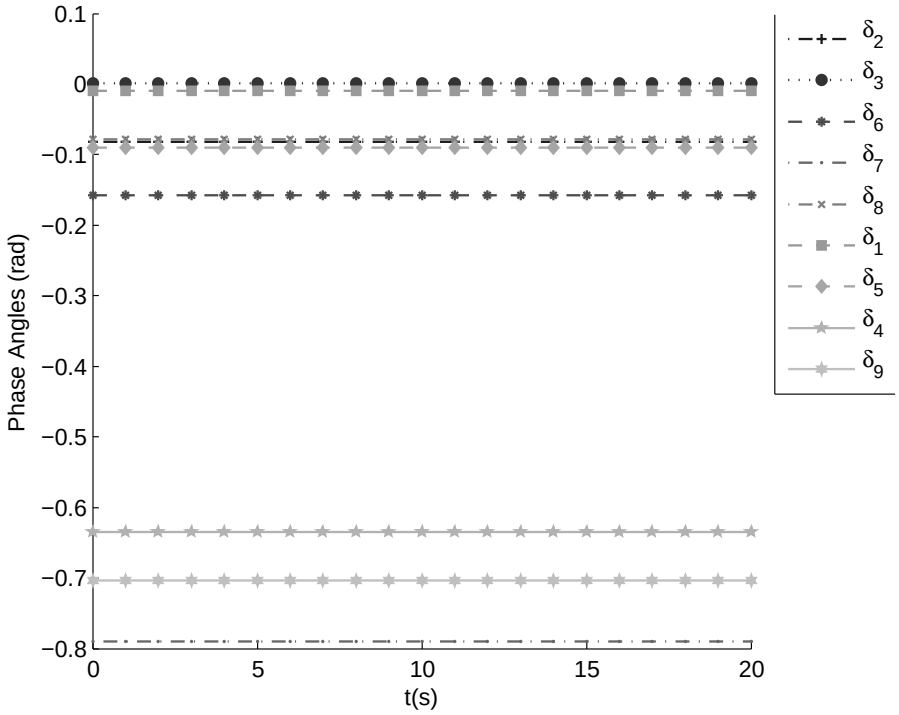


Figure 4.3: Phase-angles of the power system

In Figure 4.4(a) we have the estimates from the observer which uses the only the frequency measurements from nodes 6, 7 and 8. As it can be seen, the estimates seem to be close to the real state, but in Figure 4.4(b) we see that the norm of the estimation error $e_1 = x_1 - \hat{x}_1$ does not converge to zero. This implies that the observer is not asymptotically stable, which is indeed the fact, since the state matrix of such observer has some zero eigenvalues. Thus, having Condition i) from Theorem 2.2.1 satisfied by measuring the frequency on the boundary buses does not guarantee the asymptotic stability of the observer.

The UIO with both the frequency and phase-angle measurements from the boundary buses has an asymptotically stable state matrix, with only negative eigenvalues. The estimates of such observer can be seen in Figure 4.5(a) and the norm of the corresponding estimation error is presented in Figure 4.5(b). We see that the estimation error converges to zero, which validates the asymptotic stability of the observer.

With these two examples, we conclude that having only the frequency measurements from the boundary buses is not enough to ensure the asymptotic stability of the UIO. In fact, the structure of the dynamical system, which changes according to the set of available measurements, plays an important role in this matter. Furthermore, from the second example presented in Figure 4.5

we verify that all the information from the system may not always be needed. However, the knowledge of which information is needed or not is not yet available and an analytical analysis of Condition ii) from Theorem 2.2.1 must be performed in order to obtain this knowledge.

Disturbance in area 2. Now the response of the decentralized state estimator in area 1 will be analyzed under the effect of a sinusoidal disturbance in area 2 at $t = 0.5$, which is unknown to the observer. The phase-angles of the entire system are presented in Figure 4.6 and in Figure 4.7 we show the state estimates and the norm of the estimation error of the observer with the set of measurements composed by the phase-angles and frequency measurements from the boundary buses.

As it can be seen, the state estimation error converges to zero, even under the disturbance at bus 1 from area 2, validating that the UIO for area 1 is indeed decoupled from its neighboring areas.

4.4.2 Decentralized Fault Detection and Isolation

Some examples of the decentralized FDI method discussed in Section 4.3 will be presented and analyzed in this section.

We begin by analyzing the response of one UIO of the observer bank in the FDI scheme to a the fault it should be insensitive to, using two different sets of measurements. Both sets contain the phase-angles and frequency measurements from the boundary buses, which are necessary to decouple the FDI system from the effect of neighboring areas. One set also has the phase-angle and frequency measurements from the faulty bus, while the other has only the frequency measurement.

Then we present the response of the FDI system to three faults in area 1, one in an inner bus and the others in different boundary buses. As discussed in Section 4.3, it will be shown that the FDI system can detect and isolate faults in the inner buses, while faults in the boundary buses are usually not detectable or isolable.

UIO insensitive to a fault in bus 2. In Figure 4.8 we have the phase-angles of the area 1 when there is a sinusoidal fault signal in bus 2 at $t = 0.5s$. The norm of the estimation error of the UIO with only the frequency measurement from bus 2 is presented in Figure 4.9(a), while in Figure 4.9(b) we have the norm of the state estimation error when both the phase-angle and the frequency measurements of bus 2 are available.

As it is observable in Figure 4.9(a), the UIO with only the frequency measurement from bus 2 available does not converge to zero, thus the UIO is not insensitive to the fault as it should be.

However, when both the phase-angle and the frequency are available, the state estimation error converges to zero, as it can be seen in Figure 4.9(b), validating that the UIO is insensitive to the fault.

FDI in area 1. Here we will present the response of the FDI system proposed in Section 4.3 to several faults within area 1, in both inner and boundary buses. In this scenario, the set of available

measurements for each UIO contains all phase-angle and frequency measurements from all the buses in area 1.

The residuals during a fault in the inner bus 3 are shown in Figure 4.10, where it is visible that only the residual corresponding to a fault in bus 3 is zero, hence such fault can be successfully detected and isolated.

In Figure 4.11 we have the residuals when a fault in the boundary bus 8 is present, being clear that all the residuals are small, which indicates that this fault cannot be detected. In fact this is due to the decoupling from the neighboring areas, since the distribution matrix of a fault in bus 8, b_1^8, and the distribution matrix of the interactions of neighboring areas, E_1, are linearly dependent. We can see this fact looking at both matrices:

$$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 44.5287 & 0 & 3.0581 \\ 0 & 0 & 0 \\ 30.8072 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 34.8291 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; b_1^8 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2.4207 \\ 0 \end{bmatrix}.$$

This way all the UIOs in the bank composing the FDI system are insensitive to faults in bus 8, as they all are decoupled from E_1.

The residuals with a fault in bus 6 are presented in Figure 4.12, where we see that the residuals corresponding to faults in buses 6 and 7 are both zero, while the others are not. Since two residuals are zero, the fault cannot be isolated, but it can be detected and it is possible to say that it is either on bus 6 or 7.

Once again, this behavior is due to the decoupling from neighboring areas. The columns of the distribution matrices E_1^6 and E_1^7 are linearly dependent and, furthermore, those matrices span the same subspace:

$$E_1^6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 44.5287 & 0 & 3.0581 \\ 0 & 0 & 0 \\ 30.8072 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 34.8291 & 0 \\ 0 & 0 & 0 \end{bmatrix}, E_1^7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 44.5287 & 0 & 0 \\ 0 & 0 & 0 \\ 30.8072 & 0 & 2.3935 \\ 0 & 0 & 0 \\ 0 & 34.8291 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Thus, both UIO are decoupled from the same fault signals, which makes it impossible to distinguish between both faults.

Although both buses are boundary buses, the residuals corresponding to the rest of the buses do not remain zero when a fault is present in either of them, unlike what happens with bus 8, as seen previously. This can be explained by looking at the distribution matrices of the faults and of the interactions with neighbor areas, b_1^6, b_1^7 and E_1, respectively. The vector b_1^6 is the third column of E_1^6 and b_1^7 is the third column of E_1^7, while E_1 corresponds to the two first columns of both matrices. It is clear that b_1^6 and E_1 are linearly independent, as well as b_1^7 and E_1, which explains why the residuals sensitive to other faults are not zero - they are not decoupled from fault signals in buses 6 and 7.

This difference of sensitivity to faults in boundary buses can be explained by observing closely the topological properties of the network, particularly the connections between area 1 and the rest of the power grid. We see that area 3 is connected to area 1 only by bus 8, thus making an observer insensitive to area 3 will also decouple it from a fault signal in bus 8. As for buses 6 and 7, they are both connected to a single bus from area 2, which means that the interaction with area 2 affects both buses in a certain way, which is known to the observer decoupled from area 2. If a fault occurs in either bus 6 or 7, the trajectory of the states of both buses will not correspond to the expected interaction, thus the fault can be detected. However, we cannot say where the fault is, since we also do not know how area 2 is affecting those buses.

4.5 Summary

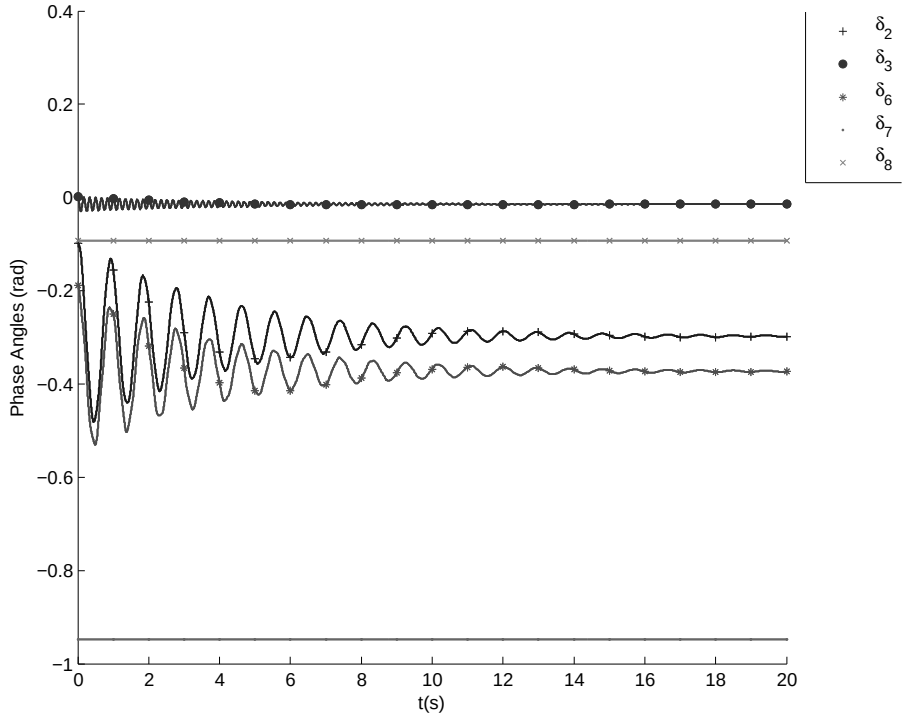
A decentralized state estimator and a decentralized FDI system applied to a power system, both based on the UIO theory, were proposed and analyzed in this chapter. It was first shown that such system can be seen as NMAS, similarly to the system analyzed in Chapter 3, although with some more complex dynamics, which were also modeled. Besides the dynamics, both systems also differ on the nature of the interactions between agents, since the one in Chapter 3 has its interactions based on communications while the interactions within the power system are due to physical coupling. Thus, a different approach from that in Chapter 3 was followed here.

Since the power systems are usually large-scale systems which are divided in smaller subsystems, the so-called "*areas*", we started by proposing a decentralized state estimation of each area, based on an UIO which decouples the area of interest from the rest of the power grid. Some comments were made regarding the general necessary and sufficient condition for such UIO to exist, given the specific problem at hands. It was shown that one of the conditions is met if frequency measurements from boundary buses are available.

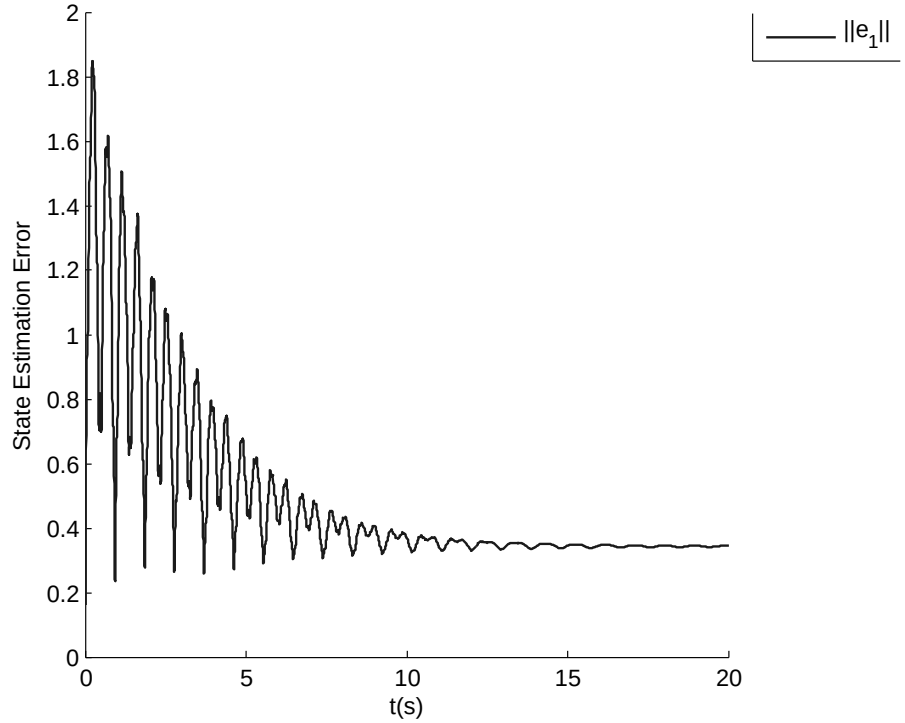
A decentralized FDI scheme based on a bank of such UIO was then proposed, which should be able to detect and isolate faults within the area of interest. As before, it was proved that the rank condition for the existence of the bank of UIO is met if the frequency measurements from the boundary buses and the faulty bus are available.

Finally some simulation result on the decentralized state estimations were presented, based on a 9 bus power grid with 3 areas. The results verified that meeting the rank condition alone does not ensure the asymptotic stability of the UIO and provided some insight on what kind of measurements are needed - both the phase-angle and frequency measurements of the boundary buses.

The same was performed for the decentralized FDI system, were we verify that such system is able to detect and isolate faults in the inner buses, but that this is not always possible for the boundary buses. Some insight on what reasons lead to this fact were also provided.

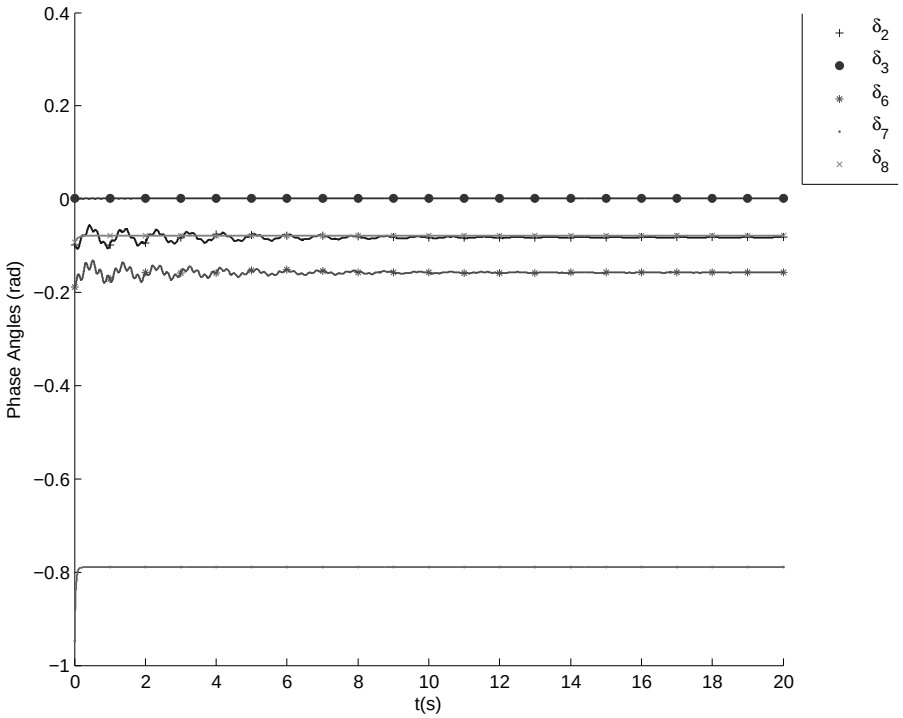


(a) State estimates

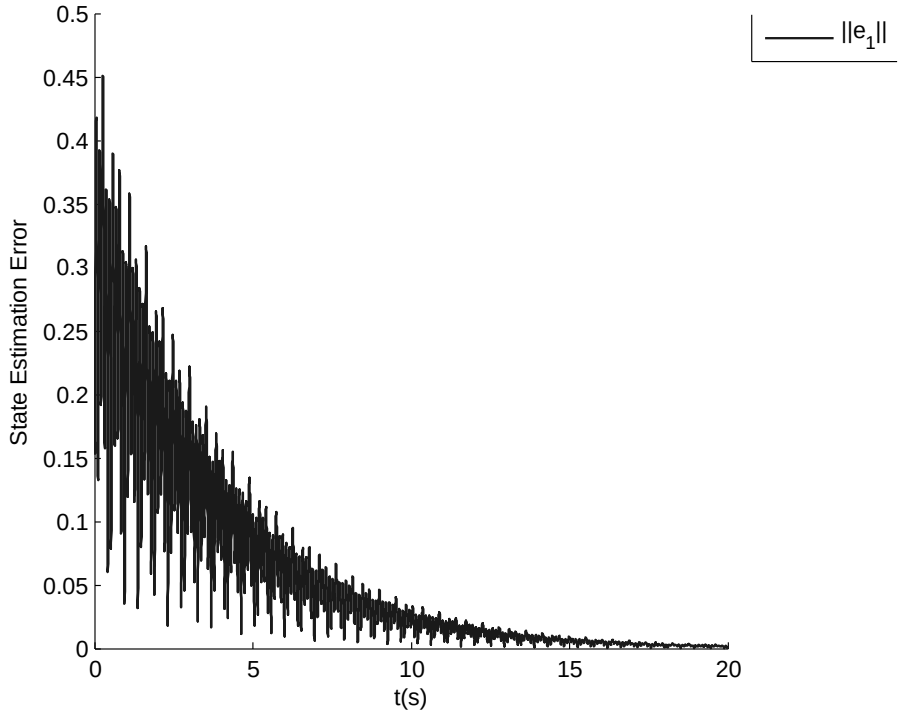


(b) State estimation error

Figure 4.4: Decentralized state estimator for area 1 with frequency measurements at the boundary buses



(a) State estimates



(b) State estimation error

Figure 4.5: Decentralized state estimation of area 1 with frequency and phase-angle measurements at the boundary buses

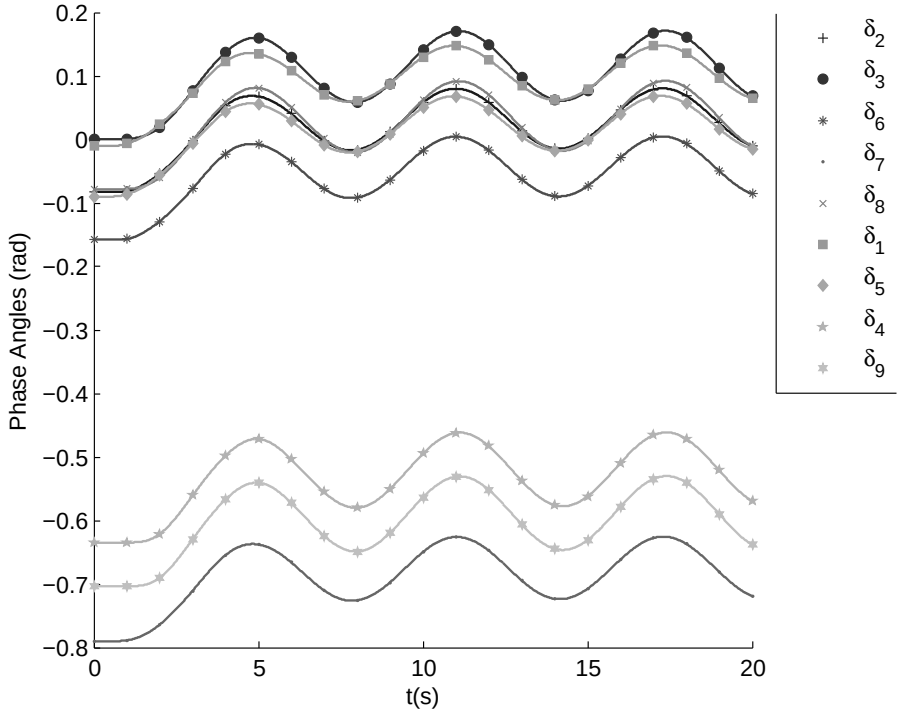
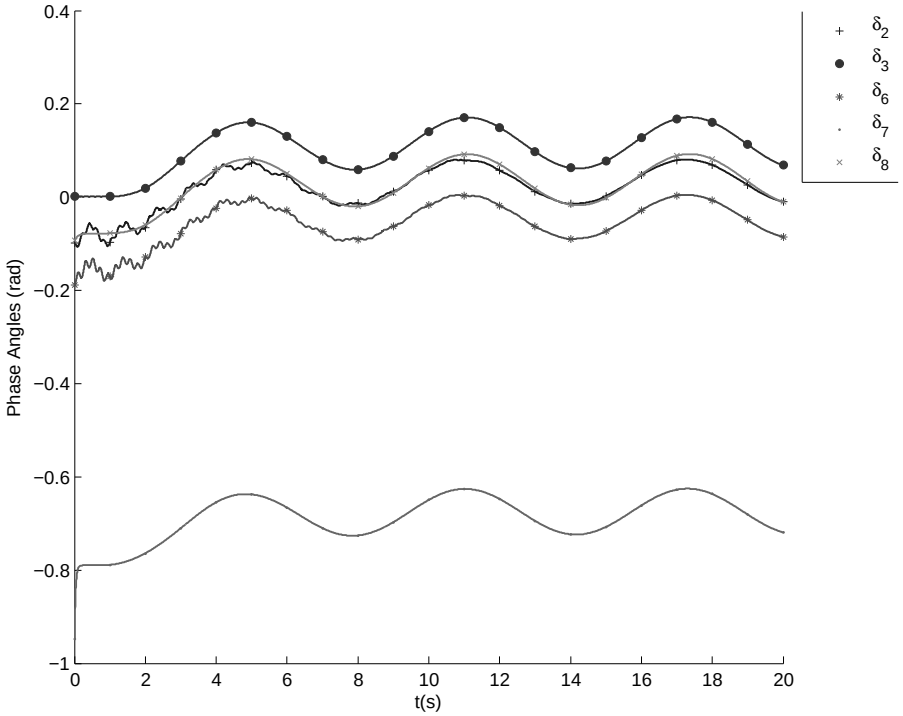
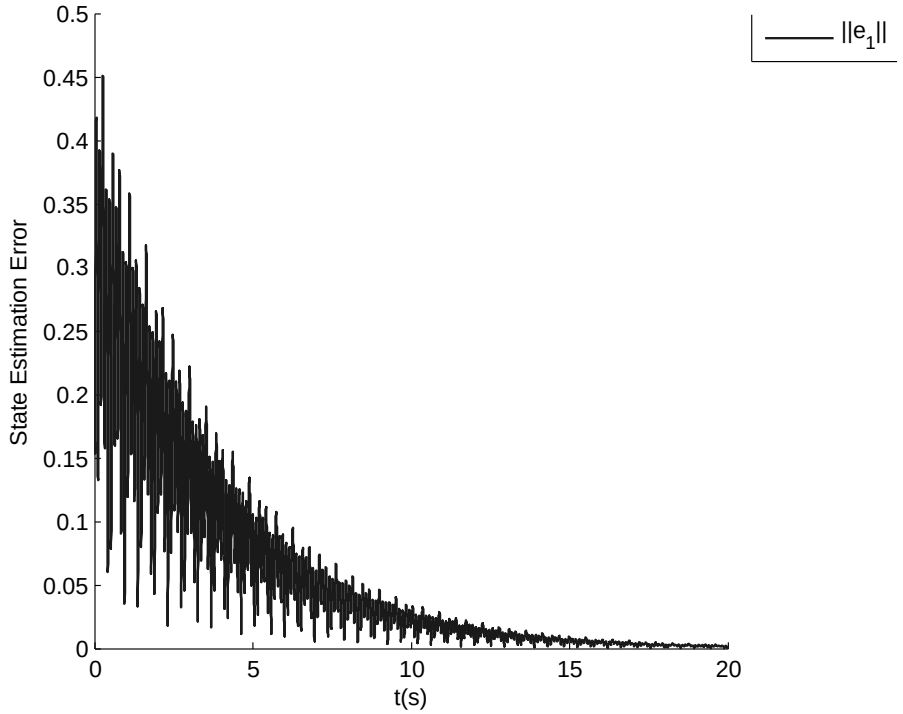


Figure 4.6: Phase-angles of the power system disturbed at bus 1



(a) State estimates



(b) State estimation error

Figure 4.7: Decentralized state estimation of area 1 under the effect of a disturbance with frequency and phase-angle measurements at the boundary buses

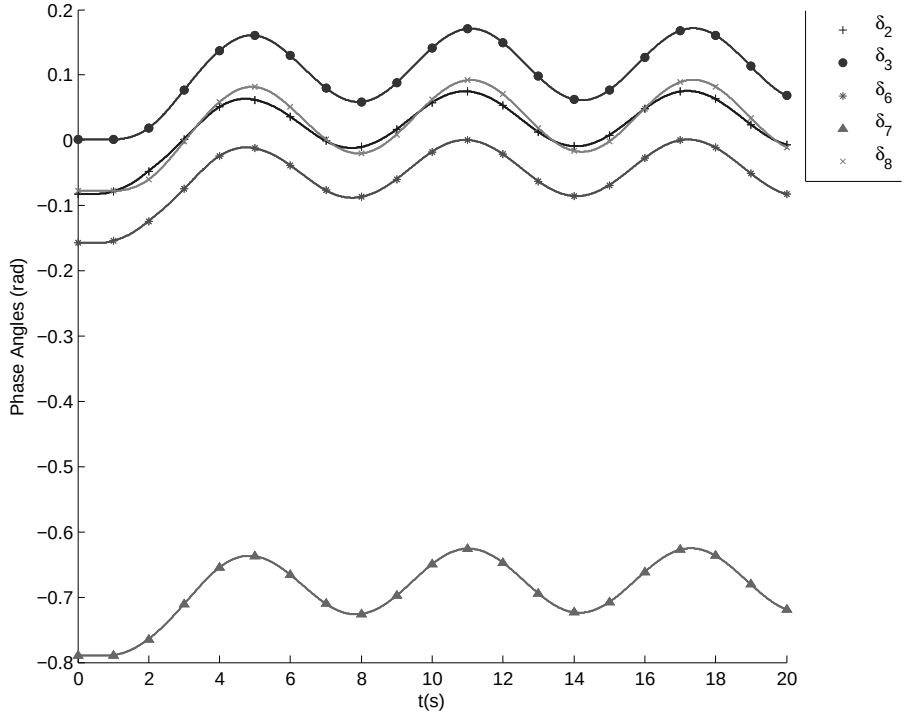
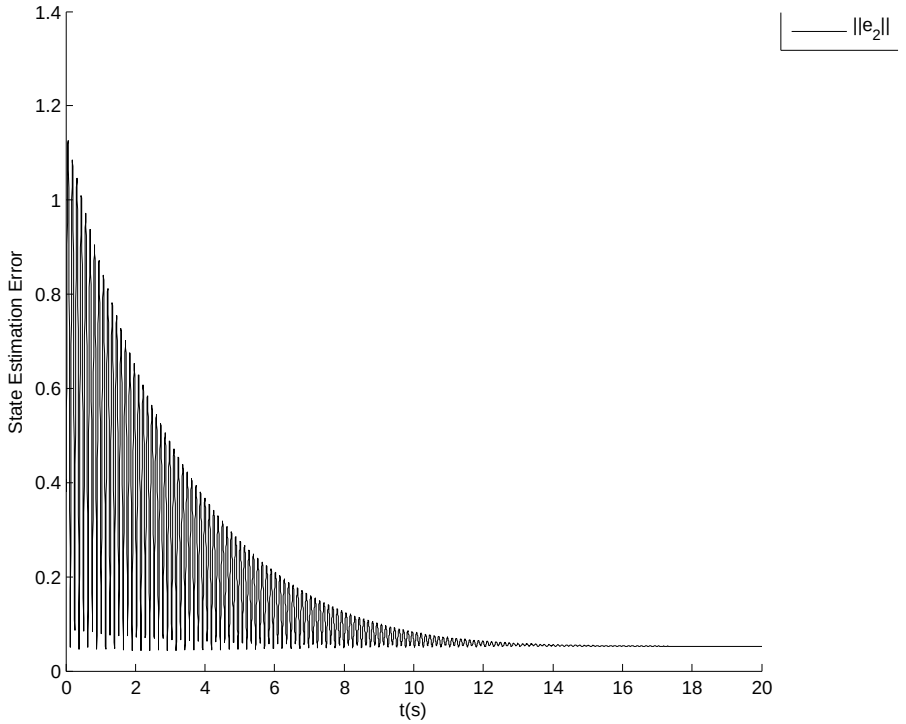
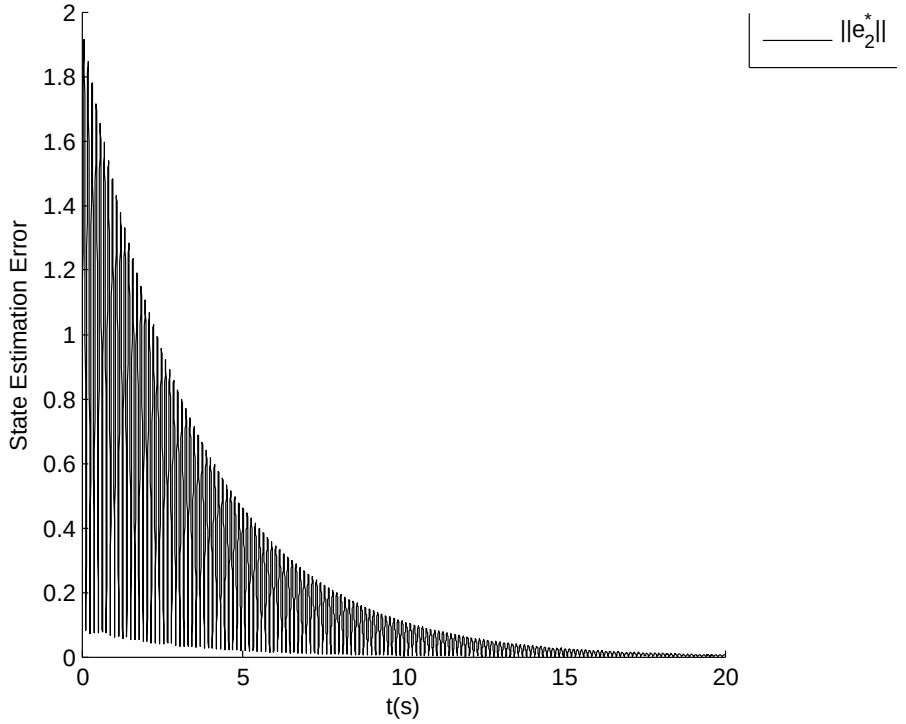


Figure 4.8: Phase-angles of the power system with a fault in bus 2



(a) State estimation error with frequency measurement from bus 2



(b) State estimation error with phase-angle and frequency measurements from bus 2

Figure 4.9: State estimation error of an UIO insensitive to a fault in bus 2 with different measurement sets

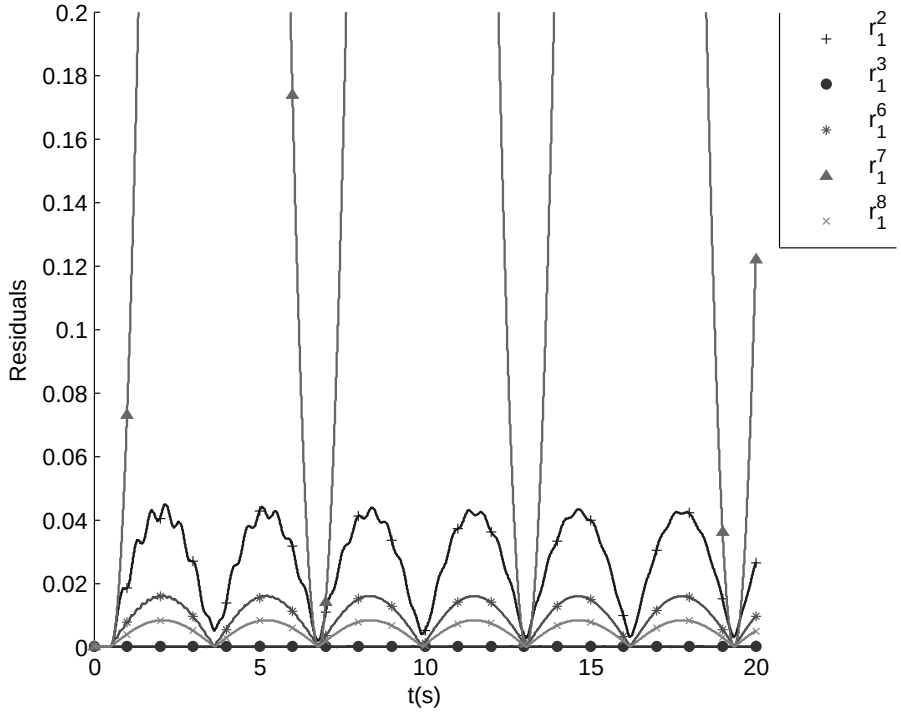


Figure 4.10: Residuals with a fault in bus 3

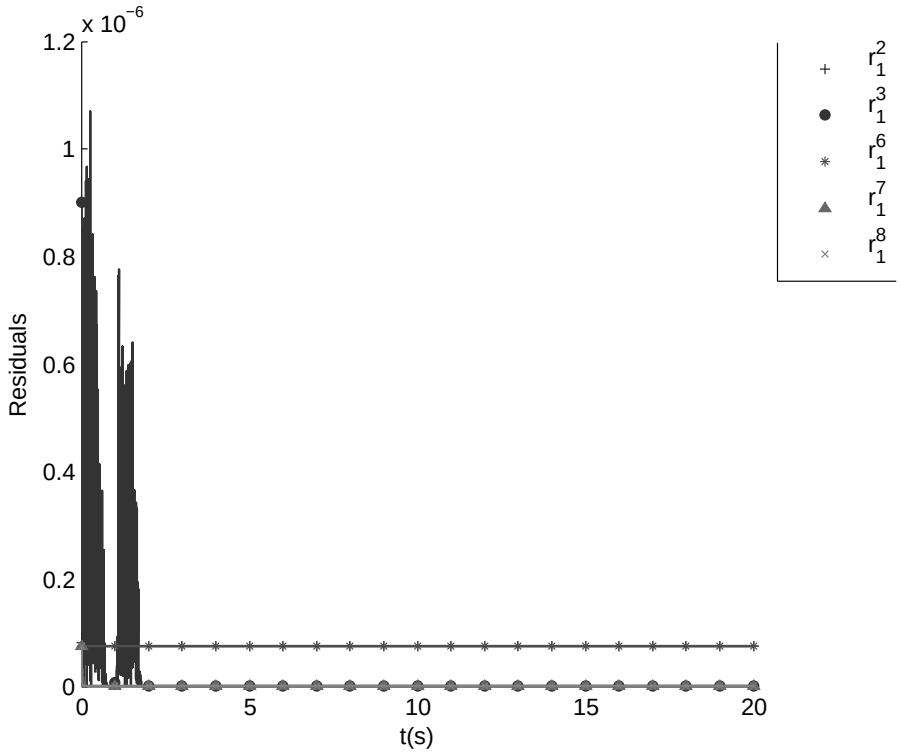


Figure 4.11: Residuals with a fault in bus 8

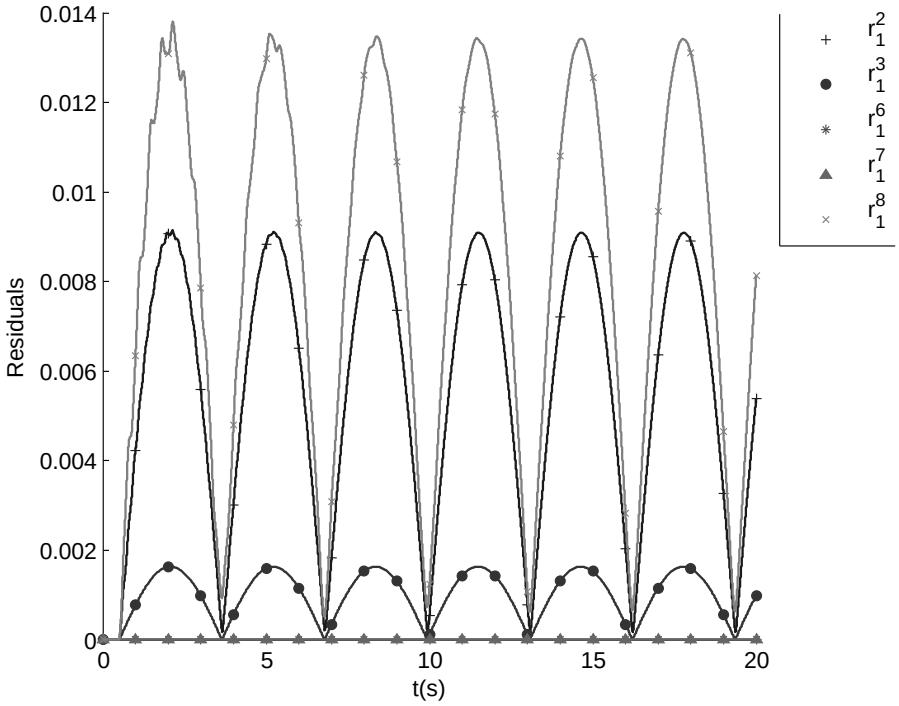


Figure 4.12: Residuals with a fault in bus 6

Chapter 5

Conclusions and Future Work

This is the final chapter of the thesis, where we present a brief summary of all the previous chapters, followed by some conclusions regarding the content of the respective chapter.

As discussed in Chapter 1, the framework needed to deal with the many challenges arising in the distributed systems is still immature and there are many tools yet to be developed. Furthermore, cooperation in this type of systems is crucial for their global performance, since these systems may involve several different agents with different local information and local objectives. Thus the importance of developing tools to tackle faults and security breaches within distributed systems, as they are a sort of non-cooperative behavior - which is the focus of the thesis.

In Chapter 3 we studied a well-known distributed system, the consensus problem, under the effect of faults and attacks on the out-going communications of a single agent. A method to detect and isolate the faulty node in a distributed fashion was proposed, based on the UIO theory. According to this method, each node should monitor its own neighbors based on the information they are sending and its own local measurement. This is a quite redundant and expensive method, since all nodes in the network have to execute a FDI scheme and the same node is monitored by all its neighbors. In order to reduce the number of monitor nodes, two different formulations of the set cover problem were made, one achieving the minimum number of monitor nodes and the other reducing the number of observers but ensuring that each node is observed by at least another one.

Sufficient conditions for the feasibility of such FDI system were also given, based on the set of available measurements and the topological properties of the network encoded in the Laplacian matrix.

Some special cases where the FDI system may give a false alarm were also discussed and some considerations on how this could be prevented were made.

The scenario where the communications of a node are being tampered was described and formulated. It was shown that the “healthy” part of the network can detect the misbehavior and

isolate the misbehaving node, but it cannot distinguish between a fault and an attack scenario. However, under the assumption that the attacked node has a reliable local measurement of its own state, it can realize the attack on its own communications by adding one more UIO to the bank monitoring faults in its neighbors.

As before, some cases where such UIO may give a false alarm were discussed and a way of preventing or minimizing this was mentioned.

Although the consensus problem may seem to be a simple problem, it turns out that it is a very interesting problem to start with and gain some useful intuition on distributed systems and how they are affected by faults and communication attacks. For instance, the necessity of cooperation in order to achieve a common goal and the fact that the "healthy" network cannot distinguish a fault from an attack seem to be intuitive aspects and can easily be proved in the consensus problem.

Regarding the proposed FDI scheme, it is quite redundant and computationally heavy, even with the reduction of monitor nodes, since each UIO estimates the state of the entire network. Although it would be desirable to have a less heavy FDI scheme, this scheme based on a bank of UIOs is quite understandable and its implementation is straightforward. Moreover, it turns out that the necessary and sufficient conditions for the existence of the UIOs can be characterized based on the properties of the network, providing a useful characterization of which information is needed from the network and how the topology affects the detectability of misbehaviors.

As for Chapter 4, there an example of a more complex NMAS was studied - the power system. First the active power flow in a power grid was modeled and it was shown how this model resembles the one in the consensus problem, but here each agent has second order dynamics and the interactions come from the existing physical coupling. However, due to the different nature of the interactions and the system itself, in this chapter the approach from Chapter 3 was not followed. Instead, the decentralized state estimation and FDI within each area of the power grid was considered, based once again in the UIO theory.

The decentralized state estimation of each area is achieved by designing an UIO decoupled from the interactions of the neighbor areas. The decentralized FDI uses a bank of UIO such that each UIO is insensitive to only one fault within a bus of the area and all the UIOs in the bank are also decoupled from the neighbor areas, as in the decentralized state estimation.

In both the decentralized state estimation and FDI, the only condition for the existence of the UIO that was successfully analyzed in a theoretical way was the rank condition, which directly constraints the set of available measurements. The detectability condition is harder to analyzed in this system, since it depends on its dynamics, which are more complex than in the consensus problem. However, the simulation results gave some insight on how this condition affects the requirements on the set of measurements - which reveals the need for phase-angle measurements - but a formal proof is still needed.

As discussed in the previous part of this chapter, the work done in this thesis tackles the problem of security in multi-agent systems, but it is not intended to be complete within itself. There is a

great amount of research to be done in this field, not only in detecting and isolating faults, but also in designing control and estimation algorithms that are inherently secure, private and robust. Furthermore, different types of faults and attacks that are not considered here may occur as well, such as faults in edges, more general attacks on the communications, among several others.

Although there are many aspects of the proposed methods to improve and many other issues to address, this thesis provides some valuable comments on the effect of faults and communication attacks in distributed systems, as well as a framework to address these issues in a distributed fashion.

Appendix A

Proof of some Lemmas

Lemma A.0.1 (Lemma 3.2.4) *If an undirected graph $\mathcal{G}$ is connected, then any partition of its Laplacian matrix $\mathcal{L}$, induced by a strict subset of nodes $\bar{F} \subset \mathcal{V}$, is invertible.*

Proof of Lemma 3.2.4. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph and $F \in \mathcal{V}(\mathcal{G})$ be a subset of the vertex set. Consider $\mathcal{G}' = \mathcal{G}/F$ to be the induced subgraph of $\mathcal{G}$ obtained by removing the set of vertices in F. After a suitable permutation of nodes, the Laplacian matrix $\mathcal{L}(\mathcal{G}) \in \mathbb{R}^{N \times N}$ can be written as

$$\mathcal{L}(\mathcal{G}) = \begin{bmatrix} \mathcal{L}_F & l \\ l^T & \mathcal{L}_{\bar{F}} \end{bmatrix}$$

with $\mathcal{L}_F \in \mathbb{R}^{|F| \times |F|}$, $\mathcal{L}_{\bar{F}} \in \mathbb{R}^{|\bar{F}| \times |\bar{F}|}$ and we have that

$$\mathcal{L}_{\bar{F}} = \mathcal{L}\left(\mathcal{G}'\right) + \Delta_{\bar{F}}$$

where $\Delta_{\bar{F}}$ is a diagonal matrix with nonnegative elements, thus it is a positive semi-definite matrix, and $\mathcal{L}(\mathcal{G}')$ is the Laplacian of the induced subgraph $\mathcal{G}'$. Furthermore, since $\mathcal{G}$ is connected, $\Delta_{\bar{F}} \neq 0$ and $[\Delta_{\bar{F}}]_{ii} > 0$ if and only if $i \in N_F$ with $N_F = \bigcup_{j \in F} N_j$.

The first property is easy to prove by a simple contradiction argument: suppose $\mathcal{G}$ is connected, $F \neq \emptyset$ and $\Delta_{\bar{F}} = 0$. Then this means that $\mathcal{L}_{\bar{F}} = \mathcal{L}(\mathcal{G}')$, which implies that $(i, j) \notin \mathcal{E}(\mathcal{G})$, $\forall i \in F \; \forall j \in \bar{F}$, *i.e.* the graph $\mathcal{G}$ is not connected.

As for the second property, it can be easily verified by observing that the diagonal entries of $\mathcal{L}_{\bar{F}}$ and $\mathcal{L}(\mathcal{G}')$ are the same for all the nodes that were not neighbors of the removed set F, meaning that those nodes did not suffer any change since they kept all their neighbors. Furthermore, the entries corresponding to nodes whose neighborhood was affected are positive since those nodes have a smaller degree in the induced graph $\mathcal{G}$ that in the original graph.

We now continue the proof by considering the two possible scenarios, $\mathcal{G}'$ is connected and $\mathcal{G}'$ is disconnected.

For a connected $\mathcal{G}'$, it is well-known that $\mathcal{L}(\mathcal{G}')$ is a positive semi-definite matrix that contains a single zero eigenvalue with the associated eigenvector $\mathbf{1}$ having all its entries set to one. Since both $\mathcal{L}(\mathcal{G}')$ and $\Delta_{\bar{F}}$ are positive-semidefinite matrices, $\mathcal{L}_{\bar{F}}$ is singular if and only if $\Delta_{\bar{F}}$ is singular

and $\mathscr{L}(\mathscr{G}')$ and $\Delta_{\bar{F}}$ share at least one eigenvector associated with the zero eigenvalue. However it can be easily seen that $c^2\mathbf{1}^T\Delta_{\bar{F}}\mathbf{1} \neq 0$, due to the properties of $\Delta_{\bar{F}}$ mentioned previously, resulting that $\mathscr{L}_{\bar{F}}$ is invertible in this case.

In the other situation, where $\mathscr{G}'$ is disconnected with n connected components, we can again perform a permutation on the nodes of $\mathscr{G}'$ and rewrite its Laplacian in a block diagonal form:

$$\mathscr{L}\left(\mathscr{G}'\right) = \begin{bmatrix} \mathscr{L}_{C_1} & 0 & \cdots & 0 \\ 0 & \mathscr{L}_{C_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathscr{L}_{C_n} \end{bmatrix}$$

with $\mathscr{L}_{C_k}$ being the Laplacian of the k^{th} connected component. It is easy to see that the block diagonal structure comes from the fact that the nodes from the subset C_i are not connected with the nodes belonging to C_j, $i \neq j$, thus C_i and C_j do not interact.

Applying the permutation to $\mathscr{L}_{\bar{F}}$ as well, we can rewrite it as:

$$\mathscr{L}_{\bar{F}} = \begin{bmatrix} \mathscr{L}_{C_1} & 0 & \cdots & 0 \\ 0 & \mathscr{L}_{C_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathscr{L}_{C_n} \end{bmatrix} - \begin{bmatrix} \Delta_{C_1} & 0 & \cdots & 0 \\ 0 & \Delta_{C_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Delta_{C_n} \end{bmatrix}$$

Under this setting we have that $x^T\mathscr{L}_{\bar{F}}x = x_{C_1}^T(\mathscr{L}_{C_1} + \Delta_{C_1})x_{C_1} + \cdots + x_{C_n}^T(\mathscr{L}_{C_n} + \Delta_{C_n})x_{C_n}$. Note that, as before, $\mathscr{L}_{C_i}$ and Δ_{C_i} are positive semi-definite matrices, thus $\mathscr{L}_{\bar{F}}$ is singular if and only if Δ_{C_i} is singular and $\mathscr{L}_{C_i}$ and Δ_{C_i} share the same eigenvector associated with the zero eigenvalue for at least one connected component i.

At this point, it is easily seen that the previous properties of Δ_{C_i} still hold with the same arguments for each connected component i and therefore we conclude that each matrix $\mathscr{L}_{\bar{F}_i} = \mathscr{L}_{C_i} + \Delta_{C_i}$ is invertible, using the same arguments as when we assume $\mathscr{G}'$ to be connected, which proves that $\mathscr{L}_{\bar{F}}$ is invertible. ■

References

[1] S. Amin, A.A. Cárdenas, and S.S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In Rupak Majumdar and Paulo Tabuada, editors, *HSCC*, volume 5469 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2009.

[2] G.P. Picco, A.L. Murphy, L. Mottola, M. Ceriotti, S. Guna, and L. Fernandes. Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment. In *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN/SPOTS)*, pages 277–288, April 2009.

[3] R. Olfati-Saber and J.S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of the IEEE Conf. on Decision and Control and European Control Conference*, pages 6698–6703, 2005.

[4] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. pages 8179–8184, 2005.

[5] R.M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

[6] Meng Ji. *Graph-Based Control of Networked Systems*. PhD thesis, Georgia Institute of Technology - School of Electrical and Computer Engineering, August 2007.

[7] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE*, volume 95, pages 215–233, January 2007.

[8] Tanya Gazelle Roosta. *Attacks and Defenses of Ubiquitous Sensor Networks*. PhD thesis, EECS Department, University of California, Berkeley, May 2008.

[9] F. Pasqualetti, A. Bicchi, and F. Bullo. Distributed intrusion detection for secure consensus computations. In *Proceedings of the IEEE Conf. on Decision and Control*, pages 5594–5599, New Orleans, LA, December 2007. IEEE.

[10] S. Sundaram and C.N. Hadjicostis. Distributed function calculation via linear iterations in the presence of malicious agents - part i: Attacking the network. In *Proceedings of the American Control Conference*, pages 1350–1355, June 2008.

[11] S. Sundaram and C.N. Hadjicostis. Distributed function calculation via linear iterations in the presence of malicious agents - part ii: Overcoming malicious behavior. In *Proceedings of the American Control Conference*, pages 1356–1361, June 2008.

[12] S. Ozdemir and Y. Xiao. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, March 2009.

[13] D. Bauso, L. Giarre, and R. Pesenti. Lazy consensus for networks with unknown but bounded disturbances. In *Proceedings of the IEEE Conf. on Decision and Control*, pages 2283–2288, New Orleans, LA, December 2007.

[14] N.A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1st edition, 1997.

[15] A.A. Cárdenas, S. Amin, and S.S. Sastry. Secure control: Towards survivable cyber-physical systems. In *First International Workshop on Cyber-Physical Systems (WCPS2008)*. IEEE, June 2008.

[16] A.A. Cárdenas, S. Amin, and S.S. Sastry. Research challenges for the security of control systems. In *Proceedings of the 3rd USENIX Workshop on Hot topics in security*, page Article 6. USENIX, July 2008.

[17] C. Godsil and G. Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer, 1st edition, 2001.

[18] Jie Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

[19] R. Isermann. Model-based fault-detection and diagnosis-status and applications. In *Annual Reviews in Control*, volume 29, pages 71–85, 2005.

[20] D. Koenig. Unknown Input Proportional Multiple-Integral Observer Design for Linear Descriptor Systems: Application to State and Fault Estimation. *IEEE Transactions on Automatic Control*, 50(2):212–217, February 2005.

[21] S. Hui and S.H. Zak. Observer design for systems with unknown inputs. In *Int. J. Appl. Math. Comput. Sci.*, volume 15, pages 431–446, 2005.

[22] M. Franceschelli, M. Egerstedt, and A. Giua. Motion probes for fault detection and recovery in networked control systems. In *Proceedings of the American Control Conference*, pages 4358–4363, June 2008.

[23] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[24] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *J. Parallel Distrib. Comput.*, 7(2):279–301, 1989.

[25] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[26] R. Olfati-Saber and R.M. Murray. Consensus protocols for networks of dynamic agents. In *Proceedings of the American Control Conference*, volume 2, pages 951–956, June 2003.

[27] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

[28] Meng Ji and M. Egerstedt. A graph-theoretic characterization of controllability for multi-agent systems. In *Proceedings of the American Control Conference*, pages 4588–4593, New York, NY, July 2008.

[29] Meng Ji and M. Egerstedt. Observability and estimation in distributed sensor networks. In *Proceedings of the IEEE Conf. on Decision and Control*, pages 4221–4226, New Orleans, LA, December 2008.

[30] J. Sandhu, M. Mesvahi, and T. Tsukamaki. Relative sensing networks: Observability, estimation, and the control structure. In *Proceedings of the IEEE Conf. on Decision and Control and European Control Conference*, pages 6400–6405, December 2005.

[31] M. Egerstedt, P. Kingston, and E. Verriest. Health monitoring of networked systems. In *Mathematical Theory of Networks and Systems*, Blacksburg, VA, July 2008.

[32] F. Grandoni. A note on the complexity of minimum dominating set. *J. Discrete Algorithms*, 4(2):209–214, July 2006.

[33] A. Abur and A.G. Exposito. *Power System State Estimation: Theory and Implementation*. Marcel-Dekker, 2004.

[34] M. Chenine and Kun Zhu. PMU-based application requirements: A survey in the nordic region. Technical report, Royal Institute of Technology (KTH), November 2008.

[35] E. Scholtz and B.C. Lesieutre. Graphical observer design suitable for large-scale DAE power systems. In *Proceedings of the IEEE Conf. on Decision and Control*, pages 2955–2960, Cancun, December 2008.

[36] M. Aldeen and F. Crusca. Observer-based fault detection and identification scheme for power systems. In *IEE Proceedings - Generation, Transmission and Distribution*, volume 153, pages 71–79, January 2006.

[37] K.-H. Lau and M. Aldeen. A decentralised estimation scheme for an interconnected power-system. In *Proceedings of the Interational Conf. on APSCOM*, volume 2, pages 689–694, Hong Kong, December 1993.

[38] Weiqing Jiang, V. Vittal, and G.T. Heydt. Diakoptic state estimation using phasor measurement units. In *IEEE Transactions on Power Systems*, volume 23, pages 1580–1589, Atlanta, GA, November 2008.

[39] Xue-Bo Chen, S.S. Stankovic, and D.D. Siljak. Decentralized state estimation of multi-area interconnected power systems. In *Proceedings of the American Control Conference*, volume 6, pages 4879–4880, Anchorage, AK, May 2002.

[40] Prabha Kundur. *Power System Stability and Control*. McGraw-Hill Professional, 1994.

[41] Göran Andersson. Modelling and analysis of electric power systems. Lecture Notes ITET ETH Zürich, March 2003.

[42] A. Abur, Jun Zhu, M.J. Rice, G.T. Heydt, and S. Meliopoulos. Enhanced state estimators - final project report. Technical Report 06-45, PSERC - Power Systems Engineering Research Center, November 2006.