UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
CONTROL SYSTEMS ENGINEERING**

"An Unknown Input Observer based approach to Fault Detection for Linear Multi-Agent Systems"

Relatore: Prof.ssa Maria Elena Valcher

Laureando: Giulio Fattore

ANNO ACCADEMICO 2022 - 2023

Data di laurea 13 luglio 2023

Abstract

This thesis deals with the problem of fault detection and isolation (FDI) in multi-agent systems (MAS) with linear dynamics. After a brief introduction of MAS, the relative problems and applications, the problem of FDI is introduced. A model in which two kinds of faults (one on the actuator and one on the sensor) is presented, and an unknown input observers (UIO) technique is used in order to generate the residual signal that is necessary to detect the presence of a fault. Subsequently, the consensus problem is solved simultaneously with the FDI problem. The thesis investigates both the scenario in which the connection topology among the agents is directed and when it is undirected. Finally, some simulations with MATLAB are performed in order to shown the effectiveness of the proposed approach.

Contents

Acronyms

FD Fault Detection.

FDI Fault Detection and Isolation.

LMI Linear Matrix Inequality.

MAS Multi Agent System.

MRC Model Reference Consensus.

RMS Root Mean Square.

SC State Consensus.

UIO Unknown Input Observer.

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Multi Agent Systems

The expression Multi Agent System (MAS) refers to a group of intelligent agents, connected through a network and interacting with each other in order to reach a common objective. In the past decades MASs have received considerable attention in the control system community due to their potential applications in many areas, such as formation of unmanned aerial vehicles (UAVs), unmanned underwater vehicle (UUVs), satellite formation, distributed optimization of multiple robotic systems, distributed filtering and many other topics.
There are several advantages in using a MAS, such as the capability to complete tasks in a distributed manner and this leads to an increase of the performance in terms of computational times and energy harvesting. Furthermore, a decentralised system is more resilient against external attacks with respect to a centralized one.

1.1.1 Dynamics of a MAS

A Multi Agent System network is a particular kind of network in which each node (agent) includes a certain dynamics that can be in discrete or continuous time, time variant or time invariant, linear or non-linear. In this thesis the linear time invariant (LTI) continuous time (CT) case is take into account. In the following, the state space representation of the i^{th} agent is given by:

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + B_u u_i(t), \\ y_i(t) &= Cx_i(t), \quad i = 1, \dots, N, \end{aligned} \tag{1.1}$$

where $x_i(t) \in \mathbb{R}^n$, $u_i(t) \in \mathbb{R}^{n_u}$, $y_i(t) \in \mathbb{R}^{n_y}$ are the state vector, the input and the output of the system, respectively. Moreover, $A \in \mathbb{R}^{n \times n}$ is the state matrix, $B \in \mathbb{R}^{n \times n_u}$ is the input matrix and $C \in \mathbb{R}^{n_y \times n}$ is the output matrix.

The dynamics of whole system can be written in "condensed" form; define the state vector, input and output of the whole system stacking the state, input and output of each agent as follows:

$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix}, \quad U(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{bmatrix}, \quad Y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{bmatrix} \tag{1.2}$$

where $X(t) \in \mathbb{R}^{Nn}$, $U(t) \in \mathbb{R}^{Nn_u}$ and $Y(t) \in \mathbb{R}^{Nn_y}$. With this vector notation it holds that:

$$\begin{aligned} \dot{X}(t) &= \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} X(t) + \begin{bmatrix} B_u & & \\ & \ddots & \\ & & B_u \end{bmatrix} U(t), \\ Y(t) &= \begin{bmatrix} C & & \\ & \ddots & \\ & & C \end{bmatrix} X(t) \end{aligned} \tag{1.3}$$

Using the Kronecker product ($\otimes$), in order to compact the equation, the three real valued, block diagonal matrices can be written as:

$$(I_N \otimes A) = \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix}, \quad (I_N \otimes B) = \begin{bmatrix} B_u & & \\ & \ddots & \\ & & B_u \end{bmatrix}, \quad (I_N \otimes C) = \begin{bmatrix} C & & \\ & \ddots & \\ & & C \end{bmatrix} \tag{1.4}$$

Hence Eq.(1.3) becomes:

$$\begin{aligned} \dot{X}(t) &= \underbrace{(I_N \otimes A)}_{:=A_N} X(t) + \underbrace{(I_N \otimes B_u)}_{:=B_{uN}} U(t), \\ Y(t) &= \underbrace{(I_N \otimes C)}_{:=C_N} X(t) \end{aligned} \tag{1.5}$$

where $dim(A_N) = Nn \times Nn$, $dim(B_{uN}) = Nn_u \times Nn$ and $dim(C_N) = Nn \times Nn_y$.
From now on, every matrix that is the Kronecker product between the identity matrix and a generic matrix will be indicated with the name of the matrix and as subscript the dimension of the identity matrix (e.g. $A_N := I_N \otimes A$).

The connection topology among N agents is represented trough a graph $\mathscr{G} = (V, E)$, where $V = \{1, 2, ..., N\}$ is the set of nodes and the edge set is $E \subseteq V \times V$. Moreover, the adjacency matrix of the graph $\mathscr{A} = a_{ij} \in \mathbb{R}^{N \times N}$ is the matrix whose entries are defined as follows: $a_{ij} = 1$ if $(v_j, v_i) \in E$ and $a_{ij} = 0$ if $(v_j, v_i) \notin E$. Condition $a_{ij} > 0$ indicates that the agent i has access to the information from agent j. The out-degree matrix is defined as $D = diag(d_1, d_2, \dots, d_N)$ with $d_i = \sum_{j=1}^{N} a_{ij}$ and the Laplacian matrix is defined as $L = D - \mathscr{A}$.

In a MAS, agents do not known their own outputs or the outputs of their neighbors but they can measure the difference between them; this is due to the fact that a distributed approach is used and there is no centralized node that stores all the data outputs. With this in mind, it is reasonable to replace the output dynamics of Eq. (1.5) with:

$$W(t) = \begin{bmatrix} \sum_{j=1}^{N} a_{1j}(y_1(t) - y_j(t)) \\ \vdots \\ \sum_{j=1}^{N} a_{Nj}(y_N(t) - y_j(t)) \end{bmatrix} \tag{1.6}$$

and substituting $y_i(t)$ with the definition in Eq. (1.1) the equation becomes

$$W(t) = \begin{bmatrix} \sum_{j=1}^{N} a_{1j}C(x_1(t) - x_j(t)) \\ \vdots \\ \sum_{j=1}^{N} a_{Nj}C(x_N(t) - x_j(t)) \end{bmatrix} \tag{1.7}$$

Note that the i^{th} entry of vector $W(t)$, $[W(t)]_i$, is the weighted sum of the differences between the output $y_i(t)$ of the agent i and the output $y_j(t)$ for every j such that there exists an edge from j to i (i.e. all the neighbors agents j of the agent i); the weight of the edge interconnection is encoded in the value of $a_{ij} \neq 0$.
With the introduction of the Laplacian matrix L the system of Eq. (1.5), using Eq. (1.6) as output dynamics, can be rewritten as:

$$\begin{aligned} \dot{X}(t) &= A_N X(t) + B_{uN} U(t), \\ W(t) &= (L \otimes C) X(t) \end{aligned} \tag{1.8}$$

Technological advances in ad hoc networks and the availability of low-cost reliable computing, data storage, and sensing devices have made it possible to envision scenarios where the coordination of many subsystems extends the range of human capabilities. In these applications, the ability of a network system to fuse information (in a decentralized fashion), compute common estimates of unknown quantities, and agree on a common view of the world is critical. These problems can be formulated as agreement problems on linear combinations of dynamically changing reference signals or local parameters. The dynamic average consensus problem is for a group of agents to cooperate to track the average of locally available time-varying reference signals, assuming that each agent is capable only of local computations and communicates with local neighbors. The difficulty of the dynamic average consensus problem is that the information is distributed across the network. There are several approaches to this problem, the main ones are discussed below. The *centralized* solution to the dynamic average consensus problem is the straightforward one, since all the information are stored in a single place and the computation is performed there. Despite its simplicity, there are several drawbacks that make it an unused method, such as:

1. The algorithm is not robust to failures of the centralized agent;
2. the method is not scalable because the amount of communication and memory required for each agent scales with the size of the network;

3. each agent must have a unique identifier (so that the centralized agent counts each value only once);
4. the calculated average is delayed by an amount that grows with the size of the network;
5. the reference signals from each agent are exposed over the entire network (which is unacceptable in applications involving sensitive data).

The centralized solution is fragile due to the existence of a single failure point in the network. This can be overcome by having every agent act as the centralized agent. In this approach, referred to as *flooding*, agents transmit the values of the reference signals across the entire network until each agent knows each reference signal. While flooding fixes the issue of robustness to agent failures, it is still subject to many of the drawbacks of the centralized solution. Although this approach works reasonably well for small size networks, its communication and storage costs scale poorly in terms of the network size and may cause, depending on how it is implemented, costs of order $O(N^2)$ per agent. This motivates the interest in developing *distributed* solutions for the dynamic average consensus problem that involve only local interactions and decisions among the agents. [8]
Although the use of the distributed approach gives a lot of advantages in terms of performance and robustness, the breakage of a MAS component remains a problem that must not be underestimated. The faults that can occur in this kind of networks are essentially two: the faults in the agents (nodes) and the faults in the communication links among them (edges). It is immediate to understand how the breakage or failure of one or more agents can cause the misbehavior of the whole system or in the worst cases its breakage. Whenever an agent breaks down, the neighboring agents that share information with it could misbehave as well and cause damage to the entire system which may remain in this condition permanently. If a communication path fails, the exchange of information between the agents is affected; in some lucky situations the topology of the network can guarantee that the communication graph remains connected, however this is not in general true since in some circumstances the MAS network split in two or more subsystems that reach their goal (e.g. consensus) with different results. In both the plight, edge or node disconnection/ failure, the final results that the MAS have reached is typically different from the one that is expected in the case in which the fails did not happened.
To avoid this it is necessary to implement a strategy called fault detection and isolation (FDI), to prevent this unwanted ends. The advantage of FDI for MAS systems is its ability to reduce the system's unplanned downtime by detecting a fault before the system suffers severs damage and has to be shut-down. Then preventive maintenance can be carried out and the time during which the system is out of work is considerably reduced. [14]

In this thesis only the case in which the agents have a malfunction in their dynamics will be taken into account. Instead, for what concerns the communication among the agents, it is assumed that remains unchanged. In first place the case in which only one agent at a time can breaks will be considered, subsequently, adding some hypotheses necessary for the resolution of the problem, the situation where several agents have simultaneous faults affecting the dynamics will be investigated. The agent faults are usually modelled through an additive l^2 signal in the state dynamics, however more details will be given in the next section where the fault in a MAS is introduced in a more rigorous way.

1.1.2 Modelling a fault in an agent of a MAS

The fault that occurs in an agent is usually represented through an additive signal, bounded in the $\mathscr{L}_2$ norm, added in the dynamics of the state; together it is also possible consider the presence of some disturbances, for example due to the noisy measure, and finally rewrite Eq. (1.1) as:

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + B_u u_i(t) + B_d d_i(t) + B_f f_i(t), \\ y_i(t) &= Cx_i(t) + D_d d_i(t) + D_f f_i(t), \quad i = 1, \dots, N, \end{aligned} \tag{1.9}$$

where $x_i(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$, $y_i(t) \in \mathbb{R}^{n_y}$ are the state vector, the input and the output of the system, respectively, as in Eq. (1.1); moreover $d_i(t) \in \mathbb{R}^{n_d}$ is an external disturbance and $f_i(t) \in \mathbb{R}^{n_f}$, $n_f \le n_y$, is the fault signal that assumes a value different from zero in the case in which the i^{th} agent is faulty. The matrices of the state space model are $A \in \mathbb{R}^{n \times n}$, $B_u \in \mathbb{R}^{n \times n_u}$, $B_d \in \mathbb{R}^{n \times n_d}$, $B_f \in \mathbb{R}^{n \times n_f}$, $C \in \mathbb{R}^{n_y \times n}$, $D_d \in \mathbb{R}^{n_y \times n_d}$, $D_f \in \mathbb{R}^{n_y \times n_f}$.

Another, more specific, way to model the fault, showed in [15], is to suppose that it can only affect the agent actuator; in this case it is convenient to distinguish two types of fault: outage and partial loss of effectiveness. Both of them are modelled and embedded in the following equation:

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + B_u^f u_i(t) + B_d d_i(t) \\ y_i(t) &= Cx_i(t) + D_d d_i(t), \quad i = 1, \dots, N \end{aligned} \tag{1.10}$$

where B_u^f is defined as:

$$B_u^f = B_u \cdot diag(k_{i1(t)}, k_{i2(t)}, \dots, k_{in_u(t)}), \quad i = 1, \dots, N \tag{1.11}$$

with B_u a full column rank constant matrix; k_{il}, $l = 1, \dots, n_u$ is a time-varying coefficient that indicates the operational status of the l^{th} actuator of the i^{th} agent. Each agent $i = 1, \dots, N$ has a number of actuators n_u whose correlation with the state dynamics is described by the matrix B_u^f.

Until the coefficient is equal to 1 the actuator works correctly, whenever this coefficient is smaller than one it possible to distinguish two kinds of fault: the case in which there is complete breakage of the actuator (i.e. any input given to the agent does not influence in any way the dynamics of the state) and the "intermediate" case in which there is a partial loss of effectiveness of the actuator (i.e. in order to have the same dynamics it is necessary a larger input with respect to the case in which there is no fault).

The fault-free status and two types of actuator faults that are all covered by Eq. (1.11), can be summarized in the following cases:

1. $k_{il}(t) = 1$: the l^{th} actuator of agent i is free of faults;
2. $0 < k_{il}(t) < 1$: the l^{th} actuator of agent i loses part of its effectiveness;
3. $k_{il}(t) = 0$: the outage case. The l^{th} actuator of agent i completely loses its effectiveness and its output is stuck at zero.

Remark 1. In the case in which $B_f f_i(t) = (B_u^f - B_u)u_i(t)$, Eq. (1.9) and Eq. (1.10) are equivalent. This means that the second representation Eq. (1.10) is a particular case of Eq. (1.9).

In this thesis we consider two types of faults that can affect the agents' dynamics: one on the actuator, and the second on the sensor. The two addictive signals are independent and affect two different parts of the dynamics of the agent. The i^{th} agent's actuator fault signal $f_i^a(t) \in \mathbb{R}^{n_u}$ modifies the dynamics of the state in a way similar to the one stated in (1.10), however it is convenient make the presence of the fault explicit, by writing $B_u^f u_i(t)$ as the sum of the input $u_i(t)$ and the fault signal $f_i^a(t)$ multiplied by B_u. For what concerns the sensor fault signal $f_i^s(t) \in \mathbb{R}^{n_y}$, it acts as an addictive signal to the output dynamics of the i^{th} agent. The dynamics of the i^{th} agent hence becomes:

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + B_u u_i(t) + B_d d_i(t) + B_u f_i^a(t), \\ y_i(t) &= Cx_i(t) + D_d d_i(t) + f_i^s(t), \quad i = 1, \dots, N, \end{aligned} \tag{1.12}$$

However, in the following, it will be assumed that only one type of fault at each time can affect an agent, because the scenario in which both sensor and actuator have a fault is improbable.

Now, some assumption are introduced in order to ensure the existence of an UIO. The details will be shown in the next chapter.

Assumption 1. The signals f_i^a, f_i^s and d_i, $i = 1, \dots, N$, are bounded.

Assumption 2. The fault affects only one agent actuator at each time.

Assumption 3. The fault affects only one agent sensor at each time.

Assumption 4. The matrices B_u and B_d are of full column rank.

Assumption 5. The column rank of the matrix

$$\begin{bmatrix} sI_n - A & B_d & B_u \\ C & 0 & 0 \end{bmatrix} \tag{1.13}$$

is full for $\Re(s) > 0$, or at least the matrices

$$\begin{bmatrix} sI_n - A & B_u \\ C & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} sI_n - A & B_d \\ C & 0 \end{bmatrix} \tag{1.14}$$

are not column rank deficient for $\Re(s) > 0$.

Assumption 6. The pair (A,C) is detectable.

Assumption 7. The following relations hold for the matrices B_u, B_d and C:

1. $rank(CB_d) = rank(B_d)$;
2. $rank(CB_u) = rank(B_u)$.

Before starting with the analysis of the various cases, it is convenient to define, as it was done with (1.2), the vectors $F_a(t) \in \mathbb{R}^{Nn_u}$, $F_s(t) \in \mathbb{R}^{Nn_y}$ and $D(t) \in \mathbb{R}^{Nn_d}$ associated to the fault on the actuator, the fault on the sensor and the disturbance signal of the system, respectively:

$$F_a(t) = \begin{bmatrix} f_1^a(t) \\ f_2^a(t) \\ \vdots \\ f_N^a(t) \end{bmatrix}, \quad F_s(t) = \begin{bmatrix} f_1^s(t) \\ f_2^s(t) \\ \vdots \\ f_N^s(t) \end{bmatrix}, \quad D(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \\ \vdots \\ d_N(t) \end{bmatrix} \tag{1.15}$$

Considering this new output definition (with the sensor fault signal $f_i^s(t)$), Eq.(1.6) becomes:

$$\begin{aligned} W(t) &= \begin{bmatrix} \sum_{j=1}^N a_{1j}(y_1(t) - y_j(t)) \\ \vdots \\ \sum_{j=1}^N a_{Nj}(y_N(t) - y_j(t)) \end{bmatrix} \\ &= \begin{bmatrix} \sum_{j=1}^N a_{1j}(Cx_1(t) + D_d d_1(t) + f_1^s(t) - Cx_j(t) - D_d d_i(t) - f_j^s(t)) \\ \vdots \\ \sum_{j=1}^N a_{Nj}(Cx_N(t) + D_d d_N(t) + f_N^s(t) - Cx_j(t) - D_d d_i(t) - f_j^s(t)) \end{bmatrix} \end{aligned} \tag{1.16}$$

and rewriting Eq. (1.16) in a more "useful" form.

$$W(t) = \begin{bmatrix} \sum_{j=1}^N [a_{1j}C(x_1(t) - x_j(t)) + a_{1j}(f_1^s(t) - f_j^s(t)) + a_{1j}D_d(d_1(t) - d_i(t))] \\ \vdots \\ \sum_{j=1}^N [a_{1j}C(x_N(t) - x_j(t)) + a_{1j}(f_N^s(t) - f_j^s(t)) + a_{1j}D_d(d_N(t) - d_i(t))] \end{bmatrix} \tag{1.17}$$

It is now possible to see that

$$\begin{aligned} W(t) = &\begin{bmatrix} \sum_{j=1}^N a_{1j}C(x_1(t) - x_j(t)) \\ \vdots \\ \sum_{j=1}^N a_{1j}C(x_N(t) - x_j(t)) \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^N a_{1j}(f_1^s(t) - f_j^s(t)) \\ \vdots \\ \sum_{j=1}^N a_{1j}(f_N^s(t) - f_j^s(t)) \end{bmatrix} + \\ &+ \begin{bmatrix} \sum_{j=1}^N a_{1j}D_d(d_1(t) - d_i(t)) \\ \vdots \\ \sum_{j=1}^N a_{1j}D_d(d_N(t) - d_i(t)) \end{bmatrix} \end{aligned} \tag{1.18}$$

where it easy to see that the first matrix is identical to the one given in Eq. (1.8), namely $(L \otimes C)X(t)$, instead the other two components are equal to $(L \otimes I_{n_y})F_s(t)$ and to $(L \otimes D_d)D(t)$. Finally, we get

$$W(t) = (L \otimes C)X(t) + (L \otimes I_{n_y})F_s(t) + (L \otimes D_d)D(t) \tag{1.19}$$

Using this new definition of $W(t)$, adding the disturbance and the fault signal (1.15) it is possible to write (1.8) as

$$\begin{aligned} \dot{X}(t) &= A_N X(t) + B_{uN}(U(t) + F_a(t)) + B_{dN} D(t), \\ W(t) &= (L \otimes C) X(t) + (L \otimes I_{n_y}) F_s(t) + (L \otimes D_d) D(t) \end{aligned} \tag{1.20}$$

In the literature alternative ways to detect a fault on an agent can be found, however the most common one is the use of observers in order to generate a residual signal and use it as a warning. In this thesis we will use an Unknown Input Observer (UIO) in order to solve the FDI problem; an introduction to them is presented in the next paragraph.

1.2 Unknown Input Observers

In this section Unknown Input Observer are introduced; following the Chapter 3 of [3] a full order observer for a generic n-dimensional state space is build. In the second part of this section some constraints on the matrices are imposed, in order to ensure convergence to zero of the estimation error, and a theorem that state the conditions for the existence of an Unknown Input Observer for the system is given. Moreover, robust fault detection scheme based on an UIO is shown; this will be taken into account in the next chapter where the method is used in the MAS network.

Starting with a generic n-dimensional state space in which all the matrix are known

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_d d(t), \\ y(t) &= Cx(t) + D_d d(t) \end{aligned} \tag{1.21}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $y(t) \in \mathbb{R}^{n_y}$ is the output vector, $x(t) \in \mathbb{R}^n$ is the input vector and $x(t) \in \mathbb{R}^n$ is the unknown input vector.
We introduce some consideration in order to made next analysis more simple without loosing generality in the statement of the problem.

Remark 2. There is no loss of generality in assuming that the unknown input matrix B_d is of full column rank. When this is not the case, the following rank decomposition can be applied to the matrix B_d:

$$B_d d(t) = B_d^1 B_d^2 d(t)$$

where B_d^1 is a full column rank matrix, B_d^2 is of full row rank and $B_d^2 d(t)$ can now be considered as a new unknown output.

Assumption 8. It is assumed, for the sake of simplicity, that the disturbance can only affect the state update equation and not the output one (i.e. $D_d = 0$).

To sum up, the UIO definition we will introduce below will refer to the following agent dynamics:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_d d(t), \\ y(t) &= Cx(t) \end{aligned} \tag{1.22}$$

We define the state estimation error as the difference between the estimated state $\hat{x}(t)$ and the true value of the state $x(t)$

$$e(t) = x(t) - \hat{x}(t) \tag{1.23}$$

Definition 1.2.1. Unknown Input Observer (UIO). An observer is defined as an unknown input observer for the system described by Eq. (1.22), if its state estimation error vector $e(t)$ approaches zero asymptotically, for every initial condition and control input, regardless of the presence of the unknown input in the system.

A full-order observer for the system (1.22) is described as:

$$\begin{aligned} \dot{z}(t) &= Jz(t) + MB_u u(t) + Ky(t) \\ \hat{x}(t) &= z(t) + Hy(t) \end{aligned} \tag{1.24}$$

where $\hat{x}(t) \in \mathbb{R}^n$ is the estimated state vector, $z(t) \in \mathbb{R}^n$ is the state of this full-order observer, and J, M, K, H are matrices designed for achieving unknown input de-coupling and other design requirement. The observer described by Eq. (1.24) is illustrated in Fig.1.1.

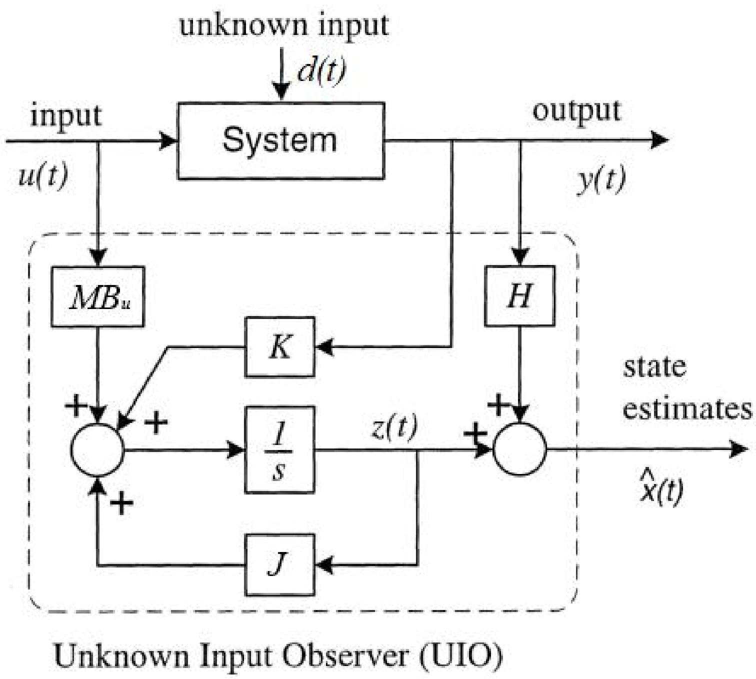


Figure 1.1: Structure of a full-order Unknown Input Observer. [3]

Suppose to split K in two parts $K = K_1 + K_2$: the details about how these two gains are selected will be shown in the following. When the observer (1.24) is applied to the system (1.22), the estimation error ($e(t) = x(t) - \hat{x}(t)$) is governed by the equation:

$$\begin{aligned} \dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\ &= (I - HC)\dot{x}(t) - \dot{z}(t) \end{aligned} \tag{1.25}$$

By substituting the definition of $\dot{x}(t)$ and $\dot{z}(t)$, given respectively by (1.22) and (1.24), in (1.25) we get

$$\dot{e}(t) = (I - HC)(Ax(t) + B_u u(t) + B_d d(t)) - (Jz(t) + MB_u u(t) + (K_1 + K_2)y(t)) \tag{1.26}$$

and after some manipulation, the final result is given by

$$\begin{aligned} \dot{e}(t) = &((I - HC)A - K_1C)e(t) + [J - ((I - HC)A - K_1C)]z(t) \\ &+ [K_2 - ((I - HC)A - K_1C)H]y(t) \\ &+ [M - (I - HC)]B_u u(t) + (HC - I)B_d d(t) \end{aligned} \tag{1.27}$$

If one imposes the following conditions (the cases under which these equalities can be met are discussed below):

$$(HC - I)B_d = 0 \tag{1.28}$$

$$M = I - HC \tag{1.29}$$

$$J = A - HCA - K_1C \tag{1.30}$$

$$K_2 = JH \tag{1.31}$$

The estimation error dynamics then becomes:

$$\dot{e} = Je(t) \tag{1.32}$$

If all the eigenvalues of J are stable, $e(t)$ will approach zero asymptotically, i.e. $\hat{x} \rightarrow x(t)$. This means that the observer (1.24) is an unknown input observer for the system (1.22) according to *Definition 1.2.1.* The design of this UIO consists of solving Eqs. (1.28)-(1.31) and making all the eigenvalues of the system matrix J stable.

Lemma 1.2.1. *Eq.* (1.28) *in the unknown matrix H is solvable if and only if*

$$rank(CB_d) = rank(B_d) \tag{1.33}$$

and a special solution is

$$H^* = B_d[(CB_d)^T CB_d]^{-1}(CB_d)^T \tag{1.34}$$

Now a necessary and sufficient condition, derived in [3], for the existence of an UIO are reported.

Theorem 1.2.2. *Necessary and sufficient conditions for* (1.24) *the existence of a UIO for the system defined by* (1.22) *are:*

1. $rank(CB_d) = rank(B_d)$
2. $(C, A - H^*CA)$ *is a detectable pair.*

Remark 3. Since it was assumed that B_d is of full column rank, this means that the rank of the matrix coincides with the number of its columns. In light of this, condition *1.* of *Theorem 1.2.2* is equal to saying that the rank of the matrix CB_d must be equal to the number of columns of the matrix B_d.

Remark 4. Condition 2 of *Theorem 1.2.2* is identical to the condition that the column rank of $\begin{bmatrix} sI_N - A & B_d \\ C & 0 \end{bmatrix}$ is full when $\Re(s) \geq 0$. [1] [3]

1.2.1 Robust Fault Detection schemes based on UIO

The main task of robust fault detection is to generate a residual which is robust to the system uncertainty. To detect a particular fault, the residual has to be sensitive to this fault. Consider model (1.22) in which the fault signal is added and where the disturbance in the output equation is neglected, due to *Assumption* 8:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_d d(t) + B_u f_a(t), \\ y(t) &= Cx(t) + f_s(t) \end{aligned} \tag{1.35}$$

with the same dimension as (1.22) and with $f_a(t) \in \mathbb{R}^{n_u}$ that represents the fault in the system's actuator and $f_s(t) \in \mathbb{R}^{n_y}$ that is the fault signal associated to the sensor. To generate a robust residual, a UIO described by Eq. (1.24) is required. Suppose that the state estimate is available, the residual is defined as:

$$r(t) = y(t) - C\hat{x}(t) \tag{1.36}$$

when this UIO-based residual generator is applied to the system described in Eq. (1.35), the residual and state estimation error ($e(t)$) will be

$$\begin{aligned} \dot{e}(t) &= ((I - HC)A - K_1 C)e(t) + MB_u f_a(t) - K_1 f_s(t) - H\dot{f}_s(t) \\ r(t) &= Ce(t) + f_s(t) \end{aligned} \tag{1.37}$$

From Eq. (1.37), it can be seen that the disturbance effect has been decoupled from the residual. In order to detect the fault it is necessary to ensure that the fault in the j^{th} actuator will affect the residual if and only if $M[B_u]_j \neq 0$. Similarly, the residual has to be made sensitive to $f_s(t)$ if sensor faults are to be detected. This condition is normally satisfied, as the sensor fault vector $f_s(t)$ has a direct effect on the residual $r(t)$. In order to detect the fault, the residual is used and the following threshold logic is applied

$$\begin{cases} \|r(t)\| \leq \tau. & \text{fault-free case} \\ \|r(t)\| > \tau. & \text{faulty case} \end{cases} \tag{1.38}$$

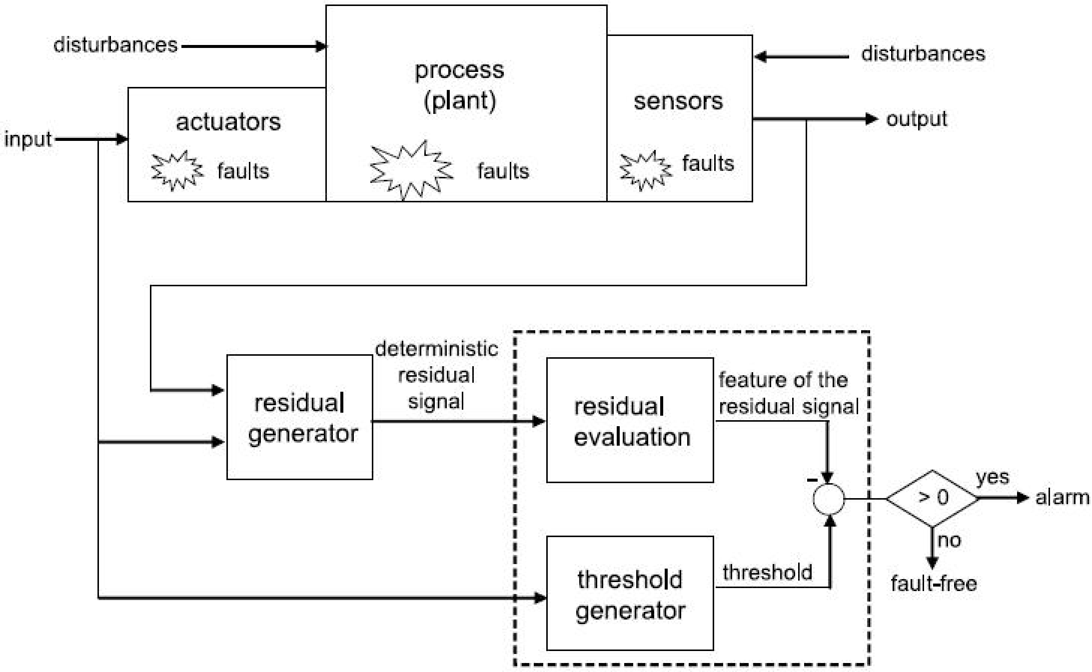


Figure 1.2: Schematic description of residual evaluation and threshold generation. [5]

Choice of threshold value

The residual error, generated by the observer, is generally corrupted with disturbance and uncertainties caused by parameter changes. Since the aim of this thesis is to achieve a successful fault detection based on the available residual signal, it is necessary to take into account this kind of issues. A way to be able to distinguish the faults from the disturbances and uncertainties is the residual evaluation and threshold setting. A decision on the possible occurrence of a fault will then be made by means of a simple comparison between the residual feature and the threshold.

The main task of robust fault detection is to generate a residual which is robust to the system uncertainty. To detect a particular fault, the residual has to be sensitive to this fault. Consider model (1.22) in which the fault signal is added and where the disturbance in the output equation is neglected, due to *Assumption* 8. Depending on the type of the system under consideration, there exist two residual evaluation strategies. The statistic testing is one of them, which is well established in the framework of *statistical methods*. Another one is the so-called *norm-based residual evaluation*. Besides the reduced on-line calculation, the norm-based residual evaluation allows a systematic threshold computation using the well-established robust control theory. [5]

In this thesis we give an hint on the *norm-based evaluation* method in order to choose the best value for the threshold logic scheme. For the purpose of fault detection, an evaluation function is first defined, is based on some mathematical feature of the signal, and, based on it, a threshold is established. The last step is then the decision making.

In the following, the most common evaluation functions are proposed.

- **Peak value**: The peak value of residual signal $r(t)$ is defined and denoted by

$$J_{peak} = \|r(t)\|_{peak} = \sup_{t \geq 0} \|r(t)\| \qquad \|r(t)\| = \sqrt{\sum_{i=1}^{n} r_i^2(t)}$$

so the so-called threshold is defined by

$$J_{th,peak} = \sup_{fault-free} \|r(t)\|_{peak}$$

Also, we can use the peak value of $\dot{r}(t)$ to reformulate the trend analysis. Let

$$J_{trend} = \|\dot{r}(t)\|_{peak} = \sup_{t \geq 0} \|\dot{r}(t)\|$$

consequently

$$J_{th,trend} = \sup_{fault-free} \|\dot{r}(t)\|_{peak}$$

- **RMS value**: The RMS value of $r(t)$ is defined by

$$J_{RMS} = \|r(t)\|_{RMS} = \sqrt{\frac{1}{T} \int_{t}^{t+T} \|r(\tau)\|^2 d\tau}$$

J_{RMS} measures the average energy of $r(t)$ over time interval $(t, t+T)$. Remember that the RMS of a signal is related to the $\mathscr{L}_2$ norm of this signal. In fact, it holds

$$\|r(t)\|_{RMS}^2 \leq \frac{1}{T} \|r(t)\|_2^2$$

then the threshold is defined as

$$J_{th,RMS} = \sup_{fault-free} \|r(t)\|_{RMS}$$

From an engineering viewpoint, the determination of a threshold requires to find out the tolerance limits for disturbances and model uncertainties under fault-free operation conditions. There are a number of factors that can significantly influence this procedure. Among them one can list:

1. the dynamics of the residual generator;
2. the way of evaluating the unknown inputs (disturbances) and model uncertainties;
3. the bounds on the unknown inputs and model uncertainties.

The threshold is understood as the tolerance limit on the unknown inputs and model uncertainties during the fault-free system operation. Based on this consideration, the threshold can be generally defined as

$$J_{th} = \sup_{f=0,\ d,\ \Delta} \|r(t)\| \tag{1.39}$$

With Δ denoting the model uncertainties of the model. Also, the way of evaluating the unknown inputs plays an important role in the determination of thresholds. Typically, the energy level and the maximum value of unknown inputs are adopted in practice for this purpose.

1.3 Consensus problem

Among the several problems that have been investigated for networks of agents, the most notable are the consensus problem and the synchronization problem. In the discussion of both problems, below, the MAS dynamics of Eq. (1.8) will be taken into account.
Suppose to have a network of N agents, represented trough a graph $\mathscr{G} = (V, E)$, under the assumption that there are no self-loops in the graph, and the agents know only the value of their relative output with respect to all their neighbours (i.e. the nodes with which they share an edge). Denote the agent i^{th} relative output information with its neighbours by:

$$w_i(t) = \sum_{j=1}^{N} a_{ij}(y_i(t) - y_j(t)), \quad i = 1, \ldots, N \tag{1.40}$$

where a_{ij} is the element in position (i, j) of the adjacency matrix $\mathscr{A}$ defined previously. Notice that this formulation is identical to the one introduced in Eq. (1.6). Usually the goal of this kind of problems, assuming that the information on the relative output of the agents is available, is to bring the states or the outputs of all agents to be identical, moreover in the case of the synchronization problem the states of the agents must also follow some reference model dynamics.

Remark 5. The relative state information is a particular case of relative output information in which the output matrix is $C = I_n$ (i.e. the state and the output coincide).

Even if the consensus problem and the synchronization problem are the ones on which this thesis mainly focuses, there are many other applications of the consensus schemes to MAS coordination that we now briefly mention [13]:

- **Vehicle Formations**: consensus schemes have been extensively applied to achieve vehicle formations.
- **Attitude Alignment**: in general applied to spacecrafts, the aim of this problem is to have all the agents adopt an identical attitude/orientation.
- **Rendezvous Problem**: the rendezvous problem requires that all agents reach a certain location simultaneously.

- **Coordinated Decision Making**: in MAS, distributed decision making has an advantage over centralized decision making in the sense that a decision maker is not required to access information from all the other decision makers.

- **Flocking**: flocking is a form of collective behavior of a large number of interacting agents with a common group objective. For many decades, scientists from rather diverse disciplines including animal behavior, physics, biophysics, social sciences, and computer science, have been fascinated by the emergence of flocking, swarming, and schooling in groups of agents with local interactions. [12]

- **Coupled Oscillators**: The coupled dynamics of oscillators is described by an ordinary differential equation which comprises of the oscillator's dynamic state with an additional weak coupling term. Therefore, understanding the mutual interactions of coupled oscillators and ensuring consensus and phase locking of the states of these oscillators is a key challenge. [7]

- **Robot Position Synchronization**: Path following problems entail designing control algorithms that drive the output of a control system along a given path in its output space with no timing law assigned to the desired traversal of the path. [9]

In the following the *State consensus (SC)* problem and *Model reference consensus (MRC)* problem (also known as synchronization problem) are presented in a more formal way.

1.3.1 Consensus problem

The aim of this requirement is to drive the state (or part of it) of all the agents to a common value that can be either the average of all the states or not. In order to reach consensus, a common approach is to define a controlled output function $z_{SC_i}(t) = x_i(t) - \frac{1}{N}\sum_{j=1}^{N} x_j(t)$, $i = 1,\dots,N$, to measure the distance of $x_i(t)$ from the average value of the states of all the agents. Note that if $z_{SC_i}(t) = 0$ $\forall i \in V$, then $x_i(t) = x_j(t)$ $\forall i, j \in V$, which implies that the consensus is achieved. Whit this formulation the case where all the agents reach the same state is considered; however it possible to consider only a portion of the state, for example in a scenario in which only the velocity of the agent must be the same. [4]

1.3.2 Model Reference Consensus

The request in this case is not only to achieve the state consensus, but also to ensure that the state follow a reference ("leader") model with dynamics:

$$\begin{aligned} \dot{x}_r(t) &= Ax_r(t) + B_u u_r(t), \\ y_r(t) &= Cx_r(t) \end{aligned} \tag{1.41}$$

where $x_r(t) \in \mathbb{R}^n$, $u_r(t) \in \mathbb{R}^{n_u}$, $y_r(t) \in \mathbb{R}^{n_y}$ are the state vector, the input and the output of the system, respectively, moreover the matrices A, B, C are the same as in Eq. (1.1).

The aim of this problem is that the agent's states (or combination of them) are required to converge to the reference state $x_r(t)$ satisfying Eq. (1.41); in light of this, it is convenient to define the performance outputs $z_{mrc_i}(t) = H(x_i(t) - x_r(t))$, $i = 1, \dots, N$, where $z_{mrc_i} \in \mathbb{R}^{n_z}$, $n_z \leq n_y$ and H is a constant matrix of appropriate dimensions that is defined according to the designer's specifications. [4]

In this thesis the synchronization problem is addressed, simultaneously trying to detect the faults that might affect a MAS.

1.3.3 Connectivity proprieties of graphs

In the next chapters, both undirected and directed topologies for the connection graph will be considered, and also some properties of connected graph will be used. For this reason in this subsection some definition will be given and some theorems state. To make it clear the subparagraph will be split in two parts, the firs refers to the undirected graphs the second one to the digraph. All the definition are taken from [2]. Before to the definition and the theorems, the following assumption is done, and it hold from now on.

Assumption 9. Without loss of generality, the eigenvalues of a matrix are supposed to be ordered so that $Re(\lambda_i) \leq Re(\lambda_j)$, if $i < j$.

Undirected graphs

Definition 1.3.1 (Connected graph)**.** A graph $\mathscr{G}$ is connected if there exists a walk connecting any node to any other node.

Lemma 1.3.1 (Zero row-sums)**.** *Let $\mathscr{G}$ be a weighted undirected graph with Laplacian L and N nodes. Then*

$$L\mathbb{1}_N = \mathbb{0}_N$$

Lemma 1.3.2. *For a weighted undirected graph $\mathscr{G}$ with symmetric Laplacian L:*

(i) $\mathscr{G}$ is connected if and only if $\lambda_i > 0$ for every $i = 2, \dots, N$;

(ii) The multiplicity of 0 as an eigenvalue of L is equal to the number of connected components of $\mathscr{G}$.

Directed graphs (Digraphs)

Definition 1.3.2 (Strongly connected graph)**.** A graph $\mathscr{G}$ is strongly connected if there exists a directed walk from any node to any other node.

Definition 1.3.3 (Spanning subgraphs)**.** A digraph (V', E') is a subgraph of a digraph (V, E) if $V' \subseteq V$ and $E' \subseteq E$. A digraph is a spanning subgraph of (V, E) if it is a subgraph and $V' = V$.

Definition 1.3.4 (Directed spanning tree)**.** A directed tree is an acyclic digraph with the following property: there exists a node, called the root, such that any other node of the digraph can be reached by one and only one directed walk starting at the root. A directed spanning tree of a digraph is a spanning subgraph that is a directed tree.
A digraph $\mathscr{G}$ possesses a directed spanning tree if one of its nodes is the root of directed walks to every other node.

Theorem 1.3.3 (Simplicity of the zero eigenvalue of the Laplacian matrix)**.** *Let L be the Laplacian matrix of a weighted digraph $\mathscr{G}$ with N nodes. Then the following statement are equivalent:*

(i) *$\mathscr{G}$ has a spanning tree;*

(ii) *the eigenvalue* 0 *is simple;*

(iii) $rank(L) = N - 1$.

Lemma 1.3.4 (Spectrum of the Laplacian matrix)**.** *Given a weighted digraph $\mathscr{G}$ with Laplacian L, the eigenvalues of L different from* 0 *have positive real part.*

Chapter 2

Fault Detection and Isolation for MAS with undirected graph

As mentioned before, from now on the disturbance will appear only in the state dynamics, instead for what concerns the fault two possible scenarios will be considered:

(i) the fault acts on an agent's actuator, a case which is represented by the state space equation

$$\begin{aligned} \dot{X}(t) &= A_N X(t) + B_{uN} U(t) + B_{uN} F_a(t) + B_{dN} D(t), \\ W(t) &= (L \otimes C) X(t) \end{aligned} \tag{2.1}$$

(ii) the fault affects the sensor of an agent, a situation that is represented through the state space model

$$\begin{aligned} \dot{X}(t) &= A_N X(t) + B_{uN} U(t) + B_{dN} D(t), \\ W(t) &= (L \otimes C) X(t) + (L \otimes I_{n_y}) F_s(t) \end{aligned} \tag{2.2}$$

Both cases can be represented by means of a unique state space representation given in Eq. (1.20), and at the occurrence one can impose that one of the two faults is identically zero for the whole evolution of the dynamics of the system.
In this chapter the case of an MAS connected with a network described by an undirected graph is taken into account. First, it is shown that an MAS of homogeneous systems (i.e. all the agents have the same dynamics) is not observable; then an UIO approach is used in order to generate a residual signal that will be used for the detection and finally a threshold logic is defined in order to detect the fault.

Assumption 10. The undirected graph $\mathscr{G}$ is connected and hence L has a single eigenvalue in 0 and all the others are positive and real.

2.1 Non observability of a homogeneous MAS

The study of FD for a homogeneous MAS whose agents are connected through an undirected network is complicated by the fact that the whole system Eq. (1.8) is not observable. This result comes from [11] where, after some mild assumption, the *Lemma 2.1.1* is stated.

Assumption 11. The triple (A, B_u, C) is a minimal realisation of the agent dynamics, with B_u and C of full column and row rank, respectively, and $n_u \leq n_y$.

Lemma 2.1.1. *The pair ($L \otimes C$, $I_N \otimes A$) associated with the dynamical system at network level described in* (1.8) *is not observable.*

Proof. The pair ($L \otimes C$, $I_N \otimes A$) is observable if and only if the associated Popov-Belevitch-Hautus (PBH) matrix pencil, given by

$$\mathscr{O}(s) := \begin{bmatrix} (L \otimes C) \\ sI_{Nn} - (I_N \otimes A) \end{bmatrix} = \begin{bmatrix} (L \otimes C) \\ I_N \otimes (sI_n - A) \end{bmatrix}, \tag{2.3}$$

is of full column rank for all $s \in \mathbb{C}$. Since L is positive semi-definite (and singular) and the graph $\mathscr{G}$ associated to it is connected, by Lemma 1.3.1, its N eigenvalues, that we assume ordered, due to *Assumption* 9, satisfy $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$. Also, L admits a spectral decomposition $L = V \Lambda V^T$, where $V \in \mathbb{R}^{N \times N}$ is an orthonormal matrix (i.e., $V^T V = I_N$) and Λ is a diagonal matrix with diagonal entries $\lambda_1, \lambda_2, \ldots, \lambda_N$. Define two orthonormal scaling matrices

$$T_l := \begin{bmatrix} (V^T \otimes I_{n_y}) & 0 \\ 0 & (V^T \otimes I_n) \end{bmatrix} \quad \textit{and} \quad T_r := (V^T \otimes I_n) \tag{2.4}$$

Pre and post multiply the PBH matrix $\mathscr{O}(s)$ in (2.3) by T_l and T_r, respectively. Since the scaling matrices T_l and T_r are non-singular, $T_l \mathscr{O}(s) T_r$ is full column rank if and only if $\mathscr{O}(s)$ is full column rank. Since $V^T V = I_N$ and $V^T L V = \Lambda$, it follows from the properties of the Kronecker product that $(V^T \otimes I_{n_y})(L \otimes C)(V \otimes I_n) = (V^T L V \otimes C) = (\Lambda \otimes C)$ and similarly $(V^T \otimes I_{n_y})(I_N \otimes (sI_n - A))(V \otimes I_n) = (I - N \otimes (sI_n - A))$. From these equalities one gets:

$$T_l \mathscr{O}(s) T_r = \begin{bmatrix} (\Lambda \otimes C) \\ I_N \otimes (sI_n - A) \end{bmatrix} \tag{2.5}$$

Since Λ is diagonal, $(\Lambda \otimes C)$ can be written as $diag(\lambda_1 C, \ldots, \lambda_N C)$ and therefore from (2.5)

$$rank(T_l \mathscr{O}(s) T_r) = \sum_{i=1}^{N} rank\left(\begin{bmatrix} \lambda_i C \\ sI_n - A \end{bmatrix} \right) \tag{2.6}$$

By *Assumption 11*, the pair (C, A) is observable, implying $[C^T (sI - A)^T)]^T$ is full column rank for every $s \in \mathbb{C}$. However $\lambda_1 = 0$, while the other λ_i, $i = 2, \ldots, N$ are strictly positive. Consequently

$$rank \begin{bmatrix} \lambda_1 C \\ sI_n - A \end{bmatrix} = rank(sI_n - A) \tag{2.7}$$

and hence, for every $s \in \mathbb{C}$, $rank(T_l \mathscr{O}(s) T_r) = (N-1)\cdot n + rank(sI - A)$. Because $T_l \mathscr{O}(s) T_r$ (and hence $\mathscr{O}(s)$) loses rank if (and only if, due to the observability assumption on the pair (C,A)) s is an eigenvalue of A, the pair $(L \otimes C,\, I_N \otimes A)$ is not observable. ■

Remark 6. Since the matrix V is orthonormal then $V^T = V^{-1}$.

Remark 7. The unobservable modes of $(L \otimes C,\, I_N \otimes A)$ are the eigenvalues of A and consequently if the system matrix A is stable, then $(L \otimes C,\, I_N \otimes A)$ is detectable.

To extract the observable-subspace from

$$\begin{aligned} \dot{X}(t) &= A_N X(t) + B_{uN}(U(t) + F_a(t)) + B_{dN} D(t), \\ W(t) &= (L \otimes C) X(t) + (L \otimes I_{n_y}) F_s(t) \end{aligned} \tag{2.8}$$

define a change of coordinates transformation matrix T^{-1} in order to have $X \to T^{-1}X = X_o$ where

$$T^{-1} := T_s^{-1} \otimes I_n \quad and \quad T_s^{-1} = \begin{bmatrix} 1 & \mathbb{0}_{N-1}^T \\ -\mathbb{1}_{N-1} & I_{N-1} \end{bmatrix} \tag{2.9}$$

Applying the transformation T^{-1} to the vector state X yields

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \to \quad T^{-1}X = X_o = \begin{bmatrix} x_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_N \end{bmatrix} = \begin{bmatrix} x_1 \\ \bar{X} \end{bmatrix} \tag{2.10}$$

Where $\bar{x}_i = x_i - x_1$ for $i = 2, \ldots, N$.
Using the new state definition, it is possible to rewrite the system matrices as

$$\begin{aligned} A_o := T^{-1} \underbrace{(I_N \otimes A)}_{:=A_N} T &= (T_s^{-1} \otimes I_n)(I_N \otimes A)(T_s \otimes I_n) \\ &= (T_s^{-1} I_N T_s \otimes I_n A I_n) = A_N \end{aligned} \tag{2.11}$$

$$B_{uo} \;:=\; T^{-1} \underbrace{(I_N \otimes B_u)}_{:=B_{uN}} = (T_s^{-1} \otimes B_u) = B_{uN}(T_s^{-1} \otimes I_{n_u}) \tag{2.12}$$

$$B_{do} \;:=\; T^{-1} \underbrace{(I_N \otimes B_d)}_{:=B_{dN}} = (T_s^{-1} \otimes B_d) = B_{dN}(T_s^{-1} \otimes I_{n_d}) \tag{2.13}$$

Finally, by applying the transformation matrix T to $(L \otimes C)$, we obtain

$$C_o := (L \otimes C) T = (L \otimes C)(T_s \otimes I_n) = (L T_s \otimes C) \tag{2.14}$$

For the sake of symmetry, we apply the same transformation T to the sensor fault signal $F_s(t)$ in order to have

$$F_s^o(t) = T^{-1}F_s(t) = \begin{bmatrix} f_1^s(t) \\ \bar{f}_2^s(t) \\ \vdots \\ \bar{f}_N^s(t) \end{bmatrix} = \begin{bmatrix} f_1^s(t) \\ \bar{F}_s(t) \end{bmatrix} \tag{2.15}$$

Where, as before, $\bar{f}_i^s(t) = f_i^s(t) - f_1^s(t)$ for $i = 2, \dots, N$.
In order to decouple the relative output signal, so that the output components do not depend on each other, we pre-multiply $W(t)$ by $(T_s^T \otimes I_{n_y})$ to obtain

$$\begin{aligned} W_o(t) :=& (T_s^T \otimes I_{n_y})W(t) \\ =& (T_s^T \otimes I_{n_y})(LT_s \otimes C)X_o(t) + (T_s^T \otimes I_{n_y})(L \otimes I_{n_y})F_s(t) \\ =& (T_s^T LT_s \otimes C)X_o(t) + (T_s^T L \otimes I_{n_y})(LT_s \otimes I_{n_y})F_s^o(t) \\ =& (T_s^T LT_s \otimes C)X_o(t) + (T_s^T LT_s \otimes I_{n_y})F_s^o(t) \end{aligned} \tag{2.16}$$

From the definition of T_s and the the fact that $L\mathbb{1}_N = 0$ by *Lemma 1.3.1* it is easy to check that

$$T_s^T LT_s = \begin{bmatrix} 0 & 0 \\ 0 & L_s \end{bmatrix} \tag{2.17}$$

where $L_s \in \mathbb{R}^{(N-1)\times(N-1)}$ is a symmetric positive definite matrix; moreover it is a sub-matrix of the original Laplacian matrix with entries obtained by setting the entire first column and row to zero. Hence the relative output measurements $W(t)$ in the new coordinate system can be written as

$$W_o(t) = \begin{bmatrix} 0 \\ (L_s \otimes C) \end{bmatrix} \bar{X}(t) + \begin{bmatrix} 0 \\ (L_s \otimes I_{n_y}) \end{bmatrix} \bar{F}_s(t) \tag{2.18}$$

From the expression of B_{uo} given in (2.12) we obtain

$$B_{uo}U(t) = B_{uN}(T_s^{-1} \otimes I_{n_u})U(t) = B_{uN}U_o(t) \tag{2.19}$$

similarly

$$B_{uo}F_a(t) = B_{uN}(T_s^{-1} \otimes I_{n_u})F_a(t) = B_{uN}F_a^o(t) \tag{2.20}$$

and finally, by (2.13)

$$B_{do}D(t) = B_{dN}(T_s^{-1} \otimes I_{n_d})D(t) = B_{dN}D_o(t) \tag{2.21}$$

where $U_o(t)$, $F_a^o(t)$ and $D_o(t)$ are defined, respectively, as

$$U_o = \begin{bmatrix} u_1 \\ \bar{u}_2 \\ \vdots \\ \bar{u}_N \end{bmatrix} = \begin{bmatrix} u_1 \\ \bar{U} \end{bmatrix}, \quad F_a^o = \begin{bmatrix} f_1^a \\ \bar{f}_2^a \\ \vdots \\ \bar{f}_N^a \end{bmatrix} = \begin{bmatrix} f_1^a \\ \bar{F}_a \end{bmatrix} \quad D_o = \begin{bmatrix} d_1 \\ \bar{d}_2 \\ \vdots \\ \bar{d}_N \end{bmatrix} = \begin{bmatrix} d_1 \\ \bar{D} \end{bmatrix} \tag{2.22}$$

and, as before $\bar{u}_i = u_i - u_1$, $\bar{f}_i^a = f_i^a - f_1^a$ and $\bar{d}_i = d_i - d_1$ for every $i = 2, \dots, N$. To sum up, the whole system, after the change of coordinates, is rewritten in the following way

$$\begin{aligned} \dot{X}_o(t) &= A_N X_o(t) + B_{uN} U_o(t) + B_{uN} F_a^o(t) + B_{dN} D_o(t) \\ W_o(t) &= (T_s^T L T_s \otimes C) X_o(t) + (T_s^T L T_s \otimes I_{n_y}) F_s^o(t) \end{aligned} \tag{2.23}$$

The structure just introduced allows to isolate the observable sub-system from the remaining one; hence the observable subsystem can be written as

$$\begin{aligned} \dot{\bar{X}}(t) &= \underbrace{(I_{N-1} \otimes A)}_{:=A_{N-1}} \bar{X}(t) + \underbrace{(I_{N-1} \otimes B_u)}_{:=B_{uN-1}} \bar{U}(t) + \underbrace{(I_{N-1} \otimes B_u)}_{:=B_{uN-1}} \bar{F}_a(t) + \underbrace{(I_{N-1} \otimes B_d)}_{:=B_{dN-1}} \bar{D}(t) \\ \breve{W}(t) &= (L_s \otimes C) \bar{X}(t) + (L_s \otimes I_{n_y}) \bar{F}_s(t) \end{aligned} \tag{2.24}$$

Since the sub-matrix L_s exhibits the same eigenvalues as the Laplacian matrix L, except for $\lambda_1 = 0$, there exists a change of coordinates matrix $V \in \mathbb{R}^{(N-1)\times(N-1)}$ such that

$$L_s = V \Lambda_s V^T = V \begin{bmatrix} \lambda_2 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix} V^T \tag{2.25}$$

where $0 < \lambda_2 \leq \cdots \leq \lambda_N$ are the eigenvalues of the matrix L_r.
Define, again, a change of coordinates for the observable subsystem, so that $\bar{X} \to T_v \bar{X} = \tilde{X}$ where $T_v = (V^T \otimes I_n)$ and consequently:

$$\begin{aligned} \tilde{A} := T_v A_{N-1} T_v^{-1} &= (V^T \otimes I_n)(I_{N-1} \otimes A)(V \otimes I_n) \\ &= (V I_{N-1} V^T \otimes I_n A I_n) = A_{N-1} \end{aligned} \tag{2.26}$$

$$\tilde{B}_u := T_v B_{uN-1} = (V^T \otimes B_u) = B_{uN-1}(V^T \otimes I_{n_u}) \tag{2.27}$$

$$\tilde{B}_d := T_v B_{dN-1} = (V^T \otimes B_d) = B_{dN-1}(V^T \otimes I_{n_d}) \tag{2.28}$$

where $\tilde{U} = (V^T \otimes I_{n_u})\bar{U}$, $\tilde{F}_a = (V^T \otimes I_{n_u})\bar{F}_a$ and $\tilde{D} = (V^T \otimes I_{n_d})\bar{D}$. Moreover apply the same transformation $(V^T \otimes I_{n_y})$ to the sensor fault signal $\bar{F}_s(t)$ so that $\tilde{F}_s = (V^T \otimes I_{n_y})\bar{F}_s$.
Finally, pre-multiply $\breve{W}$ by $(\Lambda_s^{-1} V^T \otimes I_{n_y})$ to obtain

$$\tilde{W} = (\Lambda_s^{-1} V^T \otimes I_{n_y}) \breve{W} \tag{2.29}$$

which is, by (2.24) and the fact that $\bar{X} = T_v^{-1}\tilde{X}$ equal to

$$\begin{aligned}
\tilde{W}(t) &= (\Lambda_s^{-1}V^T \otimes I_{n_y})[(L_s \otimes C)\bar{X}(t) + (L_s \otimes I_{n_y})\bar{F}_s(t)] \\
&= (\Lambda_s^{-1}V^T \otimes I_{n_y})(L_r \otimes C)(V \otimes I_{n_y})\tilde{X}(t) + (\Lambda_s^{-1}V^T \otimes I_{n_y})(L_s \otimes I_{n_y})(V \otimes I_n)\tilde{F}_s(t) \\
&= (I_{N-1} \otimes C)\tilde{X}(t) + (I_{N-1} \otimes I_{n_y})\tilde{F}_s(t) \\
&= C_{N-1}\tilde{X}(t) + I_{(N-1)n_y}\tilde{F}_s(t)
\end{aligned} \tag{2.30}$$

and thus system (2.24), in the new coordinates, becomes

$$\begin{aligned}
\dot{\tilde{X}}(t) &= A_{N-1}\tilde{X}(t) + B_{uN-1}\tilde{U}(t) + B_{uN-1}\tilde{F}_a(t) + B_{dN-1}\tilde{D}(t) \\
\tilde{W}(t) &= C_{N-1}\tilde{X}(t) + I_{(N-1)n_y}\tilde{F}_s(t)
\end{aligned} \tag{2.31}$$

In the new coordinates, system (2.24) is equivalent to the decoupled system, in the sense that the dynamics of each of the $N-1$ agents is described in a way that does not depend on the dynamics of the other agents.

$$\begin{aligned}
\dot{\tilde{x}}_i(t) &= A\tilde{x}_i(t) + B_u\tilde{u}_i(t) + B_u\tilde{f}_i^a(t) + B_d\tilde{d}_i(t) \\
\tilde{w}_i(t) &= C\tilde{x}_i(t) + \tilde{f}_i^s(t), \quad i = 2, \ldots, N
\end{aligned} \tag{2.32}$$

If instead of the decoupled system we want to work with the one that represents the relative measures with respect to the agent 1, we need to perform the following transformation on the Laplacian matrix:

$$T_s^{-1}LT_s = \begin{bmatrix} 0 & l^T \\ 0 & L_r \end{bmatrix} \tag{2.33}$$

and define, as usual, $\bar{w}_i(t) = w_i(t) - w_1(t)$. It is possible to verify that

$$\begin{aligned}
\dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{uN-1}\bar{U}(t) + B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) \\
\bar{W}(t) &= (L_r \otimes C)\bar{X}(t) + (L_r \otimes I_{n_y})\bar{F}_s(t)
\end{aligned} \tag{2.34}$$

this second model will be used both for the analysis of the FDI of the actuator and in the resolution of the synchronization problem. Instead the "decoupled" model (2.31) will be used in the following for the FD in the sensors fault case.

2.2 Actuator Fault Detection and Isolation

The Fault Detection and Isolation (FDI) problem is the problem of locating the fault, i.e. of determining at which node the fault has occurred. One of the approaches to facilitate fault isolation is to design a residual set which is designed to be sensitive to a certain group of faults and insensitive to others. The sensitivity and insensitivity properties make isolation possible. The ideal situation is to make each residual only sensitive to a particular fault and insensitive to all other faults. However, this ideal situation is normally difficult to achieve. Even when the ideal situation can be achieved, the design freedom will be used and no freedom will be left to achieve robustness. To exploit the maximum design freedom for robustness, a commonly accepted scheme in fault isolation, used also in this thesis, is to make each residual sensitive to faults for all but one agents [3].

2.2.1 Residual signal generator for a fault of an agent actuator

As mentioned before, it is assumed that only one type of fault can occur at each time. In this subsection an actuator fault is considered (i.e. we assume $f_i^s(t) = 0 \ \forall\ i = 1,\dots,N, \ \forall\ t$), and hence the dynamics of the considered MAS is

$$\begin{aligned} \dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{uN-1}(\bar{U}(t) + \bar{F}_a(t)) + B_{dN-1}\bar{D}(t) \\ \bar{W}(t) &= (L_r \otimes C)\bar{X}(t) \end{aligned} \tag{2.35}$$

To detect whether or not there is a fault, an UIO for the system is designed. Before doing that, a wise representation of the fault signal is adopted, namely the fault signal $\bar{F}_a(t)$ is divided in two parts, in order to highlight the generic i^{th} fault signal $\bar{f}_i^a$ and the associated transfer matrix that must be excluded in the FD. We assume

$$B_{uN-1}\bar{F}_a(t) = B_{uN-1}^{\;i}\bar{f}_i^a + B_{uN-1}^{\;-i}\bar{F}_a^{-i}(t) \tag{2.36}$$

Where $\bar{F}_a^{-i}(t) \in \mathbb{R}^{(N-2)n_u}$ is obtained from the vector $\bar{F}_a(t)$ by removing the entry corresponding to the i^{th} agent, $\bar{f}_i^a \in \mathbb{R}^{n_u}$. On the other hand, $B_{uN-1}^{\;i} \in \mathbb{R}^{(N-1)n \times n_u}$ is the block of columns of B_{uN-1} with indices ranging from $(i-1)n_u + 1$ to in_u and $B_{uN-1}^{\;-i} \in \mathbb{R}^{(N-1)n \times (N-2)n_u}$ is what remains of B_{uN-1} after removing $B_{uN-1}^{\;i}$. In a more graphical way

$$\bar{F}_a^{-i}(t) = \begin{bmatrix} \bar{f}_2^a \\ \vdots \\ \bar{f}_{i-1}^a \\ \bar{f}_{i+1}^a \\ \vdots \\ \bar{f}_n^a \end{bmatrix} \quad B_{uN-1}^{\;i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ B_u \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad B_{uN-1}^{\;-i} = \begin{bmatrix} B_u & & & & & \\ & \ddots & & & & \\ & & B_u & & & \\ & & & B_u & & \\ & & & & \ddots & \\ & & & & & B_u \end{bmatrix} \tag{2.37}$$

Remark 8. The following relation holds: $B_{uN-1}^{\;i} = (e_i \otimes B_u)$, where e_i is the i^{th} canonical base vector (i.e. the vector with all zeros, but one 1 in position i).

As a result, (2.31) becomes

$$\begin{aligned} \dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{uN-1}\bar{U}(t) + B_{uN-1}^{\;i}\bar{f}_a^i + B_{uN-1}^{\;-i}\bar{F}_a^{-i}(t) + B_{dN-1}\bar{D}(t), \\ \bar{W}(t) &= (L_r \otimes C)\bar{X}(t) \end{aligned} \tag{2.38}$$

Since the matrix B_{uN-1} is shared by both the actuator fault signal $\bar{F}_a(t)$ and the input signal

$\bar{U}(t)$, the same partition adopted above is performed on this last term, and hence we obtain:

$$B_{uN-1}\bar{U}(t) = B_{uN-1}^{\;i}\bar{u}_i + B_{uN-1}^{\;-i}\bar{U}^{-i}(t) \tag{2.39}$$

where as before $\bar{U}^{-i}(t) \in \mathbb{R}^{(N-2)n_u}$ is obtained from the vector $\bar{U}(t)$ by deleting the i^{th} agent's input. In light of this, it is possible to group some terms in Eq. (2.38), in particular we define

$$\begin{aligned} B_d^i D^i(t) =& B_{dN-1}\bar{D}(t) + B_{uN-1}^{\;i}\bar{u}_i + B_{uN-1}^{\;i}\bar{f}_i^a \\ =& B_{dN-1}\bar{D}(t) + B_{uN-1}^{\;i}(\bar{u}_i + \bar{f}_i^a) \\ =& \begin{bmatrix} B_{dN-1} & B_{uN-1}^{\;i} \end{bmatrix} \begin{bmatrix} \bar{D}(t) \\ (\bar{u}_i + \bar{f}_i^a) \end{bmatrix} \end{aligned} \tag{2.40}$$

With this representation the fault signal corresponding to the i^{th} agent is embedded in the disturbance term of the equation. Substituting (2.40) in Eq. (2.38) we obtain:

$$\begin{aligned} &\dot{\bar{X}}(t) = A_{N-1}\bar{X}(t) + B_{uN-1}^{\;-i}\bar{U}^{-i}(t) + B_{uN-1}^{\;-i}\bar{F}_a^{-i}(t) + B_d^i D^i(t), \\ &\bar{W}(t) = (L_r \otimes C)\bar{X}(t) \end{aligned} \tag{2.41}$$

In order to detect which agent is the faulty one, a UIO is built for every agent $i = 2, \dots, N$ considering as unknown input the augmented signal $D^i(t)$ defined in (2.40). Following the same steps as in the previous chapter, a full order observer will be

$$\begin{aligned} &\dot{z}^i(t) = J^i z^i(t) + M^i B_{uN-1}^{\;-i}\bar{U}^{-i}(t) + K^i\bar{W}(t) \\ &\hat{X}^i(t) = z^i(t) + H^i\bar{W}(t) \quad\quad i = 2, \dots, N \end{aligned} \tag{2.42}$$

Define the estimation error, as before, as

$$e_a^i(t) = \bar{X}(t) - \hat{X}^i(t) \tag{2.43}$$

Its update equation results to be the same as Eq.(1.27), namely

$$\begin{aligned} \dot{e}_a^i(t) &= \dot{\bar{X}}(t) - \dot{\hat{X}}^i(t) \\ &= \dot{\bar{X}}(t) - (\dot{z}^i(t) + H^i\dot{\bar{W}}(t)) \\ &= \dot{\bar{X}}(t) - (\dot{z}^i(t) + H^i(L_r \otimes C)\dot{\bar{X}}(t) \\ &= (I_{(N-1)n} - H^i(L_r \otimes C))\dot{\bar{X}}(t) - \dot{z}^i(t) \end{aligned} \tag{2.44}$$

By replacing the expression (2.41) and (2.42) in the equation (2.44) we obtain

$$\begin{aligned} \dot{e}_a^i(t) = (I_{(N-1)n} - H^i(L_r \otimes C))(A_{N-1}\bar{X}(t) + B_{uN-1}^{\;-i}\bar{U}^{-i}(t) + B_{uN-1}^{\;-i}\bar{F}_a^{-i}(t) + B_d^i D^i(t)) \\ - (J^i z^i(t) + M^i B_{uN-1}^{\;-i}\bar{U}^{-i}(t) + K^i\bar{W}(t)) \end{aligned} \tag{2.45}$$

Substitute, now, the expression of K^i with $K_1^i + K_2^i$, and substitute $\bar{W}(t)$ with its expression given in Eq. (2.41) only for the components that multiply matrix K_1^i, to obtain

$$\begin{aligned}\dot{e}_a^i(t) = &[(I_{(N-1)n} - H^j(L_r \otimes C))A_{N-1} - K_1^i(L_r \otimes C)]\bar{X}(t)\\&+(I_{(N-1)n} - H^j(L_r \otimes C) - M^i)B_{u_{N-1}}^{-i}\bar{U}^{-i}(t)\\&+(I_{(N-1)n} - H^j(L_r \otimes C))B_{u_{N-1}}^{-i}\bar{F}_a^{-i}(t)\\&+(I_{(N-1)n} - H^j(L_r \otimes C))B_d^i D^i(t)\\&-J^i z^i(t) - K_2^i \bar{W}(t)\end{aligned} \tag{2.46}$$

Moreover, if the following matrix relations are imposed for $i = 2, \ldots, N$

$$(H^i(L_r \otimes C) - I_{(N-1)n})B_d^i = 0 \tag{2.47}$$

$$M^i = I_{(N-1)n} - H^i(L_r \otimes C) \tag{2.48}$$

$$J^i = M^i A_{N-1} - K_1^i(L_r \otimes C) \tag{2.49}$$

$$K_2^i = J^i H^i \tag{2.50}$$

$$K^i = K_1^i + K_2^i \tag{2.51}$$

the error dynamics of the i^{th} observer becomes

$$\begin{aligned}\dot{e}_a^i(t) &= (M^i A_{N-1} - K_1^i(L_r \otimes C))\bar{X}(t) + M^i B_{u_{N-1}}^{-i}\bar{F}_a^{-i}(t) - J^i z^i(t) - J^i H^i \bar{W}(t)\\&= J^i(\bar{X}(t) - \hat{X}^i(t)) + M^i B_{u_{N-1}}^{-i}\bar{F}_a^{-i}(t)\\&= J^i e_a^i(t) + M^i B_{u_{N-1}}^{-i}\bar{F}_a^{-i}(t)\end{aligned} \tag{2.52}$$

Remark 9. In order to ensure the convergence to zero of the estimation error, in the case in which there is no fault, it is necessary that J^i is asymptotically stable for every $i = 2, \ldots, N$ (i.e. $\Re(\sigma(J^i)) < 0, \forall\, i = 2, \ldots, N$).

Notice that if there are no faults in the system, the part of the error dynamics $M^i B_{u_{N-1}}^{-i}\bar{F}_a^{-i}(t)$ is zero, and hence since J^i is stable, the error converges to zero. On the other hand, if a fault occurs at one of the agents $j \neq i$, the error will increase as its derivative is no longer identically zero.

Define, finally, the residual associated to the i^{th} observer as

$$\begin{aligned}r_a^i(t) &= \bar{W} - (L_r \otimes C)\hat{X}(t)\\&= (L_r \otimes C)\bar{X} - (L_r \otimes C)\hat{X}(t)\\&= (L_r \otimes C)e^i(t)\end{aligned} \tag{2.53}$$

Observe that the residual $r_a^i(t)$ directly depends on the error $e_a^i(t)$. Therefore it follows that the reasoning previously made for error holds also for the residual signal.

With the strategy used to build the i^{th} UIO, we are able, when a fault affects the system, by looking the associated residual $r_a^i(t)$, to detect if such fault occurred at one of the agent of the network $j = 1, \ldots, N$ except for the i^{th} one. For this reason $N-1$ observers are built in order to

generate as many residual signals as possible, that will be used for the FDI.
Now it is necessary to verify through *Theorem 1.2.2* if the observer described by (2.42) is a UIO for the system (2.41), more precisely if it is a UIO for the system

$$\begin{aligned}\dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{u_{N-1}}^{-i}\bar{U}^{-i}(t) + B_d^i D^i(t),\\ \bar{W}(t) &= (L_r \otimes C)\bar{X}(t)\end{aligned} \tag{2.54}$$

for which it is straightforward to realise that $D^i(t)$ is the unknown input. The first condition that must be verified is that

$$rank((L_r \otimes C)B_d^i) = rank(B_d^i) \tag{2.55}$$

By developing the equation one gets

$$\begin{aligned}(L_r \otimes C)B_d^i &= (I_{N-1} \otimes C)\begin{bmatrix} B_{dN-1} & B_{u_{N-1}}^{i}\end{bmatrix}\\ &= (L_r \otimes C)\begin{bmatrix}(I_{N-1} \otimes B_d) & (e_i \otimes B_u)\end{bmatrix}\\ &= \begin{bmatrix}(L_r \otimes C)(I_{N-1} \otimes B_d) & (L_r \otimes C)(e_i \otimes B_u)\end{bmatrix}\\ &= \begin{bmatrix}(L_r \otimes CB_d) & (L_r * e_i \otimes CB_u)\end{bmatrix}\end{aligned} \tag{2.56}$$

since by *Assumption 7* the matrices CB_d and CB_u have the same rank as B_d and B_u respectively and the reduced Laplacian L_r is full rank by construction, condition (2.55) holds.

For what concerns the second requirement of *Theorem 1.2.2*, in light of *Remark 4* we have to verify that

$$\begin{bmatrix} sI_{(N-1)n} - A_{N-1} & B_d^i \\ (L_r \otimes C) & 0 \end{bmatrix} \quad \text{is full column rank when} \quad \Re(s) \geq 0. \tag{2.57}$$

The previous matrix can be rewritten as follows

$$\begin{aligned}&\begin{bmatrix}(I_{N-1} \otimes sI_n) - (I_{N-1} \otimes A) & (I_{N-1} \otimes B_d) & (e_i \otimes B_u)\\ (L_r \otimes C) & 0 & 0\end{bmatrix} =\\ &\begin{bmatrix}(I_{N-1} \otimes (sI_n - A)) & (I_{N-1} \otimes B_d) & (e_i \otimes B_u)\\ (L_r \otimes C) & 0 & 0\end{bmatrix}\end{aligned} \tag{2.58}$$

Consider the first Nn column, and notice that the following inequality holds true

$$\begin{aligned}\begin{bmatrix}(I_{N-1} \otimes sI_n) - (I_{N-1} \otimes A)\\ (L_r \otimes C)\end{bmatrix} &= \begin{bmatrix}(VI_{N-1}V^T \otimes (sI_n - A))\\ (V\Lambda_r V^T \otimes C)\end{bmatrix}\\ &= \begin{bmatrix}(V \otimes I_n)(I_{N-1} \otimes (sI_n - A))(V^T \otimes I_n)\\ (V \otimes I_{n_y})(\Lambda_r \otimes C)(V^T \otimes I_n)\end{bmatrix}\end{aligned} \tag{2.59}$$

and finally get

$$\begin{bmatrix}(V\otimes I_n) & \\ & (V\otimes I_{n_y})(\Lambda_r\otimes I_{n_y})\end{bmatrix}\begin{bmatrix}I_{N-1}\otimes(sI_n-A)\\ I_{N-1}\otimes C\end{bmatrix}(V^T\otimes I_n) \tag{2.60}$$

'since the matrices V, V^T and Λ_r are non-singular by construction, also their Kronecker product with the identity matrix is non-singular, and hence evaluating the rank of the first Nn column of (2.58) is equivalent to evaluate the column rank of

$$\begin{bmatrix}(I_{N-1}\otimes(sI_n-A))\\ (I_{N-1}\otimes C)\end{bmatrix} \tag{2.61}$$

Consequently, the original problem has now been converted into the problem of verifying that the column rank of

$$\begin{bmatrix}(I_{N-1}\otimes(sI_n-A)) & (I_{N-1}\otimes B_d) & (e_i\otimes B_u)\\ (I_{N-1}\otimes C) & 0 & 0\end{bmatrix} \tag{2.62}$$

is not deficient when $\Re(s)\geq 0$.

Through the use of permutation matrices, which do not alter the rank of the matrix to which they are applied, it is possible to bring the matrix (2.62) to block diagonal form consisting of one diagonal block (the first one) equal to

$$\begin{bmatrix}sI_n-A & B_d & B_u\\ C & 0 & 0\end{bmatrix} \tag{2.63}$$

and the remaining $N-2$ blocks equal to

$$\begin{bmatrix}sI_n-A & B_d\\ C & 0\end{bmatrix} \tag{2.64}$$

Consider, first, matrix (2.63). It can be observed that if the matrix in (2.63) is of full column rank for $\Re(s)\geq 0$, then also (2.64) has this property, since (2.64) coincides with the first $n+n_d$ column of (2.63). By *Assumption 5*, (2.63) is of full column rank for $\Re(s)\geq 0$.

Hence in the first part of the proof it has been proven that $rank((L_r\otimes C)B_d^i)=rank(B_d^i)$. In the second part of the proof, instead, it has been shown that the column rank of $\left[\begin{smallmatrix}sI_{(N-1)n}-A_{N-1} & B_d^i\\ (L_r\otimes C) & 0\end{smallmatrix}\right]$ is full when $\Re(s)\geq 0$ and by *Remark 4* this is equivalent to the fact that the pair $((L_r\otimes C), A_{N-1}-H^*(L_r\otimes C)A_{N-1})$, where $H^*=B_d^1[((L_r\otimes C)B_d^i)^T(L_r\otimes C)B_d^{-i}]^{-1}((L_r\otimes C)B_d^i)^T$, is detectable. These are the two necessary and sufficient conditions of *Theorem 1.2.2* that guarantee the existence of an UIO for the system (2.41). This concludes the proof.

Remark 10. Imposing that the columns of the matrices $B_d\in\mathbb{R}^{n\times n_d}$ and $B_u\in\mathbb{R}^{n\times n_u}$ are linearly independent implies that $n_d+n_u\leq n$.

Remark 11. Considering the disturbance $d_i(t)$, affecting the dynamics of the system, as an unknown input is not always convenient. Indeed it is necessary to ensure that the matrix $\left[\begin{smallmatrix}B_u & B_d\end{smallmatrix}\right]$

is of full column rank in order to guarantee the existence of an UIO for the system. To overcome this issue, the simplest solution is to not consider the disturbance of the system as an unknown input. It follows that the conditions that must be verified are:

1. $rank(CB_u) = rank(B_u)$;
2. The column rank of $\begin{bmatrix} sI_n - A & B_u \\ C & 0 \end{bmatrix}$ is full when $\Re(s) > 0$.

It follows that the residual error in this case could converge to values that are different from zero even if there is no fault at the actuator. This behavior will be considered when the threshold values are designed in order to avoid false alarms in the detection of malfunctioning in the system.

2.2.2 Actuator Fault Detection and Isolation using residual signals

Based on the work done in [1], a strategy that uses the residual signals (2.53) generated by an UIO is presented.
From previous computation the matrix J^i twas assumed to be Hurwitz, under this assumption, it was shown that, in the case of actuator fault the error dynamics $\dot{e}_a^i(t)$ is driven by the fault signals $\tilde{F}_a^{-i}$. It follows that if the fault signals are equal to zero, the error $e_a^i(t)$ converges to zero asymptotically. By *Assumption 2*, there is at most one faulty agent in the system, hence it is possible to distinguish two situations: agent 1 has a fault or one of the agents $i = 2, \dots, N$ has a fault (i.e. $f_1^a(t) \neq 0$ or $\exists\, i \neq 1$ such that $f_i^a(t) \neq 0$).

1. if there is a fault at the agent 1, this implies that $f_1^a(t) \neq 0$. By *Assumption 2* there can only be one fault, consequently $\forall i = 2, \dots, N$, we have that $f_i^a(t) = 0$. It follows that $\tilde{f}_i^a(t) = f_i^a(t) - f_1^a(t) \neq 0\ \forall i = 2, \dots, N$, and also, by definition, $\tilde{F}_a^{-i}(t)$ is different from zero. This implies that the estimation error $e_a^i(t)$ does not tend to zero end hence also

$$\lim_{t\to\infty} r_a^i(t) \neq 0 \quad \forall\, i = 2, \dots, N \tag{2.65}$$

2. if the faults occur at an agent $i \neq 1$, the fault signal associated to the i^{th} agent will be $f_i^a \neq 0$, thereby also the vector $\tilde{f}_i^a$ will be different from zero; instead, by definition, we have that $\tilde{F}_a^{-i} = 0$. It follows that

$$\begin{cases} \lim_{t\to\infty} r_a^i(t) = 0 & \exists\, i, i \neq 1 \\ \lim_{t\to\infty} r_a^j(t) \neq 0 & j = 2, \dots, i-1, i+1, \dots, N \end{cases} \tag{2.66}$$

Therefore, the residual signal can be used in order to identify and exclude the faulty agent of the MAS.
After proving, in the previous sections, the existence of an UIO and have just shown that it is possible to distinguish different scenarios depending on where the fault is located looking only

at the value of the residual signal, a logic threshold is introduced to cover all possible cases.

$$
\begin{aligned}
&\begin{cases} \|r_a^i(t)\| < \tau, & \exists\, i, i \neq 1 \\ \|r_a^j(t)\| \geq \tau, & \forall j \neq i \end{cases} && \Longrightarrow f_i \neq 0 \\
&\begin{cases} \|r_a^i(t)\| \geq \tau, & \forall\, i = 2, \dots, N \end{cases} && \Longrightarrow f_1 \neq 0 \\
&\begin{cases} \|r_a^i(t)\| < \tau, & \forall\, i = 2, \dots, N \end{cases} && \Longrightarrow \text{fault-free}
\end{aligned}
\tag{2.67}
$$

Where τ_a is the threshold associated to the logic.

Remark 12. In the case in which, instead of relative output information, the system collects relative state information the reasoning does not change. In fact, this special case is equivalent to solve the problem above for $C = I_n$.
For what concerns the existence of an UIO, the two requirements of *Theorem 1.2.2* hold since the first one is satisfied by the equalities $rank(CB_d) = rank(I_n B_d) = rank(B_d)$ and $rank(CB_u) = rank(I_n B_u) = rank(B_u)$; instead, the second requirement holds by *Lemma A.0.1*.
The rest of the FDI procedure, therefore, remains the same.

Algorithm 1 FDI for MASs with undirected topology.

1. Number N agents and select agent number 2, that is, begin from $i = 2$.
2. Solve the equation $(H^i C_{N-1} - I_{(N-1)n}) B_d^i = 0$ to get a solution of H^i, then set $M^i = I_{(N-1)n} - H^i C_{N-1}$;
3. Determine K_1^i such that $J^i = M^i A_{N-1} - K_1^i C_{N-1}$ is stable.
4. Compute $K_2^i = J^i H^i$ and $K^i = K_1^i + K_2^i$. Hence observer (2.42) for agent i con be constructed.
5. Get the corresponding residual $r_i(t)$ as in (2.53).
6. Implement the threshold logic as shown in (2.67). Check if $\|r_2(t)\| < \tau$: if yes, then the agent 2 is faulty and stop, else move to the next step.
7. Choose agent number 3, that is, select $i = 3$, repeat steps 2-5. Check whether $\|r_3(t)\| < \tau$: if yes, then agent 3 is faulty. Stop the algorithm. If not, proceed with another agent and repeat until $\exists k$, $\|r_k(t)\| < \tau$. Then agent k is faulty and stop.
8. If $\forall i = 2, \dots, N$, $\|r_i(t)\| > \tau$, then a fault affected agent 1.

2.2.3 Distributed implementation

As mentioned in the Introduction, being able to implement the algorithm, that will be applied to an MAS in a distributed way, has several advantages, such as the scalability property and the lower cost due to the smaller size of the implemented structures (such as observers). Based on the main result presented in [1], a distributed implementation of the previous result is proposed.

The observers design until now has required the collective relative output $\bar{W}(t)$, which is not distributed and needs a lot of information. Since we are dealing with an undirected connected graph (*Assumption 10*) for every choice of three nodes of the graph that are connected we have either one of the two elementary structures depicted in Fig.2.1.

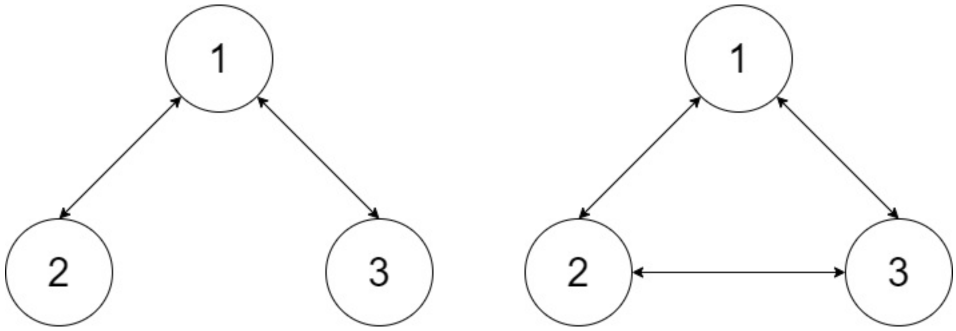


Figure 2.1: Two basic structure of tree connected node of an undirected graph.

In both cases it is possible to apply the FDI method shown before. Indeed two UIOs are built for agents 2 and 3, moreover the following logic threshold is presented in order to find if one of the tree agents has a malfunctioning:

$$\begin{cases} \|r_2\|(t) \geq \tau,\ \|r_3\|(t) \geq \tau \implies f_1(t) \neq 0 \quad \text{(Fault in 1);} \\ \|r_2\|(t) < \tau,\ \|r_3\|(t) \geq \tau \implies f_2(t) \neq 0 \quad \text{(Fault in 2);} \\ \|r_2\|(t) \geq \tau,\ \|r_3\|(t) < \tau \implies f_3(t) \neq 0 \quad \text{(Fault in 3);} \\ \|r_2\|(t) < \tau,\ \|r_3\|(t) < \tau \implies \text{fault-free.} \end{cases} \tag{2.68}$$

Where τ is the same threshold as in (2.67). Starting from this idea, it is possible to develop an FDI algorithm in a distributed manner for a network of $n \geq 3$ agents.

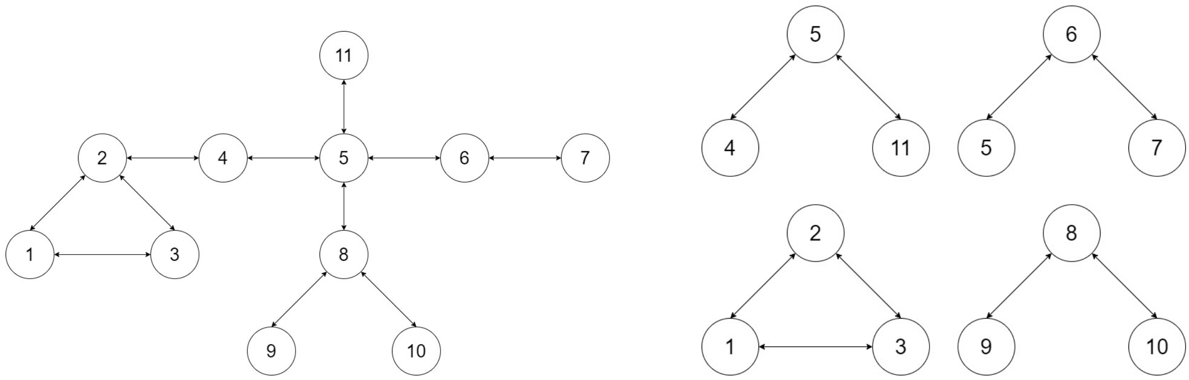
Algorithm 2 Distributed FDI for MAS with undirected topology.

1. Separate the whole system into several small groups consisting of 3 connected agents. This type of division is not necessarily a partition, in fact in two different groups there may be common nodes. The goal is to cover all agents with the minimum number of groups. Choose one group first.
2. Label the tree agents in the group with $1,2,3$. Solve equations (2.47)-(2.51) to get parameter matrices. Then construct the observer as (2.42) or (2.72) for agents 2 and 3. Get the residual and implement the Fault Detection and Isolation logic in (2.68). Check if there exist a faulty agent in this group, if yes then terminate the algorithm. Conversely, go on.
3. Select another group, return the above step until all agents have been checked and the fault has been isolated.

By the way we implemented *Algorithm 2*, the previous *Assumption 2* that imposes that only

one agent is faulty can be relaxed. This is due to the fact that, in this case, *Assumption 2* must hold only for the three-agent groups, and hence, for each group at most one fault can occur. This implies that, with this strategy, if more than one fault occurs in the whole system at each time we are still able to detected them, as far as there is a triple of agents for which a single agent is faulty.
Moreover it can be noticed that the number and the dimension of the UIO built for the task is drastically reduced. In fact, for each subset of three agents only two observers are built and their dimension is $2n$ (compared to the $N-1$ observers of dimension $(N-1)n$ each, built in the non-distributed scenario).

Example 1. *We show now an example on how to separate in groups a general undirected graph.*



(a) A general connected and undirected graph with $N = 11$ nodes.

(b) Four group of three node obtained performing the separation on the connected graph.

Figure 2.2: Example of separation procedure.

It can be noticed that some groups share common agents, since the way different groups are built does not represent a partition.

2.2.4 Directed graph scenario

For what concerns the scenario in which the Multi Agent System (MAS) has a communication network described by a directed graph the result does not change too much. This case is partially covered by [1], in particular the case in which the fault affects the state dynamics and the disturbance is not present (the effects of possible disturbance are accounted for in the choice of the threshold value). However, in addiction, the following assumption is made:

Assumption 12. The graph $\mathscr{G}$ of the MAS is directed and has a spanning tree.

If the setup of (1.20) is considered, the construction of an UIO does not change with respect to the one shown in the previous sections, and consequently also the evaluation of the residual signal will be identical.

2.3 Sensor Fault Detection and Isolation

In this section we try to apply the same techniques seen before to detect a fault in the sensors of the agents. In order to exclude the fault and perform the isolation it is necessary to neglect one output from the evaluation.

2.3.1 Residual signal generator for a fault of an agent sensor

In this subsection the case in which the fault affects one of the sensors of the agents is considered. To perform the computation, the model of Eq. (2.24) is taken into account by imposing that the actuator fault signal $\bar{F}_a(t)$ is identically zero at each time, namely the system is described by

$$\begin{aligned} \dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{uN-1}\bar{U}(t) + B_{dN-1}\bar{D}(t) \\ \bar{W}(t) &= (L_r \otimes C)\bar{X}(t) + (L_r \otimes I_{n_y})\bar{F}_s(t) \end{aligned} \tag{2.69}$$

Notice that also in this case the disturbance signal acts only on the state update equation, as in the previous case.

In order to detect the faulty agent, a smart representation of the Eq. (2.69) is derived; take the generic i^{th} agent and isolate its relative output $\bar{w}_i(t)$ from $\bar{W}(t)$ in order to obtain an equation of the type

$$\begin{aligned} \dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{uN-1}\bar{U}(t) + B_{dN-1}\bar{D}(t) \\ \bar{W}^{-i}(t) &= (L_r \otimes C)^{-i}\bar{X}(t) + (L_r \otimes I_{n_y})^{-i}\bar{F}_s(t) \\ \bar{W}^{i}(t) &= (L_r \otimes C)_i\bar{X}(t) + (L_r \otimes I_{n_y})_i\bar{F}_s(t) \end{aligned} \tag{2.70}$$

Where $(L_r \otimes C)_i \in \mathbb{R}^{n_y \times (N-1)n}$ is the block of rows of $(L_r \otimes C)$ with indices ranging from $(i-1)n_y+1$ to in_y and $(L_r \otimes C)^{-i} \in \mathbb{R}^{(N-2)n_y \times (N-1)n}$ is what remains of $(L_r \otimes C)$ after removing $(L_r \otimes C)^i$. It can be noticed that the matrices of the output equations can be rewritten as

$$\begin{aligned} (L_r \otimes C)^{-i} &= (L_r^{-i} \otimes C) \\ (L_r \otimes I_{n_y})^{-i} &= (L_r^{-i} \otimes I_{n_y}) \\ (L_r \otimes C)_i &= ([L_r]_i \otimes C) \\ (L_r \otimes I_{n_y})_i &= ([L_r]_i \otimes I_{n_y}) \end{aligned} \tag{2.71}$$

Where L_r^{-i} indicates the matrix L_r in which the i^{th} row was removed, instead $[L_r]_i$ indicates the i^{th} row of L_r.

In order to detect which agent's sensor is the faulty one, it is necessary to design a UIO for each agent $i = 2, \dots, N$. Following the reasoning about the construction of the observer previously discussed, a full order observer is described by

$$\begin{aligned}\dot{z}^i(t) &= J^i z^i(t) + M^i B_{uN-1}\tilde{U} + K^i \tilde{W}^{-i}(t) \\ \hat{X}^i(t) &= z^i(t) + H^i \tilde{W}^{-i}(t) \quad\quad i = 2, \ldots, N\end{aligned} \tag{2.72}$$

As previously done, we compute the error dynamics

$$\begin{aligned}\dot{e}^i(t) &= \dot{\bar{X}}(t) - \dot{\hat{X}}^i(t) \\ &= \dot{\bar{X}}(t) - (\dot{z}^i(t) + H^i \dot{\bar{W}}^{-i}(t)) \\ &= \dot{\bar{X}}(t) - (\dot{z}^i(t) + H^i((L_r^{-i} \otimes C)\dot{\bar{X}}(t) + (L_r^{-i} \otimes I_{n_y})^{-i}\dot{\bar{F}}_s(t)) \\ &= (I_{(N-1)n} - H^i(L_r^{-i} \otimes C))\dot{\bar{X}}(t) - \dot{z}^i(t) - H^i(L_r^{-i} \otimes I_{n_y})^{-i}\dot{\bar{F}}_s(t)\end{aligned} \tag{2.73}$$

By replacing the expressions (2.70) and (2.72) in the equation (2.73), we obtain

$$\begin{aligned}\dot{e}^i(t) &= (I_{(N-1)n} - H^i(L_r^{-i} \otimes C))(A_{N-1}\bar{X}(t) + B_{uN-1}\bar{U}(t) + B_{dN-1}\bar{D}(t)) \\ &- (J^i z^i(t) + M^i B_{uN-1}\bar{U} + K^i \bar{W}^{-i}(t)) - H^i(L_r^{-i} \otimes I_{n_y})\dot{\bar{F}}_s(t)\end{aligned} \tag{2.74}$$

Substitute, also in this case, the expression of K^i with $K_1^i + K_2^i$, and substitute the expression of $\bar{W}^{-i}(t)$ only for component that multiply matrix K_1^i

$$\begin{aligned}\dot{e}^i(t) &= ((I_{(N-1)n} - H^i(L_r^{-i} \otimes C))A_{N-1} - K_1^i(L_r^{-i} \otimes C))\bar{X}(t) \\ &+ ((I_{(N-1)n} - H^i(L_r^{-i} \otimes C) - M^i)B_{uN-1}\bar{U}(t) \\ &+ (I_{(N-1)n} - H^i(L_r^{-i} \otimes C))B_d\bar{D}(t) \\ &- J^i z^i(t) - K_1^i(L_r^{-i} \otimes I_{n_y})\bar{F}_s(t) - K_2^i \bar{W}^{-i}(t) - H^i(L_r^{-i} \otimes I_{n_y})\dot{\bar{F}}_s(t)\end{aligned} \tag{2.75}$$

Imposing the following constraints

$$(H^i(L_r^{-i} \otimes C) - I_{(N-1)n})B_{dN-1} = 0 \tag{2.76}$$

$$M^i = I_{(N-1)n} - H^i(L_r^{-i} \otimes C) \tag{2.77}$$

$$J^i = M^i A_{N-1} - K_1^i(L_r^{-i} \otimes C) \tag{2.78}$$

$$K_2^i = J^i H^i \tag{2.79}$$

$$K^i = K_1^i + K_2^i \tag{2.80}$$

the error dynamics of the i^{th} observer becomes

$$\begin{aligned}\dot{e}^i(t) &= (M^i A_{N-1} - K_1^i(L_r^{-i} \otimes C))\bar{X}(t) - J^i z^i(t) - K_2^i \bar{W}^{-i}(t) \\ &- K_1^i(L_r^{-i} \otimes I_{n_y})\bar{F}_s(t) - H^i(L_r^{-i} \otimes I_{n_y})\dot{\bar{F}}_s(t) \\ &= J^i(\bar{X}(t) - \hat{X}^i(t)) - K_1^i(L_r^{-i} \otimes I_{n_y})\bar{F}_s(t) - K_2^i \bar{W}^{-i}(t) - H^i(L_r^{-i} \otimes I_{n_y})\dot{\bar{F}}_s(t) \\ &= J^i e^i(t) - K_1^i(L_r^{-i} \otimes I_{n_y})\bar{F}_s(t) - K_2^i \bar{W}^{-i}(t) - H^i(L_r^{-i} \otimes I_{n_y})\dot{\bar{F}}_s(t)\end{aligned} \tag{2.81}$$

Finally, in order to ensure the convergence of the estimation error, in the case in which $F_s^{-i}(t)$ is identically zero, J^i must be stable for every $i = 2, \ldots, N$. Also in this case a residual signal is built for each agent $i = 2, \ldots, N$ as

$$\begin{aligned} r_s^i(t) &= \bar{W}^{-i} - (L_r^{-i} \otimes C)\hat{X}(t) \\ &= (L_r^{-i} \otimes C)\bar{X}(t) + (L_r^{-i} \otimes I_{n_y})\bar{F}_s(t) - (L_r^{-i} \otimes C)\bar{X}(t) \\ &= (L_r^{-i} \otimes C)e^i(t) + (L_r^{-i} \otimes I_{n_y})\bar{F}_s(t) \end{aligned} \tag{2.82}$$

In this case, it is easy to understand that the presence of a fault in one sensor of the network influences all the residuals with a combination of them and the isolation in this case is very difficult.
Now it is necessary to verify that the condition that ensures the existence of a UIO is satisfied. First of all, we must verify that $rank((L_r^{-i} \otimes C)B_{uN-1}) = rank(B_{uN-1})$. Developing the matrix $(L_r^{-i} \otimes C)B_{uN-1}$, we get that

$$\begin{aligned} (L_r^{-i} \otimes C)B_{uN-1} &= (L_r^{-i} \otimes C)(I_{N-1} \otimes B_u) \\ &= (L_r^{-i} \otimes CB_u) \end{aligned} \tag{2.83}$$

This implies that even if, by hypothesis $rank(CB_d) = rank(B_d)$, whenever the Kronecker product is performed with a matrix L_r^{-i}, the resulting matrix has always lower rank with respect to the matrix B_{dN-1}, and hence the first condition for the existence of a UIO for the system of Eq. (2.70) does not hold.

2.4 Sensor Fault Detection

Since, in the previous subsection it was shown that it is not possible to achieve the fault isolation, we try at least to find a way to ensure the Fault Detection (FD). To do this we consider the decoupled system of (2.31), namely

$$\begin{aligned} \dot{\tilde{X}}(t) &= A_{N-1}\tilde{X}(t) + B_{uN-1}\tilde{U}(t) + B_{dN-1}\tilde{D}(t) \\ \tilde{W}(t) &= C_{N-1}\tilde{X}(t) + I_{(N-1)n_y}\tilde{F}_s(t) \end{aligned} \tag{2.84}$$

and we build a UIO for the whole system, considering again as unknown input the disturbance $\tilde{D}(t)$.

First of all, we verify that the necessary and sufficient conditions given in *Theorem 1.2.2* are matched. in this case it is necessary to verify that

$$rank(C_{N-1}B_{dN-1}) = rank(B_{dN-1}) \tag{2.85}$$

By developing the computation, we get that

$$\begin{aligned} (C_{N-1}B_{dN-1}) &= (I_{N-1} \otimes C)(I_{N-1} \otimes B_d) \\ &= (I_{N-1} \otimes CB_d) \end{aligned} \tag{2.86}$$

Hence the equality is satisfied if and only if $rank(CB_d) = rank(B_d)$ as stated in *Assumption 7*.
For what concerns the second requirement, we must verify that the pair $(C_{N-1}, A_{N-1} - H^* C_{N-1} A_{N-1})$ is detectable, where $H^* = B_{dN-1}[(C_{N-1}B_{dN-1})^T C_{N-1}B_{dN-1}]^{-1}(C_{N-1}B_{dN-1})^T$. As shown before, the product $C_{N-1}B_{dN-1}$ can be written as $(I_{N-1} \otimes CB_d)$ and hence we get that:

$$\begin{aligned} H^* &= B_{dN-1}[(I_{N-1} \otimes CB_d)^T (I_{N-1} \otimes CB_d)]^{-1} (I_{N-1} \otimes CB_d)^T \\ &= B_{dN-1}[(I_{N-1} \otimes (CB_d)^T)(I_{N-1} \otimes CB_d)]^{-1} (I_{N-1} \otimes (CB_d)^T) \\ &= B_{dN-1}[(I_{N-1} \otimes (CB_d)^T CB_d)]^{-1} (I_{N-1} \otimes (CB_d)^T) \\ &= (I_{N-1} \otimes B_d)(I_{N-1} \otimes [(CB_d)^T CB_d)]^{-1})(I_{N-1} \otimes (CB_d)^T) \\ &= I_{N-1} \otimes (B_d[(CB_d)^T CB_d)]^{-1} (CB_d)^T) \end{aligned} \tag{2.87}$$

Before applying the PBH criterion A.0.2 in order to verify whether the pair $(C_{N-1}, A_{N-1} - H^* C_{N-1} A_{N-1})$ is detectable or not, define $H_1^* = B_d[(CB_d)^T CB_d)]^{-1}(CB_d)^T)$. Then the matrix H^* is equivalent to

$$H^* = (I_{N-1} \otimes H_1^*) \tag{2.88}$$

We need to verify if for $\Re(s) \geq 0$ the column rank of the following matrix is full:

$$\begin{aligned} &\begin{bmatrix} A_{N-1} - H^* C_{N-1} A_{N-1} - sI_{(N-1)n} \\ C_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} A_{N-1} - (I_{N-1} \otimes H_1^*) C_{N-1} A_{N-1} - sI_{(N-1)n} \\ C_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} (I_{N-1} \otimes A) - (I_{N-1} \otimes H_1^*)(I_{N-1} \otimes C)(I_{N-1} \otimes A) - s(I_{N-1} \otimes I_n) \\ I_{N-1} \otimes C \end{bmatrix} \\ &= \begin{bmatrix} I_{N-1} \otimes (A - H_1^* CA - sI_n) \\ I_{N-1} \otimes C \end{bmatrix} \end{aligned} \tag{2.89}$$

It is immediate to realise that the condition on the whole system holds if the pair $(A - H_1^* CA - sI_n, C)$ is detectable. By *Remark 4* this condition is equivalent to impose that the column rank of $\left[\begin{smallmatrix} sI_N - A & B_d \\ C & 0 \end{smallmatrix}\right]$ is full for $\Re(s) \geq 0$ and this is equivalent to *Assumption 5*.

Now the UIO for system (2.84) is presented. In order to detect which agent's sensor is the faulty one, it is necessary to design a UIO for each agent $i = 2, \ldots, N$. Following the reasoning about the construction of the observer previously discussed, a full order observer is described by

$$\begin{aligned} \dot{z}(t) &= Jz(t) + MB_{uN-1}\tilde{U} + K\tilde{W}(t) \\ \hat{X}^i(t) &= z(t) + H\tilde{W}(t) \end{aligned} \tag{2.90}$$

As previously done, we compute the error dynamics

$$\begin{aligned}\dot{e}_s(t) &= \dot{\tilde{X}}(t) - \dot{\hat{X}}(t) \\ &= \dot{\tilde{X}}(t) - (\dot{z}(t) + H\dot{\tilde{W}}(t)) \\ &= \dot{\tilde{X}}(t) - (\dot{z}(t) + H(C_{N-1}\dot{\tilde{X}}(t) + \dot{\tilde{F}}_s(t)) \\ &= (I_{(N-1)n} - HC_{N-1})\dot{\tilde{X}}(t) - \dot{z}(t) - H^i\dot{\tilde{F}}_s(t)\end{aligned} \tag{2.91}$$

By replacing the expressions (2.84) and (2.90) in the equation (2.91), we obtain

$$\begin{aligned}\dot{e}_s(t) = (I_{(N-1)n} - HC_{N-1})(A_{N-1}\tilde{X}(t) + B_{uN-1}\tilde{U}(t) + B_{dN-1}\tilde{D}(t)) \\ - (Jz(t) + MB_{uN-1}\tilde{U} + K\tilde{W}(t)) - H\dot{\tilde{F}}_s(t)\end{aligned} \tag{2.92}$$

Substitute, also in this case, the expression of K with $K_1 + K_2$, and develop the expression of $\tilde{W}(t)$ only for the matrix K_1

$$\begin{aligned}\dot{e}_s(t) = ((I_{(N-1)n} - HC_{N-1})A_{N-1} - K_1C_{N-1})\tilde{X}(t) \\ + (I_{(N-1)n} - HC_{N-1} - M)B_{uN-1}\tilde{U}(t) \\ + (I_{(N-1)n} - HC_{N-1})B_dD(t) \\ - Jz(t) - K_1\tilde{F}_s(t) - K_2\tilde{W}(t) - H\dot{\tilde{F}}_s(t)\end{aligned} \tag{2.93}$$

Imposing the following constraints

$$(HC_{N-1} - I_{(N-1)n})B_{dN-1} = 0 \tag{2.94}$$

$$M = I_{(N-1)n} - HC_{N-1} \tag{2.95}$$

$$J = MA_{N-1} - K_1C_{N-1} \tag{2.96}$$

$$K_2 = JH \tag{2.97}$$

$$K = K_1 + K_2 \tag{2.98}$$

the error dynamics of the observer becomes

$$\begin{aligned}\dot{e}_s(t) &= (MA_{N-1} - K_1C_{N-1})\tilde{X}(t) - Jz(t) - K_1\tilde{F}_s(t) - JH\tilde{W}(t) - H\dot{\tilde{F}}_s(t) \\ &= J(\tilde{X}(t) - \hat{X}(t)) - K_1\tilde{F}_s(t) - H\dot{\tilde{F}}_s(t) \\ &= Je(t) - K_1\tilde{F}_s(t) - H\dot{\tilde{F}}_s(t)\end{aligned} \tag{2.99}$$

Finally, in order to ensure the convergence of the estimation error, in the case in which $F_s(t)$ is identically zero, J must be stable. Then the residual signal is equal to

$$\begin{aligned}r_s(t) &= \tilde{W} - C_{N-1}\hat{X}(t) \\ &= C_{N-1}\tilde{X}(t) + \tilde{F}_s(t) - C_{N-1}\hat{X}(t) \\ &= C_{N-1}e_s(t) + \tilde{F}_s(t)\end{aligned} \tag{2.100}$$

by the way it has been previously defined $\tilde{F}_s(t)$ can be written as

$$\tilde{F}_s(t) = (V^{-1} \otimes I_n)\bar{F}_s(t) \tag{2.101}$$

where $\bar{f}_i^s = f_i^s - f_1^s(s)$.

2.4.1 Case of 3 agents

In the following, we try to analyze the case in which the system is composed only by three nodes, and verify if it is possible to perform the isolation of the fault knowing the topology of the graph.
If the graph is an undirected, connected, graph $\mathscr{G}$ composed by only $N = 3$ nodes is considered, four different types of connections between the agents exist, all described by the following adjacency matrices. In particular we can notice that there exists only one graph that has all the edges (i.e., $|E| = 3$), instead the other three graphs have $|E| = 2$.

$$A_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.102}$$

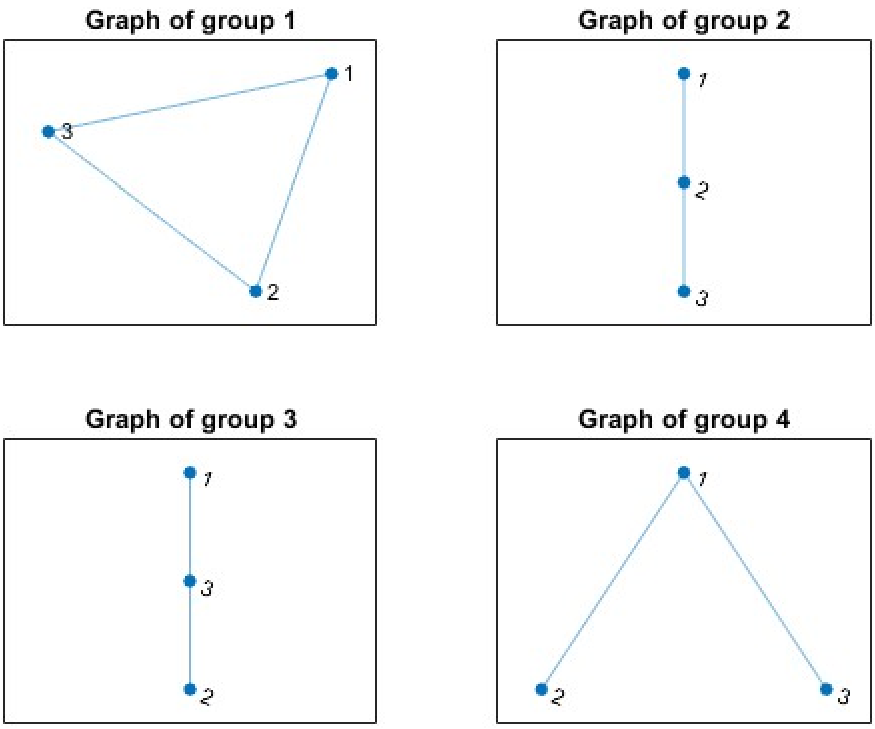


Figure 2.3: Undirected connected graph associated whit the adjacency matrices (2.102).

From the adjacency matrix (2.102), applying the definition of Laplacian matrix we obtain

$$
\begin{aligned}
L_1 &= \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad L_2 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \\
L_3 &= \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad L_4 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2.103}
$$

In order to extract the reduced Laplacian L_r from L, we define a change of coordinates transformation matrix T_s as in (2.33), obtaining

$$
\begin{aligned}
T_s^{-1}L_1T_s &= \begin{bmatrix} 0 & -1 & -1 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad T_s^{-1}L_2T_s = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 3 & -1 \\ 0 & 0 & 1 \end{bmatrix} \\
T_s^{-1}L_3T_s &= \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 1 & 3 \end{bmatrix} \quad T_s^{-1}L_4T_s = \begin{bmatrix} 0 & -1 & -1 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}
\end{aligned} \tag{2.104}
$$

Extracting the matrix L_r from the matrices in (2.104) we get

$$
Lr_1 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \quad Lr_2 = \begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix} \quad Lr_3 = \begin{bmatrix} 1 & 0 \\ 3 & -1 \end{bmatrix} \quad Lr_4 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{2.105}
$$

Remark 13. It can be noticed that the reduced Laplacian matrices Lr_2 e Lr_3 are substantially the same, and hence only one case will be considered.

The system considered is described by Eq. (2.34) in the case in which N=3, hence, the resulting system is

$$
\begin{aligned}
\dot{\bar{X}}(t) &= A_2\bar{X}(t) + B_{u2}\bar{U}(t) + B_{d2}\bar{D}(t) \\
\bar{W}(t) &= (L_r \otimes C)\bar{X}(t) + (L_r \otimes I_{n_y})\bar{F}_s(t)
\end{aligned} \tag{2.106}
$$

Note that differently from before we do not apply transformations to decouple the system; this is because the system is small and therefore it is quite simple to work with it. Since it will be valid for all the cases, a unique derivation of a UIO for the system (2.106) is performed.
The dynamics of the observer is the following

$$
\begin{aligned}
\dot{z}(t) &= Jz(t) + MB_{u2}\tilde{U} + K^i\bar{W}(t) \\
\hat{X}(t) &= z(t) + H\bar{W}(t)
\end{aligned} \tag{2.107}
$$

The existence of the UIO is guaranteed by *Assumptions 5 and 7.* Define the estimation error, as before, as

$$
e_s(t) = \bar{X}(t) - \hat{X}(t) \tag{2.108}
$$

Its update equation results to be

$$\begin{aligned}
\dot{e}_s(t) &= \dot{\bar{X}}(t) - \dot{\hat{X}}(t) \\
&= \dot{\bar{X}}(t) - (\dot{z}(t) + H\dot{\bar{W}}(t)) \\
&= \dot{\bar{X}}(t) - (\dot{z}(t) + H((L_r \otimes C)\dot{\bar{X}}(t) + (L_r \otimes I_{n_y})\dot{\bar{F}}_s(t)) \\
&= (I_{2n} - H(L_r \otimes C))\dot{\bar{X}}(t) - \dot{z}(t) - H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t))
\end{aligned} \tag{2.109}$$

By replacing the expression (2.106) and (2.107) in the equation (2.109), we obtain

$$\begin{aligned}
\dot{e}_s(t) = &(I_{2n} - H(L_r \otimes C))(A_2\bar{X}(t) + B_{u2}\bar{U}(t) + B_{d2}D(t)) \\
&- (Jz(t) + MB_{u2}\bar{U}(t) + K\bar{W}(t)) - H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t)
\end{aligned} \tag{2.110}$$

Substitute, now, the expression of K with $K_1 + K_2$, and develop the expression of $\bar{W}(t)$ only for the matrix K_1

$$\begin{aligned}
\dot{e}_s(t) = &[(I_{2n} - H(L_r \otimes C))A_2 - K_1(L_r \otimes C)]\bar{X}(t) \\
&+ (I_{2n} - H(L_r \otimes C) - M)B_{u2}\bar{U}(t) \\
&+ (I_{2n} - H(L_r \otimes C))B_{d2}D(t) \\
&- Jz(t) - K_1(L_r \otimes I_{n_y})\bar{F}_s(t) - K_2\bar{W}(t) \\
&- H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t)
\end{aligned} \tag{2.111}$$

Moreover, if the following matrix relations are imposed for

$$(H(L_r \otimes C) - I_{2n})B_d = 0 \tag{2.112}$$
$$M = I_{2n} - H(L_r \otimes C) \tag{2.113}$$
$$J = MA_2 - K_1(L_r \otimes C) \tag{2.114}$$
$$K_2 = JH \tag{2.115}$$
$$K = K_1 + K_2 \tag{2.116}$$

the error dynamics of the i^{th} observer becomes

$$\begin{aligned}
\dot{e}_s(t) &= (MA_2 - K_1(L_r \otimes C))\bar{X}(t) - Jz(t) - JH\bar{W}(t) \\
&\quad - K_1(L_r \otimes I_{n_y})\bar{F}_s(t) - H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t) \\
&= J(\bar{X}(t) - \hat{X}(t)) - K_1(L_r \otimes I_{n_y})\bar{F}_s(t) - H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t) \\
&= Je_s(t) - K_1(L_r \otimes I_{n_y})\bar{F}_s(t) - H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t)
\end{aligned} \tag{2.117}$$

Define, finally, the residual associated to the observer as

$$\begin{aligned}
r_s(t) &= \bar{W} - (L_r \otimes C)\hat{X}(t) \\
&= (L_r \otimes C)\bar{X}(t) + (L_r \otimes I_{n_y})\bar{F}_s(t) - (L_r \otimes C)\hat{X}(t) \\
&= (L_r \otimes C)e^i(t) + (L_r \otimes I_{n_y})\bar{F}_s(t)
\end{aligned} \tag{2.118}$$

To sum up the error dynamics and the residual are equal to

$$\begin{aligned} \dot{e}_s(t) &= Je_s(t) - K_1(L_r \otimes I_{n_y})\bar{F}_s(t) - H(L_r \otimes I_{n_y})\dot{\bar{F}}_s(t) \\ r_s(t) &= (L_r \otimes C)e_s(t) + (L_r \otimes I_{n_y})\bar{F}_s(t) \end{aligned} \tag{2.119}$$

Case $L_r = Lr_1$

Consider the matrix Lr_1;

$$Lr_1 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \tag{2.120}$$

Upon expanding Eq. (2.119) and substituting the expression of Lr_1, the error dynamics and the residual generated by the UIO are equal to

$$\begin{aligned} \dot{e}_s(t) &= Je_s(t) - K_1\left(\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \otimes I_{n_y}\right)\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - H\left(\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \otimes I_{n_y}\right)\begin{bmatrix} \dot{\bar{f}}_2^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\ &= Je_s(t) - 3K_1\begin{bmatrix} I_{n_y} & 0 \\ 0 & I_{n_y} \end{bmatrix}\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - 3H\begin{bmatrix} I_{n_y} & 0 \\ 0 & I_{n_y} \end{bmatrix}\begin{bmatrix} \dot{\bar{f}}_2^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\ &= Je_s(t) - 3K_1\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - 3H\begin{bmatrix} \dot{\bar{f}}_2^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \end{aligned} \tag{2.121}$$

$$\begin{aligned} r_s(t) &= \left(\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \otimes C\right)e^i(t) + \left(\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \otimes I_{n_y}\right)\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} \\ &= 3\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}e_s(t) + 3\begin{bmatrix} I_{n_y} & 0 \\ 0 & I_{n_y} \end{bmatrix}\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} \\ &= 3\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}e_s(t) + 3\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} \end{aligned} \tag{2.122}$$

It is possible to distinguish four cases, depending on where the fault is located

1. No Fault

$$\begin{cases} \dot{e}_s(t) = Je_s(t) \to 0 \\ r_s(t) = 3\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}e_s(t) \to 0 \end{cases} \tag{2.123}$$

2. Fault in agent 1

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - 3K_1\begin{bmatrix} -f_1^s(t) \\ -f_1^s(t) \end{bmatrix} - 3H\begin{bmatrix} -\dot{f}_1^s(t) \\ -\dot{f}_1^s(t) \end{bmatrix} \\ r_s(t) = 3\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}e_s(t) + 3\begin{bmatrix} -f_1^s(t) \\ -f_1^s(t) \end{bmatrix} \end{cases} \tag{2.124}$$

3. Fault in agent 2

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - 3K_1 \begin{bmatrix} f_2^s(t) \\ 0 \end{bmatrix} - 3H \begin{bmatrix} \dot{f}_2^s(t) \\ 0 \end{bmatrix} \\ r_s(t) = 3 \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} e_s(t) + 3 \begin{bmatrix} f_2^s(t) \\ 0 \end{bmatrix} \end{cases} \tag{2.125}$$

4. Fault in agent 3

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - 3K_1 \begin{bmatrix} 0 \\ f_3^s(t) \end{bmatrix} - 3H \begin{bmatrix} 0 \\ \dot{f}_3^s(t) \end{bmatrix} \\ r_s(t) = 3 \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} e_s(t) + 3 \begin{bmatrix} 0 \\ f_3^s(t) \end{bmatrix} \end{cases} \tag{2.126}$$

For this particular case ($L_r = Lr_1$), we try to describe the dynamics of the error $e_s(t)$ isolating each single component, namely $e_2^s(t) = \bar{x}_2(t) - \hat{x}_2(t)$ and $e_3^s(t)\bar{x}_3(t) - \hat{x}_3(t)$.

Consider the construction of the matrix J. From Eq. (2.114) we have that

$$J = MA_2 - K_1(L_r \otimes C) = [I_{2n} - H(L_r \otimes C)]A_2 - K_1(L_r \otimes C) \tag{2.127}$$

Considering the case Lr_1, a particular solution of H is H^* described by (2.88); substituting it in the previous equation we get

$$\begin{aligned} J &= [I_{2n} - (I_2 \otimes (B_d[(CB_d)^T CB_d)]^{-1}(CB_d)^T))(I_2 \otimes C)]A_2 - K_1(I_2 \otimes C) \\ &= [(I_2 \otimes I_n) - (I_2 \otimes H_1^*)(I_2 \otimes C)](I_2 \otimes A) - K_1(I_2 \otimes C) \\ &= (I_2 \otimes (I_n - H_1^*C)A) - K_1(I_2 \otimes C) \end{aligned} \tag{2.128}$$

Since J is block diagonal, and there exists K_1 such that J is stable, this implies that all the eigenvalue of J have real part smaller than zero. It is sufficient then that both blocks on the diagonal have stable eigenvalues. Since the two blocks are identical then there exist a controller taking the form $K_1 = I_2 \otimes K^*$ that ensures this result. Hence

$$\begin{aligned} J &= (I_2 \otimes (I_n - H_1^*C)A) - K_1(I_2 \otimes C) \\ &= (I_2 \otimes (I_n - H_1^*C)A) - (I_2 \otimes K^*)(I_2 \otimes C) \\ &= I_2 \otimes ((I_n - H_1^*C)A - K^*C) \end{aligned} \tag{2.129}$$

Where $J^* = (I_n - H_1^*C)A - K^*C \in \mathbb{R}^{n\times n}$ is Hurwitz stable. Since the matrix J is block diagonal, it is possible to decouple Eq. (2.121), and distinguish the two components of the error

$$\begin{aligned}
\dot{e}_s(t) &= Je_s(t) - 3K_1 \begin{bmatrix} \bar{f}_2(t) \\ \bar{f}_3^s(t) \end{bmatrix} - 3H \begin{bmatrix} \dot{\bar{f}}_2(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\
&= (I_2 \otimes J^*) \begin{bmatrix} e_2^s(t) \\ e_3^s(t) \end{bmatrix} - 3(I_2 \otimes K^*) \begin{bmatrix} \bar{f}_2(t) \\ \bar{f}_3^s(t) \end{bmatrix} - 3(I_2 \otimes H_1^*) \begin{bmatrix} \dot{\bar{f}}_2(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\
&= \begin{bmatrix} J^* e_2^s(t) - 3K^* \bar{f}_2(t) - 3H_1^* \dot{\bar{f}}_2(t) \\ J^* e_3^s(t) - 3K^* \bar{f}_3^s(t) - 3H_1^* \dot{\bar{f}}_3(t) \end{bmatrix}
\end{aligned} \tag{2.130}$$

with this new definition of the error dynamics, we rewrite also the residual

$$\begin{bmatrix} r_2^s(t) \\ r_2^s(t) \end{bmatrix} = 3 \begin{bmatrix} Ce_2^s(t) + \bar{f}_2(t) \\ Ce_3^s(t) + \bar{f}_3^s(t) \end{bmatrix} \tag{2.131}$$

So, it is possible to decouple the system and rewrite the conditions for which the error dynamics and the residual goes to zero or tends to a value different from zero.

1. No Fault

$$\begin{cases} \dot{e}_2^s(t) = J^* e_2^s(t) \to 0 \\ \dot{e}_3^s(t) = J^* e_3^s(t) \to 0 \\ r_2^s(t) = 3Ce_2^s(t) \to 0 \\ r_3^s(t) = 3Ce_3^s(t) \to 0 \end{cases} \tag{2.132}$$

2. Fault in agent 1

$$\begin{cases} \dot{e}_2^s(t) = J^* e_2^s(t) + 3K^* f_1^s(t) + 3H_1^* \dot{f}_1^s(t) \\ \dot{e}_3^s(t) = J^* e_3^s(t) + 3K^* f_1^s(t) + 3H_1^* \dot{f}_1^s(t) \\ r_2^s(t) = 3Ce_2^s(t) - 3f_1^s(t) \\ r_3^s(t) = 3Ce_3^s(t) - 3f_1^s(t) \end{cases} \tag{2.133}$$

3. Fault in agent 2

$$\begin{cases} \dot{e}_2^s(t) = J^* e_2^s(t) - 3K^* f_2^s(t) - 3H_1^* \dot{f}_2^s(t) \\ \dot{e}_3^s(t) = J^* e_3^s(t) \to 0 \\ r_2^s(t) = 3Ce_2^s(t) + 3f_2^s(t) \\ r_3^s(t) = 3Ce_3^s(t) \to 0 \end{cases} \tag{2.134}$$

4. Fault in agent 3

$$\begin{cases} \dot{e}_2^s(t) = J^* e_2^s(t) \to 0 \\ \dot{e}_3^s(t) = J^* e_3^s(t) - 3K^* f_3^s(t) - 3H_1^* \dot{f}_3^s(t) \\ r_2^s(t) = 3Ce_2^s(t) \to 0 \\ r_3^s(t) = 3Ce_3^s(t) + 3f_3^s(t) \end{cases} \tag{2.135}$$

And hence in this case it is possible to isolate the fault by evaluating the residuals generated by the UIO. In particular, introduce the following logic threshold

$$\begin{cases} \|r_2^s\|(t) \geq \tau,\ \|r_3^s\|(t) \geq \tau \implies f_1^s(t) \neq 0 & \text{(Fault in 1);} \\ \|r_2^s\|(t) > \tau,\ \|r_3^s\|(t) \leq \tau \implies f_2^s(t) \neq 0 & \text{(Fault in 2);} \\ \|r_2^s\|(t) \leq \tau,\ \|r_3^s\|(t) > \tau \implies f_3^s(t) \neq 0 & \text{(Fault in 3);} \\ \|r_2^s\|(t) < \tau,\ \|r_3^s\|(t) < \tau \implies \text{fault-free.} \end{cases} \tag{2.136}$$

where τ is the threshold value.

In this particular case ($L = L_1$) it was shown that also isolation is possible. Looking at the previous computation for the case Lr_1, it is possible to see that the isolation is possible in the cases in which the reduced Laplacian L_r obtained by Eq. (2.33) is diagonal. Starting from this assumption, we want to find what is the structure of the graph that leads to this type of reduced matrix L_r.

Starting from $L_r = NI_{N-1}$, where N is the number of nodes, it holds that

$$T_s^{-1} L T_s = \begin{bmatrix} 0 & l^T \\ 0 & NI_{N-1} \end{bmatrix} \tag{2.137}$$

it follows that

$$\begin{aligned} L &= T_s \begin{bmatrix} 0 & l^T \\ 0 & NI_{N-1} \end{bmatrix} T_s^{-1} \\ &= \begin{bmatrix} 1 & \mathbb{0}_{N-1}^T \\ \mathbb{1}_{N-1} & I_{N-1} \end{bmatrix} \begin{bmatrix} 0 & l^T \\ 0 & \alpha I_{N-1} \end{bmatrix} \begin{bmatrix} 1 & \mathbb{0}_{N-1}^T \\ -\mathbb{1}_{N-1} & I_{N-1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & \mathbb{0}_{N-1}^T \\ \mathbb{1}_{N-1} & I_{N-1} \end{bmatrix} \begin{bmatrix} -l^T \mathbb{1}_{N-1} & l^T \\ -N\mathbb{1}_{N-1} & NI_{N-1} \end{bmatrix} \\ &= \begin{bmatrix} -l^T \mathbb{1}_{N-1} & l_2 & l_3 & \dots & l_n \\ (-l^T \mathbb{1}_{N-1} - N) & l_2 + N & l_3 & \dots & l_n \\ (-l^T \mathbb{1}_{N-1} - N) & l_2 & l_3 + N & \dots & l_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-l^T \mathbb{1}_{N-1} - N) & l_2 & l_3 & \dots & l_n + N \end{bmatrix} \end{aligned} \tag{2.138}$$

From the Laplacian matrix L, it is possible to obtain the adjacency matrix A assuming $a_{ii} = 0\ \forall i$ and $a_{ij} = -l_{ij}$

$$A = \begin{bmatrix} 0 & -l_2 & -l_3 & \dots & -l_n \\ (l^T \mathbb{1}_{N-1} + N) & 0 & -l_3 & \dots & -l_n \\ (l^T \mathbb{1}_{N-1} + N) & -l_2 & 0 & \dots & -l_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (l^T \mathbb{1}_{N-1} + N) & -l_2 & -l_3 & \dots & 0 \end{bmatrix} \tag{2.139}$$

Since the graph is undirected the adjacency matrix A is symmetric, and hence it can be written as

$$A = \begin{bmatrix} 0 & -l_2 & -l_3 & \dots & -l_n \\ -l_2 & 0 & -l_3 & \dots & -l_n \\ -l_3 & -l_2 & 0 & \dots & -l_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -l_n & -l_2 & -l_3 & \dots & 0 \end{bmatrix} \tag{2.140}$$

And since by *Assumption 10* the graph is connected, each l_i must be different from zero. If by absurd this was not true, take for example $l_2 = 0$, this would imply that the second column of the adjacency matrix is equal to zero, and hence, by definition of adjacency matrix, there would be no edge that connects the agent 2 to another one, and this contradicts the assumption that the graph is connected.
Since we have deal with a binary adjacency matrix, each $l_i = -1\ \forall i$, and we get

$$A = \begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \tag{2.141}$$

Namely the graph with all the possible connections among the agents.

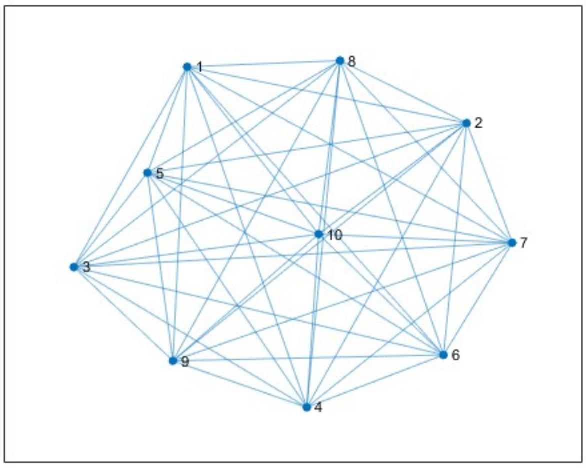


Figure 2.4: Example of an undirected graph with 10 nodes and all the possible edges.

With the same objective that motivated this computation, that is to see if it is possible to isolate the faulty agent, the other two cases can be verified.

Case $L_r = Lr_2$

Consider the matrix Lr_2

$$Lr_2 = \begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix} \tag{2.142}$$

Expand Eq. (2.119) and substitute the expression of Lr_2, in order to obtain the error dynamics and the residual generated by the UIO

$$\begin{aligned}
\dot{e}_s(t) &= Je_s(t) - K_1\left(\begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix} \otimes I_{n_y}\right)\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - H\left(\begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix} \otimes I_{n_y}\right)\begin{bmatrix} \dot{\bar{f}}_2^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\
&= Je_s(t) - K_1\begin{bmatrix} 3I_{n_y} & -I_{n_y} \\ 0 & I_{n_y} \end{bmatrix}\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - H\begin{bmatrix} 3I_{n_y} & -I_{n_y} \\ 0 & I_{n_y} \end{bmatrix}\begin{bmatrix} \dot{\bar{f}}_2^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\
&= Je_s(t) - K_1\begin{bmatrix} 3\bar{f}_2^s(t) - \bar{f}_3^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - H\begin{bmatrix} 3\dot{\bar{f}}_2^s(t) - \dot{\bar{f}}_3^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix}
\end{aligned} \tag{2.143}$$

$$\begin{aligned}
r_s(t) &= \left(\begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix} \otimes C\right)e_s(t) + \left(\begin{bmatrix} 3 & -1 \\ 0 & 1 \end{bmatrix} \otimes I_{n_y}\right)\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} \\
&= \begin{bmatrix} 3C & -C \\ 0 & C \end{bmatrix} e_s(t) + \begin{bmatrix} 3I_{n_y} & -I_{n_y} \\ 0 & I_{n_y} \end{bmatrix}\begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} \\
&= \begin{bmatrix} 3C & -C \\ 0 & C \end{bmatrix} e_s(t) + \begin{bmatrix} 3\bar{f}_2^s(t) - \bar{f}_3^s(t) \\ \bar{f}_3^s(t) \end{bmatrix}
\end{aligned} \tag{2.144}$$

Also in this case it is possible to distinguish four cases, associated with the sensors faults in the system.

1. No Fault

$$\begin{cases} \dot{e}_s(t) = Je_s(t) \to 0 \\ r_s(t) = \begin{bmatrix} 3C & -C \\ 0 & C \end{bmatrix} e_s(t) \to 0 \end{cases} \tag{2.145}$$

2. Fault in agent 1

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - K_1\begin{bmatrix} -2f_1^s(t) \\ -f_1^s(t) \end{bmatrix} - H\begin{bmatrix} -2\dot{f}_1^s(t) \\ -\dot{f}_1^s(t) \end{bmatrix} \\ r_s(t) = \begin{bmatrix} 3C & -C \\ 0 & C \end{bmatrix} e_s(t) + \begin{bmatrix} -2f_1^s(t) \\ -f_1^s(t) \end{bmatrix} \end{cases} \tag{2.146}$$

3. Fault in agent 2

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - K_1 \begin{bmatrix} 3f_2^s(t) \\ 0 \end{bmatrix} - H \begin{bmatrix} 3\dot{f}_2^s(t) \\ 0 \end{bmatrix} \\ r_s(t) = \begin{bmatrix} 3C & -C \\ 0 & C \end{bmatrix} e_s(t) + \begin{bmatrix} 3f_2^s(t) \\ 0 \end{bmatrix} \end{cases} \tag{2.147}$$

4. Fault in agent 3

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - K_1 \begin{bmatrix} -f_3^s(t) \\ f_3^s(t) \end{bmatrix} - H \begin{bmatrix} -\dot{f}_3^s(t) \\ \dot{f}_3^s(t) \end{bmatrix} \\ r_s(t) = \begin{bmatrix} 3C & -C \\ 0 & C \end{bmatrix} e_s(t) + \begin{bmatrix} -f_3^s(t) \\ f_3^s(t) \end{bmatrix} \end{cases} \tag{2.148}$$

In this scenario, as expected, it is possible to perform the FD since the only case in which the residual converges to zero is the one in which there is no fault.
For what concerns isolation, instead, it is possible to perform it only when the fault affects the second agent; in fact, in this case, the first component of the residual r_2^s is different from zero, instead r_3^s tends to zero as $t \to \infty$.
It can be noticed that, if the mirror case was considered, namely $L_r = Lr_3$, we would have the same behaviour but associated with the fault on the third agent, namely $r_2^s \to 0$ and r_3^s different from zero.

To conclude, with the structures of types 2 and 3 as in Fig. 2.3, it is possible to isolate the fault only if it affects the agent that shares both edges with the other nodes, (i.e. the one in the middle).

Case $L_r = Lr_4$

Consider now the matrix Lr_4

$$Lr_4 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{2.149}$$

Expand Eq. (2.119) and substitute the expression of Lr_4 in order to obtain the error dynamics and the residual associated to the UIO

$$\begin{aligned} \dot{e}_s(t) &= Je_s(t) - K_1 \left(\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \otimes I_{n_y} \right) \begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - H \left(\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \otimes I_{n_y} \right) \begin{bmatrix} \dot{\bar{f}}_2^s(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\ &= Je_s(t) - K_1 \begin{bmatrix} 2I_{n_y} & I_{n_y} \\ I_{n_y} & 2I_{n_y} \end{bmatrix} \begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} - H \begin{bmatrix} 2I_{n_y} & I_{n_y} \\ I_{n_y} & 2I_{n_y} \end{bmatrix} \begin{bmatrix} \dot{\bar{f}}_2(t) \\ \dot{\bar{f}}_3^s(t) \end{bmatrix} \\ &= Je_s(t) - K_1 \begin{bmatrix} 2\bar{f}_2^s(t) + \bar{f}_3^s(t) \\ \bar{f}_2^s(t) + 2\bar{f}_3^s(t) \end{bmatrix} - H \begin{bmatrix} 2\dot{\bar{f}}_2^s(t) + \dot{\bar{f}}_3^s(t) \\ \dot{\bar{f}}_2^s(t) + 2\dot{\bar{f}}_3^s(t) \end{bmatrix} \end{aligned} \tag{2.150}$$

$$\begin{aligned} r_s(t) &= \left(\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \otimes C \right) e^i(t) + \left(\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \otimes I_{n_y} \right) \begin{bmatrix} \bar{f}_2^s(t) \\ \bar{f}_3^s(t) \end{bmatrix} \\ &= \begin{bmatrix} 2C & C \\ C & 2C \end{bmatrix} e_s(t) + \begin{bmatrix} 2I_{n_y} & I_{n_y} \\ I_{n_y} & 2I_{n_y} \end{bmatrix} \begin{bmatrix} \bar{f}_2(t) \\ \bar{f}_3^s(t) \end{bmatrix} \\ &= \begin{bmatrix} 2C & C \\ C & 2C \end{bmatrix} e_s(t) + \begin{bmatrix} 2\bar{f}_2^s(t) + \bar{f}_3^s(t) \\ \bar{f}_2^s(t) + 2\bar{f}_3^s(t) \end{bmatrix} \end{aligned} \tag{2.151}$$

Also in this last scenario, it is possible to distinguish four cases corresponding to the different sensor fault locations

1. No Fault

$$\begin{cases} \dot{e}_s(t) = Je_s(t) \to 0 \\ r_s(t) = \begin{bmatrix} 2C & C \\ C & 2C \end{bmatrix} e_s(t) \to 0 \end{cases} \tag{2.152}$$

2. Fault in agent 1

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - K_1 \begin{bmatrix} -3f_1^s(t) \\ -3f_1^s(t) \end{bmatrix} - H \begin{bmatrix} -3\dot{f}_1^s(t) \\ -3\dot{f}_1^s(t) \end{bmatrix} \\ r_s(t) = \begin{bmatrix} 2C & C \\ C & 2C \end{bmatrix} e_s(t) + \begin{bmatrix} -3f_1^s(t) \\ -3f_1^s(t) \end{bmatrix} \end{cases} \tag{2.153}$$

3. Fault in agent 2

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - K_1 \begin{bmatrix} 2f_2^s(t) \\ f_2^s(t) \end{bmatrix} - H \begin{bmatrix} 2\dot{f}_2^s(t) \\ \dot{f}_2^s(t) \end{bmatrix} \\ r_s(t) = \begin{bmatrix} 2C & C \\ C & 2C \end{bmatrix} e_s(t) + \begin{bmatrix} 2f_2^s(t) \\ f_2^s(t) \end{bmatrix} \end{cases} \tag{2.154}$$

4. Fault in agent 3

$$\begin{cases} \dot{e}_s(t) = Je_s(t) - K_1 \begin{bmatrix} f_3^s(t) \\ 2f_3^s(t) \end{bmatrix} - H \begin{bmatrix} \dot{f}_3^s(t) \\ 2\dot{f}_3^s(t) \end{bmatrix} \\ r_s(t) = \begin{bmatrix} 2C & C \\ C & 2C \end{bmatrix} e_s(t) + \begin{bmatrix} f_3^s(t) \\ 2f_3^s(t) \end{bmatrix} \end{cases} \tag{2.155}$$

In this case, as before, FD is possible since the residual tends to zero only when there is no fault, and in the other cases instead it tends to a value different from zero. Looking at this kind of result we cannot say where the fault in the system, is located namely it is no possible to perform isolation. If instead of node 1, we choose one of the other node to perform the reduced Laplacian L_r, we would have a structure like the one of the previous two examples (Lr_2 and Lr_3), and hence the fault in the node 1 could be isolated.

Chapter 3

Consensus problem

In this chapter the synchronization problem for a MAS with a leader is considered. In the first part a distributed approach that makes use of a priori information about the structure of the network is presented. Subsequently, an adaptive approach that gets rid of the a priori information is proposed.

3.1 Synchronization problem

For this problem we consider only the case of directed graph topology. For simplicity the case when the output coincides with the state, namely $C = I_n$, is considered. This choice is made in order to consider also a possible fault in the agents' sensors. In paper [10] and in *Chapter 8* of [2] a solution to the synchronization problem is presented under a milder assumption, that

Assumption 13. The graph $\mathscr{G}$ contains a directed spanning tree with the leader as the root node.

Consider N identical agents with the same dynamics as before, namely

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + B_u u_i(t) + B_d d_i(t) + B_u f_i^a(t), \\ y_i(t) &= x_i(t) + f_i^s(t), \quad i = 1, \ldots, N, \end{aligned} \tag{3.1}$$

Since the $C = I_n$ and there exists a directed spanning tree for the graph $\mathscr{G}$, it is possible to build an FDI scheme in order to detect the faults and implement it also in a distributed way.
The condensed system, represented via Kronecker product, is

$$\begin{aligned} \dot{X}(t) &= A_N X(t) + B_{uN}(U(t) + F_a(t)) + B_{dN} D(t), \\ Y(t) &= X(t) + F_s(t), \\ W(t) &= (L \otimes C) X(t) + (L \otimes I_{n_y}) F_s(t) \end{aligned} \tag{3.2}$$

Assume that the agent with label 1 is the leader. It is assumed that the control input of the leader is zero, i.e. $u_1(t) = 0$. As done in *Chapter 2* we define a change of coordinates transformation matrix T^{-1} in order to have $X \rightarrow T^{-1}X = X_o$, where

$$T^{-1} := T_s^{-1} \otimes I_n \quad and \quad T_s^{-1} = \begin{bmatrix} 1 & \mathbb{0}_{N-1}^T \\ -\mathbb{1}_{N-1} & I_{N-1} \end{bmatrix} \tag{3.3}$$

Applying the transformation T^{-1} to the vector state X yields

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \rightarrow \quad T^{-1}X = X_o = \begin{bmatrix} x_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_N \end{bmatrix} = \begin{bmatrix} x_1 \\ \bar{X} \end{bmatrix} \tag{3.4}$$

And we obtain the same state space representation as in Eq. (2.23):

$$\begin{aligned} \dot{X}_o(t) &= A_N X_o(t) + B_{uN} U_o(t) + B_{uN} F_a^o(t) + B_{dN} D_o(t), \\ Y_o(t) &= X_o(t) + F_s^o(t), \\ W_o(t) &= (T_s^{-1} L T_s \otimes I_n)(X_o(t) + F_s^o(t)) \end{aligned} \tag{3.5}$$

From the definition of T_s and the fact that $L\mathbb{1}_N = 0$ by *Lemma 1.3.1*, it is easy to check that

$$T_s^{-1} L T_s = \begin{bmatrix} 0 & l^T \\ 0 & L_r \end{bmatrix} \tag{3.6}$$

where $l \in \mathbb{R}^{1 \times N-1}$ and $L_r \in \mathbb{R}^{(N-1)\times(N-1)}$. Since $\mathscr{G}$ satisfies *Assumption 13*, it follows by *Theorem 1.3.3* and *Lemma 1.3.4* that all the eigenvalues of L, except $\lambda_1 = 0$, have positive real parts. Expanding Eq. (3.5) using the representation of the Laplacian (3.6), we obtain:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1(t) \\ \dot{\bar{X}}(t) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ 0 & A_{N-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ \bar{X}(t) \end{bmatrix} + \begin{bmatrix} B_u & 0 \\ 0 & B_{u_{N-1}} \end{bmatrix} \left(\begin{bmatrix} u_1(t) \\ \bar{U}(t) \end{bmatrix} + \begin{bmatrix} f_1^a(t) \\ \bar{F}_s(t) \end{bmatrix} \right) \\ &\quad + \begin{bmatrix} B_d & 0 \\ 0 & B_{d_{N-1}} \end{bmatrix} \begin{bmatrix} d_1(t) \\ \bar{D}(t) \end{bmatrix}, \\ \begin{bmatrix} y_1(t) \\ \bar{Y}(t) \end{bmatrix} &= \begin{bmatrix} x_1(t) \\ \bar{X}(t) \end{bmatrix} + \begin{bmatrix} f_1^s(t) \\ \bar{F}_s(t) \end{bmatrix}, \\ \begin{bmatrix} w_1(t) \\ \bar{W}(t) \end{bmatrix} &= \left(\begin{bmatrix} 0 & l \\ 0 & L_r \end{bmatrix} \otimes I_n \right) \left(\begin{bmatrix} x_1(t) \\ \bar{X}(t) \end{bmatrix} + \begin{bmatrix} f_1^s(t) \\ \bar{F}_s(t) \end{bmatrix} \right) \end{aligned} \tag{3.7}$$

Define the convergence error as $\varepsilon_i(t) = \|x_i(t) - x_1(t)\|$. Solving the synchronization problem in a leader-follower setting is equivalent to ensuring that

$$\lim_{t\to\infty} \varepsilon_i(t) = \lim_{t\to\infty} \|x_i(t) - x_1(t)\| = 0, \quad \forall i,\ i = 2, \ldots, N \tag{3.8}$$

In this case, since we have defined $\bar{x}_i(t) = x_i(t) - x_1(t)$, solving the problem is equal to ensuring that $\|\bar{X}(t)\| \rightarrow 0$ as $t \rightarrow \infty$.

A static consensus protocol based on the relative state between neighboring agents is

$$u_i(t) = c\tilde{K}\sum_{j=1}^{N} a_{ij}(y_i(t) - y_j(t)), \quad \forall i,\ i = 2,\ldots,N \tag{3.9}$$

where $c > 0$ is a common coupling weight among neighboring agents, $\tilde{K} \in \mathbb{R}^{n_u \times N}$ is the feedback gain matrix. It can be noticed that the above control law is equal to imposing

$$\begin{bmatrix} u_1(t) \\ \bar{U}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ -(I_{N-1} \otimes c\tilde{K})\bar{W}(t) \end{bmatrix} \tag{3.10}$$

Substituting the expression of $\bar{U}(t)$ in the equation of $\bar{X}(t)$ we get:

$$\begin{aligned}
\dot{\bar{X}}(t) &= A_{N-1}\bar{X}(t) + B_{uN-1}(-(I_{N-1} \otimes c\tilde{K})\bar{W}(t) + \bar{F}_a(t)) + B_{dN-1}\bar{D}(t) \\
&= A_{N-1}\bar{X}(t) + B_{uN-1}(-(I_{N-1} \otimes c\tilde{K})(L_r \otimes I_n)(\bar{X}(t) + \bar{F}_s(t)) + \bar{F}_a(t)) \\
&+ B_{dN-1}\bar{D}(t) \\
&= (A_{N-1} - B_{uN-1}(I_{N-1} \otimes c\tilde{K})(L_r \otimes I_n))\bar{X}(t) \\
&- B_{uN-1}(I_{N-1} \otimes c\tilde{K})(L_r \otimes I_n)\bar{F}_s(t) + B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t)
\end{aligned} \tag{3.11}$$

Using the property of the Kronecker product $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ (with A, B, C and D of appropriate dimension), we get

$$\begin{aligned}
\dot{\bar{X}}(t) &= (A_{N-1} - B_{uN-1}(L_r \otimes c\tilde{K}))\bar{X}(t) \\
&- B_{uN-1}(L_r \otimes c\tilde{K})\bar{F}_s(t) + B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) \\
&= (A_{N-1} - (L_r \otimes cB_u\tilde{K}))\bar{X}(t) \\
&- (L_r \otimes cB_u\tilde{K})\bar{F}_s(t) + B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t)
\end{aligned} \tag{3.12}$$

The last operation was performed exploiting the fact that by definition $B_{u_{N-1}} = (I_{N-1} \otimes B_u)$. If the ideal case in which there are no faults nor disturbances is considered, $\bar{X}$ converges to zero if and only if the matrix $(A_{N-1} - (L_r \otimes cB_u\tilde{K}))$ is Hurwitz stable. We introduce, now, a theorem that ensures this property.

Lemma 3.1.1. *If the graph $\mathscr{G}$ satisfies assumption 13, the $N-1$ agents described by* (3.1) *reach leader-follower consensus under the protocol* (3.9) *with $\tilde{K} = -B_u^T P^{-1}$ and $c \geq \frac{1}{\min_{i=2,\ldots,N} \Re(\lambda_i)}$, where λ_i, $i = 2,\ldots,N$, are the non zero eigenvalues of L_r, and $P > 0$ is the solution to the following Linear Matrix Inequality (LMI):*

$$AP + PA^T - 2B_uB_u^T < 0 \tag{3.13}$$

Define $g(\bar{F}_a(t), \bar{F}_s(t), \bar{D}(t)) := -(L_r \otimes cB_u\tilde{K})\bar{F}_s(t) + B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t)$ which is a linear function on the variables $\bar{F}_a(t)$, $\bar{F}_s(t)$, $\bar{D}(t)$, then we can rewrite Eq. (3.12) as:

$$\dot{\bar{X}}(t) = (A_{N-1} - (L_r \otimes cB_u\tilde{K}))\bar{X}(t) + g(\bar{F}_a(t), \bar{F}_s(t), \bar{D}(t)) \tag{3.14}$$

Algorithm 3 High-gain LMI design. [2]

Input: a stabilizable pair (A, B_u)
Output: a control gain matrix $\bar{K}$ and a coupling gain c

1. set $P \leftarrow$ any solution to the Linear Matrix Inequality (LMI) $AP + PA^T - 2B_uB_u^T < 0$
2. set $\bar{K} \leftarrow B_u^T P^{-1}$
3. set $c \leftarrow 1/\min_{i=2,\ldots,N} \Re(\lambda_i)$

Since the two fault signals $\bar{F}_a(t)$, $\bar{F}_s(t)$ and the disturbance signal $\bar{D}(t)$ are bounded by *Assumption 1*, it is possible to compute the maximum convergence error between the values of $x_1(t)$ and $x_i(t)$ as:

$$\varepsilon_{MAX} = \max_{\substack{i=2,\ldots,N \\ \|\bar{D}\|\leq\delta_d, \\ \|\bar{F}_a\|\leq\delta_a,\ \|\bar{F}_s\|\leq\delta_s.}} [(A_{N-1} - (L_r \otimes cB_u\tilde{K}))^{-1} g(\bar{F}_a, \bar{F}_s, \bar{D})]_i \tag{3.15}$$

It is worth noticing that the value $\min_{i=2,\ldots,N} \Re(\lambda_i)$ is a global information, in the sense that each follower has to know the topology of the entire graph $\mathscr{G}$ in order to compute it. It follows that the consensus protocol (3.9) is implemented in a distributed way, but with the need to know a priori information about the structure of the network. In the following an adaptive distributed consensus protocol that does not need any information about the eigenvalues of the reduced Laplacian L_r is presented.

3.1.1 Distributed Adaptive Consensus Protocol Design

In this section, consider the case in which each agent has access to the relative states of its neighbors with respect to itself (i.e, $C = I_n$). Based on the relative states of the neighboring agents, in [10] the following distributed adaptive consensus protocol with Time-Varying (TV) coupling weights is proposed:

$$\begin{aligned} u_i(t) &= c_i(t)\rho_i(\bar{w}_i(t)^T P^{-1}\bar{w}_i(t))K\bar{w}_i(t), \\ \dot{c}_i(t) &= \bar{w}_i(t)^T \Gamma \bar{w}_i(t), \quad i = 2, \ldots, N \end{aligned} \tag{3.16}$$

where $\bar{w}_i(t) := \sum_{j=1}^N a_{ij}(x_i(t) - x_j(t)) + \sum_{j=1}^N a_{ij}(f_i^s(t) - f_j^s(t))$, $c_i(t)$ denotes the time-varying coupling weight associated with the i^{th} follower with $c_i(0) \geq 1$, $P > 0$ is a solution to the LMI (3.16), $K \in \mathbb{R}^{n_u \times n}$ and $\Gamma \in \mathbb{R}^{n \times n}$ are the feedback gain matrices to be designed, $\rho_i(\cdot)$ are smooth and monotonically increasing functions to be determined later which satisfy the condition $\rho_i(s) \geq 1$ for $s > 0$, and the rest of variables are defined as in (3.9).
It can be noticed that, since it is imposed that $C = I_n$ the definition of W is the same as in *Chapter 2* and hence it holds that:

$$\bar{W}(t) = (L_r \otimes I_n)\bar{X}(t) + (L_r \otimes I_n)\bar{F}_s(t) \tag{3.17}$$

It easy to see that the condition that ensures consensus, namely Eq. (3.8), is equivalent to impose that $\bar{W}(t)$ asymptotically converges to zero (in the scenario in which $\bar{F}_s(t) = 0$).
In light of (3.1) and (3.16) we obtain the dynamics of $\bar{W}(t)$:

$$\begin{aligned} \dot{\bar{W}}(t) &= (L_r \otimes I_n)(\dot{\bar{X}}(t) + \dot{\bar{F}}_s(t)) \\ &= (L_r \otimes I_n)[A_{N-1}\bar{X}(t) + B_{uN-1}\bar{U}(t) + B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) + \dot{\bar{F}}_s(t)] \end{aligned} \tag{3.18}$$

Since $u_1(t) = 0$ this implies that $\bar{u}_i(t) = u_i(t) - u_1(t) = u_i(t)$, hence if we define $\hat{C} := diag(c_2, \ldots, c_N)$, $\hat{\rho}(W) := diag(\rho_2(\bar{w}_2(t)^T P^{-1}\bar{w}_2(t)), \ldots, \rho_N(\bar{w}_N(t)^T P^{-1}\bar{w}_N(t)))$, it is possible to write $\bar{U}(t)$ as:

$$\bar{U}(t) = \hat{C}\hat{\rho}(W)(I_{N-1} \otimes K)\bar{W}(t) \tag{3.19}$$

Substituting this expression inside (3.18), we obtain:

$$\begin{aligned} \dot{\bar{W}}(t) &= (L_r \otimes I_n)[A_{N-1}\bar{X}(t) + B_{uN-1}(\hat{C}\hat{\rho}(W)(I_{N-1} \otimes K)\bar{W}(t))] \\ &+ (L_r \otimes I_n)[B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) + \dot{\bar{F}}_s(t)] \\ &= (L_r \otimes I_n)[(I_{N-1} \otimes A)\bar{X}(t) + (I_{N-1} \otimes B_u)(\hat{C}\hat{\rho}(W) \otimes 1)(I_{N-1} \otimes K)\bar{W}(t))] \\ &+ (L_r \otimes I_n)[B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) + \dot{\bar{F}}_s(t)] \\ &= (L_r \otimes I_n)(I_{N-1} \otimes A)\bar{X}(t) + (L_r \otimes I_n)(\hat{C}\hat{\rho}(W) \otimes B_u K)\bar{W}(t) \\ &+ (L_r \otimes I_n)[B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) + \dot{\bar{F}}_s(t)] \end{aligned} \tag{3.20}$$

By definition (3.17), we can write $\bar{X}(t)$ as

$$\bar{X}(t) = (L_r \otimes I_n)^{-1}(\bar{W}(t) - (L_r \otimes I_n)\bar{F}_s(t)) \tag{3.21}$$

Using the property of the inverse of the Kronecker product, namely $(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$, we get

$$\bar{X}(t) = (L_r^{-1} \otimes I_n)\bar{W}(t) - \bar{F}_s(t) \tag{3.22}$$

Notice that it is possible to determine the inverse of L_r since it non singular by definition. Substituting (3.22) inside (3.20) we obtain

$$\begin{aligned} \dot{\bar{W}}(t) &= (L_r \otimes A)((L_r^{-1} \otimes I_n)\bar{W}(t) - \bar{F}_s(t)) + (L_r\hat{C}\hat{\rho}(W) \otimes B_u K)\bar{W}(t) \\ &+ (L_r \otimes I_n)[B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t)] + (L_r \otimes I_n)\dot{\bar{F}}_s(t) \\ &= (A_{N-1} + L_r\hat{C}\hat{\rho}(W) \otimes B_u K)\bar{W}(t) \\ &+ (L_r \otimes I_n)[B_{uN-1}\bar{F}_a(t) + B_{dN-1}\bar{D}(t) + \dot{\bar{F}}_s(t) - A_{N-1}\bar{F}_s(t)] \end{aligned} \tag{3.23}$$

Lemma 3.1.2. *[10] There exists a positive diagonal matrix G such that $GL_r + Lr_r^T G > 0$. One possible G is given by $diag(q_2, \ldots, q_N)$, where $q = [q_2, \ldots, q_n]^T = (L_r^T)^{-1}\mathbb{1}_{N-1}$.*

Now we are able to state the main theorem that ensures synchronization of the system in the case in which there is no disturbance or faults.

Theorem 3.1.3. *[10] Suppose that the graph $\mathcal{G}$ satisfies Assumption 13. Then the leader follower consensus problem of the agents in* (3.1) *is solved by the adaptive protocol in* (3.16)*, with $K = -B_u^T P^{-1}$, $\Gamma = P^{-1} B_u B_u^T P^{-1}$, and $\rho_i(\bar{w}_i(t)^T P^{-1} \bar{w}_i(t)) = (1 + \bar{w}_i(t)^T P^{-1} \bar{w}_i(t))^3$, where $P > 0$ is a solution to the LMI* (3.13)*. Moreover, each coupling weight c_i converges to some finite steady-state value.*

Remark 14. [10] A necessary and sufficient condition for the existence of a solution $P > 0$ of the LMI (3.13) is that the pair (A, B_u) is stabilizable. Therefore a sufficient condition for the existence of an adaptive protocol (3.16) satisfying *Theorem 3.1.3* is that (A, B) is stabilizable.

Remark 15. [10] The consensus protocol (3.16) can also be designed by solving the algebraic Riccati equation: $A^T Q + QA + I - QB_u B_u^T Q = 0$. In this case matrix K in (3.16) can be chosen as $K = -B_u^T Q$, $\Gamma = QB_u B_u^T Q$ and $\rho_i = (1 + \bar{w}_i(t)^Q \bar{w}_i(t))^3$. The solvability of the above Riccati equation is equivalent to that of LMI (3.13).

Chapter 4

Simulations

In order to verify and validate the algorithms presented in the previous chapter some simulation are performed using MATLAB Simulink.

As an example we consider a slightly modified version of the one presented in [1]. Consider a network of 10 aircraft connected to each other. The simulations are performed taking into account the case of vertical take off and landing, that is associated, after the linearization in vertical plane, to a model of the type (3.2) with

$$x = \begin{bmatrix} \text{horizontal velocity} \\ \text{vertical velocity} \\ \text{pitch rate} \\ \text{pitch angle} \end{bmatrix} \qquad u = \begin{bmatrix} \text{combined pitch control} \\ \text{longitudinal cyclic pitch angle control} \end{bmatrix} \tag{4.1}$$

where the longitudinal cyclic pitch input controls horizontal movement of the aircraft, while the combined pitch input controls the vertical movement. In the case of a standard flight with a 135 knots airspeed, the system matrices are chosen as

$$A = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & 1.0100 & 0.0024 & 4.0208 \\ 0.1002 & 0.3681 & 0.7070 & 1.4200 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B_u = B_d = \begin{bmatrix} 0.4420 & 0.1761 \\ 3.5446 & 7.5922 \\ 5.5200 & 4.4900 \\ 0 & 0 \end{bmatrix} \tag{4.2}$$

Unlike the example from which it takes inspiration from, the output matrix C is assumed to be equal to the identity ($C = I_4$).

In this context we simulate both the case in which the communication is undirected and the case in which is directed. Assume that the connections between the aircraft are described by the graph $\mathscr{G}$ depicted in figure 4.1,

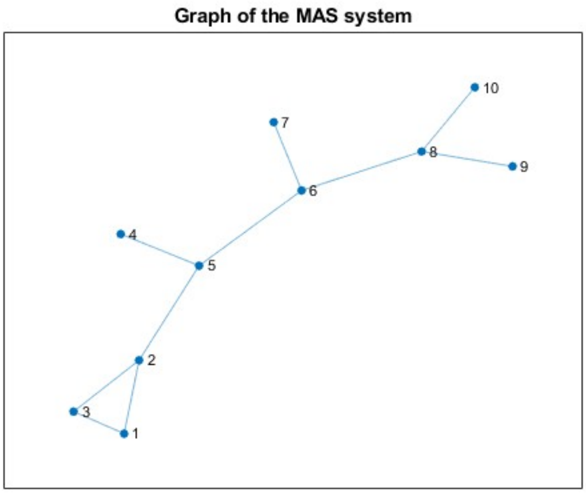


Figure 4.1: Graph of the system.

the associated adjacency matrix is

$$A_{\mathscr{G}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.3}$$

Consequently, the Laplacian associated to the graph is

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \tag{4.4}$$

In order to ensure synchronization, a controller for the whole system is built following (3.9). Since the graph is undirected, the Laplacian is symmetric, and hence all the eigenvalue are real. Since by Assumption 9 the eigenvalue are ordered, the smallest positive eigenvalue is $\lambda_2 = 0.1792$, hence we must impose that

$$c \geq \frac{1}{\lambda_2} \tag{4.5}$$

in this case we impose $c = \frac{10}{\lambda_2} = 55.811$.
Moreover, solving the LMI 3.13 the following possible solution was found

$$P = \begin{bmatrix} 0.457 & 0.021 & 0.401 & 0.171 \\ 0.021 & 6.681 & 4.011 & -0.540 \\ 0.401 & 4.011 & 5.051 & -0.584 \\ 0.171 & -0.540 & -0.584 & 0.711 \end{bmatrix} \tag{4.6}$$

and the corresponding value of the feedback matrix K is

$$K = -B^T P^{-1} = \begin{bmatrix} 0.824 & 0.283 & -1.526 & -1.238 \\ 0.114 & -1.163 & -0.088 & -0.982 \end{bmatrix} \tag{4.7}$$

In order to perform the FDI, we follow the steps of *Algorithm 2*. First, we divide the network in groups of three agents, obtaining four groups, see Fig. 4.2.

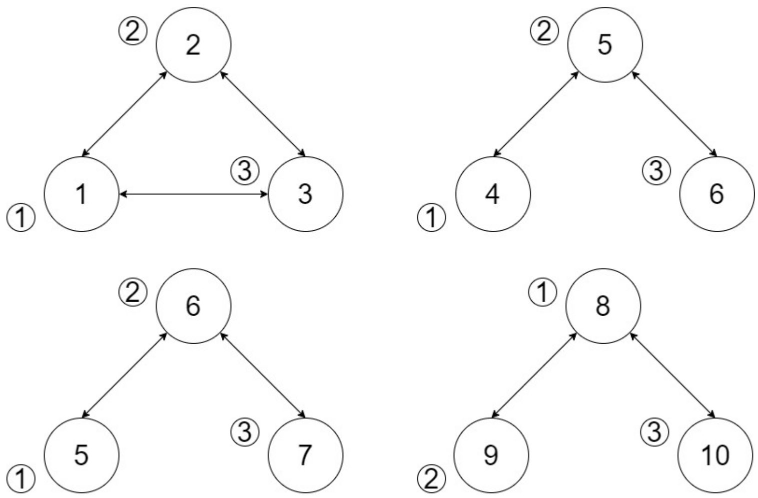


Figure 4.2: Four groups obtained from the division of the graph.

Since each group is composed by three nodes, in order to perform the FDI it is necessary to build only two UIOs. Consider first the case in which the fault of the second node must be excluded, then the matrices are

$$B_{u_2}^{-2} = \begin{bmatrix} \mathbb{0}_{4\times 2} \\ B_u \end{bmatrix} \quad B_{u_2}^{2} = \begin{bmatrix} B_u \\ \mathbb{0}_{4\times 2} \end{bmatrix} \tag{4.8}$$

So, the matrices of the observer are computed, following Eq. (2.47); a particular solution for

H^2 is given by Eq. (1.34)

$$\begin{aligned} H^2 &= B_u^2[(C_2B_u^2)^T C_2B_u^2]^{-1}(C_2B_u^2)^T \\ &= B_u^2[(B_u^2)^T B_u]^{-1}(B_u)^T \\ &= \begin{bmatrix} 0.012 & -0.038 & 0.103 & \mathbb{0}_5^T \\ -0.038 & 0.998 & 0.004 & \vdots \\ 0.103 & 0.004 & 0.989 & \mathbb{0}_5^T \\ \mathbb{0}_5 & \dots & \mathbb{0}_5 & \mathbb{0}_{5\times 5} \end{bmatrix} \end{aligned} \tag{4.9}$$

Then it is possible to compute M^2 as

$$\begin{aligned} M^2 &= I_8 - H^2 I_8 \\ &= \begin{bmatrix} 0.9876 & 0.0385 & -0.1038 & \mathbb{0}_5^T \\ 0.0385 & 0.0015 & -0.0040 & \vdots \\ -0.1038 & -0.0040 & 0.0109 & \mathbb{0}_5^T \\ \mathbb{0}_5 & \dots & \mathbb{0}_5 & I_5 \end{bmatrix} \end{aligned} \tag{4.10}$$

Then it is necessary to find K_1^2, such that J^2 is stable. In order to do this we decided to allocate all the poles of J^i in -5 and obtain

$$K_1^2 = \begin{bmatrix} 4.955 & 0.027 & -0.054 & -0.442 & & & & \\ -0.002 & 5.001 & -0.002 & -0.017 & & & & \\ 0.005 & -0.003 & 5.005 & 0.046 & & & & \\ 0 & 0 & 1 & 5 & & & & \\ & & & & 4.9634 & 0.027 & 0.018 & -0.455 \\ & & & & 0.0482 & 6.010 & 0.002 & 4.020 \\ & & & & 0.1002 & 0.368 & 5.707 & 1.420 \\ & & & & 0 & 0 & 1 & 5 \end{bmatrix} \tag{4.11}$$

After that, it was checked that all the poles were on the real negative half of the plane, performing

$$J^2 = M^2 A_2 - K_1^2 \tag{4.12}$$

Finally it is computed

$$\begin{aligned} K_2^2 &= J^2 H^2 \\ &= \begin{bmatrix} -0.062 & 0.192 & -0.518 & \mathbb{0}_5^T \\ 0.192 & -4.992 & -0.020 & \vdots \\ -0.518 & -0.020 & -4.945 & \mathbb{0}_5^T \\ \mathbb{0}_5 & \dots & \mathbb{0}_5 & \mathbb{0}_{5\times 5} \end{bmatrix} \end{aligned} \tag{4.13}$$

And finally the gain K^2 is obtained as

$$
\begin{aligned}
K^2 &= K_1^2 + K_2^2 \\
&= \left[\begin{array}{cccccccc}
4.893 & 0.219 & -0.573 & -0.442 & & & & \\
0.190 & 0.008 & -0.022 & -0.017 & & & & \\
-0.514 & -0.023 & 0.06 & 0.046 & & & & \\
0 & 0 & 1 & 5 & & & & \\
& & & & 4.963 & 0.027 & 0.018 & -0.455 \\
& & & & 0.048 & 6.010 & 0.002 & 4.020 \\
& & & & 0.100 & 0.368 & 5.707 & 1.42 \\
& & & & 0 & 0 & 1 & 5
\end{array}\right]
\end{aligned}
\tag{4.14}
$$

The same computations, considering B_u^3 (that in this case corresponds to B_u^{-2}), are done in order to obtain the value of the UIO matrices associated to the residual $r_a^3(t)$.
In all the simulations, the input reference of the leader is fixed at $[1\ 1]$.

In the first simulation we assumed that there were no errors in the system and we verified that there was synchronization among the agents.

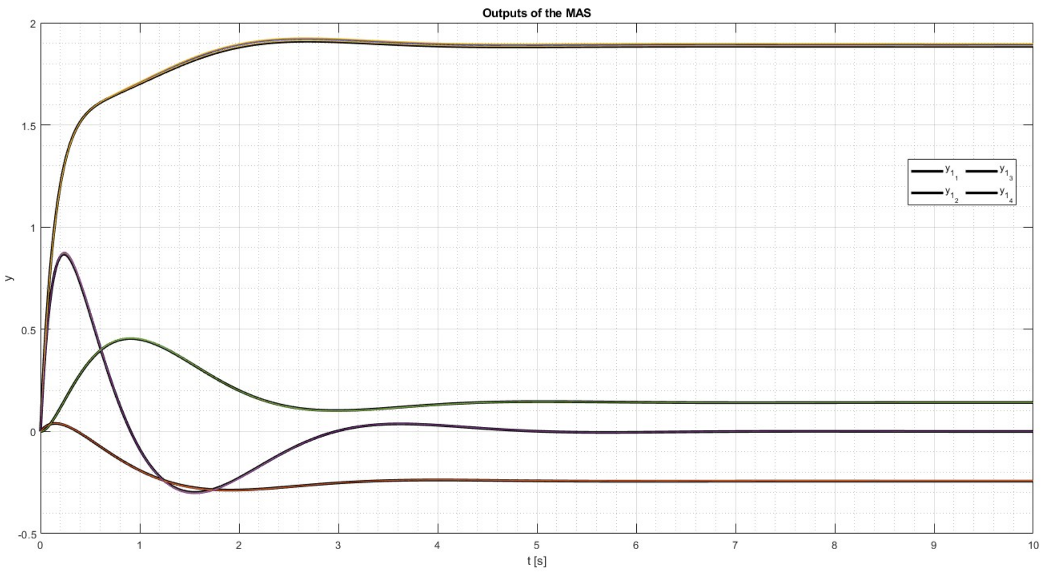


Figure 4.3: Time evolution of the output $y_i(t)$ of the agents without faults.

From Fig. 4.3, where the time evolution of the leader $y_1(t)$ is highlighted in black, it is possible to notice that there is perfect synchronization among the agents. In order to verify if the residual generator works properly, we suppose that some of the agents actuators have a fault; in particular, we suppose that at $t = 10s$ there is a fault at agent 2, at $t = 30s$ there is a fault at agent 6 and at time $t = 50s$ there is a fault at agent 8. All the three fault signals have unitary amplitude, namely $f_i = 1$.

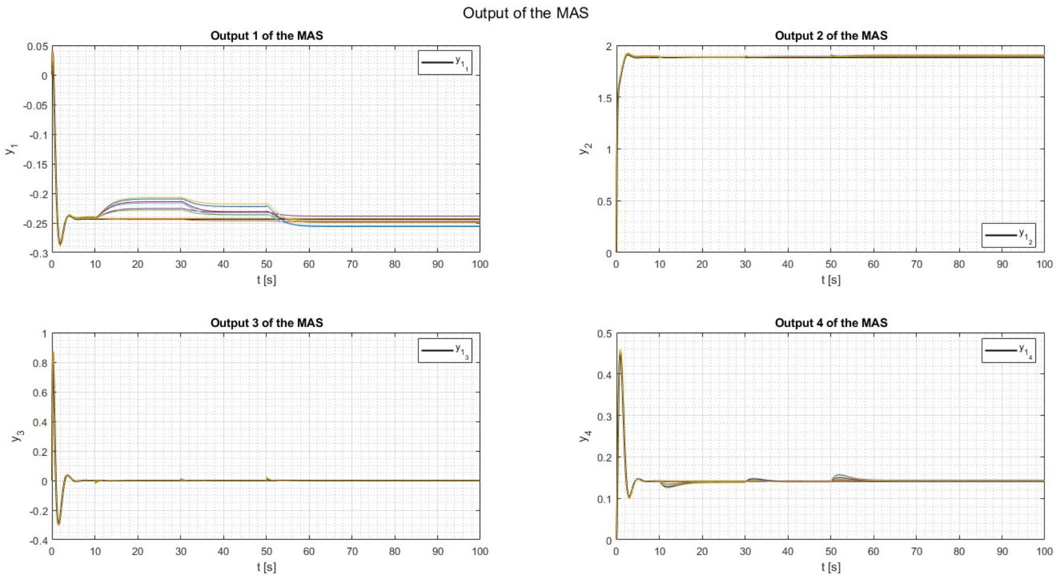


Figure 4.4: Time evolution of the output $y_i(t)$ of the agents with presence of actuator faults.

Looking at Fig. 4.4 it is possible to notice that, in the presence of faults, synchronization is not guaranteed. In fact at the time failures occur, the system starts behaving poorly.

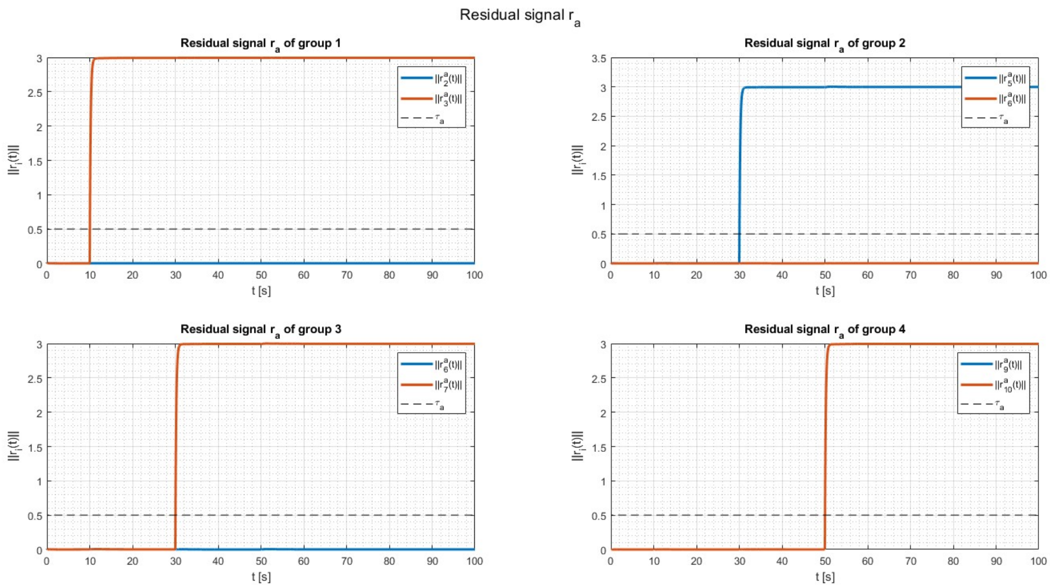


Figure 4.5: Residual generated by the four groups of UIO.

As expected, the residual behavior corresponds to the isolation logic established previously (2.68), and looking only at Fig.4.5 it is possible to identify which agent is affected by the fault.

Chapter 5

Conclusions

In this thesis the Fault Detection and Isolation (FDI) of linear homogeneous Multi Agent System (MAS) was taken into account and performed through Unknown Input Observer (UIO)s.
In the first chapter a brief introduction to MAS and to the different approaches (centralized, flooding and distributed) used to solve the problem of consensus was proposed. In addition, an overview of the different consensus problems that can be found in the literature has been given. After establishing a general model for linear, continuous time, time invariant, homogeneous agents some approaches to the modelling of a fault found in literature were discussed and finally a model that represents both the faults in the actuator and in the sensor of the agents is presented.
Since the aim of this thesis is the FDI performed through UIO, the theory behind this kind of technique was discussed and the choice of the threshold was also discussed.
In the second chapter, the main one, under the assumption that only one agent can be affected by a fault, a technique to detect the fault in the actuator that makes use of more than N-1 UIOs (where N refers to the size of the network) was presented. A distributed approach, which significantly reduces the size of the UIO used, was also presented.
For what concerns the FDI for the sensor faults, the previous approach was implemented and it was shown that, since we are dealing with homogeneous agents, it is not possible to perform the isolation. Despite this, again using a UIO, it was shown that at least the Fault Detection (FD) was possible, and the computation of a particular case in which $N = 3$ was done. It was shown that the isolation of all the agents is possible in the case in which the graph has all the connections, and this result was extended also to general MASs with N agents. For what concerns the other cases in which 3 agents are connected only with two edges it was shown that the isolation is possible only for the agent that shares both edges.

In the third chapter, a leader follower synchronization technique was presented; first in a distributed way that requires a priori information on the structure of the graph and then through an adaptive distributed approach that no longer needs this information.
Finally some simulations were performed to verify the results about synchronization and distributed FDI in the case in which an agent actuator is faulty.

Appendix A

Some useful definitions and theorems

Lemma A.0.1. *[1] The column rank of a matrix* $\begin{bmatrix} V & R \\ Z & 0 \end{bmatrix}$ *is full if R and Z are full column rank, where V, R and Z are of appropriate dimension.*

Proof. Assume that $V \in \mathbb{R}^{p\times p}$, $R \in \mathbb{R}^{p\times q}$, $Z \in \mathbb{R}^{n\times p}$. Let $V = (v_1, v_2, \ldots, v_p)$, $R = (r_1, r_2, \ldots, r_q)$, and $Z = (z_1, z_2, \ldots, z_p)$.
To verify matrix $\begin{bmatrix} V & R \\ Z & 0 \end{bmatrix}$ is full column rank, consider

$$\alpha_1 \begin{bmatrix} v_1 \\ z_1 \end{bmatrix} + \alpha_2 \begin{bmatrix} v_2 \\ z_2 \end{bmatrix} + \cdots + \alpha_p \begin{bmatrix} v_p \\ z_p \end{bmatrix} + \beta_1 \begin{bmatrix} r_1 \\ 0 \end{bmatrix} + \beta_2 \begin{bmatrix} r_2 \\ 0 \end{bmatrix} + \cdots + \beta_q \begin{bmatrix} r_q \\ 0 \end{bmatrix} = 0 \tag{A.1}$$

where α_i and β_j are coefficients with $i = 1, \ldots, p$ and $j = 1, \ldots, q$. We need to show that all α_i and β_j are equal to zero. From (A.1), it is clearly that $\alpha_1 z_1 + \alpha_2 z_2 + \cdots + \alpha_p z_p = 0$ and since Z has full column rank, those columns are linearly independent, and hence $\alpha_1 = \alpha_2 = \cdots = \alpha_p = 0$. Hence, $\beta_1 r_1 + \beta_2 + r_2 + \cdots + \beta_q r_q = 0$. As the columns of R are also linearly independent, we have $\beta_1 = \beta_2 = \cdots = \beta_q = 0$. Therefore, all α_i and β_j are equal to zero. This proves that the columns of matrix $\begin{bmatrix} V & R \\ Z & 0 \end{bmatrix}$ are linearly independent, and hence the matrix has full column rank. ∎

Theorem A.0.2 (PBH Criterion)**.** *[6] Let* $\Sigma = (A, B, C)$ *be an n-dimensional, either continuous-time or discrete-time, system. Then* Σ *is reachable if and only if the matrix* $\begin{bmatrix} A - sI_n & B \end{bmatrix}$ *is of full row rank for all* $s \in \mathbb{C}$*. If the system is not reachable, the rank of the matrix* $\begin{bmatrix} A - sI_n & B \end{bmatrix}$ *evaluated at s is less than n if and only if s is an eigenvalue of the non-reachable subsystem of* Σ*.*

Definition A.0.1 (Stabilizable pair)**.** A system whose non-controllable subsystem is asymptotically stable is called stabilisable.

Definition A.0.2 (Detectable pair)**.** [6] A system whose non-observable subsystem is asymptotically stable is called detectable.

Appendix B

MATLAB and SIMULINK

B.1 MATLAB Code

Matlab code used for the initialization of the parameter and the computation of the matrices for the UIO for the simulation.

```
clear all
close all
clc

%% simulation
satH=50;
satL=-50;
N=10; %Number of agents

A=[-0.0366 0.0271 0.0188 -0.4555;
    0.0482 1.0100 0.0024  4.0208;
    0.1002 0.3681 0.7070  1.4200;
    0.0000 0.0000 1.000   0.0000 ];

B=[ 0.4420  0.1761;
    3.5446  7.5922;
    5.5200  4.4900;
    0.0000  0.0000 ];

Bu=B; % Input matrix
Bf=B; % Actuator fault matrix

C = eye(4);

D=zeros(4,2);
```

```
Ref=[1;1];

%% check for controllability
Co = ctrb(A,B);
Contr = length(A) - rank(Co);

Co = ctrb(A',C');
Detect = length(A') - rank(Co);

%% Pole placement for stability of i=1
p=[-6.83 -1.01+1.51j -1.01-1.51j -2.55];
K1=place(A,B,p);

%% Graph
Adj=[0 1 1 0 0 0 0 0 0 0;
     1 0 1 0 1 0 0 0 0 0;
     1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 1 0 0 0 0 0;
     0 1 0 1 0 1 0 0 0 0;
     0 0 0 0 1 0 1 1 0 0;
     0 0 0 0 0 1 0 0 0 0;
     0 0 0 0 0 1 0 0 1 1;
     0 0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 0 1 0 0]; %undirected graph

Adj=[ 0 1 0 0 0 0 0 0 0 0
      0 0 1 0 1 0 0 0 0 0
      1 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0
      0 1 0 1 0 1 0 0 0 0
      0 0 0 0 0 0 1 1 0 0
      0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 1 0 0 1 1
      0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 ]'; %digraph

Ts=[           1,    zeros(1,2);
    -ones(3-1,1), eye(3-1,3-1)]; % Transformation matrix

Diag=diag(Adj*(ones(N,1)));
```

```
L=Diag-Adj; %Laplacian
eigL=sort(real(eig(L))); %eigenvalue of the laplacian, ordered

%% Set the initial condition
ic1=0*rand(1,4);
ic2=0*rand(1,4);
ic3=0*rand(1,4);
ic4=0*rand(1,4);
ic5=0*rand(1,4);
ic6=0*rand(1,4);
ic7=0*rand(1,4);
ic8=0*rand(1,4);
ic9=0*rand(1,4);
ic10=0*rand(1,4);

%% Computation of the controller for the synchronization
c_min=10/eigL(2); % Common coupling weight

G = -B*B';
Q = C'*C;
P = icare(A,[],Q,[],[],[],G);
K=-B'*P; % Controller
Gam=P*B*B'*P;
K_n=c_min*kron(eye(N),K); % Controller for the whole system

%% UIO Computation actuator fault, distributed way

% Computation for i=2
A_2=kron(eye(2),A);
Bu_2=kron(eye(2), Bu);
Bu_minus2=[Bu*0;Bu];
Bu_plus2=[Bu;Bu*0];
H_2=Bu_plus2*inv(Bu_plus2'*Bu_plus2)*Bu_plus2';
M_2=eye(length(H_2))-H_2;
p_2=-5*ones(length(M_2*A_2),1);
K_1_2=place(M_2*A_2, eye(length(M_2*A_2)), p_2);
J_2=M_2*A_2-K_1_2;
K_2_2=J_2*H_2;
K_2=K_1_2+K_2_2;

% Computation for i=3
```

```
A_2=kron(eye(2),A);
Bu_minus3=[Bu;Bu*0];
Bu_plus3=[Bu*0;Bu];
H_3=Bu_plus3*inv(Bu_plus3'*Bu_plus3)*Bu_plus3';
M_3=eye(length(H_3))-H_3;
p_3=-5*ones(length(M_3*A_2),1);
K_1_3=place(M_3*A_2, eye(length(M_3*A_2)), p_3);
J_3=M_3*A_2-K_1_3;
K_2_3=J_3*H_3;
K_3=K_1_3+K_2_3;

%% Fault signals
%Agent 1
Fa1=1;
Fat1=10;

%Agent 2
Fa2=0;
Fat2=0;

%Agent 3
Fa3=0;
Fat3=0;

%Agent 4
Fa4=0;
Fat4=0;

%Agent 5
Fa5=0;
Fat5=0;

%Agent 6
Fa6=2;
Fat6=30;

%Agent 7
Fa7=0;
Fat7=0;

%Agent 8
```

```
Fa8=3;
Fat8=50;

%Agent 9
Fa9=0;
Fat9=0;

%Agent 10
Fa10=0;
Fat10=0;
```

In the following the MATLAB code used for the plot of the graphs.

```
close all
clc

t=out.tout;
%% Plot Graph
% Whole graph
figure
plot(digraph(Adj)) % Plot the graph
title('Graph of the MAS system')

%% Plot f the relativa information W
W_t=out.W_t;
W_t_1=W_t*kron(eye(N),[1; 0; 0; 0]);
W_t_2=W_t*kron(eye(N),[0; 1; 0; 0]);
W_t_3=W_t*kron(eye(N),[0; 0; 1; 0]);
W_t_4=W_t*kron(eye(N),[0; 0; 0; 1]);

% Whole graph
figure
plot(t,W_t)
grid on
grid minor
xlabel("t [s]")
ylabel("W")
title('Relative information of the MAS')

figure
subplot(2,2,1)
plot(t,W_t_1)
```

```
title('Relative information W_1 of the MAS')
grid on
grid minor
xlabel("t [s]")
ylabel("W_1")

subplot(2,2,2)
plot(t,W_t_2)
title('Relative information W_2 of the MAS')
grid on
grid minor
xlabel("t [s]")
ylabel("W_2")

subplot(2,2,3)
plot(t,W_t_3)
title('Relative information W_3 of the MAS')
grid on
grid minor
xlabel("t [s]")
ylabel("W_3")

subplot(2,2,4)
plot(t,W_t_4)
title('Relative information W_4 of the MAS')
grid on
grid minor
xlabel("t [s]")
ylabel("W_4")

sgtitle('Relative information of the MAS')

%% Plot of the output Y
y_t=out.y_t;
y_t_1=y_t*kron(eye(N),[1; 0; 0; 0]);
y_t_2=y_t*kron(eye(N),[0; 1; 0; 0]);
y_t_3=y_t*kron(eye(N),[0; 0; 1; 0]);
y_t_4=y_t*kron(eye(N),[0; 0; 0; 1]);

% Whole graph
figure
```

```matlab
plot(t,y_t(:,[1:4]),'k', 'LineWidth',2.5)
hold on
plot(t,y_t(:,[5:40]))
L=legend('y_{1_1}','y_{1_2}','y_{1_3}','y_{1_4}')
L.AutoUpdate = 'off'
L.NumColumns = 2
legend('Location','best')
grid on
grid minor
xlabel("t [s]")
ylabel("y")
title('Outputs of the MAS')

% Graph divided by state
figure
subplot(2,2,1)
plot(t,y_t_1(:,1),'k', 'LineWidth',1.5)
hold on
plot(t,y_t_1(:,[2:10]))
title('Output 1 of the MAS')
L=legend('y_{1_1}')
L.AutoUpdate = 'off'
L.NumColumns = 1
legend('Location','best')
grid on
grid minor
legend
xlabel("t [s]")
ylabel("y_1")

subplot(2,2,2)
plot(t,y_t_2(:,1),'k', 'LineWidth',1.5)
hold on
plot(t,y_t_2(:,[2:10]))
title('Output 2 of the MAS')
L=legend('y_{1_2}')
L.AutoUpdate = 'off'
L.NumColumns = 1
legend('Location','best')
grid on
grid minor
```

```
xlabel("t [s]")
ylabel("y_2")

subplot(2,2,3)
plot(t,y_t_3(:,1),'k', 'LineWidth',1.5)
hold on
hold on
plot(t,y_t_3(:,[2:10]))
title('Output 3 of the MAS')
L=legend('y_{1_3}')
L.AutoUpdate = 'off'
L.NumColumns = 1
legend('Location','best')
grid on
grid minor
xlabel("t [s]")
ylabel("y_3")

subplot(2,2,4)
plot(t,y_t_4(:,1),'k', 'LineWidth',1.5)
hold on
plot(t,y_t_4(:,[2:10]))
title('Output 4 of the MAS')
L=legend('y_{1_4}')
L.AutoUpdate = 'off'
L.NumColumns = 1
legend('Location','best')
grid on
grid minor
xlabel("t [s]")
ylabel("y_4")

sgtitle('Output of the MAS')

%% Actuator residual plot
Ra1=out.Ra1;
Ra2=out.Ra2;
Ra3=out.Ra3;
Ra4=out.Ra4;
ThrA=0.5*ones(1,length(t));

```

```
figure
subplot(2,2,1)
plot(t,Ra1,'LineWidth',2.5)
hold on
plot(t,ThrA,'k--')
title('Residual signal r_a of group 1')
grid on
grid minor
legend('||r^a_2(t)||','||r^a_3(t)||','\tau_a')
xlabel("t [s]")
ylabel("||r_i(t)||")

subplot(2,2,2)
plot(t,Ra2,'LineWidth',2.5)
hold on
plot(t,ThrA,'k--')
title('Residual signal r_a of group 2')
grid on
grid minor
legend('||r^a_5(t)||','||r^a_6(t)||','\tau_a')
xlabel("t [s]")
ylabel("||r_i(t)||")

subplot(2,2,3)
plot(t,Ra3,'LineWidth',2.5)
hold on
plot(t,ThrA,'k--')
title('Residual signal r_a of group 3')
grid on
grid minor
legend('||r^a_6(t)||','||r^a_7(t)||','\tau_a')
xlabel("t [s]")
ylabel("||r_i(t)||")

subplot(2,2,4)
plot(t,Ra4,'LineWidth',2.5)
hold on
plot(t,ThrA,'k--')
title('Residual signal r_a of group 4')
grid on
grid minor
```

```
legend('||r_9^a(t)||','||r^a_{10}(t)||','\tau_a')
xlabel("t [s]")
ylabel("||r_i(t)||")
sgtitle('Residual signal r_a')
```

B.2 SIMULINK Scheme

In the following are depicted the various component of the SIMULINK scheme used for the simulation.

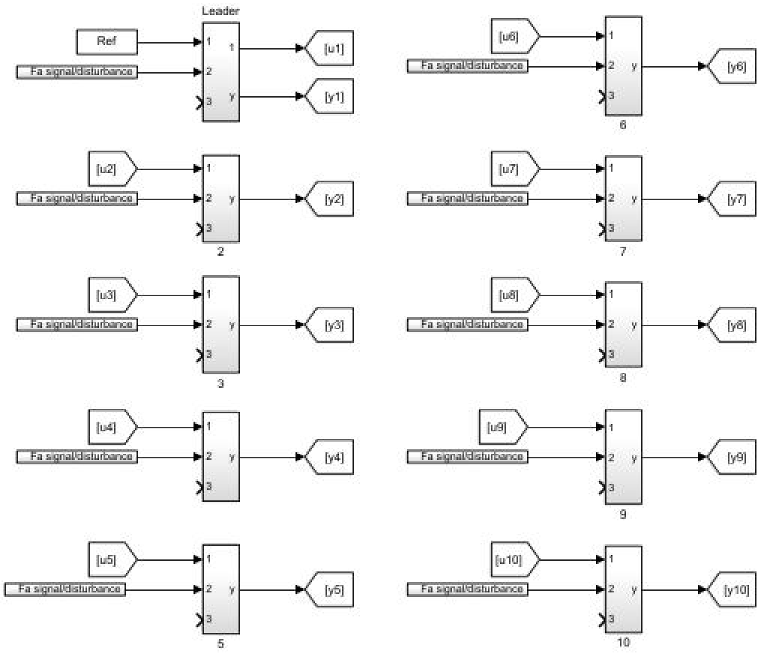


Figure B.1: Group of Multi Agent System (MAS).

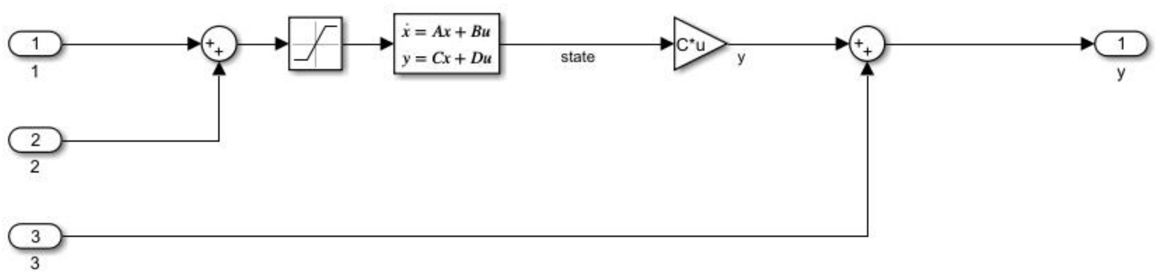


Figure B.2: Generic agent of the MAS.

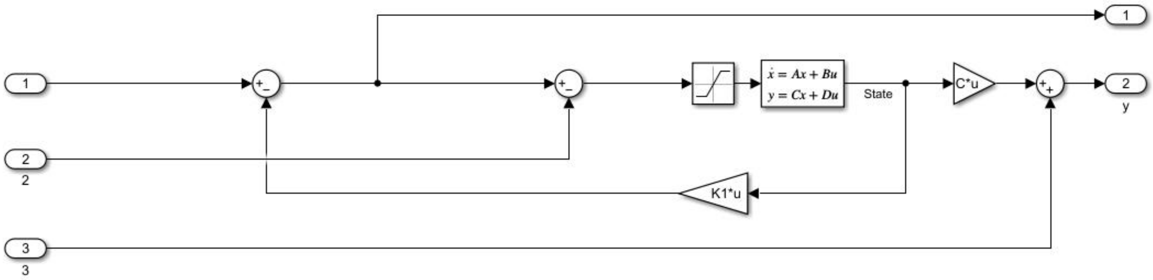


Figure B.3: Leader of the MAS.

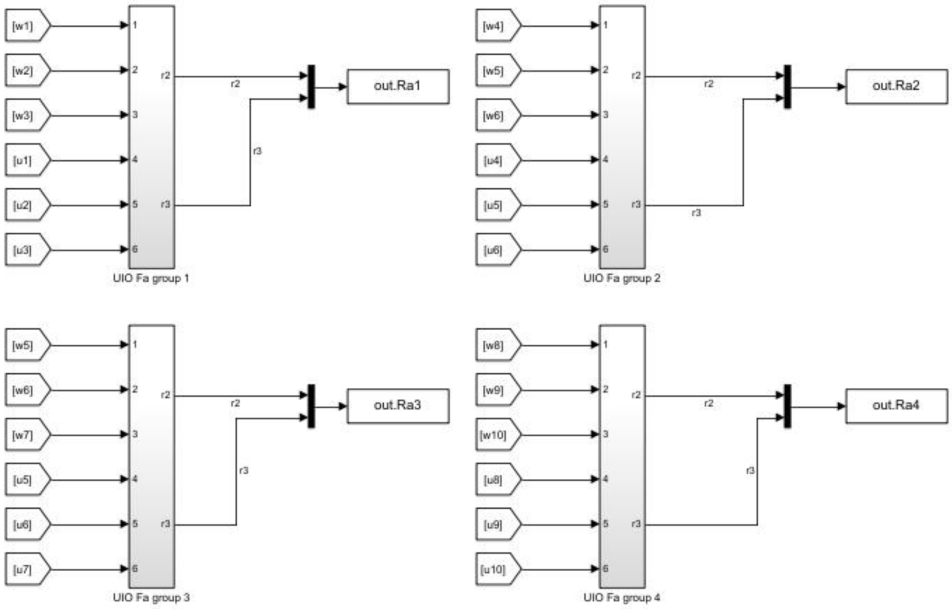


Figure B.4: Group of residual generator.

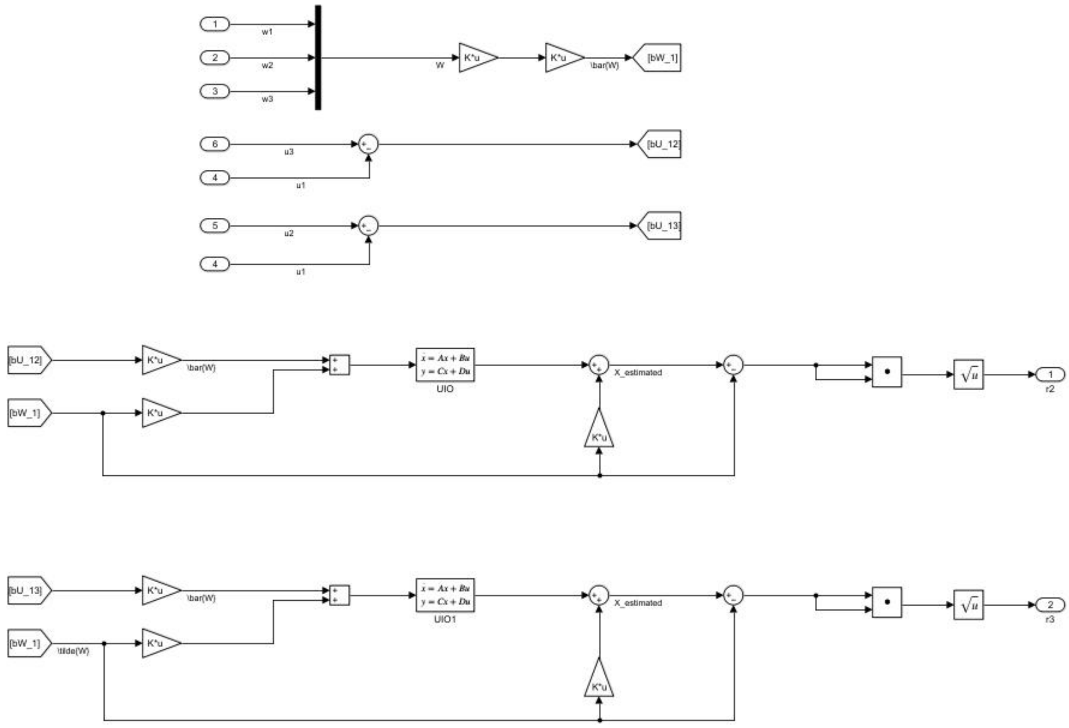


Figure B.5: Residual generator.

Bibliography

[1] Yuqi Bai and Jinzhi Wang. Fault detection and isolation using relative information for multi-agent systems. *ISA Transactions*, 116:182–190, 2021.

[2] F. Bullo. *Lectures on Network Systems*. Kindle Direct Publishing, 1.6 edition, 2022.

[3] Jie Chen. *Robust residual generation for model-based fault diagnosis of dynamic systems*. edition, 1995.

[4] Mohammadreza Davoodi, Nader Meskin, and Khashayar Khorasani. Simultaneous fault detection and consensus control design for a network of multi-agent systems. *Automatica*, 66:185–194, 2016.

[5] S.X. Ding. *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, 2008.

[6] Ettore Fornasini. *Appunti di teoria dei sistemi*. Progetto Libreria, 2011.

[7] Shyam Krishan Joshi, Shaunak Sen, and Indra Narayan Kar. Synchronization of coupled oscillator dynamics. *IFAC-PapersOnLine*, 49(1):320–325, 2016. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016.

[8] Solmaz S. Kia, Bryan Van Scoy, Jorge Cortes, Randy A. Freeman, Kevin M. Lynch, and Sonia Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, June 2019.

[9] Yuqian Li and Christopher Nielsen. Position synchronized path following for a mobile robot and manipulator. In *52nd IEEE Conference on Decision and Control*, pages 3541–3546, 2013.

[10] Zhongkui Li, Guanghui Wen, Zhisheng Duan, and Wei Ren. Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs. *IEEE Transactions on Automatic Control*, 60(4):1152–1157, 2015.

[11] Prathyush P. Menon and Christopher Edwards. Robust fault estimation using relative information in linear multi-agent networks. *IEEE Transactions on Automatic Control*, 59(2):477–482, Feb 2014.

[12] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE*

Transactions on Automatic Control, 51(3):401–420, 2006.

[13] Wei Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1859–1864 vol. 3, 2005.

[14] Anass Taoufik, Krishna Busawon, Michael Defoort, and Mohamed Djemai. An output observer approach to actuator fault detection in multi-agent systems with linear dynamics. In *2020 28th Mediterranean Conference on Control and Automation (MED)*, pages 562–567, Sep. 2020.

[15] Bo Zhou, Wei Wang, and Hao Ye. Cooperative control for consensus of multi-agent systems with actuator faults. *Computers & Electrical Engineering*, 40(7):2154–2166, 2014.