SEP

SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Tuxtla Gutiérrez

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“Fault Estimation of Convex Multi-Agents Systems”.

Estimación de Fallas en Sistemas Multi-Agentes Convexos.

Tesis que presenta:

Ing. Luis Beltrán Farrera Díaz

Como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS EN INGENIERÍA MECATRÓNICA

Directores de Tesis
Dr. Francisco Ronay López Estrada
Dr. Mohammed Chadli

Asesores
Dr. Elías Neftalí Escobar Gómez
MC. Rafael Mota Grajales

Tutor
Dr. Rubén Grajales Coutiño

Febrero 2020
Tuxtla Gutiérrez, Chiapas. México.

Tuxtla Gutiérrez, Chiapas 19/Febrero/2020

OFICIO No DEPI -165/2020

C. ING. LUIS BELTRÁN FARRERA DÍAZ
PASANTE DE LA MAESTRÍA EN INGENIERÍA MECATRÓNICA
EGRESADO DEL INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ
PRESENTE.

Habiendo recibido la comunicación, por parte de los CC. DR. FRANCISCO RONAY LÓPEZ ESTRADA, DR. MOHAMMED, DR. ELÍAS NEFTALÍ ESCOBAR GÓMEZ, MC RAFAEL MOTA GRAJALES, DR. RUBEN GRAJALES COUTIÑO. En el sentido, de que se encuentra satisfactorio el contenido de la tesis denominada **"Fault Estimation of Convex Multi – Agents System"**, elaborada por usted, como prueba escrita para obtener el Grado de Maestro en Ciencias en Ingeniería Mecatrónica, AUTORIZAMOS a que se proceda a la impresión de la misma.

Sin otro particular, le envió un cordial saludo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"Ciencia y Tecnología con Sentido Humano"

ESTADOS UNIDOS MEXICANOS

DR. JUAN JOSÉ VILLALOBOS MALDONADO
JEFE DE LA DIVISIÓN DE POSGRADO E INVESTIGACIÓN.

M. en I. JOSÉ MANUEL ROSADO PÉREZ
DIRECTOR.

SECRETARÍA DE EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO
DE Tuxtla Gutiérrez
DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

C.c.p. Servicios Escolares.
C.c.p. Archivo

DEDICATORIA

A mi padre por inculcarme, a través de su ejemplo, el gusto por la lectura, el estudio constante y la estimación de los buenos valores. A mi madre por transmitirme, a través de su amor, el deseo de superarme personal y profesionalmente. A mis hermanas, a mi sobrino, a mi cuñado y a Jazmín por amenizar esta etapa de mi vida.

AGRADECIMIENTOS

Agradezco la oportunidad que me brindaron mis directores de tesis, el Dr. Francisco Ronay López Estrada y el Dr. Mohammed Chadli, para desarrollar este trabajo de investigación. Les agradezco por todas sus enseñanzas a través de sus asesorías, sus contribuciones al desarrollo de este trabajo y de mi persona, y por los consejos que me permitieron concluir satisfactoriamente esta investigación.
De igual manera, agradezco las valiosas correcciones, sugerencias y comentarios que, durante este proceso, me han brindado mis revisores el Dr. Elías Neftalí Escobar Gómez y el M.C. Rafael Mota Grajales. Gracias también a mi tutor el Dr. Rubén Grajales Coutiño por el apoyo que me brindó durante toda la maestría.
Valoro y agradezco el tiempo que me dedicaron muchos de mis compañeros, estudiantes de la maestría y del doctorado, en pequeñas asesorías que fueron de gran ayuda en los momentos en los que parecía que los resultados aquí presentados eran inalcanzables.
Finalmente agradezco al Dr. Carlos Ríos Rojas por las aportaciones realizadas mientras fue parte de este proyecto. Le agradezco grandemente su amistad, su confianza y su humildad.

Gracias totales al CONACYT por haber financiado la investigación otorgándome la beca de maestría además de brindarme la oportunidad de enriquecer esta investigación con una estancia en el extranjero mediante una beca mixta.

RESUMEN

El trabajo de investigación presentado propone una solución para la estimación de fallas en actuadores en sistemas multi-agentes (MAS, por sus siglas en inglés) en configuración líder-seguidor modelados en espacio de estados y cuya comunicación está dada de acuerdo a la teoría de grafos. Para conseguir este objetivo, se diseña un observador Proporcional-Integral (PI) que permite estimar las variables de estado no disponibles o no medibles del sistema, es decir, que la matriz C del espacio de estados es distinta a la matriz identidad, $C \neq I$, y, de manera simultánea, permite estimar correctamente las fallas que los actuadores puedan presentar. Este observador ha sido diseñado, mediante el criterio de estabilidad de Lyapunov, para sistemas lineales. Además del observador lineal, esta investigación extiende sus resultados al campo de los sistemas no lineales a través de las técnicas utilizadas en los sistemas convexos. En ambos casos, las soluciones para encontrar las ganancias del observador se presentan en desigualdades matriciales lineales (LMI, por sus siglas en inglés).

Para diseñar este observador PI, se presentan todas las herramientas que el lector necesitará para la correcta compresión del desarrollo de este trabajo. Se abordará sobre las generalidades de los sistemas convexos, de manera que sea posible trabajar con sistemas no lineales representándolos como sistemas multimodelos lineales. Se desarrollarán observadores de estados para comprender el funcionamiento de estos así como su importancia en el control automático. De igual manera, se introducen y se explican a detalle los preliminares matemáticos necesarios para diseñar los observadores de estados y los estimadores de fallas en sistemas Multi-Agentes.

ABSTRACT

This research proposes a solution for actuator fault estimation of multi-agent systems (MAS) for leader-following configuration. Space state models represent the dynamic of each agent in the network and the graph theory provides the communication between the agents. To achieve this objective, a Proportional-Integral Observer (PI) estimates the unavailable or non-measurable state variables of the system, that is, the C matrix of the state space is different from the identity matrix, i.e., $C\ not = I$, and, simultaneously, the observer proposed estimates adequately the actuator faults. This observer uses the Lyapunov stability criterion to design the observer gain matrices and the fault estimator gain matrices. In addition to the linear version, this research extends the results to deals with non-linear systems through convex systems techniques. In both cases, the linear matrix inequalities (LMI) method gives the solution to find the gain matrices.

To design this PI observer and for the correct understanding of all developments in this research, this work provides all the theoretical basis to the reader. The explanation of the section Convex Systems allows working with non-linear systems represented with linear multi-model systems. To understand how the state observers work as well as their importance in automatic control, this work gives some examples to learn to design state observers in the section State Estimation observer-based. In the same way, in detail, the section of Multi-agent systems provides the mathematical preliminaries necessary to design the state observer and fault estimator for Multi-Agents systems.

Contents

List of Figures

Chapter 1

Project Characterization

1.1 Introduction

Cooperative control of Multiagent systems (MAS) has been recently studied in order to achieve a common objective of several agents (Lewis et al. 2013). MAS is a collection of agents cooperating or competing with each other in order to fulfill common or individual goals (Liu & Fang 2007). In that sense, an agent is defined as an element of the system that autonomously performs actions through its actuators according to the data obtained from its environment through sensors. MAS with identical agents are called homogeneous, e.g., a fleet of identical robots, or wind turbine farm, etc. On the other hand, when the MAS is integrated by nonidentical systems it is called heterogeneous or nonhomogeneous (Li et al. 2012), e.g microgrid systems. Some applications of MAS can be found in the field of mobile robotics (Koubâa et al. 2015), transportation (Okdinawati et al. 2017), microgrid systems (Zhao et al. 2015), wind turbines farms (Akbarimajd et al. 2018), among others.

Recently, fault-tolerant control systems (FTC) for MAS has been explored in order to increase the reliability and safety of such systems. For example, in Yang et al. (2011) was developed an FTC for a class of nonlinear MAS that keeps the system under a required performance region even in case of faults occurring in one of the agents. The authors in Wang et al. (2015) propose a robust adaptive controller for multiagent systems with nonidentical dynamic uncertainties and undetectable faults in actuators based on the Lyapunov approach. Davoodi et al. (2016) developed a controller for a homogeneous multiagent system by considering distributed detection filters. In Deng & Yang (2017) was proposed a nonlinear FTC MAS by considering a fuzzy algorithm to deal with the non-linearities of the system. Chadli et al. (2017) presented a filter for detection and isolation of fault in a multiagent system where each of the agents is not identical. Nevertheless, it is well known that the applicability of FTC algorithms depends on the accuracy of the mathematical model. Nonetheless, most of the algorithms were developed for MAS approximated by linear time-varying models (LTIs). LTIs models are valid only near to an operation or equilibrium point, and they did not consider nonlinear dynamics, which are present in all physical systems. To deal with this limitation, an approach that has proved to represent complex nonlinear dynamics are convex systems modeled by Linear Parameter Varying (LPV) or Takagi-Sugeno (TS) models which increase the physical representativity of a real physical system (Gómez-Peñate et al. 2018). FTC for lateral vehicle dynamics modeled as TS with sensor faults was developed in Ichalal et al. (2016) while FTC for TS with sensor and actuator faults simultaneously was proposed in Kharrat et al. (2018). Nevertheless, to design a Fault-tolerant control it is desirable to estimate the fault in the process to keep the system under operation region .epliu2011fault.

The present work proposes a methodology to develop a state and fault estimator observer for convex multi-agent systems in the presence of actuator faults. The main objective is to design the observer for linear multi-agent systems according to graph theory and find the gain matrices involved through the Lyapunov Criterion. Then, the target is to extend this development to deal with convex multi-agent systems. All developments consider directed and undirected graphs for leader-following configuration.

1.2 Hypothesis

A Proportional-Integral observer can be designed to estimate states and faults for linear multi-agent systems and convex multi-agent systems with the leader-following configuration. The observer will estimate states and faults in a shorter convergence time compared to conventional proportional observers. Further, the communication graphs can be directed and undirected indistinctly.

1.3 Objectives

1.3.1 General Objective

Design an observer to estimate states and actuator faults in multi-agent convex systems modeled with directed and undirected graphs.

1.3.2 Specific Objectives

- Study control algorithms for linear multi-agent systems with full state variables.
- Desing state observers for linear multi-agent systems modeled under graph topologies.
- Desing an observer to estimate states and faults in linear multi-agent systems.
- Extend the Fault Estimator to multi-agent convex systems.

Chapter 2

Theoretical framework

This chapter presents general concepts about the three subjects involved: Convex systems, state estimation observers, and multi-agent systems. First, convex systems are presented with techniques used to apply linear concepts in nonlinear mathematical models. Second, state estimation observer-based is studied and multiple examples are given to understand how the observer works and the method to design them. Finally, the theory of Multiagent systems is analyzed to develop a distributed controller with the objective of controls a set of systems with the same law control. The observer is distributed in the sense that it depends only on the state variables of each system and the relative state of the neighbor systems. These concepts will be properly explained below.

2.1 Convex Systems

The Takagi-Sugeno (TS) models emerge as an attempt to formalize the fuzzy control theory. In 1985, the TS modeling technique was presented, based on a set of linguistic rules that allow linear combinations of the system states Takagi & Sugeno (1985). An improved version with an approach that contains greater mathematical rigor was presented in Cao & Frank (2001) using the theory of Lyapunov, where local dynamics of discrete nonlinear systems are described. However, up to this point, TS models are described as a heuristic part technique and part based on mathematical models.

Subsequently, due to the ease of adapting the Takagi-Sugeno technique to mathematical models and their application to physical systems, these become very popular. TS systems can be seen as a linear combination of local models that are blended by nonlinear convex functions. Due to this historical question, there is a confusion about the adequate terminology to refer to the Takagi-Sugeno model-based that are not related with the field of fuzzy logic. TS systems are also called quasi-linear Parameter Varying systems. Due to the above, the title of this work uses the term "convex" to differentiate it from works that belong to the fuzzy control researches. This philosophical question was deathly discussed recently in López-Estrada et al. (2019).

The applications for this technique are diverse; it is enough to have a nonlinear mathematical model to be able to develop a TS representation that allows knowing the behavior of the system in an accurate way or through an approximation in a larger work area with respect to the LTI models or LPV, Fig. 2.1(Witczak 2014, p. 8). It is out of the scope of this research to discuss the similarity and the difference between qLPV, LPV or TS models. Then, we refer to these multi-models representations as convex systems. Stability analyses have been proposed for the use of this technique such as can see in Cao & Frank (2001), robust control Lien & Yu (2008), state observer design Marx et al. (2007), estimation of faults in sensors and actuators Youssef et al. (2017) and applications to real systems such as wind turbines Bououden et al. (2012), steam generator of a thermal power station Astorga-Zaragoza et al. (2018), controllers of UAVs Kladis et al. (2011), among many others.

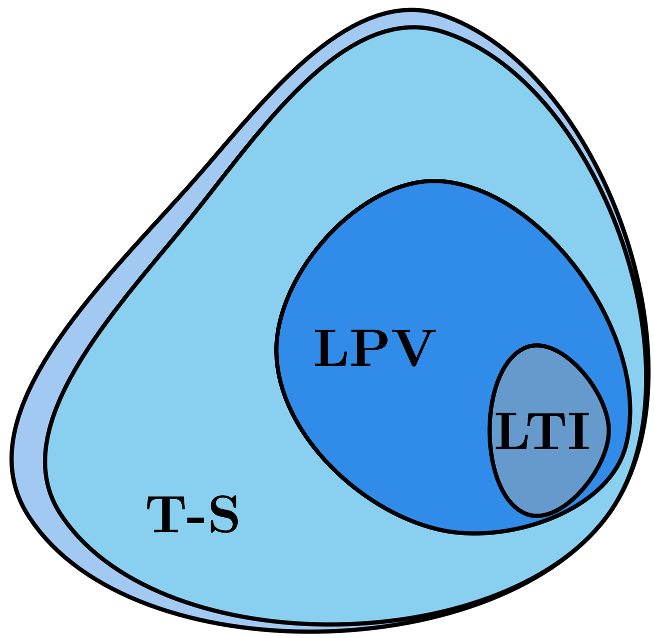

Figure 2.1: Performance for a TS model.

According to Lendek et al. (2011) and Guzmán-Rabasa et al. (2019), the general form of a convex model that describes a non-linear dynamic system in continuous time is given by:

$$\begin{aligned} \dot{x}(t) &= \sum_{j=1}^{m} \rho_j(z(t))(A_j x(t) + B_j u(t)) \\ y(t) &= Cx(t) \end{aligned} \tag{2.1}$$

where m is the number of local linear models, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ are the state, input and output vectors respectively. $A_j \in \mathbb{R}^{n\times n}$, $B_j \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ are known matrices with appropriated dimensions. $z(t)$ is the vector of scheduling variables which may depend on the states, inputs, measurements, or other exogenous variables. The scheduling variables represent the nonlinearities of the system and are chosen as $z_q(\cdot) \in [\underline{nl}_q, \overline{nl}_q]$, $q = 1 \ldots p$ where $\underline{nl}_q$ and $\overline{nl}_q$ are the minimum and maximum, respectively and p is the number of nonlinear terms in the system. $\rho_j(z(t))$ is the scheduling or membership function which is constructed with the combination of the weighting functions η^q. The weighting functions are strongly related to nonlinearities in the system and for each z_q exist two weighting functions given by:

$$\begin{aligned} &\eta_0^q(\cdot) = \frac{\overline{nl}_q - z_q(\cdot)}{\overline{nl}_q - \underline{nl}_q}, \eta_1^q(\cdot) = 1 - \eta_0^q(\cdot); \\ &q = 1...p \end{aligned} \tag{2.2}$$

where $\eta_0^q(\cdot)$ and $\eta_1^q(\cdot)$ are normalized, i.e., $\eta_0^q(\cdot) \geq 0$, $\eta_1^q(\cdot) \geq 0$, and $\eta_0^q(\cdot) + \eta_1^q(\cdot) = 1, \forall q$. Then, the number of rules or local models is given by: $m = 2^p$. The membership functions $\rho_j(z(t))$ satisfies the property of convex sum:

$$\begin{aligned} &\sum_{j=1}^{m} \rho_j(z(t)) = 1 \\ &0 \leq \rho_j(z(t)) \leq 1, \quad \forall j \in \{1, 2, ..., m\} \end{aligned} \tag{2.3}$$

Remark. The scope of this investigation is limited to consider nonlinearities in matrices A and B, then matrix C has a linear representation in all the research.

2.1.1 Linear Convex Systems examples

This section provides some examples to clarify the convex systems theory. First, some examples are introduced to calculate the scheduling variables $z(\cdot)$, the weighting functions η^q, and the membership functions $\rho_j(z(t))$. Finally, one example allows comparing the nonlinear dynamics and the representation of the convex dynamic, where it is possible to see those convex models are an exact representation of the nonlinear models.

Example 1. Consider a nonlinear system whose dynamics is represented in a space state model by:

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} x_2(t) & x_1(t) \\ 1 & -2 \end{bmatrix} x(t) + \begin{bmatrix} x_2(t) \\ 1 \end{bmatrix} u(t) \\ y(t) &= Cx(t) \end{aligned} \tag{2.4}$$

Since the output vector $y(t)$ is linear, it will be omitted in the development of the multi-models. Evidently, the nonlinear terms in this model are $x_1(t)$ and $x_2(t)$ in the matrix A and B. Then the scheduling variables can be chosen as $z_1(t) = x_2(t)$ and $z_2(t) = x_1(t)$. Note that scheduling variables are not unique for a model, the developer selects the scheduling variables ensuring the observability of the system and getting the least amount of multi-models possible. Then, the system can be rewritten as:

$$\dot{x}(t) = \begin{bmatrix} z_1(t) & z_2(t) \\ 1 & -2 \end{bmatrix} x(t) + \begin{bmatrix} z_1(t) \\ 1 \end{bmatrix} u(t) \tag{2.5}$$

The maximum and minimum range of the scheduling variables depends on the inferior limit and superior limit of the nonlinear dynamics, i.e., the minimum and maximum ranges are given by the natural dynamics of the system. We assume that $z_1(t) = x_2(t) \in [-2 \;\; 2]$ and $z_2(t) = x_1(t) \in [1 \;\; 3]$. This limits, are expressed for convex systems as:

$$\begin{aligned} \underline{nl}_1 &= -2 \\ \overline{nl}_1 &= 2 \\ \underline{nl}_2 &= 1 \\ \overline{nl}_2 &= 3 \end{aligned} \tag{2.6}$$

where, evidently, $\underline{nl}_q$ represents the minimum and $\overline{nl}_q$ the maximum. As it was explained before, there are two weighting functions $\eta^q(t)$ for each scheduling variable $z_q(t)$. Substituting the appropriate values in 2.2, the weighting functions are calculated as:

$$
\begin{aligned}
\eta_0^1(t) &= \frac{2-x_2}{2-(-2)} = \frac{2-x_2}{4} \\
\eta_1^1(t) &= 1-\eta_0^1 = 1-\frac{2-x_2}{4} = \frac{2+x_2}{4} \\
\eta_0^2(t) &= \frac{3-x_1}{3-(-1)} = \frac{3-x_1}{4} \\
\eta_1^2(t) &= 1-\eta_0^2 = 1-\frac{3-x_1}{4} = \frac{1+x_1}{4}
\end{aligned}
\tag{2.7}
$$

Since there are two scheduling variables $z_1(t)$ and $z_2(t)$, i.e., $p = 2$, the number of local models and weighting functions are given by $m = 2^p = 2^2 = 4$. Corresponding weighting functions and local models are calculated as follow:

1. A_1, B_1, and $\rho_1 z(t)$

$$
\begin{aligned}
A_1 &= \begin{bmatrix} \underline{nl}_1 & \underline{nl}_2 \\ 1 & -2 \end{bmatrix}, \; B_1 = \begin{bmatrix} \underline{nl}_1 \\ 1 \end{bmatrix} \\
A_1 &= \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}, \; B_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}
\end{aligned}
\tag{2.8}
$$

and the weighting function is computed as: $\rho_1(z) = \eta_0^1\eta_0^2$

2. A_2, B_2, and $\rho_2(z(t))$

$$
\begin{aligned}
A_2 &= \begin{bmatrix} \underline{nl}_1 & \overline{nl}_2 \\ 1 & -2 \end{bmatrix}, \; B_1 = \begin{bmatrix} \underline{nl}_1 \\ 1 \end{bmatrix} \\
A_2 &= \begin{bmatrix} -2 & 3 \\ 1 & -2 \end{bmatrix}, \; B_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}
\end{aligned}
\tag{2.9}
$$

and the weighting function is computed as: $\rho_2(z) = \eta_0^1\eta_1^2$

3. A_3, B_3, and $\rho_3(z(t))$

$$
\begin{aligned}
A_3 &= \begin{bmatrix} \overline{nl}_1 & \underline{nl}_2 \\ 1 & -2 \end{bmatrix}, \ B_1 = \begin{bmatrix} \overline{nl}_1 \\ 1 \end{bmatrix} \\
A_3 &= \begin{bmatrix} 2 & 1 \\ 1 & -2 \end{bmatrix}, \ B_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}
\end{aligned}
\tag{2.10}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_1^1 \eta_0^2$

4. A_4, B_4, and $\rho_4(z(t))$

$$
\begin{aligned}
A_4 &= \begin{bmatrix} \overline{nl}_1 & \overline{nl}_2 \\ 1 & -2 \end{bmatrix}, \ B_1 = \begin{bmatrix} \overline{nl}_1 \\ 1 \end{bmatrix} \\
A_4 &= \begin{bmatrix} 2 & 3 \\ 1 & -2 \end{bmatrix}, \ B_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}
\end{aligned}
\tag{2.11}
$$

and the weighting function is computed as: $\rho_4(z) = \eta_1^1 \eta_1^2$

Example 2. Similar to Example 1, in this example a nonlinear system with two state variables $x_1(t)$ and $x_2(t)$ and with two scheduling functions $z_1(t)$ and $z_2(t)$ is presented. The objective is to calculate the linear multi-models A_j and B_j and the corresponding weighting functions $z_q(t)$. Note that nonlinearities are present only in matrix A. Consider the nonlinear system represented in space state:

$$
\dot{x}(t) = \begin{bmatrix} x_1(t) + x_2(t) & x_2^2(t) \\ 4 & 7 \end{bmatrix} x(t) + \begin{bmatrix} 3.5 \\ 2 \end{bmatrix} u(t) \tag{2.12}
$$

Scheduling variables can be chosen as $z_1(t) = x_1(t) + x_2(t) \in [1 \;\; 2]$ and $z_2(t) = x_2^2(t) \in [-3 \;\; 2]$, where maximum $\overline{nl}_q$ and minimum $\underline{nl}_q$ are given demonstratively. Rewriting, one gets:

$$\dot{x}(t) = \begin{bmatrix} z_1(t) & z_2(t) \\ 4 & 7 \end{bmatrix} x(t) + \begin{bmatrix} 3.5 \\ 2 \end{bmatrix} u(t) \tag{2.13}$$

with

$$\begin{aligned} \underline{nl}_1 &= 1 \\ \overline{nl}_1 &= 2 \\ \underline{nl}_2 &= -3 \\ \overline{nl}_2 &= 2 \end{aligned} \tag{2.14}$$

The weighting functions are calculated as:

$$\begin{aligned} \eta_0^1 &= \frac{2-(x_1+x_2)}{2-1} = 2 - x_1 - x_2 \\ \eta_1^1 &= 1 - \eta_0^1 = 1 - (2 - x_1 - x_2) = -1 + x_1 + x_2 \\ \eta_0^2 &= \frac{2 - x_2^2}{2-(-3)} = \frac{2 - x_2^2}{5} \\ \eta_1^2 &= 1 - \eta_0^2 = 1 - \frac{2 - x_2^2}{5} = \frac{3 + x_2^2}{5} \end{aligned} \tag{2.15}$$

The two scheduling variables $z_1(t)$ and $z_2(t)$ generates $m = 2^p = 2^2 = 4$ multi-models for matrix A_j and weighting functions, consider that B matrix is linear.

1. A_1 and $\rho_1 z(t)$

$$A_1 = \begin{bmatrix} \underline{nl}_1 & \underline{nl}_2 \\ 4 & 7 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 4 & 7 \end{bmatrix} x \tag{2.16}$$

and the weighting function is computed as: $\rho_1(z) = \eta_0^1 \eta_0^2$

2. A_2 and $\rho_2(z(t))$

$$A_2 = \begin{bmatrix} \underline{nl}_1 & \overline{nl}_2 \\ 4 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 7 \end{bmatrix} x \tag{2.17}$$

and the weighting function is computed as: $\rho_2(z) = \eta_0^1 \eta_1^2$

3. A_3 and $\rho_3(z(t))$

$$A_3 = \begin{bmatrix} \overline{nl}_1 & \underline{nl}_2 \\ 4 & 7 \end{bmatrix} = \begin{bmatrix} 2 & -3 \\ 4 & 7 \end{bmatrix} x \tag{2.18}$$

and the weighting function is computed as: $\rho_3(z) = \eta_1^1 \eta_0^2$

4. A_4 and $\rho_4(z(t))$

$$A_3 = \begin{bmatrix} \overline{nl}_1 & \overline{nl}_2 \\ 4 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 7 \end{bmatrix} x \tag{2.19}$$

and the weighting functions is computed as: $\rho_4(z) = \eta_1^1 \eta_1^2$

Example 3. In order to clarify the combinations to obtain the membership functions $\rho_j(z)$, this example consider 3 scheduling variables, so that, $m = 2^p = 2^3 = 8$ multi-models are obtained. Following nonlinear model is given:

$$\dot{x}(t) = \begin{bmatrix} x_1(t)x_2(t) & 3 & -1 \\ 0 & x_1^2(t) & 2 \\ x_3(t) & 1 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ x_1^2(t) & 2 \\ 1 & x_3(t) \end{bmatrix} u(t) \tag{2.20}$$

which can be transformed to:

$$\dot{x}(t) = \begin{bmatrix} z_1(t) & 3 & -1 \\ 0 & z_2(t) & 2 \\ z_3(t) & 1 & 3 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ z_2(t) & 2 \\ 1 & z_3(t) \end{bmatrix} u(t) \tag{2.21}$$

with $z_1(t) = x_1(t)x_2(t) \in [-2 \;\; 4]$, $z_2(t) = x_1^2(t) \in [0 \;\; 3]$, and $z_3(t) = x_3(t) \in [2 \;\; 5]$. In order to calculate the weighting functions η_0^q and η_1^q, the limits $\underline{nl}_q$ and $\overline{nl}_q$, with $q = 1, 2, 3$, are defined:

$$
\begin{aligned}
\underline{nl}_1 &= -2 \\
\overline{nl}_1 &= 4 \\
\underline{nl}_2 &= 0 \\
\overline{nl}_2 &= 3 \\
\underline{nl}_3 &= 2 \\
\overline{nl}_3 &= 5
\end{aligned}
\tag{2.22}
$$

Calculating the weighting functions:

$$
\begin{aligned}
\eta_0^1 &= \frac{4 - x_1x_2}{4 - (-2)} = \frac{4 - x_1x_2}{6} \\
\eta_1^1 &= 1 - \eta_0^1 = 1 - \frac{4 - x_1x_2}{6} = \frac{2 + x_1x_2}{6} \\
\eta_0^2 &= \frac{3 - x_1^2}{3 - (0)} = \frac{3 - x_1^2}{3} \\
\eta_1^2 &= 1 - \eta_0^2 = 1 - \frac{3 - x_1^2}{3} = \frac{x_1^2}{3} \\
\eta_0^3 &= \frac{5 - x_3}{5 - 2} = \frac{5 - x_3}{3} \\
\eta_1^3 &= 1 - \eta_0^3 = 1 - \frac{5 - x_3}{3} = \frac{-2 + x_3}{3}
\end{aligned}
\tag{2.23}
$$

with the three nonlinearities $p = 3$ and the local models and weighting functions are given by $m = 8$. Following, the 8 local models and the corresponding weighting functions are given:

1. A_1, B_1, and $\rho_1(z)$

$$
\begin{aligned}
A_1 &= \begin{bmatrix} \underline{nl}_1 & 3 & -1 \\ 0 & \underline{nl}_2 & 2 \\ \underline{nl}_3 & 1 & 3 \end{bmatrix}, B_1 = \begin{bmatrix} 1 & 0 \\ \underline{nl}_2 & 2 \\ 1 & \underline{nl}_3 \end{bmatrix} \\
A_1 &= \begin{bmatrix} -2 & 3 & -1 \\ 0 & 0 & 2 \\ 2 & 1 & 3 \end{bmatrix}, B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 2 \end{bmatrix}
\end{aligned}
\tag{2.24}
$$

and the weighting function is computed as: $\rho_1(z) = \eta_0^1 \eta_0^2 \eta_0^3$

2. A_2, B_2, and $\rho_2(z)$

$$
\begin{aligned}
A_2 &= \begin{bmatrix} \underline{nl}_1 & 3 & -1 \\ 0 & \underline{nl}_2 & 2 \\ \overline{nl}_3 & 1 & 3 \end{bmatrix}, B_2 = \begin{bmatrix} 1 & 0 \\ \underline{nl}_2 & 2 \\ 1 & \overline{nl}_3 \end{bmatrix} \\
A_2 &= \begin{bmatrix} -2 & 3 & -1 \\ 0 & 0 & 2 \\ 5 & 1 & 3 \end{bmatrix}, B_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 5 \end{bmatrix}
\end{aligned}
\tag{2.25}
$$

and the weighting function is computed as: $\rho_2(z) = \eta_0^1 \eta_0^2 \eta_1^3$

3. A_3, B_3, and $\rho_3(z)$

$$
\begin{aligned}
A_3 &= \begin{bmatrix} \underline{nl}_1 & 3 & -1 \\ 0 & \overline{nl}_2 & 2 \\ \underline{nl}_3 & 1 & 3 \end{bmatrix}, B_3 = \begin{bmatrix} 1 & 0 \\ \overline{nl}_2 & 2 \\ 1 & \underline{nl}_3 \end{bmatrix} \\
A_3 &= \begin{bmatrix} -2 & 3 & -1 \\ 0 & 3 & 2 \\ 2 & 1 & 3 \end{bmatrix}, B_3 = \begin{bmatrix} 1 & 0 \\ 3 & 2 \\ 1 & 2 \end{bmatrix}
\end{aligned}
\tag{2.26}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_0^1 \eta_1^2 \eta_0^3$

4. A_4, B_4, and $\rho_4(z(t))$

$$
\begin{gathered}
A_4 = \begin{bmatrix} \underline{nl}_1 & 3 & -1 \\ 0 & \overline{nl}_2 & 2 \\ \overline{nl}_3 & 1 & 3 \end{bmatrix}, \; B_4 = \begin{bmatrix} 1 & 0 \\ \overline{nl}_2 & 2 \\ 1 & \overline{nl}_3 \end{bmatrix} \\
A_4 = \begin{bmatrix} -2 & 3 & -1 \\ 0 & 3 & 2 \\ 5 & 1 & 3 \end{bmatrix}, \; B_4 = \begin{bmatrix} 1 & 0 \\ 3 & 2 \\ 1 & 5 \end{bmatrix}
\end{gathered}
\tag{2.27}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_0^1 \eta_1^2 \eta_1^3$

5. A_5, B_5, and $\rho_5(z)$

$$
\begin{gathered}
A_4 = \begin{bmatrix} \overline{nl}_1 & 3 & -1 \\ 0 & \underline{nl}_2 & 2 \\ \underline{nl}_3 & 1 & 3 \end{bmatrix}, \; B_4 = \begin{bmatrix} 1 & 0 \\ \underline{nl}_2 & 2 \\ 1 & \underline{nl}_3 \end{bmatrix} \\
A_4 = \begin{bmatrix} 4 & 3 & -1 \\ 0 & 0 & 2 \\ 2 & 1 & 3 \end{bmatrix}, \; B_4 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 2 \end{bmatrix}
\end{gathered}
\tag{2.28}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_1^1 \eta_0^2 \eta_0^3$

6. A_6, B_6, and $\rho_6(z)$

$$
\begin{aligned}
A_6 &= \begin{bmatrix} \overline{nl}_1 & 3 & -1 \\ 0 & \underline{nl}_2 & 2 \\ \overline{nl}_3 & 1 & 3 \end{bmatrix}, \ B_6 = \begin{bmatrix} 1 & 0 \\ \underline{nl}_2 & 2 \\ 1 & \overline{nl}_3 \end{bmatrix} \\
A_6 &= \begin{bmatrix} 4 & 3 & -1 \\ 2 & 0 & 2 \\ 5 & 1 & 3 \end{bmatrix}, \ B_6 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 5 \end{bmatrix}
\end{aligned}
\tag{2.29}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_1^1 \eta_0^2 \eta_1^3$

7. A_7, B_7, and $\rho_7(z)$

$$
\begin{aligned}
A_7 &= \begin{bmatrix} \overline{nl}_1 & 3 & -1 \\ 0 & \overline{nl}_2 & 2 \\ \underline{nl}_3 & 1 & 3 \end{bmatrix}, \ B_7 = \begin{bmatrix} 1 & 0 \\ \overline{nl}_2 & 2 \\ 1 & \underline{nl}_3 \end{bmatrix} \\
A_7 &= \begin{bmatrix} 4 & 3 & -1 \\ 0 & 3 & 2 \\ 2 & 1 & 3 \end{bmatrix}, \ B_7 = \begin{bmatrix} 1 & 0 \\ 3 & 2 \\ 1 & 2 \end{bmatrix}
\end{aligned}
\tag{2.30}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_1^1 \eta_1^2 \eta_0^3$

8. A_8, B_8, and $\rho_8(z)$

$$
\begin{aligned}
A_8 &= \begin{bmatrix} \overline{nl}_1 & 3 & -1 \\ 0 & \overline{nl}_2 & 2 \\ \overline{nl}_3 & 1 & 3 \end{bmatrix}, \ B_8 = \begin{bmatrix} 1 & 0 \\ \overline{nl}_2 & 2 \\ 1 & \overline{nl}_3 \end{bmatrix} \\
A_8 &= \begin{bmatrix} 4 & 3 & -1 \\ 2 & 3 & 2 \\ 5 & 1 & 3 \end{bmatrix}, \ B_8 = \begin{bmatrix} 1 & 0 \\ 3 & 2 \\ 1 & 5 \end{bmatrix}
\end{aligned}
\tag{2.31}
$$

and the weighting function is computed as: $\rho_3(z) = \eta_1^1 \eta_1^2 \eta_1^3$

Remark. For any nonlinear system, the number of local models is given by $m = 2^p$. This method allows for obtaining an exact representation of the nonlinear model.

2.2 State Estimation Observer-Based

The modern industry requires faster processes with reduced costs. Under these conditions, engineers have the function of designing efficient automatic systems, with the least possible cost. This automation is carried out with different strategies as Fuzzy Control that focuses on gaining an intuitive understanding of how to best control the process, then this information is loaded directly into the fuzzy controller (Passino et al. 1998). Another approach is The Neural networks that have a lot of successful applications although, in general, from a theoretical point of view, it presents a lack of a firm mathematical basis in stability, robustness, and performance analysis(Ge et al. 2013). However, the main concern of this work is the model-based control where a mathematical model of the system is required.

Model-based control is limited in the complete representation of the system since an exact representation requires a very complex mathematical model. However, physicists and mathematicians provides to the engineers mathematical models that represents enough to keep in control the systems. Usually, for control purposes, the system is represented in a state space representation (Ogata 2010) where a vector of state variables provides the information of the system modeled as position, velocity, acceleration, flow, temperature, and so on, depending of the mathematical model. In continuous time, a system is presented for the following space state model:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.32}$$

where $x \in \mathbb{R}^N$ is the state vector, $u \in \mathbb{R}^m$ is the control input vector, $y \in \mathbb{R}^p$ is the measured output vector. Matrices A, B, C, and D are constant with appropriate dimensions. In order to control a system, the pair (A, B) must to be controllable, which means that the system has enough actuators to control the system. Commonly, matrix C provides the available state variables, in other words, matrix C allows to know what *physical sensors* there are in the system. Matrix D is used only if the systems considers a direct output from the control input vector u which is not common in most of the industrial applications.

Consider the following numerical example where the input vector u does not affect directly the output vector y, i.e., $D = 0$:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -5 & 3 & 2 \\ 4 & -4 & -3 \\ 0 & 2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \end{aligned} \tag{2.33}$$

The dynamics response to a step signal in the time $0s$ is illustrated in 2.2. The system 2.33 is evidently stable, and the state variables x_1, x_2, and x_3 stabilize naturally after the time $1.5s$. Note that, for this numerical examples, matrix $C = I$ which means that all state variables are available.

In practical applications, the measured output vector $y(t)$ does not provides all states variable since, physically, the cost of some sensors can be very high, i.e., $C \neq I$. Under this limitation, *state observers* allows to estimate, based only on the available state variables, the dynamics of the unknown state variables. Then, observers states can be understood as *virtual sensor* designed in order to provide the unknown state variables. The only requirement to apply this method is that the pair A, C in the space state representation must be observable, which means that, with the system in state $x(t_0)$, it is possible to determine this state from the observation of the output over a finite time interval (Ogata 2010). The system 2.32 is observable if and only if the *nmxn observability matrix*

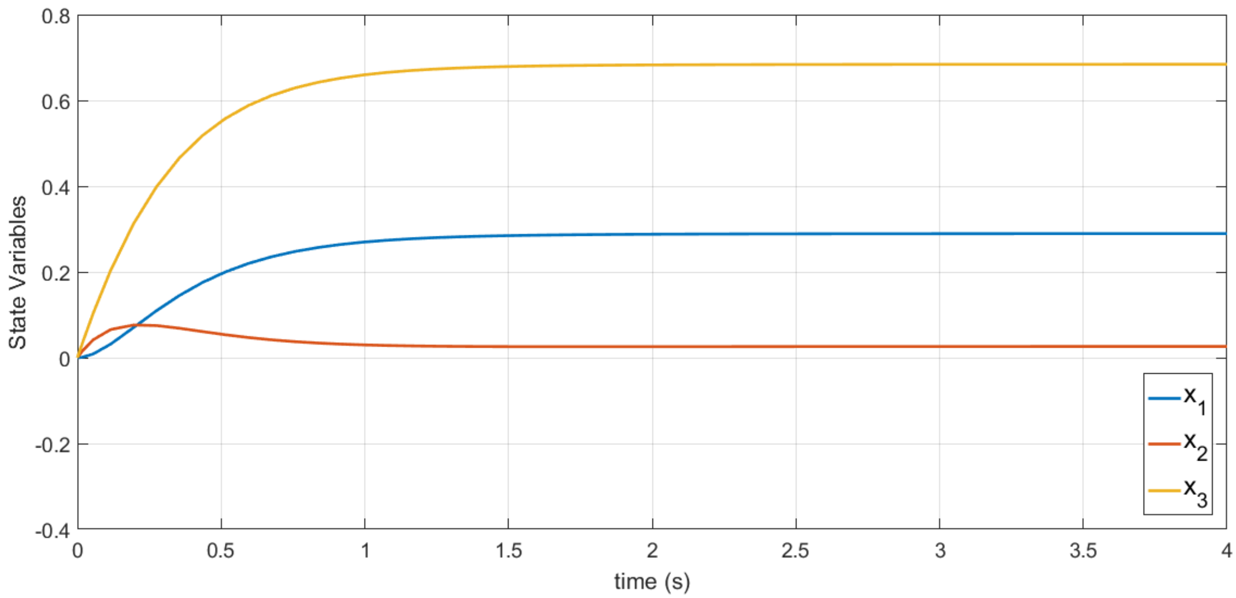

Figure 2.2: Response to step signal

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{2.34}$$

is of rank n, i.e., it has n linearly independent column vectors.

Based on 2.33, the last example, if only states variables x_1 and x_2 are available, x_3 can be estimated if and only if the observability matrix of system 2.35 is of rank n:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -5 & 3 & 2 \\ 4 & -4 & -3 \\ 0 & 2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x \end{aligned} \tag{2.35}$$

and its observability matrix is calculated as:

$$\begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \tag{2.36}$$

with rank 3. Then, this system is observable and it is possible to develop a state observer in order to estimate x_3.

Given the system 2.32, the following Luenberger observer(Radke & Gao 2006) is proposed in order to estimate states:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + R(y - \hat{y}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{2.37}$$

Where $\hat{x}$ is the estimated state vector, R is the observer gain matrix, and $\hat{y}$ is the estimated output vector. Note that D is not included in this observer since overall is not required. Henceforth, in this section, time dependence (t) will be omitted to simplify the mathematical expressions.

It is defined the *state estimation error* as:

$$e = x - \hat{x} \tag{2.38}$$

The target to develop the observer it is to design a gain matrix R such that $e \Rightarrow 0$. If we derive 2.38, one gets:

$$\dot{e} = \dot{x} - \dot{\hat{x}} \tag{2.39}$$

and substituting 2.32 and 2.37 in 2.39:

$$\begin{aligned}
\dot{e} &= Ax + Bu - A\hat{x} - Bu - R(y - \hat{y}) \\
\dot{e} &= A(x - \hat{x} - R(Cx - C\hat{x})) \\
\dot{e} &= Ae - RCe \\
\dot{e} &= (A - RC)e
\end{aligned} \tag{2.40}$$

this leads to the following lemma:

Lemma 1.(Ogata 2010) The dynamic behavior of the error vector is determined by the eigenvalues of matrix $(A-RC)e$. If matrix $(A-RC)e$ is a stable matrix the error vector will converge to zero for any initial error vector $e(0)$. That is, $\hat{x}$ will converge asymptotically to x regardless of the values of $\hat{x}(0)$ and x. If the eigenvalues of matrix $(A-RC)e$ are chosen in such a way that the dynamic behavior of the error vector is asymptotically stable and is adequately fast, then any error vector will tend to zero (the origin) with an adequate speed.

If the matrix 2.40 is stable, the error vector will converge to zero for any initial error vector $e(0)$. Then, given the dynamic behavior of the error e, the main target is to design a matrix R such that $(A-RC)$ is stable.

Then, the problem is reduced to design a matrix R such that $(A-RC)$ is stable. The first approach to design the observer gain matrix is the pole placement method, where the observer gain matrix is designed in order that the poles of the matrix $(A-RC)$ are placed anywhere in the left side of the complex plane. Let choose the poles $p = [-3,\ -3,\ -2]$ and the solving with a mathematical software, in this case MATLAB with the command $R = place(A', C', p)'$ one gets:

$$R_p = \begin{bmatrix} -2.3077 & 3.4615 \\ 4.4615 & -1.6923 \\ 0 & 2 \end{bmatrix} \tag{2.41}$$

Where R_p is the gain matrix obtained through the pole placement method. For this only development, the simulation example is provided to the reader in order to understand how the state

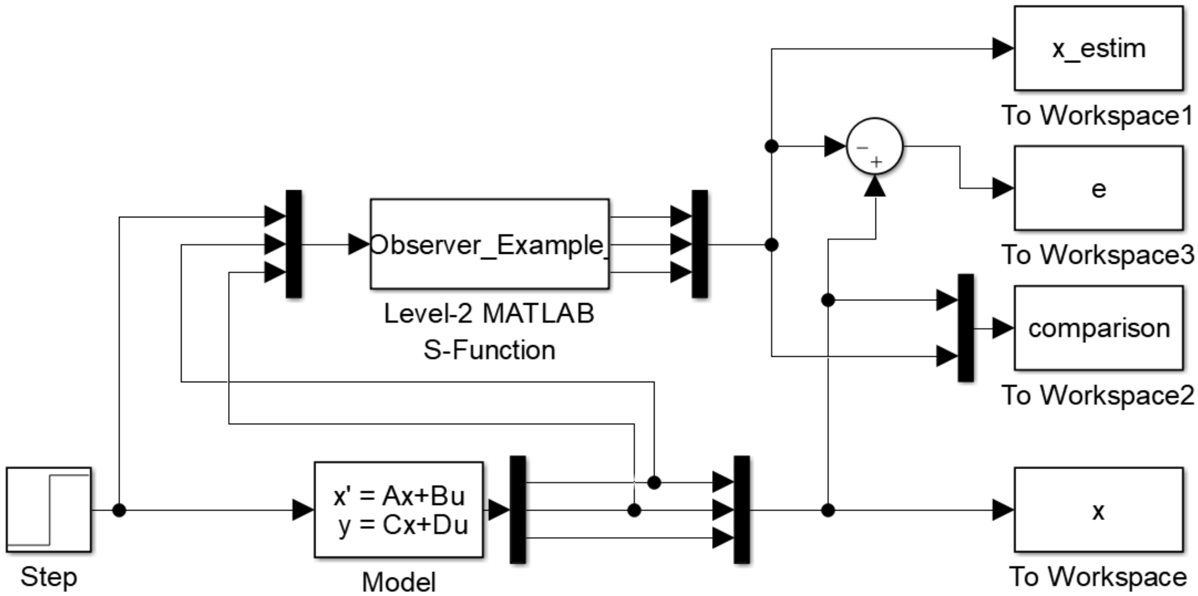

Figure 2.3: Diagram Matlab/Simulink

observer works. The simulation was developed in Matlab/Simulink as shown In Fig. 2.3. The input u is the block labeled as *Step* and whose behavior is described by:

$$u = \begin{cases} 0 & if \;\; t \leq 0 \\ 1 & if \;\; t > 0 \end{cases}$$

Initial conditions of the model are $[0 \;\; 0 \;\; 0]$ while the initial conditions of the state observer are $[0.15 \;\; -0.2 \;\; 0.2]$ in order to separate the initial dynamics of both systems and appreciate how the observer converges to the model dynamics.

The block labeled as *Model* is an state-space block to simulate linear systems and it was programmed as 2.33, this is, with $C = I$ in order to compare the measured output vector y with the estimation of the output vector $\hat{y}$. The block *Observer_Example* is a level-2 S-Function where the equation 2.37 is programmed, this is the block that contains the observer gain matrix R_p, also it is provided with the available state variables x_1 and x_2 in the input vector to be able to estimate x_3. The four blocks left are *To Workspace* blocks which sends the information to the work space in Matlab in order to create the graphics. Note the connections for each To Workspace blocks and

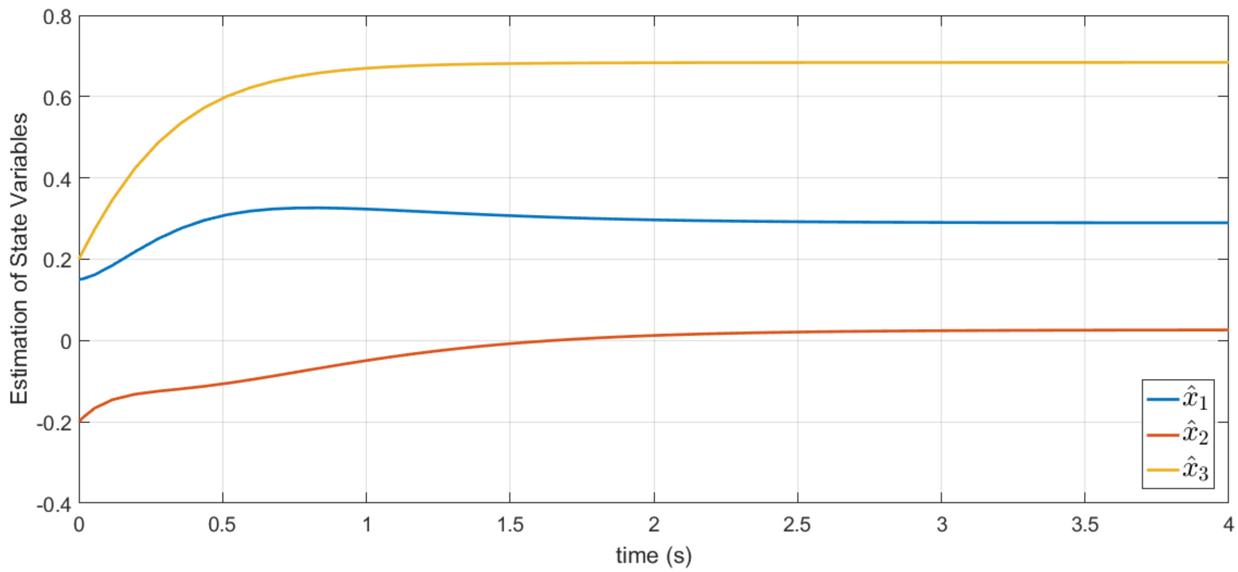

Figure 2.4: Dynamics state variable observer.

particularly in the block e, it is the difference between Model output and the Observer output, i.e., the state estimation error e.

Results are depicted in Fig. 2.4, where it is possible to see that x_3 has been estimated and the state observer also give x_1 and x_2. To compare the dynamics of the state variables x_1, x_2, and x_3 with the estimation $\hat{x_1}$, $\hat{x_2}$, and $\hat{x_3}$ the Fig 2.5 shows both systems where $\hat{3}$ estimates correctly x_3 which is not available for the observer, this means that based only on two variables states the state observer can estimate the third one for this system. Finally, the dynamics error is provided in Fig. 2.6, the error e converges asymptotically to zero, in the time $3s$ it is practical equal to zero.

With this example, the effectiveness of the state observers have been illustrated. However, pole placement method base its effectiveness on the poles chosen, which are chosen arbitrary. Then, an advanced method to calculate the observer gain matrix is through optimal quadratic Lyapunov functions (Chadli et al. 2005). The stability Lyapunov criterion is summarized in Lemma 2.

Lemma 2.(CURRAN 1993) if there exist a real, symmetric, and positive definite matrix $P \in \mathbb{R}^{nxn}$, then:

$$V_{(x)} := x^T P x > 0 \tag{2.42}$$

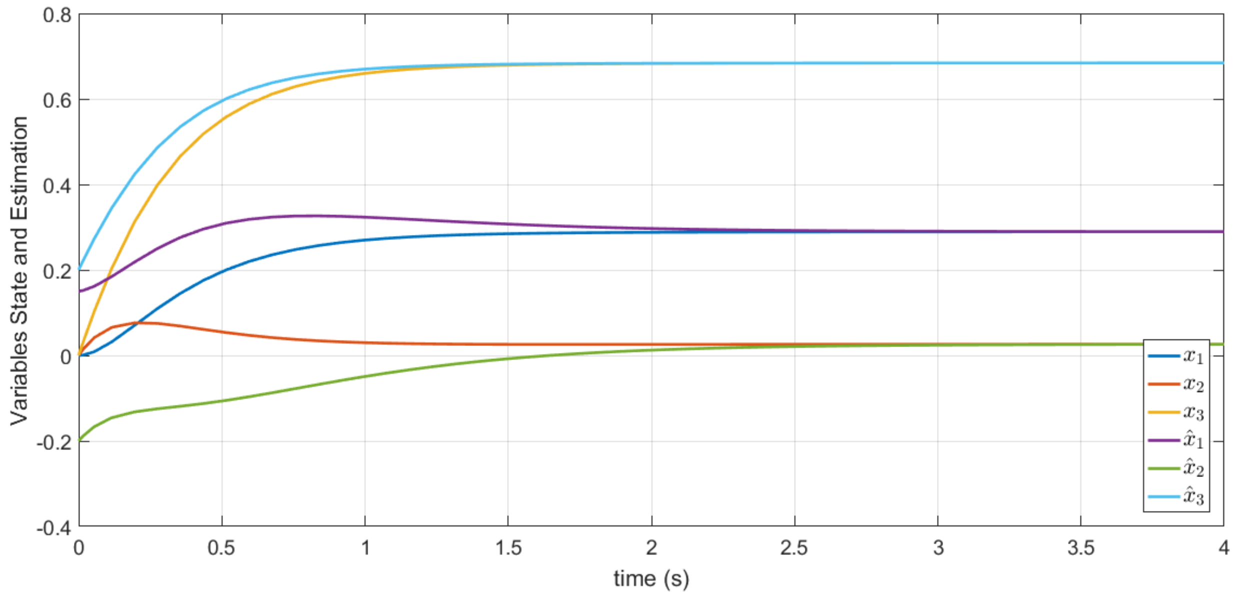

Figure 2.5: Comparison model and observer.

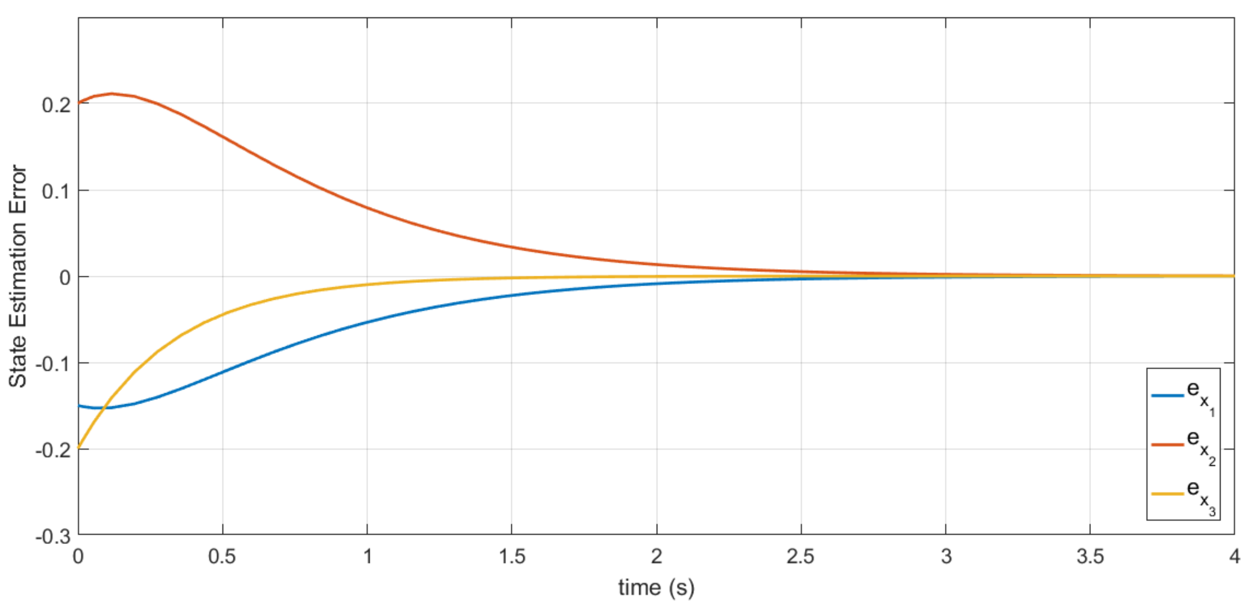

Figure 2.6: Dynamics Error.

And if:

$$\dot{V}_{(x)} := \dot{x}^T P x + x^T P \dot{x} < 0 \tag{2.43}$$

Then, $\dot{x}$ is asymptotically stable.

The objective in this method is to chose P such that $\dot{V}_(x) < 0$. This gain matrix can be calculated solving a Linear Matrix Inequalities (LMI) (Boyd et al. 1994).

Stability Lyapunov criterion is applied to system 2.35 in order to design an appropriated observer gain matrix R. Let propose the following Lyapunov function:

$$V_{(e)} := e^T P e > 0 \tag{2.44}$$

whose derivative is calculated as:

$$\dot{V}_{(e)} := \dot{e}^T P e + e^T P \dot{e} < 0 \tag{2.45}$$

Substituting 2.40 in 2.45 one gets:

$$\begin{aligned} \dot{V}_{(e)} &:= e^T (A - RC)^T P e + e^T P (A - RC) e < 0 \\ &:= e^T (A^T P - C^T R^T P) e + e^T (PA - PRC) e < 0 \\ &:= e^T (A^T P - C^T R^T P + PA - PRC) e < 0 \end{aligned} \tag{2.46}$$

Note that $e^T e$ in 2.46 is a quadratic term, hence definite positive. Then, we need to ensure that:

$$A^T P - C^T R^T P + PA - PRC < 0 \tag{2.47}$$

In order to calculate matrix R through LMI technique, the expression 2.47 must be a linear equation, this is, each term in the expression must to have only one unknown term. Note that R^tP and PR are quadratics terms where both constant matrices are unknown. Then, let define a new constant matrix $\Gamma = PR$:

$$A^T P - C^T \Gamma^T + PA - \Gamma C < 0 \tag{2.48}$$

which can be solved with a computational software as Matlab. Then, solving 2.48 one gets the following observer gain matrix:

$$R_L \begin{bmatrix} -0.0212 & -0.4303 \\ 6.0013 & 1.5661 \\ 1.7683 & -2.6485 \end{bmatrix} \tag{2.49}$$

The state estimation is shown in Fig. 2.7 where $\hat{x_1}$, $\hat{x_2}$ and $\hat{x_3}$ are provided. For this case, also $C = [100;010]$, then only x_1 and x_2 are available and the observer state estimates x_3. In order to compare both observers, initial conditions are also chosen as $x_1(0) = 0.15$, $x_2(0) = -0.2$, and $x_3(0) = 0.2$. Fig. 2.8 illustrate the effectiveness of the observer, $\hat{x}_1$, $\hat{x}_2$, and $\hat{x}_2$ converges asymptotically to x_1, x_2, and x_3 practically before to second 1, which is a better performance that the obtained by pole placement method. To compare the speed of the convergence of both state observers, state estimation error e_p for pole placement and state estimation error e_L for Lyapunov are depicted in Fig. 2.9. The vector $e_L = \begin{bmatrix} e_{L1} & e_{L2} & e_{L3} \end{bmatrix}^T$ where $e_{Li} = x_i - \hat{x}_{Li}$, $i = 1,2,3$ and the vector $e_p = \begin{bmatrix} e_{p1} & e_{p2} & e_{p3} \end{bmatrix}^T$ where $e_{pi} = x_i - \hat{x}_{pi}$, $i = 1,2,3$. It can be observed that the Lyapunov observer has a faster convergence, $e_L \approx 0$ in time $1.5s$. This difference is more evident in the estimations in e_p1 and e_p2 that are the pole placement estimation of the available state variables x_1 and x_2, note that $e_p \approx 0$ after time $2.5s$. Thus, the observer obtained by the Lyapunov criterion is more reliable and more efficient than the observer designed by pole placement.

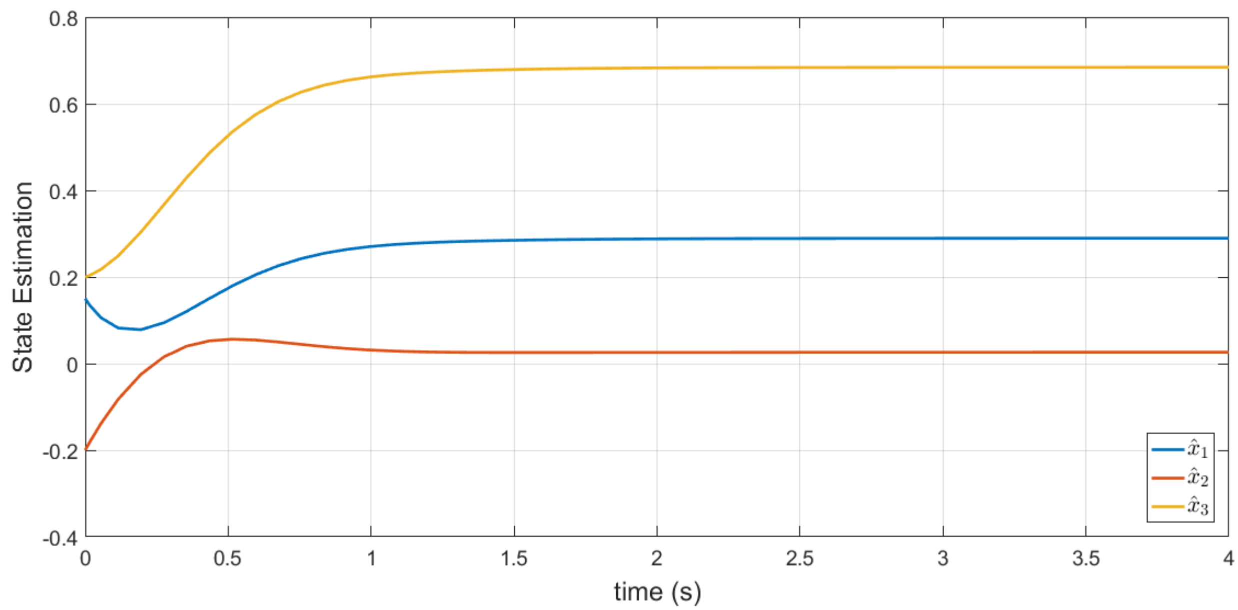

Figure 2.7: State estimation.

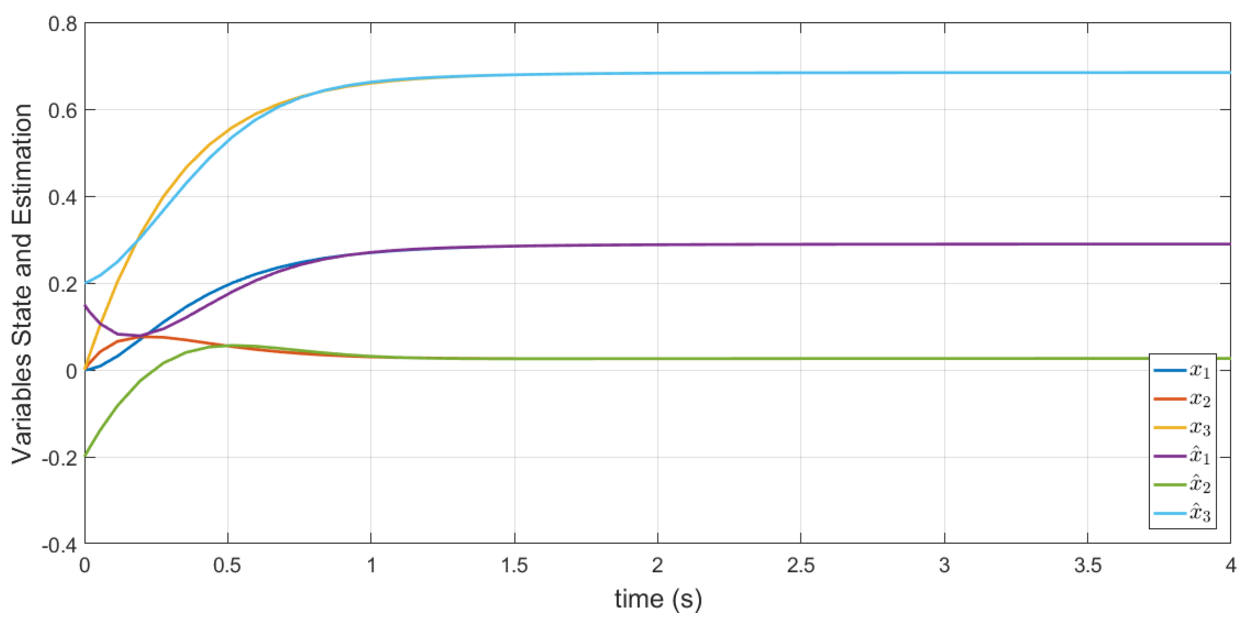

Figure 2.8: State estimation and state variables.

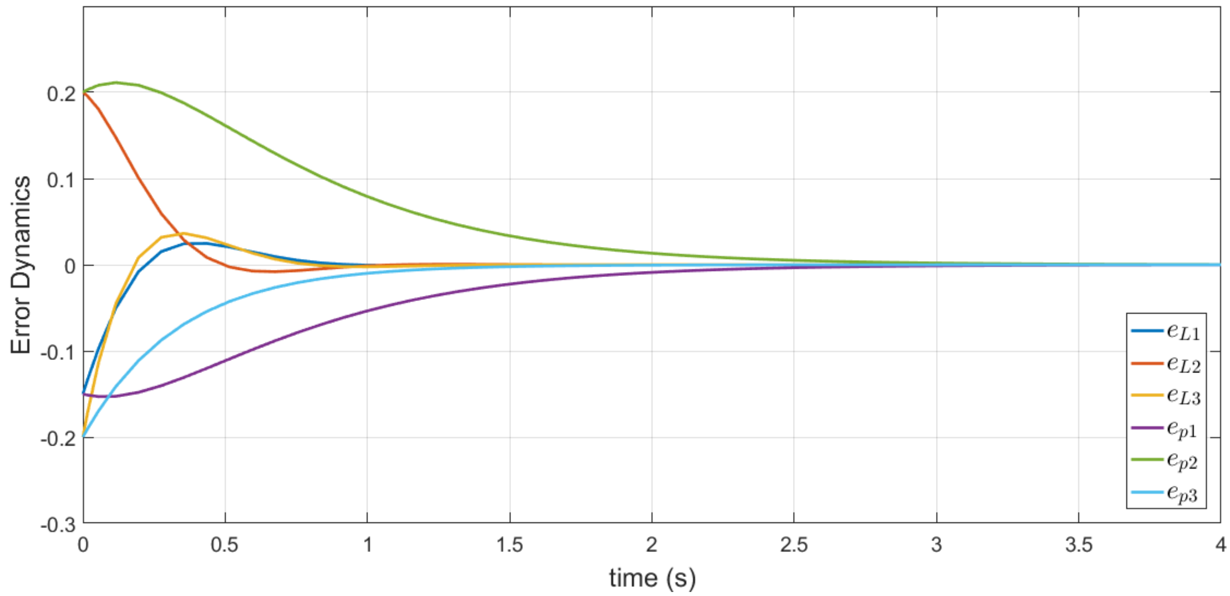

Figure 2.9: Comparison of dynamics error.

According to the analysis above, in this research the observer gain matrices will be designed through Lyapunov criterion developing LMIs.

2.2.1 Examples

In this section two examples are given to develop a state observer. First, the observability matrix of the system is calculated to decide if the observer can be developed. Next, observer gain matrices are calculated according with the LMI 2.48. Finally, the state estimation is presented to prove the effectiveness of the method.

Example 1. Consider the following system:

$$\dot{x} = \begin{bmatrix} 2 & 0 & 0 & 3 \\ 4 & -3 & -5 & 2 \\ -2 & 0 & 3 & -4 \\ 1 & 0 & -2 & -5 \end{bmatrix} x + \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x \tag{2.50}$$

System 2.50 has x_i, $i = 1, \ldots, 4$ state variables of which only x_1 and x_2 can be measured. Note that this is an unstable system, which means that produce an unbounded output for a bounded input. In this simulation the input is a vector $u = [1.5 \;\; 1]^T$

The target is to design a state observer to estimate x_3 and x_4. The first step is to prove the observability of the system in order to know if it is possible to estimate the state variables not available. The corresponding observability matrix is defined as:

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} \tag{2.51}$$

whose rank is 4, then the system 2.50 is observable.

Substituting matrices A and C in the LMI developed 2.48 and solving through software, one gets the following observer gain matrix R:

$$R = \begin{bmatrix} 5.7586 & 1.3445 \\ 5.9464 & 12.2378 \\ -7.2746 & -37.7550 \\ 6.1354 & 9.1000 \end{bmatrix} \tag{2.52}$$

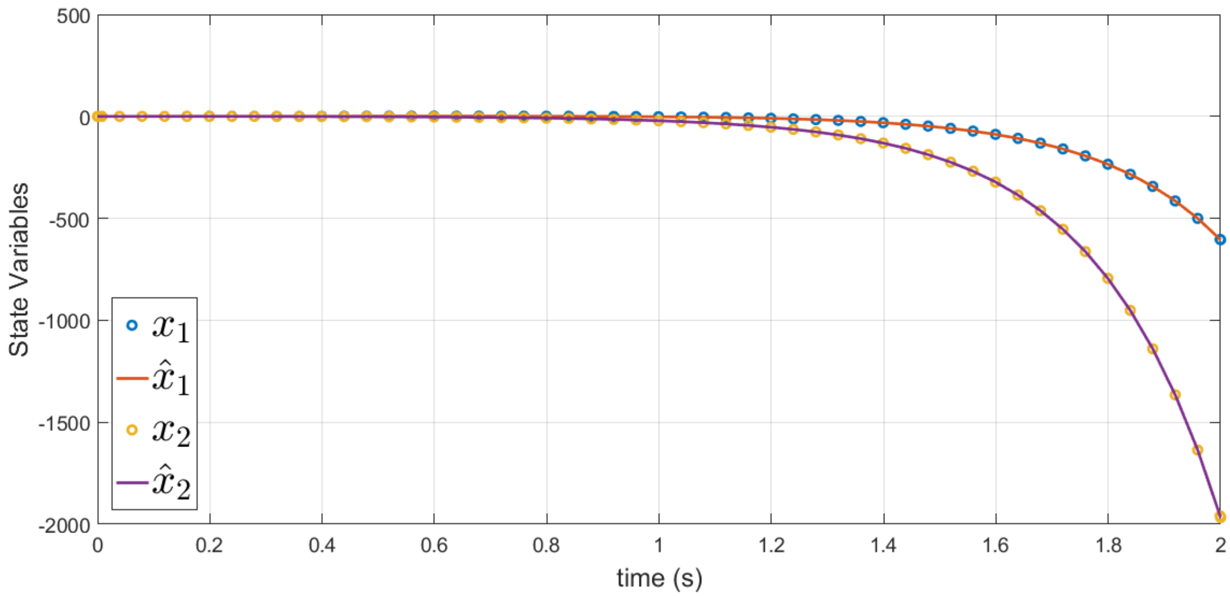

Figure 2.10: State estimation x_1 and x_2.

Simulation results are shown in Fig. 2.10, Fig.2.11, and Fig. 2.12. Measured state variables x_1 and x_2 are compared with the state estimation $\hat{x}_1$ and $\hat{x}_2$ is depicted in Fig. 2.10 where it is possible to see that unbounded outputs tends to negative infinity. The observer state 2.37 is develop with only available state variables x_1 and x_2 but to has a comparison in Fig.2.11 is provided the estimation $\hat{x}_3$ and also the state variable x_3 obtained from the model, $\hat{x}_3$ converges asymptotically to x_3 even if it has an unbounded response to the input u. Finally, the comparison between state variable x_4 and its estimation $\hat{x}_4$ which converges asymptotically is illustrated in Fig. 2.12.

Example 2. In this example a practical system is illustrated with an academical model of an Aircraft. The aircraft shown in Fig. 2.13 considers only 3 degrees of freedom x, y, and θ. The input vector u is given by $u = [F_1 \;\; F_2]^T$. The system is governed by the following differential equations:

$$m\ddot{x} = F_1 cos\theta - F_2 sin\theta - c\dot{x} \tag{2.53}$$

$$m\ddot{y} = F_1 sin\theta + F_2 cos\theta - mg - c\dot{y} \tag{2.54}$$

$$J\ddot{\theta} = rF_1 \tag{2.55}$$

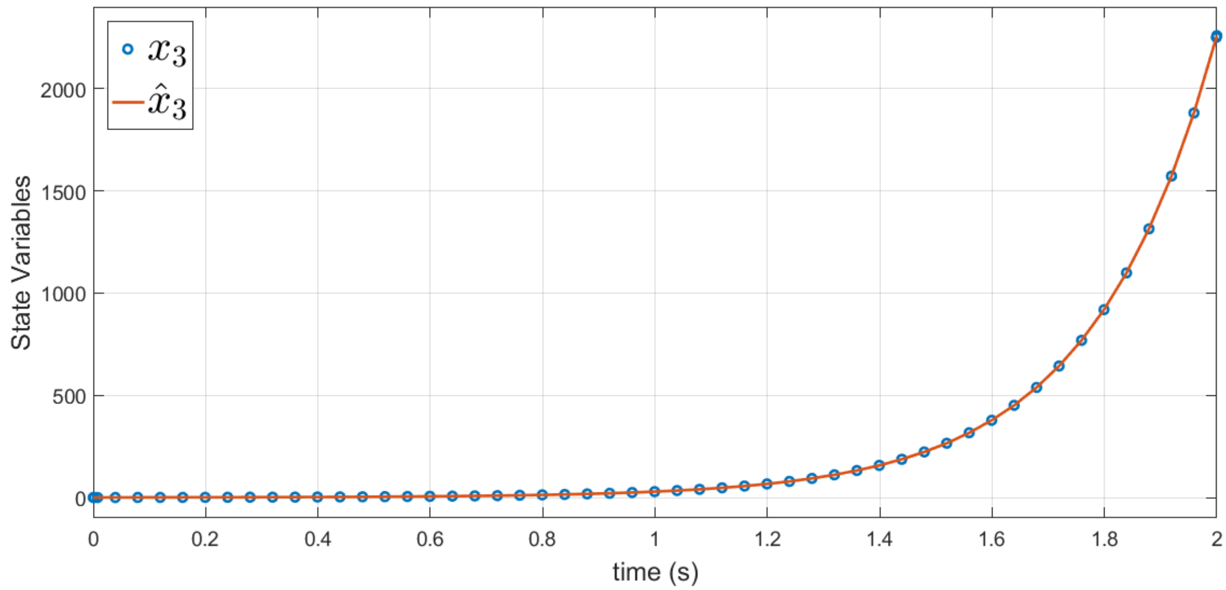

Figure 2.11: State estimation x_3.

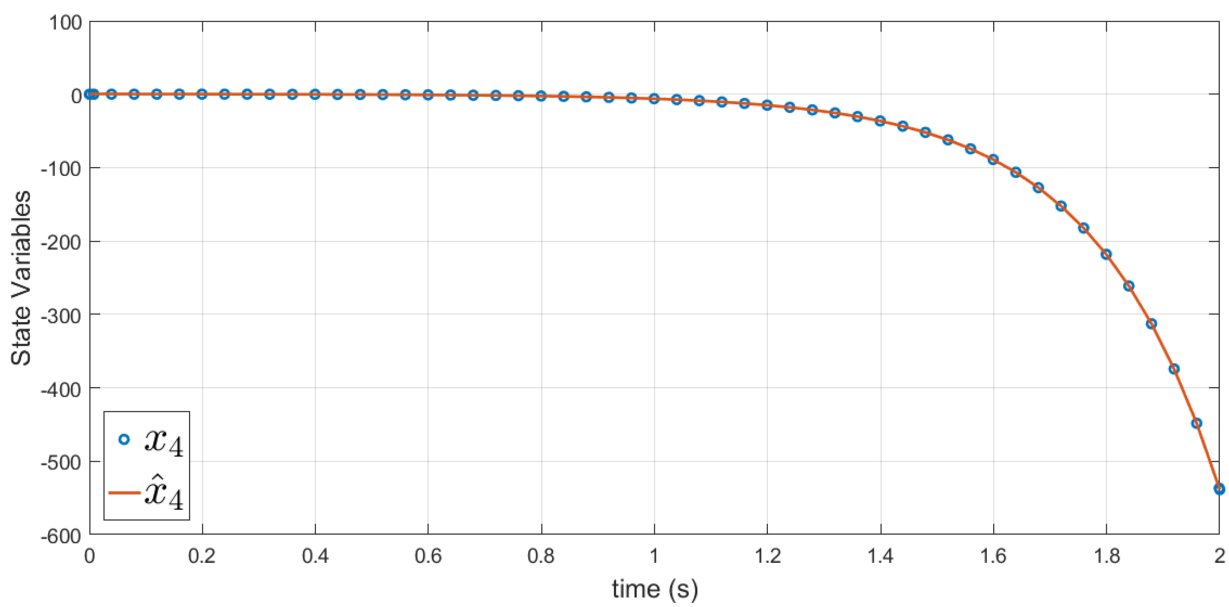

Figure 2.12: State estimation x_4.

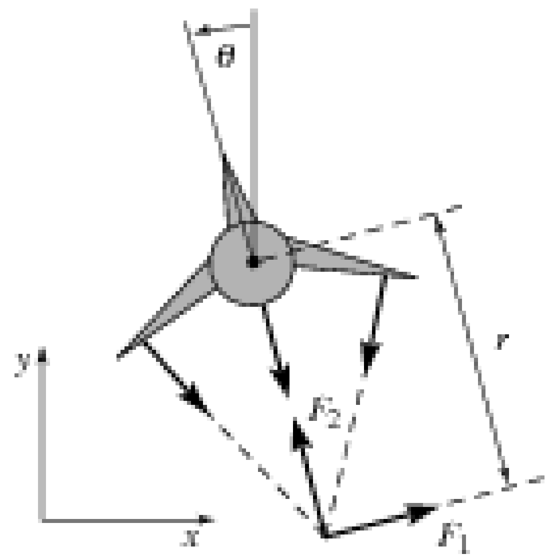

Figure 2.13: Aircraft.

Transforming the differential equations to an state space model, the dynamics of the system is described by:

$$\dot{x_1} = x_2 \tag{2.56}$$

$$\ddot{x_2} = \frac{1}{m}(U_1 cosx_5 - U_2 sinx_5 + mgsinx_5 - cx_2) \tag{2.57}$$

$$\dot{x_3} = x_4 \tag{2.58}$$

$$\ddot{x_4} = \frac{1}{m}(U_1 sinx_5 + U_2 cosx_5 + mgcosx_5 - mg - cx_4) \tag{2.59}$$

$$\dot{x_5} = x_6 \tag{2.60}$$

$$\dot{x_6} = \frac{1}{J} rU_1 \tag{2.61}$$

Note that this is a nonlinear system, hence due to the linear approach of this research, the space state model is linearized and presented with A and B matrices as follow:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{-c}{m} & 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{-c}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m} \\ 0 & 0 \\ \frac{r}{J} & 0 \end{bmatrix} \tag{2.62}$$

The measured state variable are x_1, x_3, and x_5 so that:

$$C \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.63}$$

Proving the observability of the system through its observability matrix which is defined as:

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix} \tag{2.64}$$

with $rank = 6$. Then, it is possible to find a observable gain matrix R from the Luenberger observer 2.37 to estimate the state variables x_2, x_4, and x_6. Consider the LMI 2.48 which is solved with *sedumi* in Matlab software. The gain matrix obtained is the next:

$$R = \begin{bmatrix} 0.8748 & 0.0000 & -3.6768 \\ 1.2913 & 0.0000 & -11.0349 \\ 0.0000 & 0.8748 & 0.0000 \\ 0.0000 & 1.2913 & 0.0000 \\ 3.6768 & 0.0000 & 0.8750 \\ 1.2256 & 0.0000 & 1.2917 \end{bmatrix} \tag{2.65}$$

The Aircraft is an unstable system, then its state variable gives an unbounded output for a bounded input, i.e., its state variables tends to infinity. To appreciate the performance of the observer, the input $u \in \mathbb{R}^2$ has been controlled through methods that are not part of this research study to stabilize the system output. Fig 2.14 illustrates the converge of x_2 and $\hat{x}_2$ where is shown that the observer converges before time $10s$. Fig. **??** shows the estimation of x_4, similar to the estimation of x_2, the convergence occurs before to $10s$ but, in this case, the stabilization of signal is soft. Fig. 2.16 shows the estimation of x_6, this variable is properly estimated until second $12s$. Then, the observer estimates correctly the unavailable state variables. Fig. 2.17 provides the dynamics of all state variables of the observer x_i, $i = 1, \dots, 6$ and the model $\hat{x}_i$, $i = 1, \dots, 6$, note how all state converge asymptotically.

2.3 Multi-Agent Systems

In recent years, developers of automatic controllers have paid attention to cooperative control of multi-agent systems due to the needed of distributed controllers that can synchronize the behavior of a group of systems for mobile robots, UAVs, satellites, aircraft and other systems that involve a collective task Ren et al. (2005), Kuriki & Namerikawa (2014), Cai & Huang (2014) the cooperative control of Multi-agent Systems (MAS) has been widely studied in recent years. Cooperative control studies the dynamics of multi-agent systems (MAS) linked to each other by a communication graph. The graph represents the shared information between agents . The objective of cooperative control is to devise control protocols for the individual agents that guarantee synchronized behavior of the

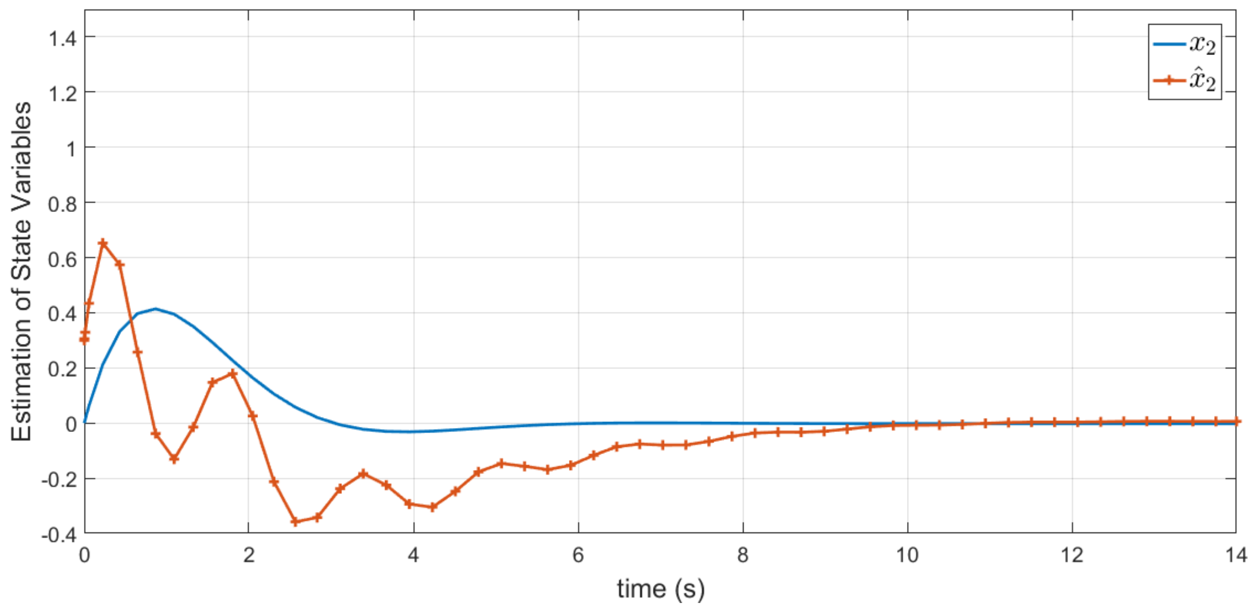

Figure 2.14: State Estimation x_2.

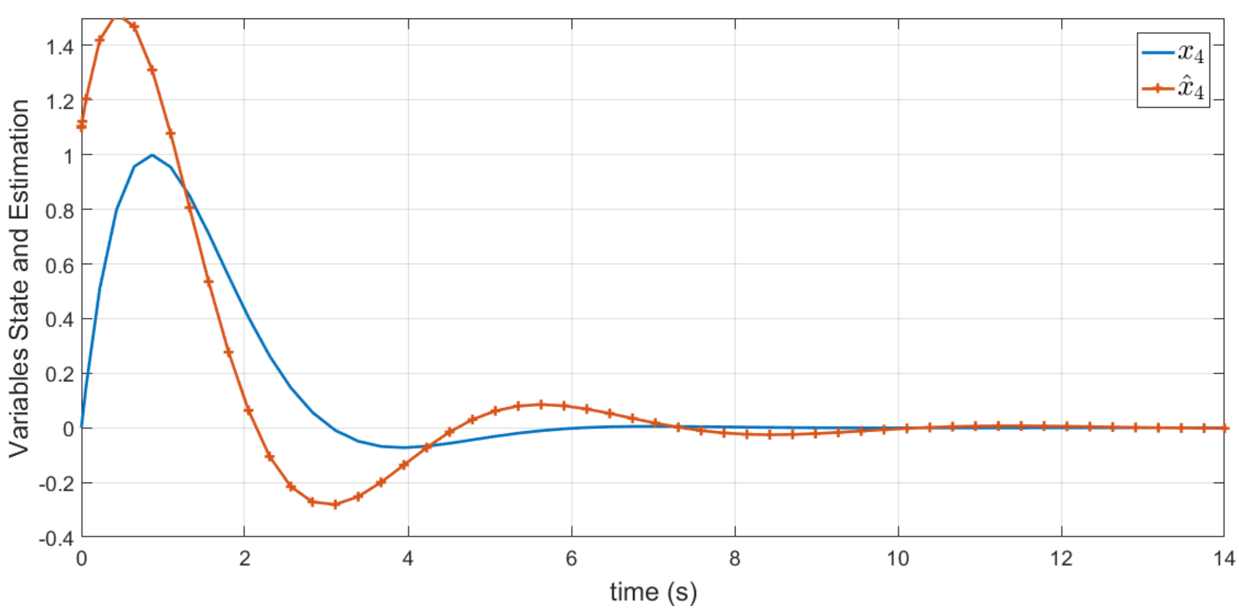

Figure 2.15: State Estimation x_4.

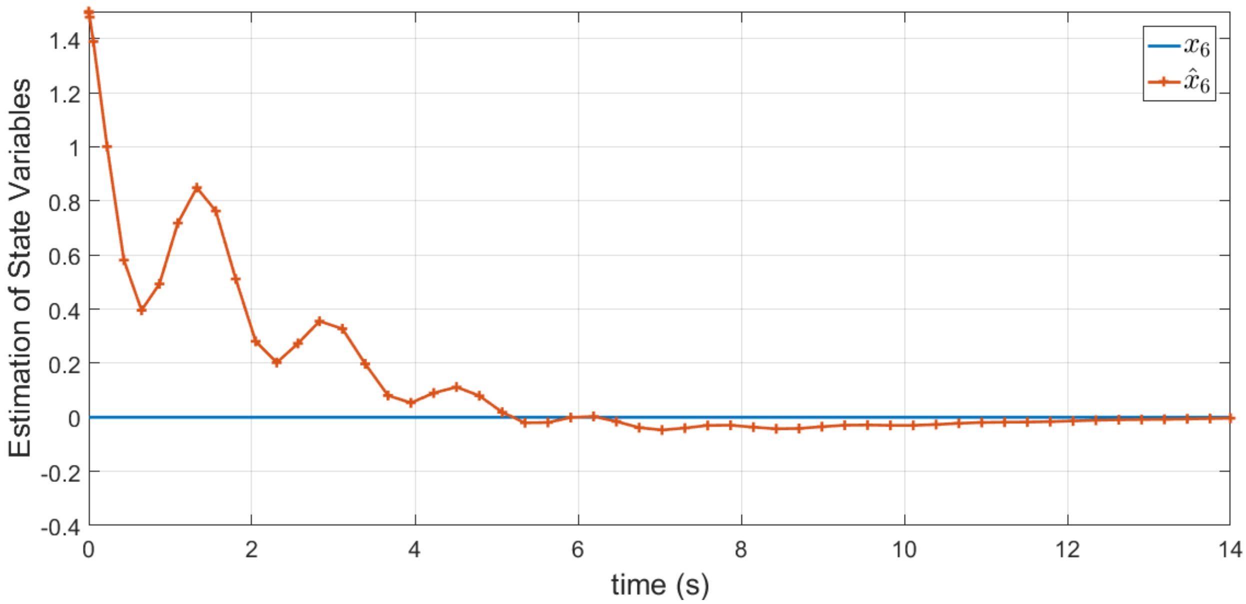

Figure 2.16: State Estimation x_6.

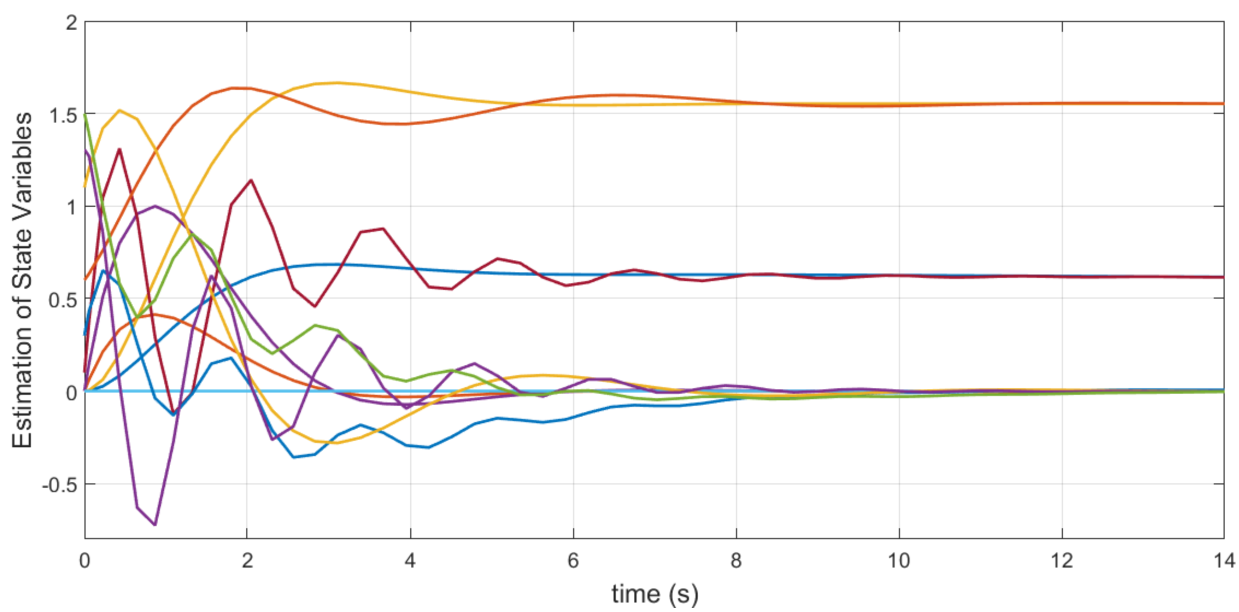

Figure 2.17: Dynamics: model and observer.

states of all the agents in some prescribed sense.

In general, cooperative control can be divided into three main approaches. The first one focuses on a physical synchronization between agents, i.e., if there exists a coordinated physical displacement between the agents or not. Thus the cooperative control can be classified in formation control and non-formation control Ren et al. (2005), respectively. Fig. 2.18 depicts a collection of 6 drones where the task is a coordinated flight, then it is a formation control of MAS; on the other hand, Fig. 2.19 shows a microgrid where the control target is to maintain continuous and constant production of the energy generated by the different systems involved. The second one is related to the existence of a leader agent. If the leader agent exists it is called *Leader-following consensus* of MAS. The leader agent acts as a command generator and provide to the *follower agents* the *desired trajectory*. This trajectory can refer to the displacement or any other state variable in the dynamics, g.e., it is a case of tracking control where the control target is to track the leader, that is why this category is also called *cooperative tracking control* (Lewis et al. 2013). If there not exist a leader agent, the case is called *leaderless consensus* or *average consensus* of MAS, and usually, it is related to the initial condition of the collection of systems Li et al. (2015). The last classification mentioned in this research is the homogeneous and non-homogeneous MAS, this classification divides the MAS depending if all the collection of systems has an identical dynamics behavior (homogeneous) or if at least one of the agents in the collection is different (non-homogeneous) Li et al. (2012), as illustrated in Fig. 2.20 and 2.21. Particularly, in this last category, there exists a remarkable difference in the mathematical model. Represented in space state, Homogeneous MAS is described by the same A and B since all agents have the same dynamical behavior as in 2.66, while non-homogeneous MAS are described by different A_i and B_i because agents can be different from each other in the collection, as presented in 2.67.

$$\dot{x}_i = Ax_i + Bu_i, \quad i = 1, \ldots, N \tag{2.66}$$

$$\dot{x}_i = A_i x_i + B_i u_i, \quad i = 1, \ldots, N \tag{2.67}$$

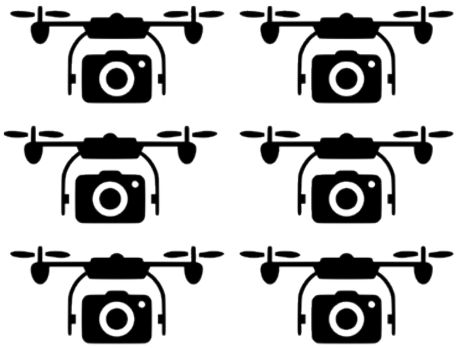

Figure 2.18: A collection of 6 identical drones.

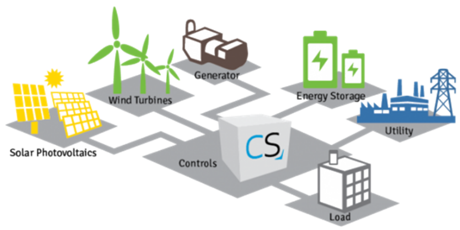

Figure 2.19: A microgrid

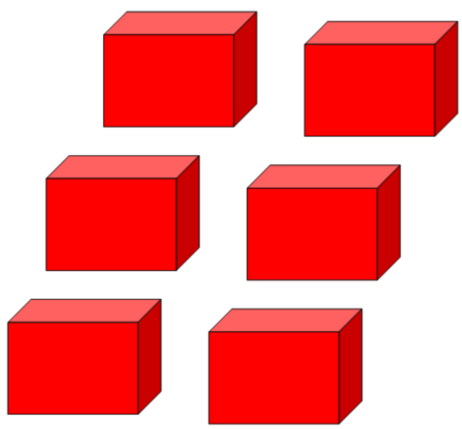

Figure 2.20: Homogeneous collection of 6 elements.

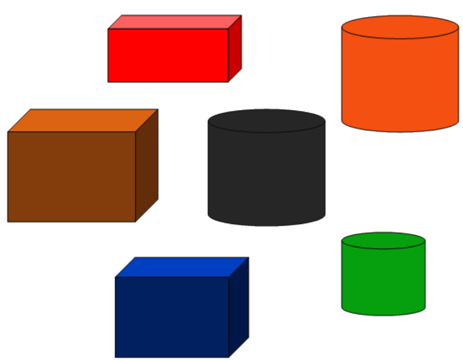

Figure 2.21: Heterogeneous collection of 6 elements.

In order to understand the development of controllers and observers of MAS, broad concepts and mathematical preliminaries should be understood. This thesis does not delve into much of the essential concepts and only provides what is necessary for the development of the state estimators and fault estimator observer-based, which are the main objective of this research.

In this research, graph theory is used to establish the connection between the agents. Then, general concepts of graph theory are introduced in section *Basic Definition and Connectivity* with the intention of clearing the link between algebra and Consensus Control. In this section, main concepts as nodes, edges, graphs, digraphs, graph Laplacian matrix, and so on are defined in order to develop system controllers in the First MAS Approach section and the state and fault estimator in Chapter 3: Materials and Methods. The next section is divided into two different approaches, leader-following graphs, and leaderless graphs, and each of them has particular definitions and mathematical preliminaries that are worth to mention. Even if both methods are important for MAS, pay careful attention to the leader-following case since the fault estimators in Chapter 3 are designed for this configuration.

2.3.1 Basic Definition and Connectivity

2.3.1.1 Leaderless Graphs

A graph is a pair $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ a set of N nodes or vertices, for MAS applications every node represents an agent or an individual system, and $\overline{\mathcal{E}} \subseteq \overline{\mathcal{V}} \times \overline{\mathcal{V}}$ is the set of edges or arcs that represents the connection between the agents. Elements of $\mathcal{E}$ are all the ordered pairs (v_i, v_j) which is an edge or arc from v_i to v_j, and it is represented with an arrow with tail at v_i and head at v_j, this arrow is also know as *edge arrow*. Graphs are usually represented by a diagram that contains all nodes and edges. Basic elements in the graph are depicted in Fig. 2.22 where A) is a node or agent, B) is a row with tail at v_i and head at v_j, this is, the row connect the agent i to the agent j, and C) is a bidirectional row edge which will be explained later.

Each agent v_i has a set of incoming edges and a set of outcoming edges, the set of incoming edges are called *neighbors* of a node v_i and is defined as $N_{in} = \{v_j : (v_j, v_i) \in \mathcal{E}\}$, i.e., the set of

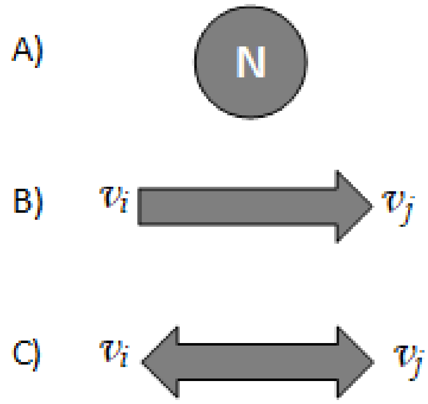

Figure 2.22: Basic elements in the graph.

nodes v_j with the edge arrow incoming to v_i. The number of neighbors N_i of node v_i is equal to its in-degree. In the opposite way, the set of outcoming edge of a node v_i is $N_{out} = \left\{ v_j : (v_i, v_j) \in \mathcal{E} \right\}$, i.e., the set of nodes v_j with edges outcoming from v_i. The number of N_{out} of node v_i is defined as its out-degree.

If the in-degree equals the out-degree for all nodes $v_i \in V$ the graph is said to be balanced. If $\left(v_i, v_j\right) \in \mathcal{E} \Rightarrow \left(v_j, v_i\right)$ the edge row is said to be bidirectional. Associate with each edge $\left(v_j, v_i\right) \in \mathcal{E}$ a weight $a_{ij} > 0$ if $\left(v_i, v_j\right) \in \mathcal{E}$. A graph is said to be undirected if $a_{ij} = a_{ji}, \forall i, j$, otherwise is an directed graph or digraph. In the Fig. 2.23 a graph $\mathcal{G} = \left\{ \mathcal{V}, \mathcal{E} \right\}$ is given, the set of N nodes and $\mathcal{E}$ edges in the graph are the following:

$$
\begin{aligned}
&\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6\} \\
&\mathcal{E} = \{(v_1, v_2), (v_1, v_3), (v_2, v_1)(v_2, v_5), (v_3, v_1), (v_3, v_4), (v_3, v_6), (v_5, v_4), (v_6, v_3)\}
\end{aligned}
\tag{2.68}
$$

A directed path is a sequence of nodes $v_0, v_1, ..., v_r$ such that $(v_i, v_{i+1}) \in E, i \in \{0, 1, \ldots, r-1\}$. Node v_i is said to be connected to node v_j if there is a directed path from v_i to v_j. Graph G is said to be strongly connected if v_i , v_j are connected for all distinct nodes $v_i, v_j \in V$.

A directed tree is a connected digraph where every node except one, called the root, has in-degree equal to one. A spanning tree is a directed tree formed by edge arrows that connects all

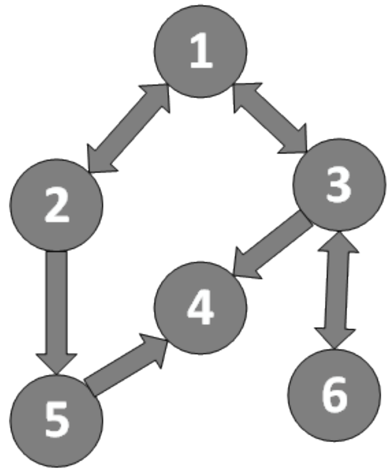

Figure 2.23: Graph Example.

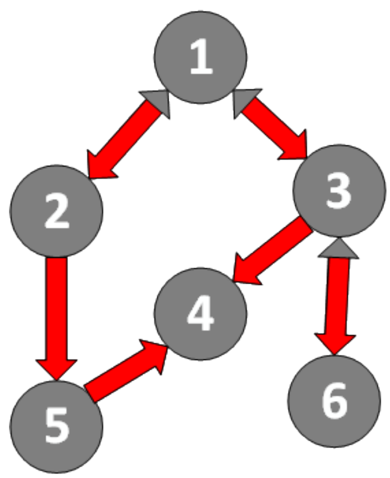

Figure 2.24: Spanning tree in red.

the nodes of the graph. A graph is said to have a spanning tree if a subset of the edges forms a directed tree. In other words, if a graph has a spanning tree means that all agents are reachable from a common agent. The graphs can contain more than one spanning tree but, in order to reach the consensus, at least one spanning tree must to exist in the topology. A spanning tree is depicted in Fig. 2.24 in red where the agent 1 is the root, note that node 1 can reach all agents in the digraph. It is worth to mention that this spanning tree is not the only one in the graph, with the exception of the agent 4, any other agent can be considered as the root of an spanning tree.

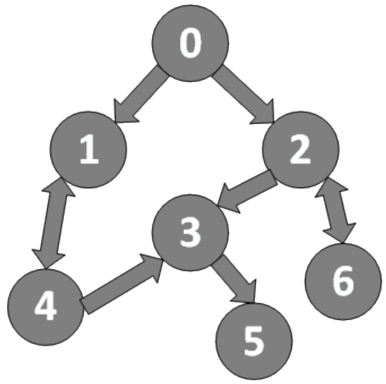

Figure 2.25: Leader-following graph example.

2.3.1.2 Leader-Following Graphs

As explained above, there exists MAS that considers a leader agent and this category is studied with small differences in the graph topology in comparison with the leaderless case explained before. In this section is explained a detail the graph theory applied to the leader-following case.

For MAS leader-following an augmented graph is defined as $\overline{\mathcal{G}} = \{\overline{\mathcal{V}}, \overline{\mathcal{E}}\}$ where $\overline{\mathcal{V}} = \{v_0, v_1, v_2, ..., v_N\}$ is the set of $N+1$ nodes or vertices (agents for MAS applications) and $\overline{\mathcal{E}} \subseteq \overline{\mathcal{V}} \times \overline{\mathcal{V}}$ is the set of edges which is the set of ordered pairs which express the connection between all agents including the leader one. Note that $\overline{\mathcal{E}} \subseteq E$. Without loss of generality, it is common to label the leader agent as v_0, hence in this work the leader agent will be labeled as 0 in the mathematical representations and in the graphs topologies as is shown in Fig. 2.25. In this theory, leader agent is not affected by any other follower agent, with this, leader agent can be considered as an a command generator exosystem that generates the desired target trajectory (Lewis et al. 2013) and then the pair v_i, v_0 with $i = 1, 2, ..., N$ does not exist.

Generally, the leader agent can reach directly just a subset of follower agents. Nevertheless, the augmented graph $\overline{\mathcal{G}}$ must to contain a spanning tree with the leader node as the root, this means that there exist a directed path from v_0 to v_i where $i = 1, .., N$, i.e., the information of the leader node is transmitted to all follower agents directly or indirectly in the graph. In the graph example in Fig. 2.25 the leader node can reach all follower nodes through multiple directed paths. If there exist a subset of k $1 \leq k < N$ nodes in the graph $\mathcal{G}$ with $S_k = \{n_1, n_2, ..., n_k\}$ and $S_k \subseteq N$ which

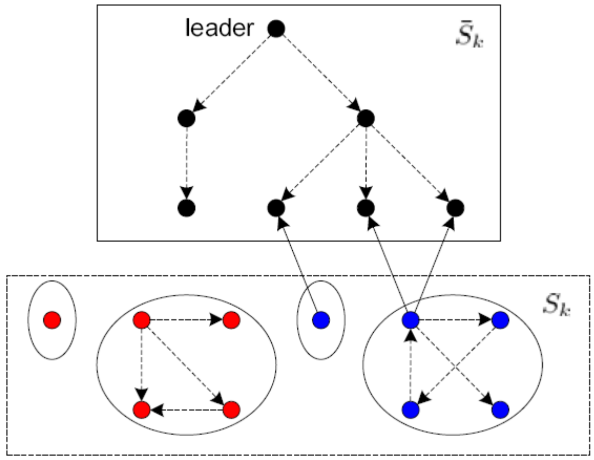

Figure 2.26: Isolation of agents in the graph topology.

do not have access to the leader information, this implies that the set S_k do not have access to the information of the other follower agents either, this set S_k is isolated of the rest of the agents and the synchronization to the leader agent cannot be achieved. This is depicted in Fig. 2.26.

Therefore, all the works that deal with Leader-following consensus need the following assumption:

Assumption 2. (Lewis et al. 2013) The augmented graph $\overline{\mathcal{G}}$ contains a spanning tree with the leader node as the root. In other words, there is a directed path (not necessarily unique) from the leader node to every follower node.

2.3.2 Mathematical Preliminaries

Given the edge weights a_{ij}, a graph can be represented by an adjacency or connectivity matrix $A = [a_{ij}]$ with weights $a_{ij} > 0$ if $(v_j, v_i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Note that $a_{ii} = 0$, i.e., there is not self loops. Therefore, most researches define the graph as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ since every graph is related to an adjacency matrix $\mathcal{A}$. A graph without self loops is called a simple graph and if there exist at least one loop, it is called a multi-graph. To illustrate those definitions, In Fig. 2.27 two different graphs with leader-following configuration are presented, a) depicts a multigraph with self loops in nodes 3 and 4 while b) depicts a simple graph, as any other presented before, without self

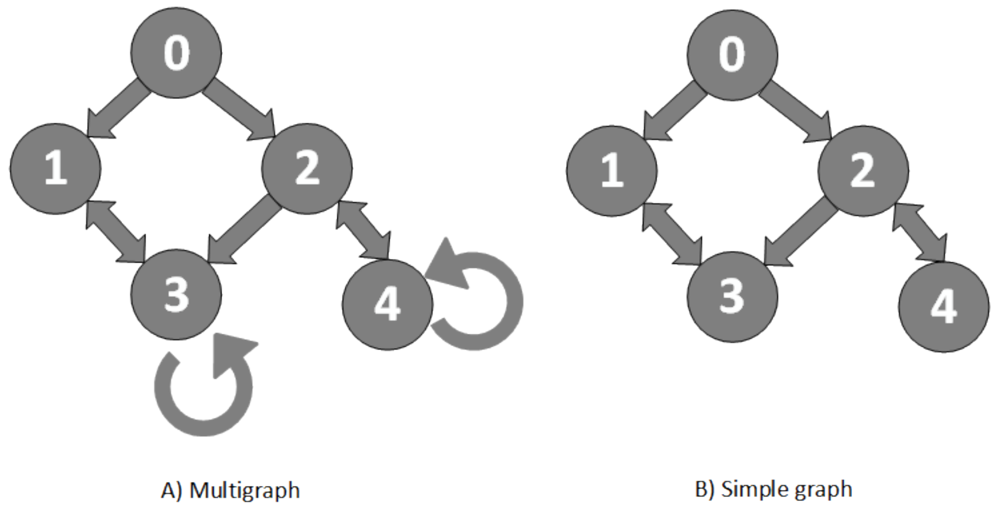

Figure 2.27: Illustration of multigraphs and simple graphs.

loops.

Define the weighted in-degree of node v_i as the i-th row sum of $\mathcal{A}$:

$$d_i = \sum_{j=1}^{N} a_{ij} \tag{2.69}$$

and the weighted out-degree of node v_i as the i-th column sum of $\mathcal{A}$ as:

$$d_i^o = \sum_{j=1}^{N} a_{ji} \tag{2.70}$$

Note that row sum is expressed with a_{ij} and the column sum with a_{ji}.

If the weights $a_{ij} = a = ji$, $\forall i, j$, i.e., the graph is undirected, the adjacency matrix $\mathcal{A}$ is symmetric, $A = A^T$. A graph is said to be weight balanced if the weighted in-degree equals the weighted out-degree for all i, this is the same definition that $a_{ij} = 1 \forall i, j$ if $\left(v_j, v_i\right) \in \mathcal{E}$. Hereinafter, to simplify the examples, all graphs will be weight balanced graphs even if it is not mentioned, i.e., all weights $a_{ij} = 1$.

It is define the diagonal in-degree matrix $D = diag\left(d_i\right)$ and the Laplacian matrix $L = D - \mathcal{A}$. Then, with a given graph as the presented in Fig. 2.23 it is possible to construct the adjacency matrix $\mathcal{A}$, the diagonal in-degree matrix D, and the Laplacian matrix L. Particularly, the Laplacian matrix is of a big interest since this one contains all the communication informacion in the graph and it has many useful properties that can be fully studied by the reader in (Lewis et al. 2013) and which leads to **Lemma 1** and **Lemma 2**.

Lemma 1. Under the assumption that eigenvalues of (L) are ordered as $\lambda_1 < \lambda_2 <, ..., < \lambda_N$, Laplacian matrix L has rank N-1, i.e., λ_0 is nonrepeated and all nonzero eigenvalues have positive real parts if and only if graph $\mathcal{G}$ has a spanning tree (Li et al. 2009, 2011).

Graphs are unnecessary if the Laplacian matrix is given. However, in the rest of this thesis, both graphs and matrices are given in order to detail all examples and developments. From Fig. 2.23, matrices $\mathcal{A}$, D and L are given:

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

Eigenvalues λ_i of Laplacian matrix L are calculated in order to show that Theorem 1 is fulfill. $\lambda_5 = 0$ and is not repited and the rest of eigenvalues λ_i are positive definite.

$$
\begin{aligned}
\lambda_1 =& 2 \\
\lambda_2 =& 1 \\
\lambda_3 =& 3.4142 \\
\lambda_4 =& 2 \\
\lambda_5 =& 0 \\
\lambda_6 =& 0.5858
\end{aligned}
\tag{2.71}
$$

For the leader-following case, given an augmented graph $\overline{\mathcal{G}} = \{\overline{\mathcal{V}}, \overline{\mathcal{E}}, \mathcal{A}\}$ it is necessary to define how the follower agents are connected to the leader. In practice, only a subset of follower agents has direct information from the leader, then, if the agent i is connected to the leader, there exist an edge (v_0, v_i) with a weight g_i and it is said that the node is pinned to the leader, thus, weights g_i are known as pinning gains (Duan et al. 2009, Li et al. 2009, Wang & Chen 2002). Remember that the leader agent is an exosystem whose is not affected by any other agent, then, $(v_i, v_0) \notin \mathcal{E}$ i.e., the node v_0 does not receive information

Remark. Only for graphs in the leader-following case, the weights a_{ij} define the connections between the follower agents and the weights g_i define the connection between the subset of follower agents that can observe the leader.

It is defined as the weighted diagonal pinning gain matrix:

$$G = diag(g_i) \tag{2.72}$$

The matrix 2.72 is related with the leader connection and leads to the Lemma 2.

Lemma 2. Under the Assumption 2, the eigenvalues λ_i of $(L+G)$ have positive real parts (Lewis et al. 2013).

The adjacency matrix $\mathcal{A}$, the diagonal in-degree matrix D, the Laplacian matrix L, and the diagonal pinning gain matrix G are obtained for the graph depicted in Fig. 2.25 which contains a leader node labeled as 0. By inspection in the graph, there exists a spanning tree with the leader agent as the root, with this, Assumption 2 is fulfilled.

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$, \quad L = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

calculating the eigenvalues of $(L+G)$ one gets:

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 2 \\ \lambda_3 &= 2.6180 \\ \lambda_4 &= 0.3820 \\ \lambda_5 &= 2.6180 \\ \lambda_6 &= 0.3820 \end{aligned} \tag{2.73}$$

Lemma 2 is satisfied with all eigenvalues $\lambda_i > 0$.

2.3.3 Examples

In this section, the main concepts studied in the theoretical framework are illustrated through multiple examples. The examples will serve to understand the basics of graph theory that will be used during the development of this research.

Examples $1-4$ are related to leaderless case, the adjacency matrix $\mathcal{A}$, the diagonal in-degree matrix $diag(d_i)$, and the Laplacian matrix L are calculated based on the graph topology given. Examples 5 and 6 are concerned to leader-following case, then, the adjacency matrix $\mathcal{A}$, the diagonal in-degree matrix $diag(d_i)$, the Laplacian matrix L, and the diagonal pinning gain matrix $diag(g_i)$ are calculated. For all examples, the main objective is the construction of the corresponding matrices and the analysis of the Lemma 1 and Lemma 2, which are fulfilled in all cases presented.

Example 1. In this example, a directed graph or digraph with 3 agents labeled as 1, 2, and 3 is depicted in Fig 2.28, this topology is a leaderless case where the condition of having a spanning tree is accomplished. By inspection, one can realize that Adjacency matrix $\mathcal{A} \in \mathbb{R}^3$, in-degree matrix $D \in \mathbb{R}^3$, and Laplacian matrix $L \in \mathbb{R}^3$ since three agents are involved.

According to $\mathcal{A} = [a_{ij}]$ with weights $a_{ij} = 1$ if $\left(v_j, v_i\right) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise, 2.70 and with $L = D - \mathcal{A}$, one can define:

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

In order to prove Lemma 1, the eigenvalues of Laplacian matrix L are given:

$$\lambda_1 = 2$$

$$\lambda_2 = 0$$

$$\lambda_3 = 3$$

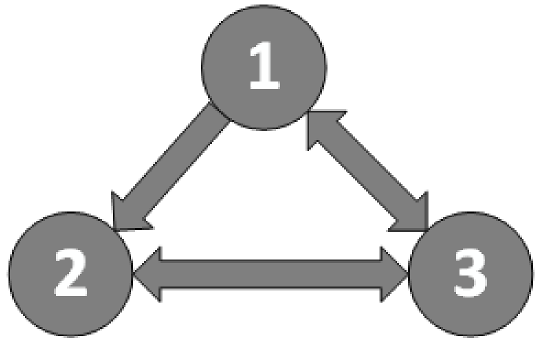

Figure 2.28: 3 agents in leaderless configuration and with directed communication.

Where is possible to see that all eigenvalues are positive definite.

Example 2. Similar to Example 1, the graph in 2.29 consider only 3 agents in a leaderless consensus case. In this graph topology, the communication between agents is undirected, i.e., for all agents the communication is bidirectional. Then, it is supossed that $\mathcal{A} = \mathcal{A}^T$. Further, $\mathcal{A} \in \mathbb{R}^3$, $D \in \mathbb{R}^3$, $L \in \mathbb{R}^3$, and all eigenvalues are positive definite.

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \; L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Calculating its eigenvalues:

$$\lambda_1 = 0$$

$$\lambda_2 = 3$$

$$\lambda_3 = 3$$

Then, Lemma 1 is fulfilled.

Example 3. Fig. 2.30 shows a directed graph with 6 agents in a leaderless configuration. It is evident that $\mathcal{A} \in \mathbb{R}^6$, $D \in \mathbb{R}^6$, $L \in \mathbb{R}^6$, and all eigenvalues are positive definite.

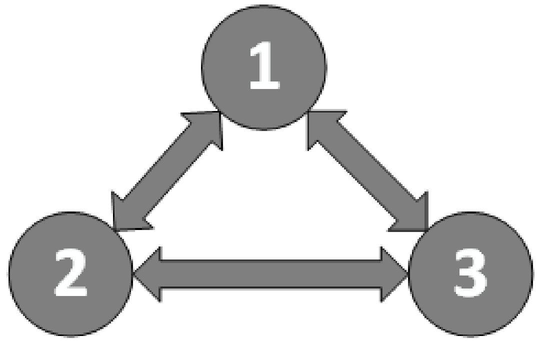

Figure 2.29: 3 agents in leaderless configuration and with undirected communication.

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; L = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 \\ -1 & -1 & 4 & -1 & 0 & -1 \\ 0 & -1 & -1 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

Eigenvalues of L are:

$$\begin{aligned} \lambda_1 &= 4.4142 \\ \lambda_2 &= 0 \\ \lambda_3 &= 1.5858 \\ \lambda_4 &= 1 \\ \lambda_5 &= 5.2361 \\ \lambda_6 &= 0.7639 \end{aligned}$$

Example 4. Fig. 2.31 depicted an almost identical graph than Example 3. Note that the only difference lies in the connection between agent 1 and agent 2, while Example 3 considers a directed connection (v_1, v_2), this graph consider a bidirectional connection $(v_1, v_2) = (v_2, v_1)$. With

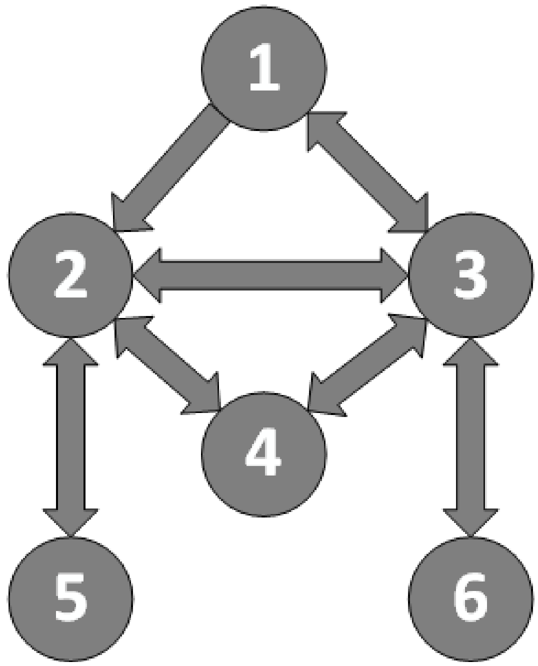

Figure 2.30: 6 agents in leaderless configuration and with directed communication.

the above, Fig. 2.31 is an undirected graph that satisfies $\mathcal{A} = \mathcal{A}^T$. Adjacency matrix $\mathcal{A}$, in-degree matrix D and Laplacian matrix L are given:

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 \\ -1 & -1 & 4 & -1 & 0 & -1 \\ 0 & -1 & -1 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

Note the difference in the eigenvalues between Example 3 and Example 4 whose change significantly changing only one edge.

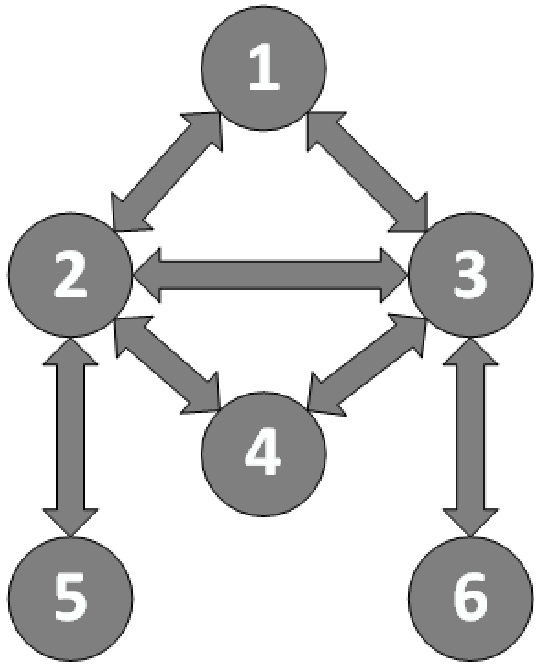

Figure 2.31: 6 agents in leaderless configuration and with undirected communication.

$$
\begin{aligned}
\lambda_1 &= 0 \\
\lambda_2 &= 0.7639 \\
\lambda_3 &= 1.2679 \\
\lambda_4 &= 2 \\
\lambda_5 &= 4.7321 \\
\lambda_6 &= 5.2361
\end{aligned}
$$

Example 5. In order to clarify the configuration Leader-following for MAS, in this example, an undirected graph with a leader agent and 3 follower agents is depicted in Fig. 2.32. Without loss generality, it is usual to label the leader agent as 0 since it can be considered as an *exosystem* and its communication edges or arcs are not part of the adjacency matrix $\mathcal{A}$, instead, it is represented by the pinning gain matrix G. Note that, only agent 1 and agent 2 has a directed communication with the

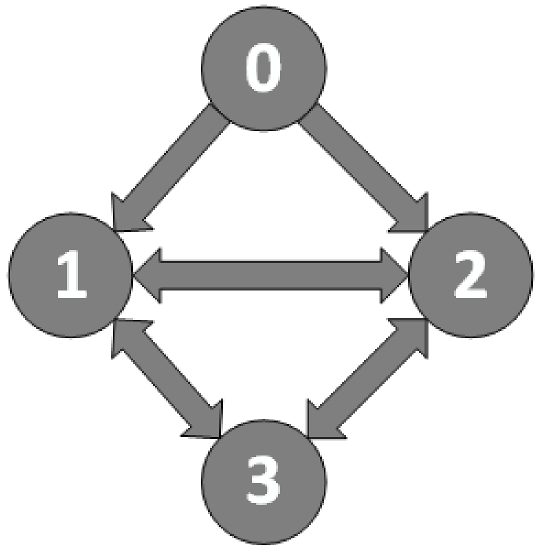

Figure 2.32: Leader agent and 3 followers with undirected communication.

leader agent although the condition of the existence of an spanning tree is fulfill, the leader agent 0 can reach any other agent in the topology directly or indirectly by following the edge arrows. With 2.72, one gets:

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

According with Lemma 2, all eigenvalues of $(L+G)$ are definite positive only if there exist a spanning tree. Since this condition is fulfill, it is possible to that Lemma 2 is accomplished when the eigenvalues are calculated:

$$\lambda_1 = 0.5858$$
$$\lambda_2 = 3.4142$$
$$\lambda_3 = 4$$

Example 6. Finally, this example presents a directed graph with a Leader-following topology. This particular graph is called *formation graph* due to that descending hierarchy in which the agent i only receives information from its immediate neighbor, the agent j, with a larger hierarchy. Note that only agents 1 and 2 have directed communication with the leader. Following, matrices $\mathcal{A}$, D, L and G are provided:

$$\mathcal{A}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&1&0&0&0&0\end{bmatrix},\ D=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{bmatrix}$$

$$,\ L=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\-1&0&1&0&0&0&0&0\\0&-1&0&1&0&0&0&0\\0&0&-1&0&1&0&0&0\\0&0&-1&0&0&1&0&0\\0&0&0&-1&0&0&1&0\\0&0&0&-1&0&0&0&1\end{bmatrix},\ G=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Eigenvalues of $(L+G)$ are given in 2.74. Note that Lemma 2 is accomplished.

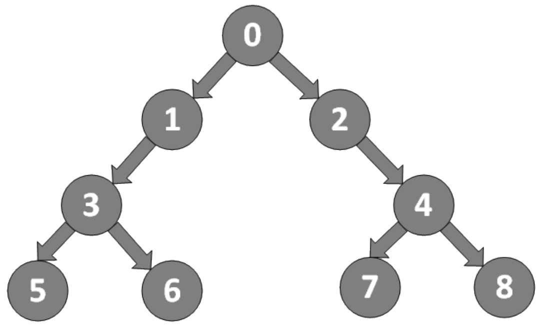

Figure 2.33: Leader agent and 8 followers with directed communication

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 1 \\ \lambda_3 &= 1 \\ \lambda_4 &= 1 \\ \lambda_5 &= 1 \\ \lambda_6 &= 1 \\ \lambda_7 &= 1 \\ \lambda_8 &= 1 \end{aligned} \tag{2.74}$$

2.3.4 First MAS Approach

2.3.4.1 Problem Statement and Motivation

Consider a system of $N+1$ identical agents with a linear dynamic, there is a group of N followers and one leader. The dynamic of i-th follower agents can be represented as:

$$\dot{x}_i = Ax_i + Bu_i \qquad i = 0, ..., N \tag{2.75}$$

Where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ is the control input. Matrices A and B are assumed to be stabilizable and detectable. Considering without loss of generality that the leader agent is indexed by 0 and the follower agents indexed by $1,\dots,N$, the control input of the leader is $u_0 = 0$. It is necessary a Lemma and an assumption in order to design a control law for the system 2.75.

Lemma 1. Laplacian matrix L has rank $N-1$, i.e., $\lambda_1 = 0$ is nonrepeated and all nonzero eigenvalues have positive real parts if and only if graph G has a spanning tree (Lewis et al. 2013).

Assumption 1. The Graph topology G contains a directed spanning tree with a leader as the root node. In other words, there is a directed path (not necessarily unique) from the leader to the N followers (Li et al. 2015).

Leader node in the Laplacian matrix L has in-degree equal to zero. Then, if the leader node is indexed as node 0 and followers as nodes $1,\dots,N$, L can be partitioned as:

$$L = \begin{bmatrix} 0 & 0_{1xN} \\ L_2 & L_1 \end{bmatrix}$$

Where $L_2 \in \mathbb{R}^{Nx1}$ and $L_1 \in \mathbb{R}^{NxN}$. In this representation of L, it will be more easy work only with the L_1 term which determines the topology of the followers. Remaining that leader agent has input control zero, the needed information to develop the controller is in L_1.

Leader-follower consensus is reached when:

$$\lim_{t\to\infty}(x_i(t) - x_0(t)) = 0, \quad \forall i \in N \tag{2.76}$$

Where $x_i(t)$ is the state of the i -th agent and $x_0(t)$ is the leader node (previously indexed as 0 agent). Thus, the main objective of the research is to design a local distributed controller u_i for all follower nodes such that 2.76 be accomplished.

In order to design u_i for cooperative control works Li et al. (2010) and Zhang et al. (2011) has been proposed a static consensus protocol that use the information between the agents and the leader node, given as:

$$u_i = cK \sum_{j=1}^{N} a_{ij}(x_i - x_j) \qquad i = 1, \ldots, N$$

Where $c > 0$ is the common coupling weight among neighboring, $K \in \mathbb{R}^{mxn}$ is the feedback gain matrix, and a_{ij} is the weight for edge (v_j, v_i) of the adjacency matrix A associated with G. It is worth to mention that if the weights a_{ij} are not relevant, then a_{ij} is set equal to 1 if $(v_j, v_i) \in E$.

It is simple to see that the task consists in to find a $c > 0$ and a $K \in \mathbb{R}^{sxn}$ that satisfies 2.76. One way to solve this problem is with the following lemma.

Lemma 2 (Li et al. (2010), Zhang et al. (2011)): Under assumption 1. The N agents described by (1) reach leader-follower consensus under the protocol (4) with $K = -B^T P^{-1}$, $c \leq \frac{1}{min_{i \in N} Re(\lambda_i)}$ where λ_i,$i = 1, \ldots, N$, are the nonzero eigenvalues of L_1 and $P > 0$ is a solution to the linear matrix inequality (LMI) proposed:

$$AP + PA^T - 2BB < 0 \tag{2.77}$$

Nevertheless, to find the $c > 0$ that satisfices 2.76 it is necessary that every single agent has global information in order to know $min_{i \in N} Re(\lambda_i)$ that corresponds to the minimum real part of the eigenvalues of the Laplacian matrix L_1.

To solve this limitation in the static protocol, some works as Li, Ren, Liu & Fu (2013), Li, Ren, Liu & Xie (2013) propose an adaptative consensus protocol that has the advantage that there exists a dynamical coupling gain. In these works, a fully distributed consensus protocol is given, in other words, the control input u_i, $i = 1, \ldots, N$ depends on only agents dynamics and the information of the neighbors for topologies that consider undirected graphs. All the same, there is not a protocol developed for directed graph topologies that overcome that limitation.

Now, in order to solve the problem on fully distributed consensus protocol for directed graph, a novel adaptative protocol is presented in this research. This protocol uses only information of the state variables of the local agent and the neighbor's information considering a zero input for the leader agent.

2.3.4.2 Distributed Adaptative Consensus Protocol Design

This development example proposes a distributed consensus protocol that use only information of the local agent and the neighbors without global information of the Laplacian matrix for the system (1). The protocol includes time-varying coupling weights and a monotonically increasing function as follows:

$$
\begin{aligned}
u_i &= c_i p_i(\xi_i^T P^{-1} \xi_i) K \xi_i \\
\dot{c}_i &= \xi_i^T P^{-1} \xi_i
\end{aligned}
\tag{2.78}
$$

Where c_i is the dynamical coupling weight with $c_i(0) \leq 1$, $p_i(\xi_i^T P^{-1} \xi_i)$ is a monotonically increasing function determined later with $p(w) \leq 1$ for $w > 0$, $P > 0$ is a solution of the LMI 2.77, and $K \in \mathbb{R}^{mxn}$ and $\Gamma \in \mathbb{R}^{nxn}$ both are matrices gain determined in Theorem 1.

The advantage of this control input is the time-varying coupling weights $c_i(t)$. This dynamical $c_i(t)$ is calculated in base of the consensus error ξ_i and is independent of any eigenvalue of the Laplacian matrix. In order to defined every factor in the consensus protocol the next Theorem is given:

Theorem 1. Under the assumption 1, the leader-follower consensus problem for the system (1) can be solved through the adaptative protocol (6) with $K = -B^T P^{-1}$,$\Gamma = P^{-1} B B^T P^{-1}$, and $p_i(\xi_i^T P^{-1} \xi_i) = (1 + \xi_i^T P^{-1} \xi_i)^3$, where $P > 0$ is a solution of the LMI 2.77. Moreover, c_i converges to some finite steady-state value.

Proof: See reference Li et al. (2015).

Remark 1. LMI 2.77 allows to determine the matrices K and Γ but both can be solved by Riccati equation $A^T Q + QA + I - QBB^T Q = 0$ as in Zhang et al. (2011), Tuna (2008), Hengster-Movric et al. (2013) with $K = -B^T Q$ and $\Gamma = QBB^T Q$ and the monotonically increasing function

$p_i(\xi_i^T P^{-1} \xi_i)$ with the same methodology that in this research for protocol (6).

Remark 2. This research provides a solution for consensus problem in fully distributed and adaptative way in the sense that every single agent has a control input u_i that depends only on the local agent and the information of the neighbors and it is independent of the global information of the Laplacian matrix. Although there exist similar works as Li, Ren, Liu & Fu (2013), Li, Ren, Liu & Xie (2013), this research focuses on solving the problem for the case of directed graph topologies.

Remark 3. Protocol 2.78 provide an extra freedom for design with respect to Li, Ren, Liu & Fu (2013), Li, Ren, Liu & Xie (2013) because of the implementation of p_i. This factor is dynamical and converges asymptotically to 1 while consensus error ξ_i converges asymptotically to zero. Then, this extra factor ensures that when the consensus is reached this factor does not affect the behavior of the system and the protocol is reduced to consensus protocols in Li, Ren, Liu & Fu (2013), Li, Ren, Liu & Xie (2013).

2.3.4.3 Simulation Example

For a graphical analyze one example is provided. The topology is shown in Fig. 2.34 where is possible to see that the leader node is the node indexed by 0 and only the node 1 can obtain information of it, also exist 5 followers that should reach the consensus only through node 1 information. This topology satisfies assumption 1.

Since Li et al. (2015) mentions that if the weights a_{ij} are not relevant, then a_{ij} is set equal to 1 if $(v_j, v_i) \in E$, for this example the weight is a set of 1. Then, the error consensus ξ_i is a difference of the state variables of the nodes x_i and x_j, that is $x_i - x_j$.

The system dynamic for this example is given by 2.79:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}; \; B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.79}$$

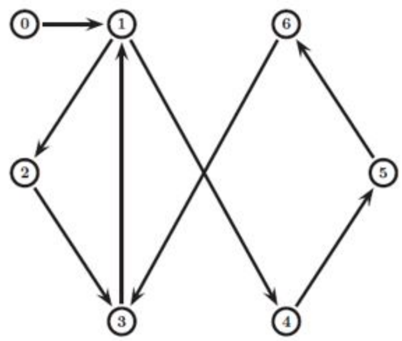

Figure 2.34: Topology for the example.

Every factor of the input control u_i is determined by Theorem 1 solving the LMI 2.77 with Matlab toolbox SeDuMi. To satisfies the characteristics of protocol 2.78 the initial state $c_i(0)$ was randomly chosen between $(1,3)$. Fig. 2.35 shows the consensus error $\xi_i = x_i - x_j$ where the error of every state variable of the x_i agent converges asymptotically to zero, thus the consensus is reached. Fig. 2.36 depicts the dynamical coupling gain c_i that satisfies the condition $c_i(0) \leq 1$, it is possible to see that c_i converges to finite steady-state values. Finally, Fig. 2.37 proves the asseveration in remark 3, that p_i converges asymptotically to 1.

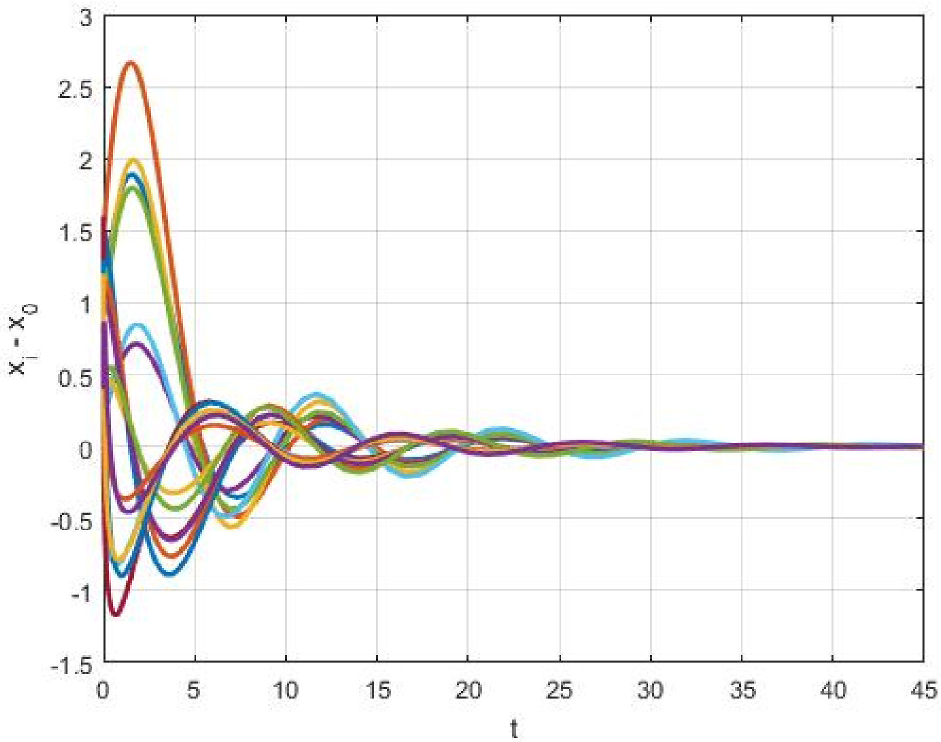

Figure 2.35: Consensus error $x_i - x_0$

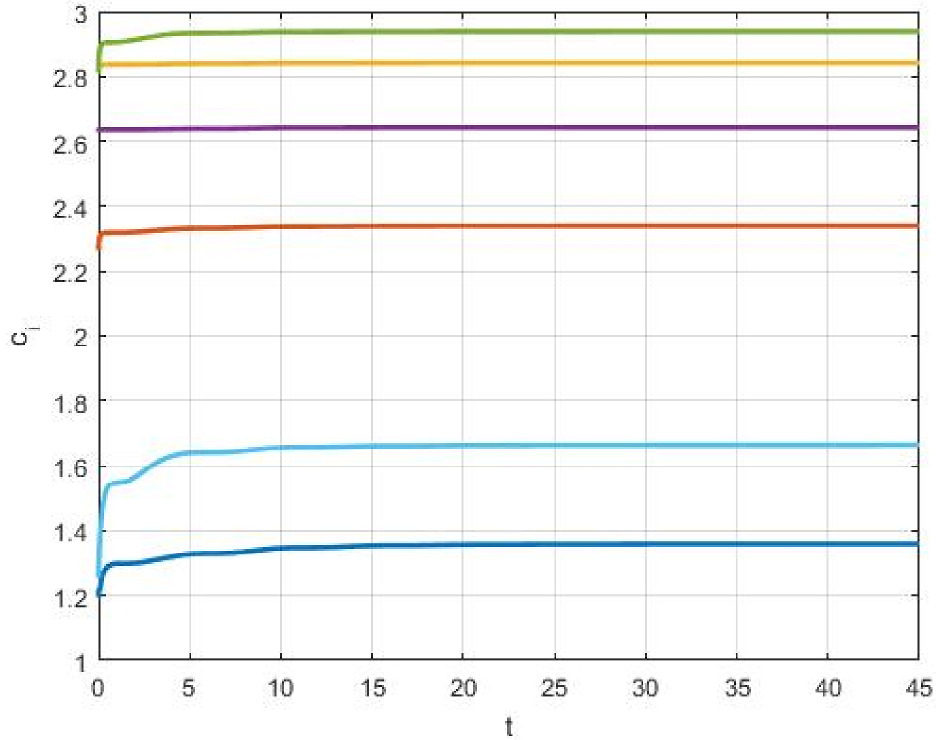

Figure 2.36: Adaptative coupling gain c_i

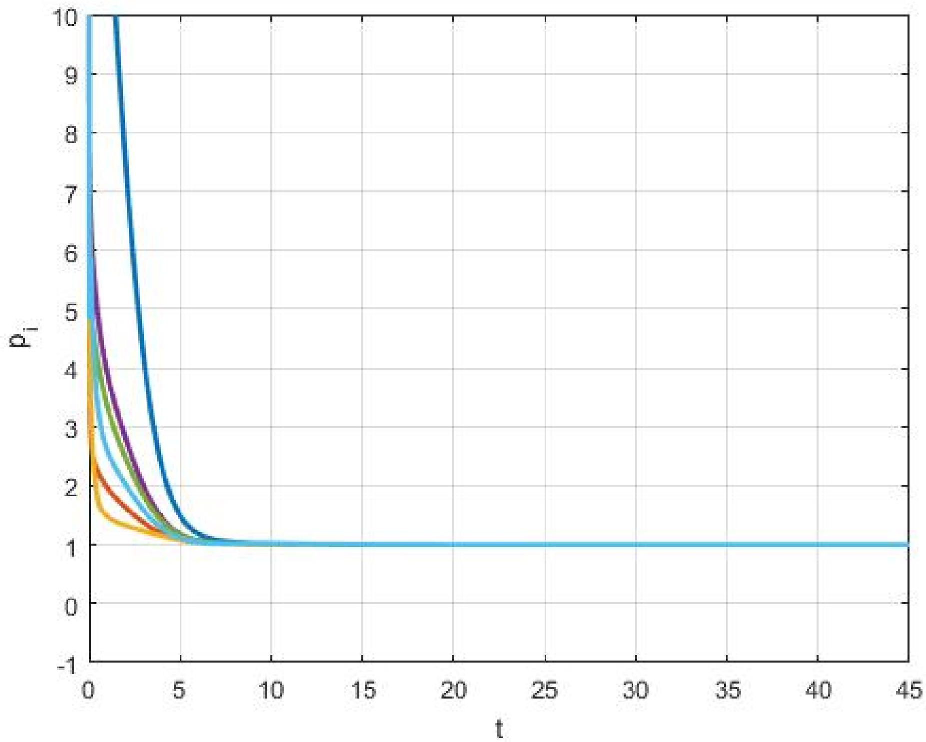

Figure 2.37: Values for p_i for each agent i

Simulation Conclusions. The consensus protocol proposed in this development example allows to reach the consensus for linear multi-agents systems with directed topologies. This is an advantage over Li, Ren, Liu & Fu (2013), Li, Ren, Liu & Xie (2013) that proposed the adaptative consensus protocol only for undirected graph topologies. In comparison with the static protocol Li et al. (2010), Zhang et al. (2011) the adaptative protocol does not use any global information and for that reason it is properly a fully distributed consensus protocol. The main objective of the distributed control is the independence of the controller of every single agent but some proposed protocols depend of the Laplacian matrix and is worth to mention that the Laplacian matrix is obtained in base of the global information and then then the controller is not completely independent.

Chapter 3

Linear fault estimation of MAS

In recent years, cooperative control for multi-agent systems (MAS) has been widely studied as a solution for problems where multiple systems have to collaborate to reach a common goal. In this scenario, an individual control law for each one of the agents cannot provide a satisfactory performance of the global control task, e.g., distributed cooperative control of microgrids Bidram, Davoudi, Lewis & Guerrero (2013), Bidram, Davoudi, Lewis & Qu (2013), Nasirian et al. (2014), cooperative formation control of autonomous underwater vehicles Das et al. (2016), formation control of unmanned aerial vehicles (UAVs) Kuriki & Namerikawa (2014), cooperative control of manipulators Li et al. (2016), to mention a few. Among different topics of MAS, the leader-following, also called cooperative tracking control, has become the most popular consensus problem. In this case, the leader sends information to the agents, then, the controller tries to reduce the error so all follower agents can track the desired trajectory generated by the leader Lewis et al. (2013). In the literature, there is extensive research about this problem, for example, in Wang & Wu (2012) a leader-following formation control for a second-order nonlinear multi-agent system under fixed and switching topologies is exposed. Authors in Cai & Huang (2014) present a leader-following control for multiple spacecraft systems. In Zhang et al. (2014), an adaptive technique for fault estimation is proposed. The works of Ma et al. (2015) and Zhao et al. (2017) present two different approaches for second-order multi-agent systems, one is an optimal strategy and the other is an event-triggered strategy for the communication graph, respectively. However, few works are dedicated to detect and isolate faults for MAS.

The purpose of a fault diagnosis system is to generate an alarm when a fault occurs as well as to detect, locate, and estimate the magnitude of the faulty element. Few works have been proposed for fault diagnosis in MAS. In particular, for leader-following cooperative control problems, in Shi et al. (2014) a bank of optimal robust observers to detect and isolate actuator faults was designed. In Chen & Song (2015) an actuator fault detection module for directed graphs was developed, in Li (2015) a controller for multi-agent systems subject to loss of actuator effectiveness with an adaptive observer was proposed. Furthermore, few works related to fault estimation have been reported, e.g., in Zhou et al. (2014), fault tolerant cooperative control is achieved with a sliding mode observer to estimate faults. In Ye et al. (2017), an adaptive observer to estimate the states and bias faults with multiple leaders is presented, an unknown input observer (UIO) is designed in Wu et al. (2018) to estimate states and faults for directed graphs in presence of exosystem disturbances, and Yang et al. (2018) presents a distributed adaptive fault estimation algorithm for undirected graphs. Nevertheless, to the best of the authors' knowledge, proportional-integrative (PI) Distributed Fault Estimation Observers (DFEO) Zhang et al. (2015) have not been fully explored for fault estimation in leader-following applications.

In this chapter, a Proportional-Integral Distributed Fault Estimation Observer (PI-DFEO) is proposed for MAS with a distributed approach. In order to reach the main goal, it is considered that the control law of each agent depends on its own information and the information provided by its neighbors. In order to show the connection between the agents, a graph topology is considered. Moreover, the PI-DFEO estimates both, the system states and actuator faults. The proposed approach can be applied to directed and indirect graphs, according to graph theory. Furthermore, to guarantee robustness against measurement noise and disturbances a H_∞ criterion was considered. As a result, sufficient conditions to compute the observer gains are given by a set of feasible Linear Matrix Inequalities. Finally, the performance of the proposed method is tested through numerical examples of formation control.

3.1 Problem statement

Consider a collection of $N+1$ identical agents where the follower nodes are represented with faults through the following space state model:

$$\begin{aligned}
\dot{x}_i(t) &= Ax_i(t) + Bu_i(t) + Hf_i(t); \\
y_i(t) &= Cx_i(t);
\end{aligned} \tag{3.1}$$

where $i = 1, \ldots, N$, indicates the i-th agent in the multi-agent system, $x_i(t) \in \mathbb{R}^n$ is the state vector, $u_i(t) \in \mathbb{R}^m$ is the control input vector, $f_i(t) \in \mathbb{R}^r$ represents the system component or actuator fault vector, and $y_i(t) \in \mathbb{R}^p$ is the output vector. The pair (A, C) is assumed to be observable and matrix H is constant with appropriate dimensions.

Without loss of generality, the leader agent labeled with the subscript 0 is modeled as follows:

$$\dot{x}_0(t) = Ax_0(t). \tag{3.2}$$

Remark 1. Note that the leader agent does not consider any input, this holds for the standard multi-agent systems theory and for the purpose of this work. However, it is worth to mention that a controller for the leader agent can be developed independently applying any other single agent control theory (Lewis et al. 2013).

For the development of this work, the following assumption and lemma are considered:

Assumption 1. In order that all follower agents can track the state of the leader, the graph must have a spanning tree with the leader node as the root, i.e., the leader node can send information directly or indirectly to all follower agents (Lewis et al. 2013).

Lemma 1. Under Assumption 1, the matrix $(L+G)$ is nonsingular. In addition, this matrix is positive defined (Lewis et al. 2013).

In order to estimate states, the following distributed observer is proposed:

$$\begin{aligned}
\dot{\hat{x}}_i(t) &= A\hat{x}_i(t) + Bu_i(t) - R\zeta_i(t) + H\hat{f}_i(t); \\
\hat{y}_i(t) &= C\hat{x}_i(t); \\
\hat{f}_i(t) &= -\Gamma F\left(\zeta_i(t) + \int_{t_f}^{t} \zeta_i(t)dt\right);
\end{aligned} \tag{3.3}$$

where $i = 1, \ldots, N$, $\hat{x}_i(t) \in \mathbb{R}^n$ is the estimated state, $\hat{y}_i(t) \in \mathbb{R}^p$ is the estimated output, $\zeta_i(t) \in \mathbb{R}^q$ is the relative output estimation error of the i-th agent in the communication graph defined later, $R \in \mathbb{R}^{n\times q}$ is the observer gain matrix to be designed, and $\hat{f}_i(t)$ is the estimated fault. To deal with fault estimation, a distributed PI fault estimator $\hat{f}_i(t)$ is proposed where the relative output estimation error ζ_i is used. Additionally, an integral term of ζ_i is added that allows the observer to have a faster convergence to the fault, $F \in \mathbb{R}^{r\times p}$ is the fault estimator gain matrix to be designed, and matrix $\Gamma = \Gamma^T > 0$ is the learning rate. Note that t_f indicates the time when the fault occurs.

Then, the problem is reformulated to find the gain matrices R and F, such that the estimation error between system (3.1) and observer (3.3) tends asymptotically to zero, for all follower agents. Communication between follower agents and the leader agent is established through the pinning gain matrix G defined in Section 3.

3.2 Main contribution

The main idea of this Chapter is to estimate states and faults of the follower agents in a MAS using the observer proposed in (3.3). Then, according to the MAS theory, the relative output estimation error ζ_i, which expresses the information exchanged between the agents, is defined as (Lewis et al.

2013):

$$
\begin{aligned}
\zeta_i(t) = \sum_{j=1}^{N} a_{ij} \Big(\big(\hat{y}_i(t) - y_i(t)\big) - \big(\hat{y}_j(t) - y_j(t)\big) \Big) \\
+ g_i \Big(\big(\hat{y}_i(t) - y_i(t)\big) - \big(\hat{y}_0(t) - y_0(t)\big) \Big); \\
i = 1, \ldots, N;
\end{aligned}
\tag{3.4}
$$

where $\hat{y}_i(t)$ is the observer output of the i-th agent, $\hat{y}_j(t)$ is the observer output of the j-th agent who constitute the neighbors of the agent i in the collection of systems, g_i represents the nodes pinned to the leader node.

From Remark 1, since the leader agent can be treated independently and an observer for the leader agent is not needed, it is reasonable to assume for the MAS model that the output estimation error of the leader agent is equal to zero, i.e., $\hat{y}_0(t) - y_0(t) = 0$. Then, the relative output estimation error is reduced to the following expression:

$$
\begin{aligned}
\zeta_i(t) = \sum_{j=1}^{N} a_{ij} \Big(\big(\hat{y}_i(t) - y_i(t)\big) - \big(\hat{y}_j(t) - y_j(t)\big) \Big) \\
+ g_i \big(\hat{y}_i(t) - y_i(t)\big); \quad i = 1, \ldots, N;
\end{aligned}
\tag{3.5}
$$

In order to develop the DFEO, it is necessary to define the dynamic error for both, state estimation and fault estimation. The state estimation error vector for the i-th agent is defined as,

$$
e_{x_i}(t) = \hat{x}_i(t) - x_i(t); \tag{3.6}
$$

and then, the dynamic state estimation error is given by,

$$
\begin{aligned}
\dot{e}_{x_i}(t) &= \dot{\hat{x}}_i(t) - \dot{x}_i(t); \\
&= A\hat{x}_i(t) + Bu_i(t) + H\hat{f_i(t)} - R\zeta_i(t) - Ax_i(t) \\
&- Bu_i(t) - Hf_i(t); \\
&= A(\hat{x}_i(t) - x_i(t)) + H(\hat{f}_i(t) - f_i(t)) - R\zeta_i(t);
\end{aligned} \tag{3.7}
$$

where the relative output estimation error ζ_i can be expressed as:

$$
\begin{aligned}
\zeta_i(t) &= \sum_{j=1}^{N} a_{ij}(C\hat{x}_i(t)) - \sum_{j=1}^{N} a_{ij}(Cx_i(t)) \\
&- \sum_{j=1}^{N} a_{ij}(C\hat{x}_j(t)) + \sum_{j=1}^{N} a_{ij}(Cx_j(t)) + g_i(C\hat{x}_i(t)) \\
&- g_i(Cx_i(t));
\end{aligned} \tag{3.8}
$$

then, from equation (2.70):

$$
\begin{aligned}
\zeta_i(t) =& d_iC\hat{x}_i(t) - d_iCx_i(t) - \begin{bmatrix} a_{i1} \dots & a_{iN} \end{bmatrix} \begin{bmatrix} C\hat{x}_1(t) \\ \vdots \\ C\hat{x}_N(t) \end{bmatrix} \\
& + \begin{bmatrix} a_{i1} & \dots & a_{iN} \end{bmatrix} \begin{bmatrix} Cx_1(t) \\ \vdots \\ Cx_N(t) \end{bmatrix} \\
& + g_iC\hat{x}_i(t) - g_iCx_i(t).
\end{aligned} \tag{3.9}
$$

Lets define the global vectors $\zeta(t) = \left[\zeta_1^T(t), \dots, \zeta_N^T(t)\right]^T \in \mathbb{R}^{qN}$, $x(t) = \left[x_1^T(t), \dots, x_N^T(t)\right]^T \in \mathbb{R}^{nN}$, $e_x(t) = \left[e_{x_1}^T(t), \dots, e_{x_N}^T(t)\right]^T \in \mathbb{R}^{nN}$. And, given the adjacency matrix $\mathcal{A} = [a_{ij}]$, the diagonal in-degree matrix $D = diag\{d_i\}$, the diagonal pinning gain matrix $G = diag\{g_i\}$, and the Laplacian matrix defined as $L = D - \mathcal{A}$, then one gets:

$$\begin{aligned}\zeta(t) =& (D \otimes C)\hat{x}(t) - (D \otimes C)x(t) - (\mathcal{A} \otimes C)\hat{x}(t) \\ & + (\mathcal{A} \otimes C)x(t) + (G \otimes C)\hat{x}(t) \\ & - (G \otimes C)x(t); \\ =& ((D - \mathcal{A} + G) \otimes C)(\hat{x}(t) - x(t)); \\ =& ((L + G) \otimes C)e_x(t).\end{aligned} \tag{3.10}$$

Then, the global distributed state estimation dynamic error for the whole MAS is given by:

$$\begin{aligned}\dot{e}_x(t) =& (I_N \otimes A)e_x(t) + (I_N \otimes H)e_f(t) \\ & - (I_N \otimes R)[((L + G) \otimes C)e_x(t)]; \\ =& (I_N \otimes A - (L + G) \otimes RC)e_x(t) \\ & + (I_N \otimes H)e_f(t).\end{aligned} \tag{3.11}$$

The fault estimation error vector for the i-th agent is defined as,

$$e_{f_i}(t) = \hat{f}_i(t) - f_i(t); \tag{3.12}$$

where the fault estimation dynamic error can be obtained with,

$$\begin{aligned}\dot{e}_{f_i(t)} &= \dot{\hat{f}}_i(t) - \dot{f}_i(t); \\ &= -\Gamma F(\zeta_i(t) + \dot{\zeta}_i(t)) - \dot{f}_i(t).\end{aligned} \tag{3.13}$$

Let $\dot{f}(t) = \left[\dot{f}_1^T(t), \ldots, \dot{f}_N^T(t)\right]^T \in \mathbb{R}^{nN}$ be the global fault vector. Then, the global distributed fault estimation dynamic error for the whole MAS is given by:

$$\dot{e}_f(t) = -((L + G) \otimes \Gamma FC)(e_x(t) + \dot{e}_x(t)) - \dot{f}(t). \tag{3.14}$$

If only faults with small variations are considered, therefore, it is assumed that $\dot{f} \approx 0$ (Estrada et al. 2015), which leads to:

$$\dot{e}_f(t) = -((L+G) \otimes \Gamma FC)(e_x(t) + \dot{e}_x(t)). \tag{3.15}$$

Now, the H_∞ criterion Hu et al. (2016) that is included in the developed DFEO in order to provide robustness to the observer is presented as:

$$J_{rd} := \dot{V}_{e(t)} + J_1 < 0; \tag{3.16}$$

where $V_{e(t)}$ is the candidate Lyapunov function, $J_1 = e_x^T(t)e_x(t) - \gamma^2 e_f^T(t)e_f(t)$, $\gamma > 0$ is a scalar value. J_1 can be expressed in matrix form:

$$J_1 = \begin{bmatrix} e_x^T(t) & e_f^T(t) \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} e_x(t) \\ e_f(t) \end{bmatrix}. \tag{3.17}$$

Theorem 1. If there exist a symmetric positive definite matrix $P \in \mathbb{R}^{n\times n}$, and matrices $Y \in \mathbb{R}^{n\times p}$ and $F \in \mathbb{R}^{r\times p}$ that satisfy:

$$\begin{bmatrix} I_N \otimes (A^T P + PA) - \Lambda & \Phi \\ * & \phi \end{bmatrix} < 0; \tag{3.18}$$

$$H^T P = FC; \tag{3.19}$$

where $\Lambda = (L+G) \otimes (YC) + (L+G)^T \otimes C^T Y^T) - I$, $\Phi = I_N \otimes PH - (L+G)^T \otimes (A^T PH + PH) + (L+G)^{2T} \otimes (C^T Y^T H)$, $\phi = -(L+G)^T \otimes (H^T PH) - (L+G) \otimes (H^T PH) - \gamma^2 I$. The observer gain matrix R can be calculated by $R = P^{-1} Y$.

Proof. Consider the following Lyapunov function:

$$V_{e(t)} = e_x^T(t)(I_N \otimes P)e_x(t) + e_f(t)^T(I_N \otimes \Gamma^{-1})e_f(t); \tag{3.20}$$

Then, by calculating its derivative, the following is obtained:

$$\dot{V}_{e(t)} = \dot{e}_x^T(t)(I_N \otimes P)e_x(t) + e_x^T(t)(I_N \otimes P)\dot{e}_x(t) + \dot{e}_f^T(t)(I_N \otimes \Gamma^{-1})e_f(t) + e_f^T(t)(I_N \otimes \Gamma^{-1})\dot{e}_f(t). \tag{3.21}$$

Substituting equations (3.11) and (3.15) in (3.21):

$$\begin{aligned}\dot{V}_{e(t)} =& \Big(e_x^T(t)(I_N \otimes A - (L+G) \otimes RC)^T + e_f^T(t)(I_N \otimes H)^T\Big)(I_N \otimes P)e_x(t) + \\ & e_x^T(t)(I_N \otimes P)\Big((I_N \otimes A - (L+G) \otimes RC)e_x(t) + (I_N \otimes H)e_f(t)\Big) - \Big((e_x(t) + \\ & \dot{e}_x(t))^T((L+G)^T \otimes \Gamma FC)^T\Big)(I_N \otimes \Gamma^{-1})e_f(t) - e_f(t)^T(I_N \otimes \Gamma^{-1})\Big(((L+G) \otimes \Gamma FC)(e_x(t) \\ & + \dot{e}_x(t))\Big) \\ =& e_x^T(t)(I_N \otimes A^T P - (L+G)^T \otimes C^T R^T P)e_x(t) + e_f^T(t)(I_N \otimes H^T P)e_x(t) + e_x^T(t)(I_N \otimes PA - \\ & (L+G) \otimes PRC)e_x(t) + e_x^T(t)(I_N \otimes PH)e_f(t) - (e_x(t) + \dot{e}_x(t))^T((L+G)^T \otimes (C^T F^T))e_f(t) \\ & - e_f^T(t)((L+G) \otimes (FC))(e_x(t) + \dot{e}_x(t)).\end{aligned} \tag{3.22}$$

with $Y = PR$ and $H^T P = FC$. From Theorem 1, equation (3.22) can be transformed to:

$$\begin{aligned}\dot{V}_{e(t)} =& e_x^T(t)(I_N \otimes (A^T P + PA) - (L+G)^T \otimes C^T Y^T - (L+G) \otimes YC)e_x(t) + 2e_x^T(t)(I_N \otimes PH)e_f(t) \\ & - 2e_x^T(t)\Big((L+G)^T \otimes PH\Big)e_f(t) \underbrace{-2\dot{e}_x^T(t)\Big((L+G)^T \otimes PH\Big)e_f(t)}_{\mathsf{e_x}},\end{aligned} \tag{3.23}$$

where:

$$\begin{aligned}\mathsf{e_x} =& -2\Big(e_x^T(t)(I_N \otimes A - (L+G) \otimes RC)^T + e_f^T(t)(I_N \otimes H)^T\Big)\Big((L+G)^T \otimes PH\Big)e_f(t) \\ =& -2e_x^T(t)\Big((L+G)^T \otimes A^T PH - (L+G)^{2T} \otimes C^T Y^T H\Big)e_f(t) \\ & -2e_f^T(t)\Big((L+G)^T \otimes H^T PH\Big)e_f(t)\end{aligned} \tag{3.24}$$

Then, by substituting (3.24) in (3.23)

$$\begin{aligned}\dot{V}_{e(t)} =& e_x^T(t)(I_N \otimes (A^T P + PA) - (L+G)^T \otimes C^T Y^T - (L+G) \otimes YC)e_x(t) + 2e_x^T(t)(I_N \otimes PH- \\ &(L+G)^T \otimes (PH + A^T PH) + (L+G)^{2T} \otimes C^T Y^T H)e_f(t) \\ &- 2e_f^T(t)((L+G)^T \otimes H^T PH)e_f(t)\end{aligned} \tag{3.25}$$

Then, J_1 term from (3.16) is added to (3.25) to apply the H_∞ criterion:

$$\begin{aligned}\dot{J}_{rd} =& e_x^T(t)(I_N \otimes (A^T P + PA) - (L+G)^T \otimes C^T Y^T - (L+G) \otimes YC)e_x(t) + 2e_x^T(t)(I_N \otimes PH- \\ &(L+G)^T \otimes (PH + A^T PH) + (L+G)^{2T} \otimes C^T Y^T H)e_f(t)- \\ &2e_f^T(t)((L+G)^T \otimes H^T PH)e_f(t) + e_x^T(t)e_x(t) - \gamma^2 e_f^T(t)e_f(t)\end{aligned} \tag{3.26}$$

which leads to the matrix representation provided in Theorem 1. This completes the proof. □

Remark. From Section 3, undirected graphs consider the matrix $(L+G) = (L+G)^T$ that is an special case of directed graphs. Theorem 1 can consider $(L+G) = (L+G)^T$ or $(L+G) \neq (L+G)^T$ indistinctly. Then, Theorem 1 can deals with directed and undirected graphs topologies.

3.3 Simulation Examples

In this section, two simulation examples are given to illustrate the effectiveness of the theoretical results.

Example 1. The system proposed in Zhang et al. (2015) is considered. The problem consist on a collection of 5 identical aircrafts. The leader agent and 4 follower agents, with the directed communication topology depicted in Fig. 3.1 (formation graph).

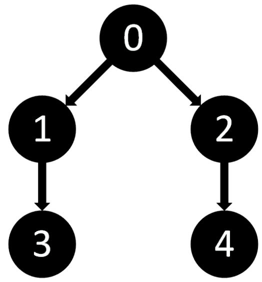

Figure 3.1: Communication graph for the numerical example.

Each aircraft is modeled in a state space representation as :

$$A = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.0100 & 0.0024 & -4.0208 \\ 0.1002 & 0.3681 & -0.7070 & 1.4200 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \tag{3.27}$$

$$B = \begin{bmatrix} 0.4422 & 0.1761 \\ 3.5446 & -7.5922 \\ -5.5200 & 4.4900 \\ 0 & 0 \end{bmatrix}; \; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \tag{3.28}$$

where the state vector is defined as $x_i(t) = \begin{bmatrix} v_h(t) & v_v(t) & q(t) & \theta(t) \end{bmatrix}^T$ and whose elements are horizontal velocity, vertical velocity, pitch angle, and pitch rate, respectively. The input vector includes collective pitch control and longitudinal cyclic pitch control.

The adjacency matrix $\mathcal{A}$, the in-degree matrix D, the Laplacian matrix L and the pinning gain matrix G are obtained from Fig. 3.1,

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}; G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

As can be seen from Fig. 3.1, a spanning tree in the graph proposed exist and all eigenvalues of $(L+G)$ are 1, i.e. $(L+G) > 0$ which fulfills Lemma 1.

For this numerical experiment, the following notation is used to define all vectors $q_i(t) = \begin{bmatrix} q_{i,b_1}(t) & q_{i,b_2}(t) & \dots & q_{i,b_n}(t) \end{bmatrix}^T$ where a is the agent label and b is the vector element. Then, as each agent has only two actuators, faults in agents 2 and 4 can be represented as $f_2(t) = \begin{bmatrix} f_{2,1}(t) & f_{2,2}(t) \end{bmatrix}^T$ and $f_4(t) = \begin{bmatrix} f_{4,1}(t) & f_{4,2}(t) \end{bmatrix}^T$, respectively. It is assumed that faults occur simultaneously in agents 2 and 4 as it is presented below:

$$f_{2,1}(t) = \begin{cases} 0 & 0s \leq t < 2s; \\ 0.2u_{2,1} & 2s \leq t < 4s; \\ 0 & 4s \leq t < 6s; \\ -0.2u_{2,1} & 6s \leq t < 8s; \\ 0 & 8s \leq t < 12s; \end{cases} \tag{3.29}$$

$$f_{4,1}(t) = \begin{cases} 0 & 0s \leq t < 2s; \\ 0.05u_{4,1} & 2s \leq t < 4s; \\ 0 & 4s \leq t < 6s; \\ -0.05u_{4,1} & 6s \leq t < 8s; \\ 0 & 8s \leq t < 12s; \end{cases} \tag{3.30}$$

Which means a 20% degradation of the pitch control $u_{2,1}$ when $2s \leq t < 4s$ and -20% when $6s \leq t <$

$8s$ for agent 2. And, a 5% degradation of the pitch control $u_{4,1}$ when $2s \leq t < 4s$ and -5% when $6s \leq t < 8s$ for agent 4. Initial conditions were randomly chosen between $0-3$ in order to be nonzero values, then initial conditions are given: $agent_1 = \begin{bmatrix} 1.2653 & 1.9672 & 2.0362 & 1.9664 \end{bmatrix}^T$, $agent_2 = \begin{bmatrix} 2.7472 & 0.1071 & 2.2732 & 0.5136 \end{bmatrix}^T$, $agent_3 = \begin{bmatrix} 2.3766 & 2.5474 & 2.2294 & 2.1181 \end{bmatrix}^T$, and $agent_4 = \begin{bmatrix} 2.8785 & 2.8020 & 1.1767 & 0.0955 \end{bmatrix}^T$. Defining $H = B$, and solving Theorem 1, the following constant matrices P, Y, R and F are obtained:

$$P = \begin{bmatrix} 1.2439 & 0.0126 & 0.1161 & 0.3070 \\ 0.0126 & 0.0378 & 0.0461 & 0.0349 \\ 0.1161 & 0.0461 & 0.0862 & 0.0470 \\ 0.3070 & 0.0349 & 0.0470 & 0.1608 \end{bmatrix};$$

$$Y = \begin{bmatrix} 0.8754 & 0.1567 & 0.3070 \\ 0.0504 & 0.0614 & 0.0782 \\ 0.1261 & 0.0765 & 0.1178 \\ -0.3325 & -0.0399 & 0.0381 \end{bmatrix};$$

$$R = \begin{bmatrix} 5.2205 & 1.1986 & 1.0049 \\ 29.1051 & 8.5784 & 6.7898 \\ -13.2428 & -3.4711 & -2.2591 \\ -14.4876 & -3.3858 & -2.4964 \end{bmatrix};$$

$$F = \begin{bmatrix} -0.0460 & -0.1149 & -0.2609 \\ 0.6444 & -0.0780 & 0.0572 \end{bmatrix};$$

$$\Gamma = \begin{bmatrix} 12 & 3 \\ 3 & 12 \end{bmatrix}.$$

The numerical results are presented in the following Figures. Fig. 3.2 presents the fault estimation $\hat{f}_{2,1}$ of actuator 1 in agent 2 and Fig. 3.3 shows the fault estimation $\hat{f}_{4,1}$ of actuator 1 in agent 4.

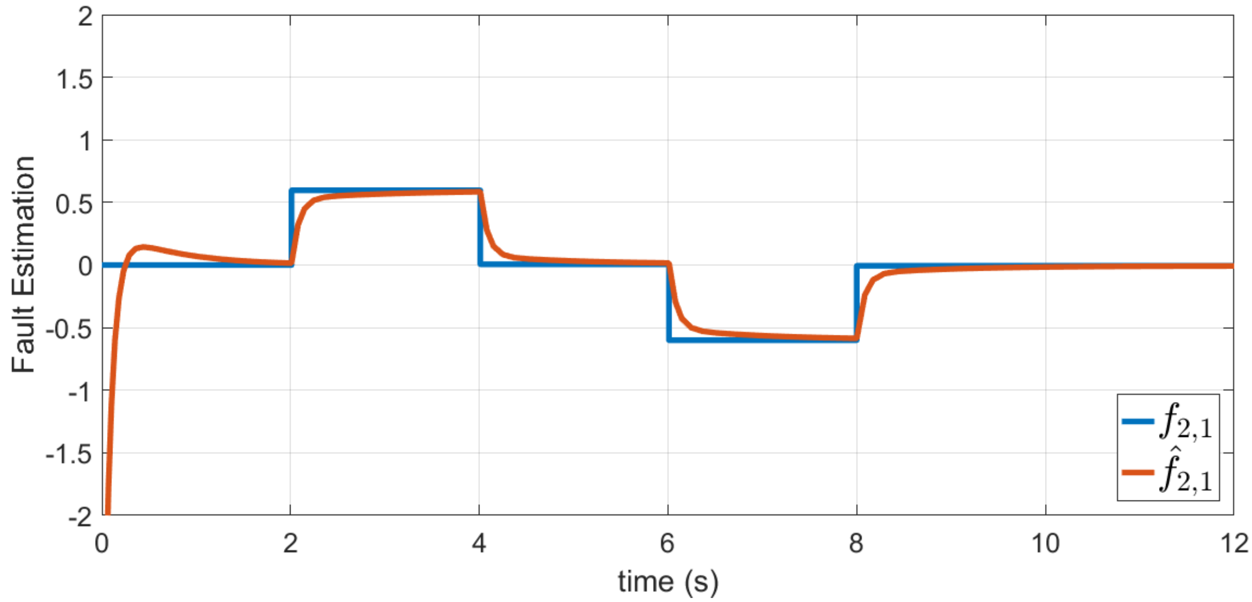

Figure 3.2: Fault estimation of agent 2.

In both figures, it is possible to see that the estimation converges asymptotically to the fault quickly even when the dynamics of the fault changes, observe the behavior of the fault estimator after the seconds 2s, 4s, 6s, and 8s. Finally, Fig. 3.4 show the state estimation error for agents 1-4, where horizontal velocity $x_{i,1}$, vertical velocity $x_{i,2}$, pitch angle $x_{i,3}$, and pitch rate $x_{i,4}$, $n = 1,\ldots,4$, are correctly estimated.

Note that state estimation error increase in the presence of faults in times 2s, 4s, 6s, and 8s for agents 2 and 4 who are the only agents with faults. Nevertheless, the observer is robust enough to estimate states even with multiple faults as it can be seen in Fig. 3.4.

Example 2. In this section a collection of coupled mass-spring systems are studied (Lewis et al. 2013, p. 97). Each agent is represented by a pair of coupled mass-spring systems as is represented in Fig. 3.5.

There exist 6 pairs of mass-spring systems in the MAS. Follower agents are represented in the space-state model as 3.31 and communication between these agents is described, with a direct topology, in Fig. 3.6. System 3.2 represents the leader dynamic.

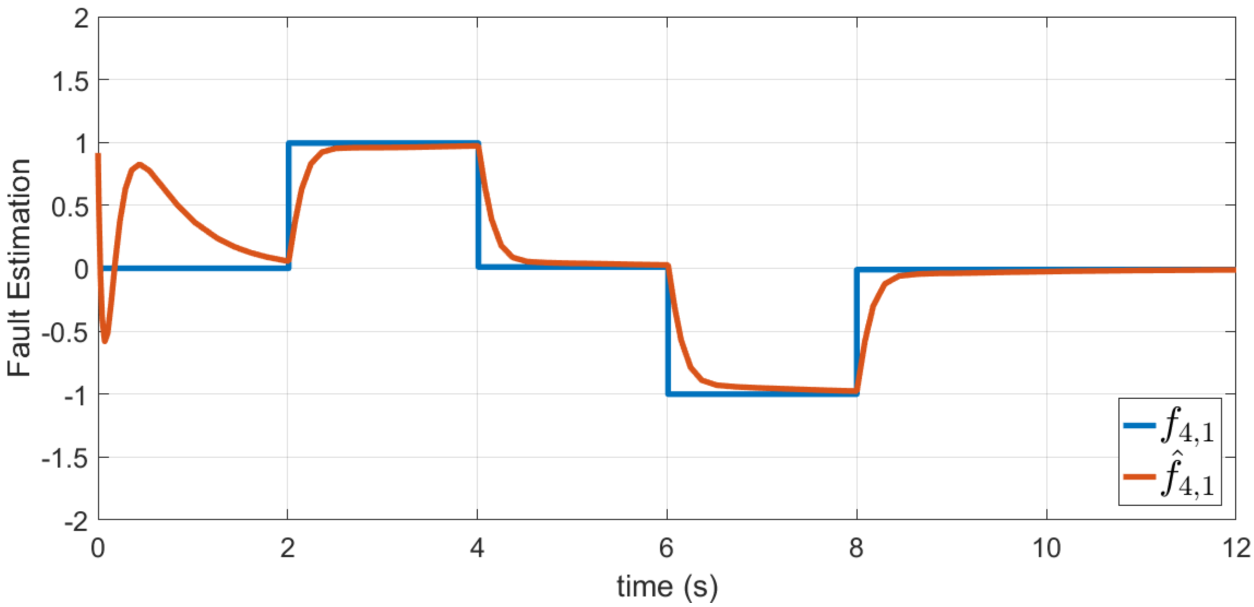

Figure 3.3: Fault estimation of agent 4.

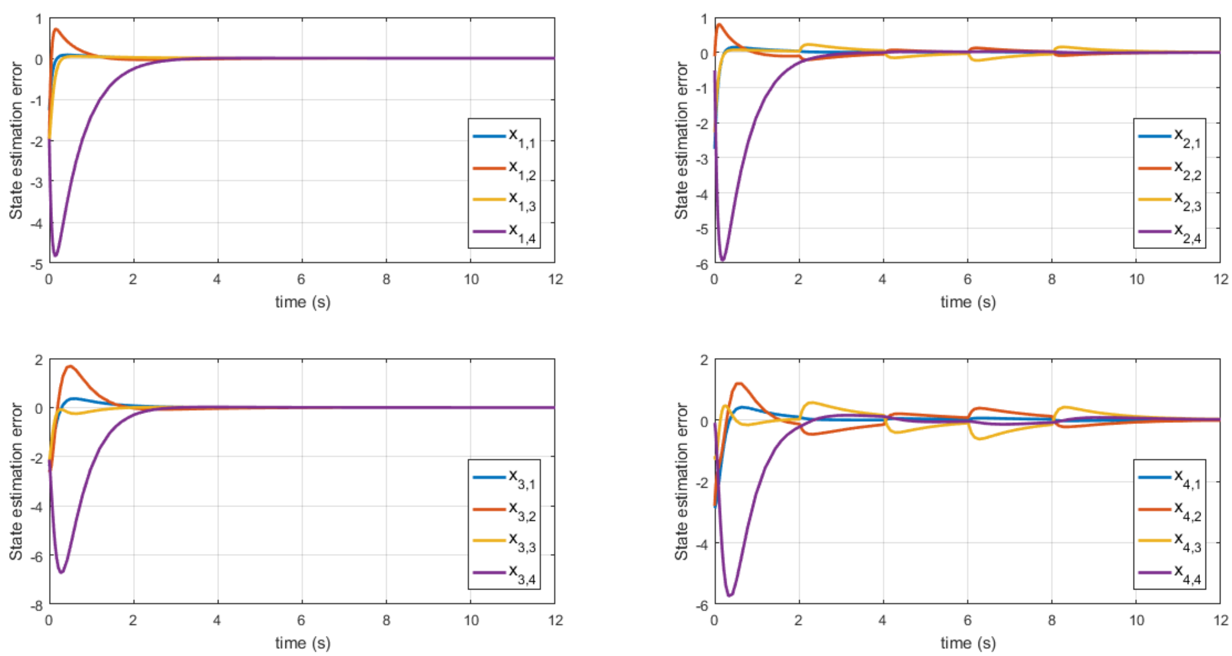

Figure 3.4: State estimation error.

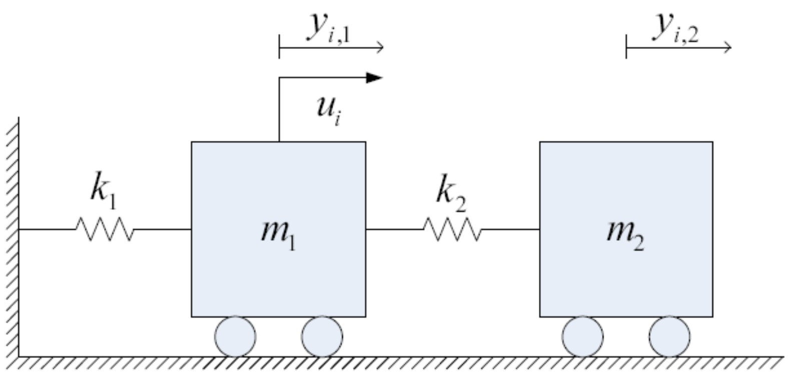

Figure 3.5: Coupled mass-spring systems.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2.2727 & 0 & 0.9190 & 0 \\ 0 & 0 & 0 & 1 \\ 1.1111 & 0 & -1.1111 & 0 \end{bmatrix}; \; B = \begin{bmatrix} 0 \\ 0.9091 \\ 0 \\ 0 \end{bmatrix}; \tag{3.31}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \tag{3.32}$$

In this simulation, 5 follower agents are considered, in other words, $i = 1, \ldots, 5$ and the leader agent is labeled as 0. The Laplacian matrix L and the pinning matrix G can be calculated through 3.6, then:

$$(L+G) = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}. \tag{3.33}$$

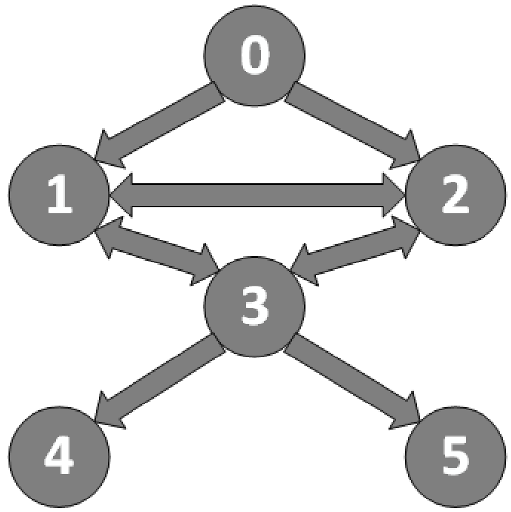

Figure 3.6: Example topology.

The fault induced to the system consists of a deviation of 25 % of the value of the input vector u_i. For this example, actuators faults of agents 4 and 5 are considered since these are further away from the leader, and therefore from the signal of the desired trajectory, then the complexity to maintain the desired behavior of these two agents is bigger.

In practice, the initial conditions are chosen according to the initial measured value of the system states variables (López-Estrada et al. 2016). The initial conditions of the system were chosen randomly between $[0-2]$ and the initial conditions of the observer as zero values, so that there was a discrepancy between both and in order to observe the dynamics of the state estimation error. Defining $H = B$, and solving the Theorem 1, the following matrices are obtained P, Y, R and F:

$$P = \begin{bmatrix} 0.3496 & -0.0125 & -0.1244 & -0.0244 \\ -0.0125 & 0.1484 & -0.0354 & 0.0000 \\ -0.1244 & -0.0354 & 0.2056 & -0.0818 \\ -0.0244 & 0.0000 & -0.0818 & 0.1727 \end{bmatrix};$$

$$Y = \begin{bmatrix} 0.2709 & 0.0333 & -0.0007 \\ -0.0703 & 0.0757 & 0.0193 \\ -0.0181 & -0.0053 & 0.3463 \\ 0.0267 & -0.0152 & 0.0028 \end{bmatrix};$$

$$R = \begin{bmatrix} 1.1170 & 0.1901 & 1.3912 \\ -0.1795 & 0.5743 & 1.0692 \\ 0.8398 & 0.2022 & 3.4448 \\ 0.7102 & 0.0349 & 1.8439 \end{bmatrix};$$

$$F = \begin{bmatrix} -0.0113 & 0.1349 & -0.0322 \end{bmatrix}; \Gamma = \begin{bmatrix} 7 \end{bmatrix}.$$

The simulation results are presented below. Figures 3.7 and 3.8 illustrate that state estimation error tends asymptotically to zero. To explain this example, the nomenclature $x_{i,a}$ is used where i corresponds to the follower agent number and a is the state variable of the agent i. Figure 3.9 shows the convergence between the fault estimated by the observer and the induced fault where it is assumed that the faults in agent 4 and 5 occur simultaneously in the second 3 of the simulation . Note that, since the estimation of the fault depends on the relative errors ζ_i in the estimation of the states, the estimation of the fault is closer to the induced fault when the state estimation error begin to stabilize at zero.

In order to illustrate the complexity of work with MAS in the Matlab/Simulink environment, it is provided the Fig. 3.10. Each agent x_i, $i = 1, \ldots, 5$ needs a linear model in space state representation, a state estimation observer and a fault estimator module, both were programmed in different blocks to appreciate each function. The communication in the graph applied though ζ_i, it is developed with arrows and Sum blocks and this communication is responsible to have this complex diagram.

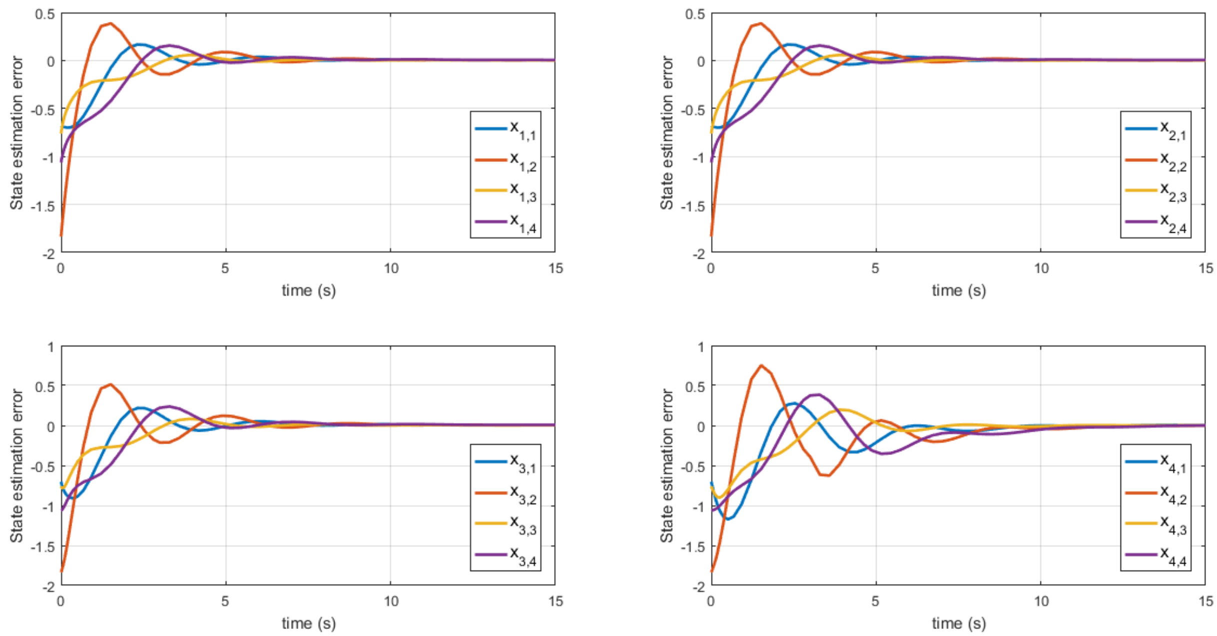

Figure 3.7: State estimation error of the agents 1-4.

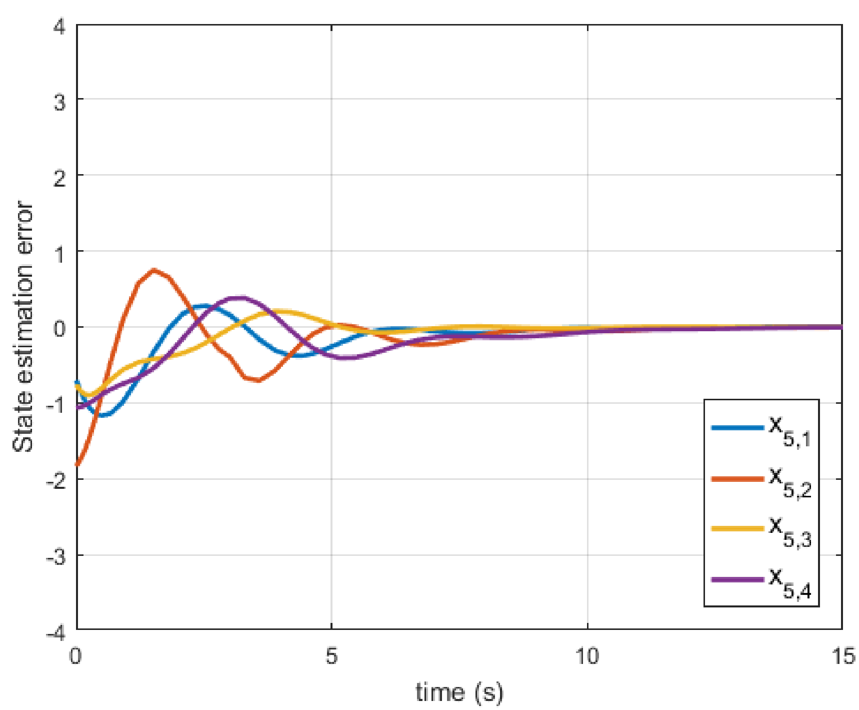

Figure 3.8: State estimation error of the agent 5.

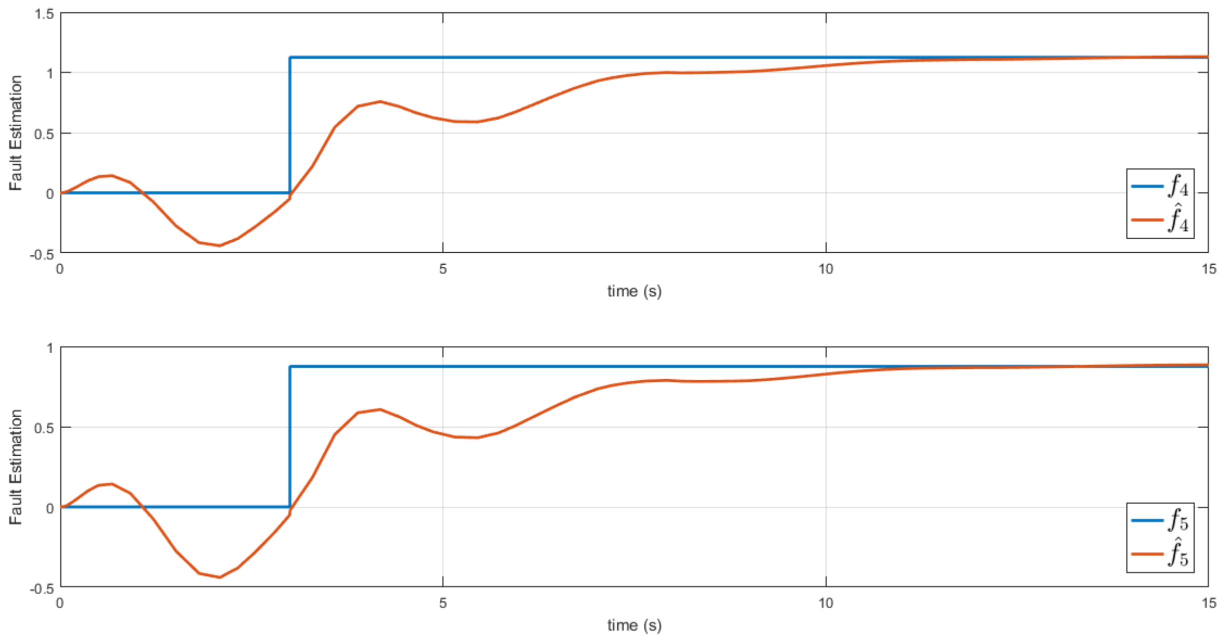

Figure 3.9: Fault estimation f_4 and f_5.

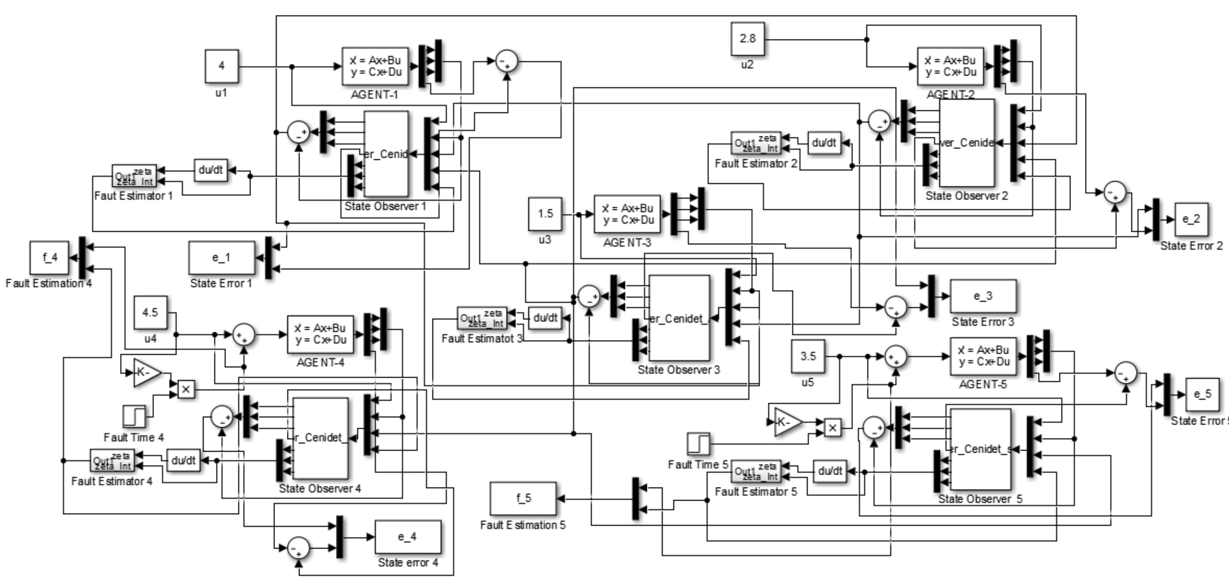

Figure 3.10: Matlab/Simulink scheme.

Chapter 4

Convex Fault Estimation of MAS

4.1 Introduction

As explained before, several papers about multi-agent systems have been elaborated since existing multiple applications for this approach in the control theory. For example, Zhang et al. (2016) propose a novel distributed control strategy where switching topologies and disturbance of the systems are considered. Liu et al. (2016) studied formation tracking problem for multiple vehicles with a motion planning approach, in this case, the time of synchronization for all agents is preassigned according to the task requirements. Zhang et al. (2013) developed observer-based controllers for event-triggered multi-agent systems, two different focus are presented, a distributed one, and a centralized one, Zhao et al. (2016) present a distributed observer-based algorithm to solve the finite-time consensus tracking problem. Further, the settling time can be estimated for second-order multi-agent systems with this control protocol. Yan et al. (2016) developed a fault detection filter applied to spring-mass systems using the optimal performance of H_∞. Finally, Zhao et al. (2016) give two algorithms for distributed average computation problem of multiple time-varying signals, static and adaptive coupling strengths are studied. Most of these works focus their efforts on developing algorithms to observe or to control the collection of systems interconnected by a communication graph. Nevertheless, if only a linear multi-agent systems approach is taken into account, a lot of information on the mathematical model is lost, due to the natural nonlinearities present in physical systems, which decreases the performance and robustness of the proposed algorithms. On the other hand, nonlinear theory, which allows describing almost all information of the mathematical model, involves a high computational cost. Under this analysis, it is introduced the convex multi-agent systems where multi-models are used to express each agent dynamics. This approach combines the low computational cost of the linear control theory, and it allows us to describe all mathematical models, even if nonlinearities are presented.

Multi-agent convex systems have been studied in order to provide solutions for nonlinear dynamical models with the advantages of the linear control theory. In this field, Gonzalez & Werner (2014) proposes a control protocol to deal with the formation control of multi-agent systems with leader-following configuration using linear fractional transformation; the control methodology is exposed in Scherer (2001) which is not observer-based. In Chen et al. (2016), it is developed a robust controller to solve the consensus problem in multi-agent systems where faults and plant disturbances are studied; in this case, only undirected graphs are considered. Chadli et al. (2016) designed a fault detection and isolation filter for a class of linear parameter varying (LPV) multi-agent systems where disturbance and noise signals are present, and sensor fault signals exist. Mesbahi & Velni (2017) study leader-following case treating the leader as an external signal (reference or disturbance), and the control law requires the solution of the Sylvester equation, in this work systems fault-free are assumed. However, few attention has been paid to fault estimation convex multi-agent system. (here you can cite FTC or FE if there exist FE observers consensus or leader). In this work, an algorithm to estimate states and faults for convex MAS is presented where is focus on the directed graphs in the leader-following case.

In this section, the PI-observer developed in the linear fault estimation section is extended to convex multi-agent systems in the leader-following configuration. In this case, actuator faults and sensor noise are included in the dynamics system representation. Finally, a new LMI is developed where H_∞ criterion is used to reduce the sensor noise. In this section, a numerical example is exposed to prove the effectiveness of this method.

4.2 Problem Statement

Consider a collection of $N+1$ identical convex multi-agent systems where follower agents dynamical model in continuous time is given by (4.1):

$$\begin{aligned} \dot{x}_i(t) &= \sum_{j=1}^{h} h_j(\xi(t))(A_j x_i(t) + B_j u_i(t)) + H f_i(i) \\ y_i(t) &= Cx(t) + F w_i(i) \end{aligned} \tag{4.1}$$

where $i = 1,\ldots,N$ and $j = 1,\ldots,h$, $x_i(t) \in \mathbb{R}^n$ is the state vector, $u_i(t) \in \mathbb{R}^m$ is the input vector, $f_i(t) \in \mathbb{R}^r$ is the actuator fault vector, $y_i(t) \in \mathbb{R}^p$ is the output vector and $w_i(t) \in \mathbb{R}^n$ is the sensor noise vector. Matrices A and C are assumed to be observable and H and F are constant and real matrices with appropriated dimensions.

Remark 1. In this mathematical model, subscripts i are used to describe the number of systems in the network, i.e., there are $i = 1,\ldots,N$ follower agents. On the other hand, subscripts j give the number of multi-models of each follower agent with $j = 1,...,h$, according to convex systems theory (Guzmán-Rabasa et al. 2019).

In order to estimate states and faults, the following convex distributed observer is proposed:

$$\begin{aligned} \dot{\hat{x}}_i(t) &= \sum_{j=1}^{h} h_j(\xi(t))\big(A_j \hat{x}_i(t) + B_j u_i(t) - R_j \zeta_i(t)\big) + H\hat{f}_i(t); \\ \hat{y}_i(t) &= C\hat{x}_i(t); \\ \hat{f}_i(t) &= -\Gamma\Phi\left(\zeta_i(t) + \int_{t_f}^{t} \zeta_i(t)dt\right); \end{aligned} \tag{4.2}$$

where $i = 1,\ldots,N$ and $j = 1,\ldots,h$, $\hat{x}_i(t) \in \mathbb{R}^n$ is the estimated state, $\hat{y}_i(t) \in \mathbb{R}^p$ is the estimated output, $\zeta_i(t) \in \mathbb{R}^q$ is the relative output estimation error of the i-th agent in the communication graph defined later, $R_j \in \mathbb{R}^{n\times q}$ is the observer gain matrix to be designed, and $\hat{f}_i(t)$ is the estimated fault. To deal with fault estimation, a distributed PI fault estimator $\hat{f}_i(t)$ is proposed where the information of relative output estimation error ζ_i is needed. Additionally, the integral term of ζ_i is added to have a faster convergence to the fault. $\Phi \in \mathbb{R}^{r\times p}$ is the fault estimator gain matrix to be designed, and matrix $\Gamma = \Gamma^T > 0$ is the learning rate. Note that t_f indicates the time when the fault occurs.

4.3 State Observer for MAS

To develop a distributed observer, it is necessary to establish a connection between agents, i.e., to explain how the information is shared between all agents. In this case, the connection is defined by the relative output estimation error ζ_i:

$$\begin{aligned}\zeta_i(t) = &\sum_{j=1}^{N} a_{ij}\Big(\big(\hat{y}_i(t) - y_i(t)\big) - \big(\hat{y}_j(t) - y_j(t)\big)\Big)\\ &+ g_i\Big(\big(\hat{y}_i(t) - y_i(t)\big) - \big(\hat{y}_0(t) - y_0(t)\big)\Big);\\ &i = 1,\dots,N;\end{aligned} \tag{4.3}$$

Now, substituting 4.1 and 4.2 in 4.3 and with the assumption $\hat{y}_0(t) - y_0(t) = 0$ (see details in linear MAS section):

$$\begin{aligned}\zeta_i(t) = &\sum_{k=1}^{N} a_{ik}\Big(\big(C\hat{x}_i(t) - Cx_i(t) - Fw_i(t)\big) - \big(C\hat{x}_k(t) - Cx_k(t) - Fw_k(t)\big)\Big)\\ &+ g_i(C\hat{x}_i(t) - Cx_i(t) - Fw_i);\\ &i = 1,\dots,N;\end{aligned} \tag{4.4}$$

that can be easily transformed to:

$$
\begin{aligned}
\zeta_i(t) =& d_i(C\hat{x}_i(t) - Cx_i(t) - Fw_i(t)) - \begin{bmatrix} a_{i1} \dots & a_{iN} \end{bmatrix} \begin{bmatrix} C\hat{x}_1(t) - Cx_1(t) - Fw_1(t) \\ \vdots \\ C\hat{x}_N(t) - Cx_N(t) - Fw_N(t) \end{bmatrix} \\
& + g_i(C\hat{x}_i(t) - Cx_i(t) - Fw_i(t)).
\end{aligned}
\tag{4.5}
$$

with the matrices $\mathcal{A} = [a_{ik}]$, $D = diag(d_i)$, and $G = diag(g_i)$ one gets:

$$
\begin{aligned}
\zeta_i(t) &= DC\hat{x}_i(t) - DCx_i(t) - DFw_i(t) - \mathcal{A}C\hat{x}_i(t) + \mathcal{A}Cx_i(t) + \mathcal{A}Fw_i(t) + GC\hat{x}_i(t) - GCx_i(t) \\
&\quad - GFw_i(t) \\
&= DC(\hat{x}_i(t) - x_i(t)) - \mathcal{A}C(\hat{x}_i(t) - x_i(t)) - (D - \mathcal{A})Fw_i(t) + GC(\hat{x}_i(t) - x_i(t)) \\
&\quad - GFw_i(t)
\end{aligned}
\tag{4.6}
$$

with $e_{x_i}(t) = \hat{x}_i(t) - x_i(t)$ and the Laplacian matrix $L = D - \mathcal{A}$:

$$
\zeta_i(t) = (L+G)Ce_{x_i}(t) - (L+G)Fw_i(t) \tag{4.7}
$$

In order to express the complete state estimation error vector $e_x(t)$ and the sensor noise vector $w(t)$ for all agents in the network, vectors $e_x(t) = \left[e_{x_1}^T(t), \dots, e_{x_N}^T(t)\right]^T \in \mathbb{R}^{nN}$, $w(t) = \left[w_1^T(t), \dots, w_N^T(t)\right]^T \in \mathbb{R}^{nN}$, and $\zeta(t) = \left[\zeta_1^T(t), \dots, \zeta_N^T(t)\right]^T \in \mathbb{R}^{qN}$ are defined. Then, global relative output estimation error ζ is expressed as:

$$
\zeta(t) = (L+G) \otimes Ce_x(t) - (L+G) \otimes Fw(t) \tag{4.8}
$$

The observer is developed to estimate states and faults, with this target, state estimation error $e_{x_i}(t)$ and fault estimation error $e_{f_i}(t)$ must to converge asymptotically to zero. Let define the dynamics of the estimation errors:

$$\dot{e}_{x_i}(t) = \dot{\hat{x}}_i(t) - \dot{x}_i(t) \tag{4.9}$$

$$\dot{e}_{f_i}(t) = \dot{\hat{f}}_i(t) - \dot{f}_i(t) \tag{4.10}$$

then, the scheduling function are not required to describe the dynamical error and if it is substituted 4.1 and 4.2 in 4.9, one gets:

$$\begin{aligned}\dot{e}_{x_i}(t) &= A_j\hat{x}_i(t) + B_ju_i(t) + H\hat{f}_i(t) - R_j\zeta_i(t) - A_jx_i(t) - B_ju_i(t) - Hf_i(t)\\ &= A_je_{x_i}(t) + He_{f_i}(t) - R_j\zeta_i(t)\end{aligned} \tag{4.11}$$

where, substituting 4.8 in 4.11, the global state estimation error is given by:

$$\begin{aligned}\dot{e}_x(t) &= (I_N \otimes A_j)e_x(t) + (I_N \otimes H)e_f(t) - R_j((L+G)\otimes Ce_x(t) - (L+G)\otimes Fw(t))\\ &= (I_N \otimes A_j - (L+G)\otimes R_jC)e_x(t) + (I_N \otimes H)e_f(t) + ((L+G)\otimes R_jF)w(t)\end{aligned} \tag{4.12}$$

developing the fault estimation error e_{f_i} it is substituted 4.2 in 4.10:

$$\dot{e}_{f_i} = -\Gamma\Phi(\zeta_i + \dot{\zeta}_i) - \dot{f}_i \tag{4.13}$$

where it is assumed that $\dot{\hat{f}}_i = 0$, i.e., dynamical faults are not considered to be estimated. Nevertheless, even when this restriction is not satisfied, it is possible to reconstruct time varying signals with slow variation (Chadli et al. 2013). Then, the global fault estimation error is given by substitution 4.8 in 4.13:

$$\dot{e}_f(t) = -\Gamma\Phi((L+G)\otimes Ce_x(t) - (L+G)\otimes Fw(t)) - \Gamma\Phi((L+G)\otimes C\dot{e}_x(t) - (L+G)\otimes F\dot{w}(t)) \tag{4.14}$$

same small variation assumption is considered for the dynamical sensor noise $\dot{w}(t) = 0$ to obtain:

$$\dot{e}_f = -((L+G)\otimes\Gamma\Phi C)e_x(t) + ((L+G)\otimes\Gamma\Phi F)w(t) - ((L+G)\otimes\Gamma\Phi C)\dot{e}_x \tag{4.15}$$

following the methodology proposed in the linear distributed observer, the H_∞ criterion is defined:

where, $V_{e(t)}$ is the candidate Lyapunov function, $J_1 = e^T(t)e(t) - \gamma^2 w^T(t)w(t)$, $\gamma > 0$ is a scalar value where:

$$J_{rd} := \dot{V}_{e(t)} + J_1 < 0; \tag{4.16}$$

where $e(t) = \begin{bmatrix} e_x(t) & e_f(t) \end{bmatrix}^T$

Then, J_1 can be expressed in matrix form:

$$J_1 = \begin{bmatrix} e_x^T(t) & e_f^T(t) & w^T(t) \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} e_x(t) \\ e_f(t) \\ w(t) \end{bmatrix}. \tag{4.17}$$

Theorem 2. If there exist a symmetric positive definite matrix $P \in \mathbb{R}^{nxn}$, and matrices $Y_j \in \mathbb{R}^{nxp}$ and $\Phi \in \mathbb{R}^{rxp}$ that satisfy conditions 4.18 and 4.19:

$$\begin{bmatrix} I_N \otimes (A_j^T P + PA_j) - (L+G)^T \otimes C^T Y_j^T - (L+G) \otimes Y_j C + I & \Pi & (L+G) \otimes Y_j C \\ * & \Xi & \Psi \\ * & * & -\gamma^2 I \end{bmatrix} < 0; \quad (4.18)$$

$$H^T P = \Phi C; \quad (4.19)$$

where $\Pi = -(L+G)^T \otimes (PH + A_j^T PH) + (L+G)^{2T} \otimes C^T Y_j^T H + I_N \otimes PH$, $\Xi = -(L+G)^T \otimes H^T PH - (L+G) \otimes H^T PH$, and $\Psi = (L+G) \otimes H^T P - (L+G)^2 \otimes H^T Y_j C$ .The PI observer converges asymptotically to the states and faults with the observer gain matrix $R_j = P^{-1} Y_j$.

Proof. Consider the following Lyapunov function candidate:

$$V_e(t) = e_x^T(t)(I_N \otimes P)e_x(t) + e_f(t)^T(I_N \otimes \Gamma^{-1})e_f(t); \quad (4.20)$$

whose derivative is:

$$\begin{aligned} \dot{V}_e(t) =& \dot{e}_x^T(t)(I_N \otimes P)e_x(t) + e_x^T(t)(I_N \otimes P)\dot{e}_x(t) \\ &+ \dot{e}_f^T(t)(I_N \otimes \Gamma^{-1})e_f(t) + e_f^T(t)(I_N \otimes \Gamma^{-1})\dot{e}_f(t). \end{aligned} \quad (4.21)$$

Remark 2. Lyapunov functions 3.20 and 4.20 are identical since this equation allows us to work with directed and undirected graphs, as it was demonstrated in the linear section. However, the difference between both approaches can be noticed in the next development.

Now, to provide clearer development, state estimation error e_x and fault estimation error e_f in 4.21 are treated separately. First, state estimation part of the Lyapunov function is calculated:

$$
\begin{aligned}
\dot{V}_{e_x}(t) &= \dot{e}_x^T(t)(I_N \otimes P)e_x(t) + e_x^T(t)(I_N \otimes P)\dot{e}_x(t) \\
&= \Big(e_x^T(t)(I_N \otimes A_j - (L+G) \otimes R_jC)^T + e_f^T(t)(I_N \otimes H)^T + w^T(t)((L+G) \otimes R_jF)^T\Big)(I_N \otimes P) \\
&\quad e_x(t) + e_x^T(t)(I_N \otimes P)\Big((I_N \otimes A_j - (L+G) \otimes R_jC)e_x(t) + (I_N \otimes H)e_f(t) + ((L+G) \otimes R_jF) \\
&\quad w(t)\Big) \\
&= e_x^T(t)(I_N \otimes A_j^TP - (L+G)^T \otimes C^TR_j^TP)e_x(t) + e_f^T(t)(I_N \otimes H^TP)e_x(t) + w^T(t)((L+G)^T \otimes \\
&\quad F^TR_j^TP)e_x(t) + e_x^T(t)(I_N \otimes PA_j - (L+G) \otimes PR_jC)e_x(t) + e_x^T(t)(I_N \otimes PH)e_f(t) + e_x^T(t) \\
&\quad ((L+G) \otimes PR_jF)w(t)
\end{aligned}
\tag{4.22}
$$

that can be transformed to:

$$
\begin{aligned}
\dot{V}_{e_x}(t) &= e_x^T(t)(I_N \otimes A_j^TP - (L+G)^T \otimes C^TR_j^TP)e_x(t) + 2e_f^T(t)(I_N \otimes H^TP)e_x(t) + 2w^T(t)((L+G)^T \otimes \\
&\quad F^TR_j^TP)e_x(t) + e_x^T(t)(I_N \otimes PA_j - (L+G) \otimes PR_jC)e_x(t) \\
&= e_x^T(t)(I_N \otimes (A_j^TP + PA_j) - (L+G)^T \otimes C^TR_j^TP - (L+G) \otimes PR_jC)e_x(t) + 2e_f^T(t)(I_N \otimes H^TP) \\
&\quad e_x(t) + 2w^T(t)((L+G)^T \otimes F^TR_j^TP)e_x(t)
\end{aligned}
\tag{4.23}
$$

The second part of the Lyapunov function, which is concerned with the fault estimation is calculated as follow:

$$
\begin{aligned}
\dot{V}_{e_f}(t) &= \dot{e}_f^T(t)(I_N \otimes \Gamma^{-1})e_f(t) + e_f^T(t)(I_N \otimes \Gamma^{-1})\dot{e}_f(t) \\
&= -e_x^T(t)((L+G)^T \otimes C^T\Phi^T)e_f(t) + w^T(t)((L+G)^T \otimes F^T\Phi^T)e_f(t) - \dot{e}_x^T(t)((L+G)^T \otimes \\
&\quad C^T\Phi^T)e_f(t) - e_f^T(t)((L+G) \otimes \Phi C)e_x(t) + e_f^T(t)((L+G) \otimes \Phi F)w(t) - e_f^T(t)((L+G)\otimes \\
&\quad \Phi C)\dot{e}_x(t)
\end{aligned}
\tag{4.24}
$$

which also can be transformed to:

$$
\begin{aligned}
\dot{V}_{e_f}(t) = &-2e_x^T(t)((L+G)^T \otimes C^T\Phi^T)e_f(t) + 2w^T(t)((L+G)^T \otimes F^T\Phi^T)e_f(t) - 2\dot{e}_x^T(t)((L+G)^T \otimes \\
&C^T\Phi^T)e_f(t)
\end{aligned}
\tag{4.25}
$$

where:

$$
\begin{aligned}
&-2\dot{e}_x^T(t)((L+G)^T \otimes C^T\Phi^T)e_f(t) = \\
&-2\Big(\big(e_x^T(t)(I_N \otimes A_j - (L+G) \otimes R_jC)^T + e_f^T(t)(I_N \otimes H)^T + w^T(t)((L+G)\otimes \\
&R_jF)^T\big)((L+G)^T \otimes C^T\Phi^T)e_f(t)\Big) = \\
&-2\Big(e_x^T(t)((L+G)^T \otimes A_j^TC^T\Phi^T - (L+G)^{2T} \otimes C^TR_j^TC^T\Phi^T)e_f(t) + e_f^T(t)((L+G)^T \otimes H^TC^T\Phi^T) \\
&e_f(t) + w^T(t)((L+G)^{2T} \otimes F^TR_j^TC^T\Phi^T)e_f(t)\Big) = \\
&-2e_x^T(t)((L+G)^T \otimes A_j^TC^T\Phi^T - (L+G)^{2T} \otimes C^TR_j^TC^T\Phi^T)e_f(t) - 2e_f^T(t)((L+G)^T \otimes H^TC^T\Phi^T) \\
&e_f(t) - 2w^T(t)((L+G)^{2T} \otimes F^TR_j^TC^T\Phi^T)e_f(t)
\end{aligned}
\tag{4.26}
$$

Therefore:

$$\begin{aligned}\dot{V}_{e_f}(t) = & -2e_x^T(t)((L+G)^T \otimes C^T\Phi^T)e_f(t) + 2w^T(t)((L+G)^T \otimes F^T\Phi^T)e_f(t) - 2e_x^T(t)((L+G)^T \otimes \\ & A_j^T C^T\Phi^T - (L+G)^{2T} \otimes C^T R_j^T C^T \Phi^T)e_f(t) - 2e_f^T(t)((L+G)^T \otimes H^T C^T \Phi^T)e_f(t) \\ & -2w^T(t)((L+G)^{2T} \otimes F^T R_j^T C^T \Phi^T)e_f(t) \\ = & -2e_x^T(t)((L+G)^T \otimes (C^T\Phi^T + A_j^T C^T \Phi^T)) - (L+G)^{2T} \otimes C^T R_j^T C^T \Phi^T)e_f(t) + 2w^T(t) \\ & ((L+G)^T \otimes F^T\Phi^T - (L+G)^{2T} \otimes F^T R_j^T C^T \Phi^T)e_f(t) - 2e_f^T(t)((L+G)^T \otimes H^T C^T \Phi^T)e_f(t)\end{aligned} \tag{4.27}$$

In order to satisfy **Theorem 2**, following restriction must be fulfilled:

$$F = C, \tag{4.28}$$

this means, that noise matrix F and output matrix C must to be the same to develop a feasible LMI. Further, with 4.19 and $PR_j = Y_j$, one gets:

$$\begin{aligned}\dot{V}_{e_x}(t) = & e_x^T(t)(I_N \otimes (A_j^T P + PA_j) - (L+G)^T \otimes C^T Y_j^T - (L+G) \otimes Y_j C)e_x(t) + 2e_x^T(t)(-(L+G)^T \otimes \\ & (PH + A_j^T PH)) + (L+G)^{2T} \otimes C^T Y_j^T H + I_N \otimes PH)e_f(t) + 2w^T(t)((L+G)^T \otimes C^T Y_j^T)e_x(t) + \\ & 2w^T(t)((L+G)^T \otimes PH - (L+G)^{2T} \otimes C^T Y_j^T H)e_f(t) + e_f^T(t)(-(L+G)^T \otimes H^T PH - (L+G) \\ & \otimes H^T PH)e_f(t)\end{aligned} \tag{4.29}$$

Note that, from 4.27:

$$
\begin{aligned}
&-2e_f^T((L+G)^T \otimes H^T PH)e_f = \\
&-e_f^T((L+G)^T \otimes H^T PH)e_f - e_f^T((L+G) \otimes H^T PH)e_f = \\
&e_f^T(-(L+G)^T \otimes H^T PH - (L+G) \otimes H^T PH)e_f
\end{aligned} \tag{4.30}
$$

Finally, according to 4.16, HJ_{rd} is completed as follow:

$$
\begin{aligned}
\dot{V}_{e_x}(t) = {} & e_x^T(t)(I_N \otimes (A_j^T P + PA_j) - (L+G)^T \otimes C^T Y_j^T - (L+G) \otimes Y_j C)e_x(t) + 2e_x^T(t)(-(L+G)^T \otimes \\
& (PH + A_j^T PH)) + (L+G)^{2T} \otimes C^T Y_j^T H + I_N \otimes PH)e_f(t) + 2w^T(t)((L+G)^T \otimes C^T Y_j^T)e_x(t) + \\
& 2w^T(t)((L+G)^T \otimes PH - (L+G)^{2T} \otimes C^T Y_j^T H)e_f(t) + e_f^T(t)(-(L+G)^T \otimes H^T PH - (L+G) \\
& \otimes H^T PH)e_f(t) + e^T(t)e(t) - \gamma^2 w^T(t)w(t)
\end{aligned} \tag{4.31}
$$

Which leads to the LMI in 4.18. This ends the proof. □

4.4 Numerical Example

In this section, a numerical example is provided to illustrate the theoretical results. The system proposed for this approach is described in 4.32. Note that this system has only two state variables, of which only the first state variable is measurable. The observer must estimate the second state variable and the actuator faults. For this simulation example, only three follower agents are considered in the graph topology labeled as 1, 2, and 3. Agents 1 and 2 have direct communication with the leader agent labeled as 0. Agent 3 is the only agent without direct communication with the leader but exists a spanning tree, according to Lewis et al. (2013), explained in detail in section **2.3 Multi-Agent Systems**. The simulation results of the PI observer are compared with the proportional observer version which is the most common observer.

The communication graph is depicted in Fig. 4.1

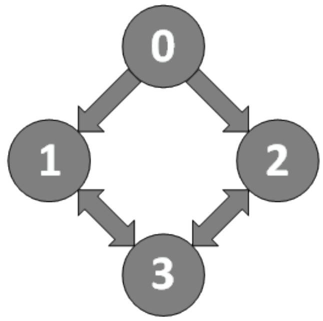

Figure 4.1: Communication graph for the numerical example.

$$A = \begin{bmatrix} -4 & 2.5+\rho \\ 4.5 & -4 \end{bmatrix}; B = \begin{bmatrix} 1.5 \\ 3 \end{bmatrix}; \; C = \begin{bmatrix} 1 & 0 \end{bmatrix}; \tag{4.32}$$

where ρ is the nonlinear parameter in the convex system. In this example $\rho = 0.35 * sin(t) + 0.45$. Then, for the nature of the signal wave described, $\rho_{max} = 0.8$ and $\rho_{min} = 0.1$. Linear multi-models for the system presented in 4.32 are obtained by substituting ρ_{max} and ρ_{min} as is presented in 4.33:

$$\begin{aligned} A_1 &= \begin{bmatrix} -4 & 3.3 \\ 4.5 & -4 \end{bmatrix} \\ A_2 &= \begin{bmatrix} -4 & 2.6 \\ 4.5 & -4 \end{bmatrix} \end{aligned} \tag{4.33}$$

Weighting functions, related to the nonlinearities, are presented in 4.34. Note that, ρ is an element that is calculated online in the simulation according to the dynamic presented in 4.32.

Solving **Theorem 2**, gain matrices are obtained for linear multi-models. Constant matrix P, and gain matrix R_i are presented following:

$$
\begin{aligned}
h_1(\xi(t)) &= \frac{\rho - \rho_{min}}{\rho_{max} - \rho_{min}} \\
h_2(\xi(t)) &= \frac{\rho_{min} - \rho}{\rho_{max} - \rho_{min}}
\end{aligned} \tag{4.34}
$$

$$
P = \begin{bmatrix} 0.1150 & -0.0356 \\ -0.0356 & 0.0178 \end{bmatrix}; \tag{4.35}
$$

$$
R_1 = \begin{bmatrix} -0.2731 \\ -0.1448 \end{bmatrix}; \tag{4.36}
$$

$$
R_2 = \begin{bmatrix} -0.6046 \\ -0.3361 \end{bmatrix}; \tag{4.37}
$$

$$
\tag{4.38}
$$

Remark. Note that nonlinear model 4.32 is transformed to the two linear models 4.33. Then, we obtain two gain matrices R_1 and R_2.

The three follower agents has induced actuator faults as is presented below. In the expression $u_{i,n}$, i represents the agent label and n is the state variable.

$$
f_1(t) = \begin{cases} 0.5u_{1,1} & 0s \leq t < 12s; \\ 0 \quad 12s \leq t < 18s; \end{cases} \tag{4.39}
$$

$$
f_2(t) = \begin{cases} 0 \quad 0s \leq t < 5s; \\ 0.5u_{2,1} & 5s \leq t < 14s; \\ 0 \quad 14s \leq t < 18s; \end{cases} \tag{4.40}
$$

$$f_3(t) = \begin{cases} 0 & 0s \leq t < 2s; \\ 0.5u_{3,1} & 2s \leq t < 9s; \\ 0 & 9s \leq t < 18s; \end{cases} \tag{4.41}$$

Simulation results are presented in the next figures. Figure 4.2 presents estimation error for Agent 1, $x_{PI:1,1}$ and $x_{PI:1,2}$ depicts the estimation error of the PI observer for state variable 1 and state variable 2, respectively. On the other hand, $x_{P:1,1}$ and $x_{P:1,2}$ shows the estimation error of the Proportional observer. The difference between both observer is that the PI converges to zero even in the presence of faults while the proportional one is several affected by the presence of faults in times $2s$, $5s$, $9s$, and $14s$. Figure 4.3 shows the dynamic of the state estimation error of the agent 2. $x_{PI:21}$ is the state estimation error of the state variable 1 with the PI observer, $x_{PI:22}$ is the state estimation error of the state variable 2 with the PI observer, and $x_{P:21}$ and $x_{P:22}$ are the state estimation error of the state variable 1 and 2 with the proportional observer, respectively. Figure 4.4 depicts the state estimation error of the agent 3 comparing the PI observer and the proportional observer. Realize that the three agents' faults affect the estimation of the observer for all agents. This can be seen in all figures in the times $2s$, $5s$, $9s$, $12s$, and $14s$ that is the times where there exists a change in the fault dynamic.

Fault estimation of the Agent 1 is shown in Fig.4.5, where few differences between both observers can be seen. In this case, the importance of the example is to appreciate the peaks in the times $2s$, $5s$, $9s$, and $14s$ which is the times where the faults f_2 and f_3 change abruptly, these faults affect all the estimation because of the distributed approach. Fig. Fault estimation of f_2 in the Agent 2 is depicted in 4.6 the convergence time to the fault is very similar for both observer, however, the fault estimation of the PI observer is closer to the real fault. Finally, in Fig. 4.7 it is possible to see the performance of both observers. PI observer estimates closer the fault and the peaks are present when the fault dynamic change abruptly.

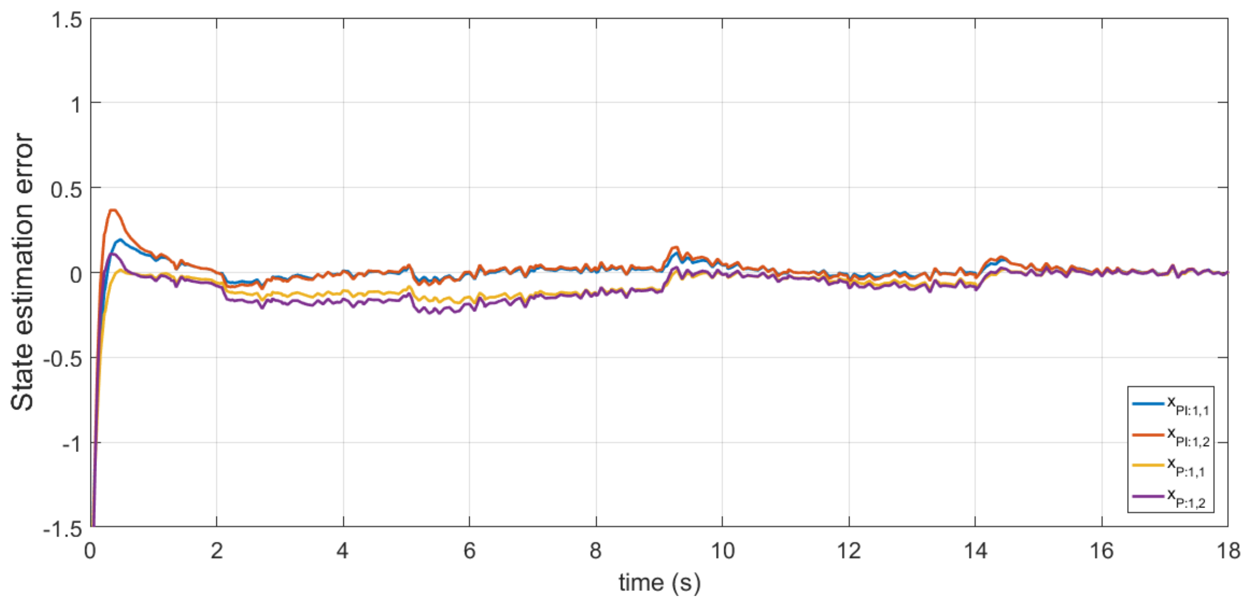

Figure 4.2: State estimation error and ρ.

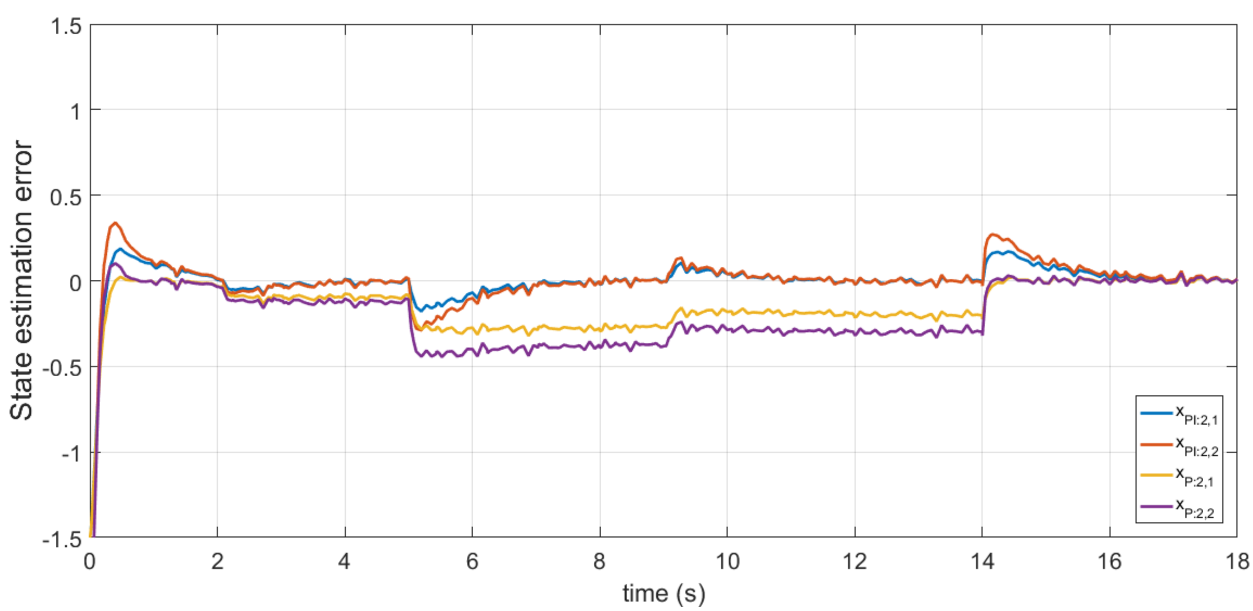

Figure 4.3: State estimation error and ρ.

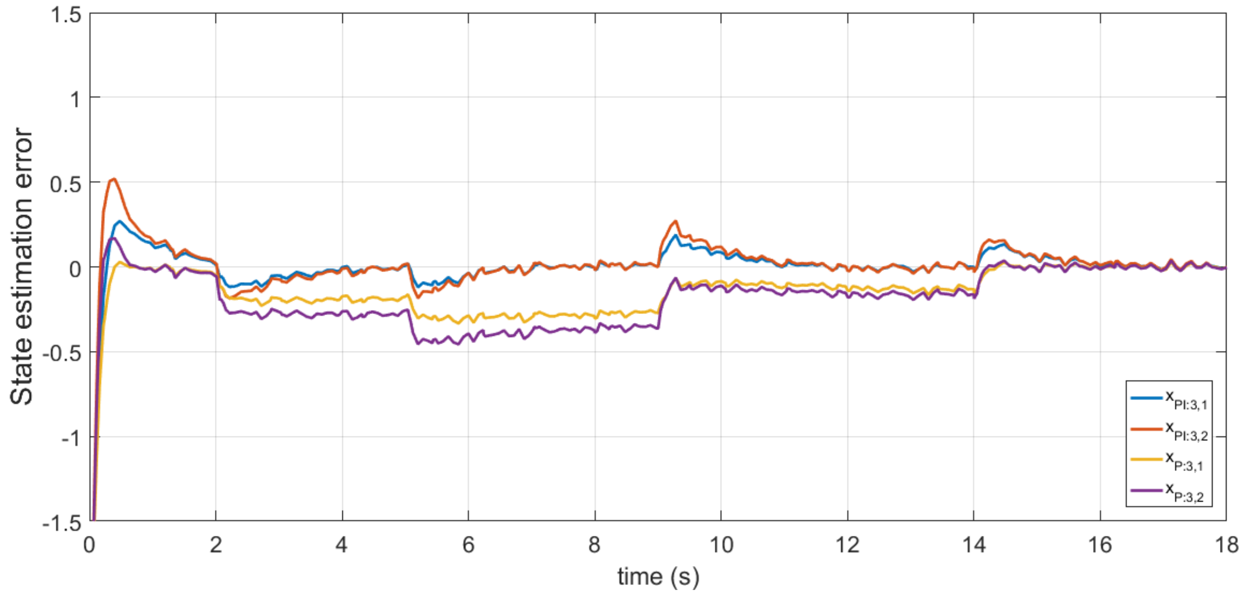

Figure 4.4: State estimation error and ρ.

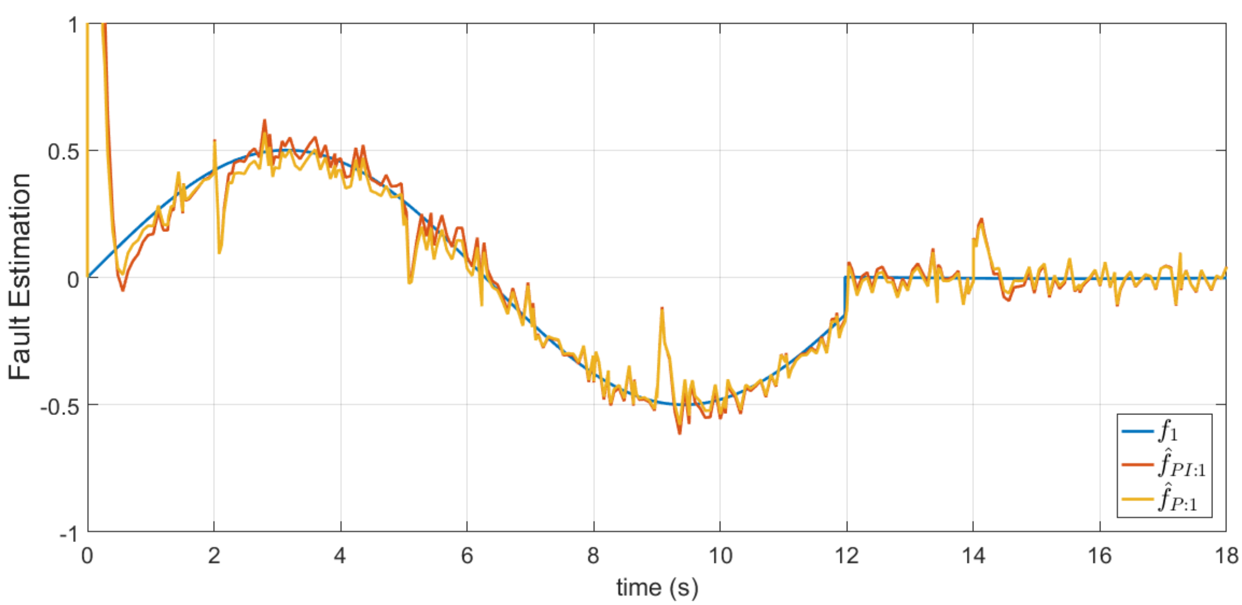

Figure 4.5: State estimation error and ρ.

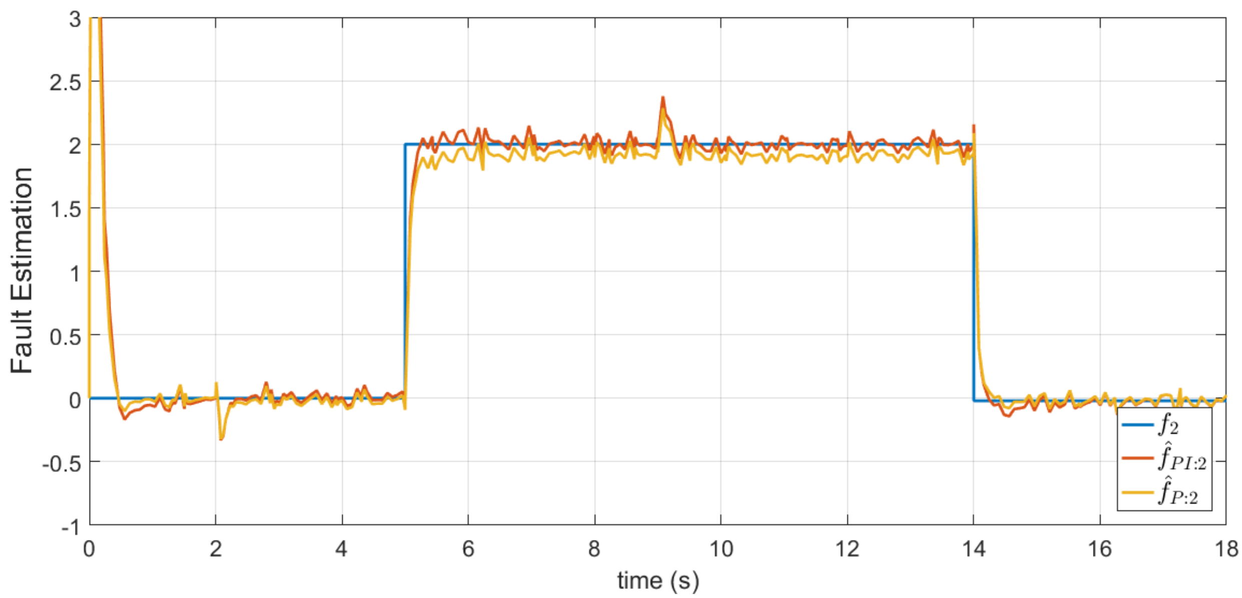

Figure 4.6: State estimation error and ρ.

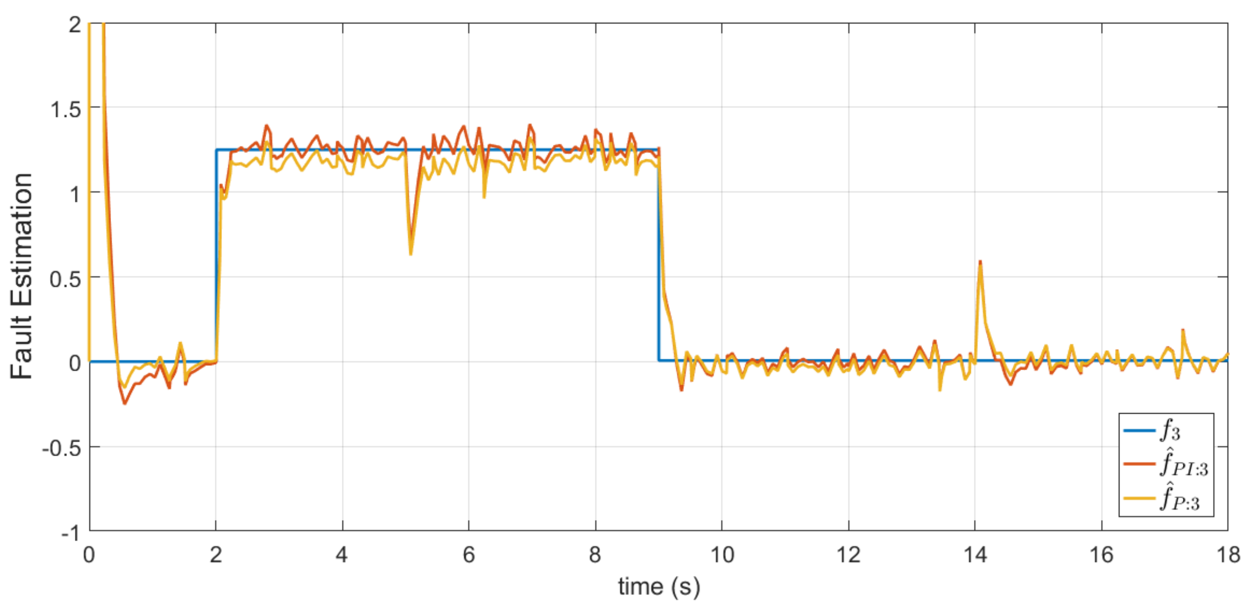

Figure 4.7: State estimation error and ρ.

Chapter 5

Conclusions

5.1 Conclusions

Regarding linear strategy and according to the results presented in Chapter 3, the proposed linear PI observer for leader-following applications estimates efficiently the actuator faults in multiple follower agents in the MAS. In this case, the learning rate in the fault estimator Γ was chosen heuristically, since there is not an established methodology to select an optimal value. Therefore, the observer was tuned with appropriated dimensions of Γ. Furthermore, the graphs can be indistinctly directed or undirected because the Laplacian matrix does not have to be symmetric in Theorem 1, which means that bidirectional or directed communication between the agents is supported. In practical applications, the connectivity involve economic financing, then reducing the edges means a cost reduction as long as the MAS is functional. Therefore, directed graphs should be used when the bidirectional communication is not required for the application. So that, Theorem 1 allows to develop state and fault estimation with reduced costs in applications when it is possible to use directed graphs. On the other hand, the linear observer developed considers only faults with small variations are considered besides the operation region is limited to the linearized model.

Regarding convex strategy, this approach solves the region limitation in the linear observer. This convex PI observer can estimate states and fault in nonlinear systems modeled as convex systems, as it was illustrated before. Further, this estimator is robust to output noise (sensor noise) since the noise vector $w(t)$ is taking into account in the development of the Theorem 2. However, similar to the linear approach, faults and noise are assumed to have only small variations.

This research is focused on the observer design and does not integrate a control algorithm for the MAS.

Other important conclusions of this research work are listed below:

- MAS systems allow developing a distributed consensus protocol with the aim of the coordination of multiple systems working together to reach objectives that can not be achieved by itself.
- MAS is a field of techniques and approaches that should be carefully studied to choose a viable method for the research purpose.
- Convex models offer two crucial advantages in the design of fault diagnosis and fault-tolerant control systems; one is the representation of nonlinear systems without loss of information and the possibility of extending techniques developed for linear systems to nonlinear systems.
- This work will provide contributions in the field of multi-agent systems using a convex model that represents the system throughout the region of work.

Regardless of the limitations that this research presents, the general objective and specific objectives of this research were satisfactorily achieved. Two observers were developed whose efficiency has been proven; both provide to the reader a general theorem (Theorem 1 and Theorem 2) for any system with a feasible solution. A detailed mathematical development is provided for both observers, and simulation examples are given to illustrate the theoretical results.

Future works should study advanced techniques in order to provide a solution which deals with dynamical faults $\dot{f}$ and dynamical noise $\dot{w}$. Likewise, the performance in Theorem 1 and Theorem 2 was not studied, then methods to allocate the poles in a specific region on the left side of the complex plane should be explored to compare the solutions provided in this research and to find better estimations of the faults and the states. Further, due to the multiple systems involve in the solution of the LMIs proposed, it is reasonable to study a poly-quadratic Lyapunov function in order to get better results for both approaches.

Publications

The results of these thesis are summarized in the following publications:

- Conference paper "Observador Proporcional-Integral para Sistemas Multi-Agentes Líder-Seguidor" submitted and accepted to "3ra Jornada de Ciencia y Tecnología Aplicada" (3a JoCyTecA) in the "Centro Nacional de Investigación y Desarrollo Tecnológico" (CENIDET).

- JCR paper "Distributed Fault Estimation of Multi-Agent Systems using a Proportional-Integral Observer: A Leader-Following Application" submitted to the **International Journal of Applied Mathematics and Computer Science (AMCS)**, that is in the "under view" status. The first review was satisfactory: acceptable with "minor reviews".

- Paper in development "Distributed Fault Estimation of convex Multi-Agent Systems for Leader-Following configuration". The target is to submit the paper to the **IET Journal of Control Theory and Applications** that publishes high-quality papers on original, theoretical and experimental research and development in the area of systems and control

Bibliography

Akbarimajd, A., Olyaee, M., Sobhani, B. & Shayeghi, H. (2018), ‘Nonlinear multi-agent optimal load frequency control based on feedback linearization of wind turbines’, *IEEE Transactions on Sustainable Energy* .

Astorga-Zaragoza, C.-M., Osorio-Gordillo, G.-L., Reyes-Martínez, J., Madrigal-Espinosa, G. & Chadli, M. (2018), ‘Takagi—sugeno observers as an alternative to nonlinear observers for analytical redundancy. application to a steam generator of a thermal power plant’, *International Journal of Fuzzy Systems* pp. 1–11.

Bidram, A., Davoudi, A., Lewis, F. L. & Guerrero, J. M. (2013), ‘Distributed cooperative secondary control of microgrids using feedback linearization’, *IEEE Transactions on Power Systems* **28**(3), 3462–3470.

Bidram, A., Davoudi, A., Lewis, F. L. & Qu, Z. (2013), ‘Secondary control of microgrids based on distributed cooperative control of multi-agent systems’, *IET Generation, Transmission & Distribution* **7**(8), 822–831.

Bououden, S., Chadli, M., Filali, S. & El Hajjaji, A. (2012), ‘Fuzzy model based multivariable predictive control of a variable speed wind turbine: Lmi approach’, *Renewable Energy* **37**(1), 434–439.

Boyd, S., El Ghaoui, L., Feron, E. & Balakrishnan, V. (1994), *Linear matrix inequalities in system and control theory*, Vol. 15, Siam.

Cai, H. & Huang, J. (2014), 'The leader-following attitude control of multiple rigid spacecraft systems', *Automatica* **50**(4), 1109–1115.

Cao, Y.-Y. & Frank, P. M. (2001), 'Stability analysis and synthesis of nonlinear time-delay systems via linear takagi–sugeno fuzzy models', *Fuzzy sets and systems* **124**(2), 213–229.

Chadli, M., Aouaouda, S., Karimi, H. R. & Shi, P. (2013), 'Robust fault tolerant tracking controller design for a vtol aircraft', *Journal of the Franklin Institute* **350**(9), 2627–2645.

Chadli, M., Davoodi, M. & Meskin, N. (2016), 'Distributed state estimation, fault detection and isolation filter design for heterogeneous multi-agent linear parameter-varying systems', *IET Control Theory & Applications* **11**(2), 254–262.

Chadli, M., Davoodi, M. & Meskin, N. (2017), Distributed fault detection and isolation filter design for heterogeneous multi-agent lpv systems, *in* 'American Control Conference (ACC), 2017', IEEE, pp. 1610–1615.

Chadli, M., Ragot, J. & Maquin, D. (2005), Quadratic stability and stabilisation of interval takagi-sugeno model: Lmi approach, *in* 'Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.', IEEE, pp. 1035–1038.

Chen, G. & Song, Y.-D. (2015), 'Robust fault-tolerant cooperative control of multi-agent systems: A constructive design method', *Journal of the Franklin Institute* **352**(10), 4045–4066.

Chen, J., Zhang, W., Cao, Y.-Y. & Chu, H. (2016), 'Observer-based consensus control against actuator faults for linear parameter-varying multiagent systems', *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47**(7), 1336–1347.

CURRAN, P. F. (1993), 'Proof of the circle criterion for state space systems via quadratic lyapunov functions—part 2', *International journal of control* **57**(4), 957–969.

Das, B., Subudhi, B. & Pati, B. B. (2016), 'Cooperative formation control of autonomous underwater vehicles: An overview', *International Journal of Automation and computing* **13**(3), 199–225.

Davoodi, M., Meskin, N. & Khorasani, K. (2016), ‘Simultaneous fault detection and consensus control design for a network of multi-agent systems’, *Automatica* **66**, 185–194.

Deng, C. & Yang, G.-H. (2017), ‘Adaptive fault-tolerant control for a class of nonlinear multi-agent systems with actuator faults’, *Journal of the Franklin Institute* **354**(12), 4784–4800.

Duan, Z., Chen, G. & Huang, L. (2009), ‘Disconnected synchronized regions of complex dynamical networks’, *IEEE Transactions on Automatic Control* **54**(4), 845–849.

Estrada, F. L., Ponsart, J. C., Theilliol, D. & Astorga-Zaragoza, C.-M. (2015), ‘Robust h-/h_∞ fault detection observer design for descriptor-lpv systems with unmeasurable gain scheduling functions’, *International Journal of Control* **88**(11), 2380–2391.

Ge, S. S., Hang, C. C., Lee, T. H. & Zhang, T. (2013), *Stable adaptive neural network control*, Vol. 13, Springer Science & Business Media.

Gómez-Peñate, S., López-Estrada, F.-R., Valencia-Palomo, G., Osornio-Ríos, R., Zepeda-Hernández, J., Rios-Rojas, C. & Camas-Anzueto, J. (2018), ‘Sensor fault diagnosis observer for an electric vehicle modeled as a takagi-sugeno system’, *Journal of Sensors* **2018**.

Gonzalez, A. M. & Werner, H. (2014), ‘Lpv formation control of non-holonomic multi-agent systems’, *IFAC Proceedings Volumes* **47**(3), 1997–2002.

Guzmán-Rabasa, J. A., López-Estrada, F. R., González-Contreras, B. M., Valencia-Palomo, G., Chadli, M. & Pérez-Patricio, M. (2019), ‘Actuator fault detection and isolation on a quadrotor unmanned aerial vehicle modeled as a linear parameter-varying system’, *Measurement and Control* p. 0020294018824764.

Hengster-Movric, K., You, K., Lewis, F. L. & Xie, L. (2013), ‘Synchronization of discrete-time multi-agent systems on graphs using riccati design’, *Automatica* **49**(2), 414–423.

Hu, C., Jing, H., Wang, R., Yan, F. & Chadli, M. (2016), ‘Robust h_∞ output-feedback control for path following of autonomous ground vehicles’, *Mechanical Systems and Signal Processing* **70**, 414–427.

Ichalal, D., Marx, B., Ragot, J., Mammar, S. & Maquin, D. (2016), ‘Sensor fault tolerant control of nonlinear takagi–sugeno systems. application to vehicle lateral dynamics’, *International Journal of Robust and Nonlinear Control* **26**(7), 1376–1394.

Kharrat, D., Gassara, H., El Hajjaji, A. & Chaabane, M. (2018), ‘Adaptive observer and fault tolerant control for takagi-sugeno descriptor nonlinear systems with sensor and actuator faults’, *International Journal of Control, Automation and Systems* **16**(3), 972–982.

Kladis, G. P., Menon, P. P. & Edwards, C. (2011), Cooperative tracking for a swarm of unmanned aerial vehicles: A distributed tagaki-sugeno fuzzy framework design, *in* ‘Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on’, IEEE, pp. 4114–4119.

Koubâa, A., Sriti, M.-F., Bennaceur, H., Ammar, A., Javed, Y., Alajlan, M., Al-Elaiwi, N., Tounsi, M. & Shakshuki, E. (2015), Coros: a multi-agent software architecture for cooperative and autonomous service robots, *in* ‘Cooperative Robots and Sensor Networks 2015’, Springer, pp. 3–30.

Kuriki, Y. & Namerikawa, T. (2014), Consensus-based cooperative formation control with collision avoidance for a multi-uav system, *in* ‘2014 American Control Conference’, IEEE, pp. 2077–2082.

Lendek, Z., Guerra, T. M., Babuska, R. & De Schutter, B. (2011), *Stability analysis and nonlinear observer design using Takagi-Sugeno fuzzy models*, Springer.

Lewis, F. L., Zhang, H., Hengster-Movric, K. & Das, A. (2013), *Cooperative control of multi-agent systems: optimal and adaptive design approaches*, Springer Science & Business Media.

Li, J. (2015), ‘Distributed cooperative tracking of multi-agent systems with actuator faults’, *Transactions of the Institute of Measurement and Control* **37**(9), 1041–1048.

Li, S., He, J., Li, Y. & Rafique, M. U. (2016), ‘Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective’, *IEEE transactions on neural networks and learning systems* **28**(2), 415–426.

Li, Z., Duan, Z. & Chen, G. (2011), 'Dynamic consensus of linear multi-agent systems', *IET Control Theory & Applications* **5**(1), 19–28.

Li, Z., Duan, Z., Chen, G. & Huang, L. (2009), 'Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint', *IEEE Transactions on Circuits and Systems I: Regular Papers* **57**(1), 213–224.

Li, Z., Duan, Z., Chen, G. & Huang, L. (2010), 'Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint', *IEEE Transactions on Circuits and Systems I: Regular Papers* **57**(1), 213–224.

Li, Z., Duan, Z., Xie, L. & Liu, X. (2012), 'Distributed robust control of linear multi-agent systems with parameter uncertainties', *International Journal of Control* **85**(8), 1039–1050.

Li, Z., Ren, W., Liu, X. & Fu, M. (2013), 'Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols', *IEEE Transactions on Automatic Control* **58**(7), 1786–1791.

Li, Z., Ren, W., Liu, X. & Xie, L. (2013), 'Distributed consensus of linear multi-agent systems with adaptive dynamic protocols', *Automatica* **49**(7), 1986–1995.

Li, Z., Wen, G., Duan, Z. & Ren, W. (2015), 'Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs', *IEEE Transactions on Automatic Control* **60**(4), 1152–1157.

Lien, C.-H. & Yu, K.-W. (2008), 'Robust control for takagi–sugeno fuzzy systems with time-varying state and input delays', *Chaos, Solitons & Fractals* **35**(5), 1003–1008.

Liu, X. & Fang, Z. (2007), An agent-based intelligent transport system, *in* 'International Conference on Computer Supported Cooperative Work in Design', Springer, pp. 304–315.

Liu, Y., Zhao, Y. & Chen, G. (2016), 'Finite-time formation tracking control for multiple vehicles: A motion planning approach', *International Journal of Robust and Nonlinear Control* **26**(14), 3130–3149.

López-Estrada, F. R., Ponsart, J.-C., Theilliol, D., Zhang, Y. & Astorga-Zaragoza, C.-M. (2016), 'Lpv model-based tracking control and robust sensor fault diagnosis for a quadrotor uav', *Journal of Intelligent & Robotic Systems* **84**(1-4), 163–177.

López-Estrada, F.-R., Rotondo, D. & Valencia-Palomo, G. (2019), 'A review of convex approaches for control, observation and safety of linear parameter varying and takagi-sugeno systems', *Processes* **7**(11), 814.

Ma, J., Zheng, Y. & Wang, L. (2015), 'Lqr-based optimal topology of leader-following consensus', *International Journal of Robust and Nonlinear Control* **25**(17), 3404–3421.

Marx, B., Koenig, D. & Ragot, J. (2007), 'Design of observers for takagi–sugeno descriptor systems with unknown inputs and application to fault diagnosis', *IET Control Theory & Applications* **1**(5), 1487–1495.

Mesbahi, A. & Velni, J. M. (2017), 'Distributed observer-based cooperative control for output regulation in multi-agent linear parameter-varying systems', *IET Control Theory & Applications* **11**(9), 1394–1403.

Nasirian, V., Moayedi, S., Davoudi, A. & Lewis, F. L. (2014), 'Distributed cooperative control of dc microgrids', *IEEE Transactions on Power Electronics* **30**(4), 2288–2303.

Ogata, K. (2010), 'Ingeniería de control moderna. quinta'.

Okdinawati, L., Simatupang, T. M. & Sunitiyoso, Y. (2017), 'Multi-agent reinforcement learning for value co-creation of collaborative transportation management (ctm)', *International Journal of Information Systems and Supply Chain Management (IJISSCM)* **10**(3), 84–95.

Passino, K. M., Yurkovich, S. & Reinfrank, M. (1998), *Fuzzy control*, Vol. 42, Citeseer.

Radke, A. & Gao, Z. (2006), A survey of state and disturbance observers for practitioners, *in* '2006 American Control Conference', IEEE, pp. 6–pp.

Ren, W., Beard, R. W. & Atkins, E. M. (2005), A survey of consensus problems in multi-agent coordination, *in* 'American Control Conference, 2005. Proceedings of the 2005', IEEE, pp. 1859–1864.

Scherer, C. W. (2001), 'Lpv control and full block multipliers', *Automatica* **37**(3), 361–375.

Shi, J., He, X., Wang, Z. & Zhou, D. (2014), 'Distributed fault detection for a class of second-order multi-agent systems: an optimal robust observer approach', *IET Control Theory & Applications* **8**(12), 1032–1044.

Takagi, T. & Sugeno, M. (1985), 'Fuzzy identification of systems and its applications to modeling and control', *IEEE Transactions on Systems, Man, and Cybernetics* (1), 116–132.

Tuna, S. E. (2008), 'Lqr-based coupling gain for synchronization of linear systems', *arXiv preprint arXiv:0801.3390* .

Wang, J.-L. & Wu, H.-N. (2012), 'Leader-following formation control of multi-agent systems under fixed and switching topologies', *International Journal of Control* **85**(6), 695–705.

Wang, X. F. & Chen, G. (2002), 'Pinning control of scale-free dynamical networks', *Physica A: Statistical Mechanics and its Applications* **310**(3-4), 521–531.

Wang, Y., Song, Y. & Lewis, F. L. (2015), 'Robust adaptive fault-tolerant control of multiagent systems with uncertain nonidentical dynamics and undetectable actuation failures', *IEEE Transactions on Industrial Electronics* **62**(6), 3978–3988.

Witczak, M. (2014), 'Fault diagnosis and fault-tolerant control strategies for non-linear systems', *Lecture Notes in Electrical Engineering* **266**, 375–392.

Wu, Y., Wang, Z., Ding, S. & Zhang, H. (2018), 'Leader–follower consensus of multi-agent systems in directed networks with actuator faults', *Neurocomputing* **275**, 1177–1185.

Yan, H., Qian, F., Zhang, H., Yang, F. & Guo, G. (2016), '*h*_{

\

infty} fault detection for networked mechanical spring-mass systems with incomplete information', *IEEE Transactions on Industrial Electronics* **63**(9), 5622–5631.

Yang, H., Staroswiecki, M., Jiang, B. & Liu, J. (2011), 'Fault tolerant cooperative control for a class of nonlinear multi-agent systems', *Systems & control letters* **60**(4), 271–277.

Yang, P., Ma, B., Dong, Y. & Liu, J. (2018), 'Fault-tolerant consensus of leader-following multi-agent systems based on distributed fault estimation observer', *International Journal of Control, Automation and Systems* **16**(5), 2354–2362.

Ye, D., Chen, M. & Li, K. (2017), 'Observer-based distributed adaptive fault-tolerant containment control of multi-agent systems with general linear dynamics', *ISA transactions* **71**, 32–39.

Youssef, T., Chadli, M., Karimi, H. & Wang, R. (2017), 'Actuator and sensor faults estimation based on proportional integral observer for ts fuzzy model', *Journal of the Franklin Institute* **354**(6), 2524–2542.

Zhang, H., Feng, G., Yan, H. & Chen, Q. (2013), 'Observer-based output feedback event-triggered control for consensus of multi-agent systems', *IEEE Transactions on Industrial Electronics* **61**(9), 4885–4894.

Zhang, H., Feng, T., Yang, G.-H. & Liang, H. (2014), 'Distributed cooperative optimal control for multiagent systems on directed graphs: An inverse optimal approach', *IEEE Transactions on Cybernetics* **45**(7), 1315–1326.

Zhang, H., Lewis, F. L. & Das, A. (2011), 'Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback', *IEEE Transactions on Automatic Control* **56**(8), 1948–1952.

Zhang, H., Yang, R., Yan, H. & Yang, F. (2016), 'H consensus of event-based multi-agent systems with switching topology', *Information Sciences* **370**, 623–635.

Zhang, K., Jiang, B. & Cocquempot, V. (2015), 'Adaptive technique-based distributed fault estimation observer design for multi-agent systems with directed graphs', *IET Control Theory & Applications* **9**(18), 2619–2625.

Zhao, B., Xue, M., Zhang, X., Wang, C. & Zhao, J. (2015), 'An mas based energy management system for a stand-alone microgrid at high altitude', *Applied Energy* **143**, 251–261.

Zhao, M., Peng, C., He, W. & Song, Y. (2017), 'Event-triggered communication for leader-following consensus of second-order multiagent systems', *IEEE transactions on cybernetics* **48**(6), 1888–1897.

Zhao, Y., Duan, Z., Wen, G. & Chen, G. (2016), 'Distributed finite-time tracking of multiple non-identical second-order nonlinear systems with settling time estimation', *Automatica* **64**, 86–93.

Zhou, B., Wang, W. & Ye, H. (2014), 'Cooperative control for consensus of multi-agent systems with actuator faults', *Computers & Electrical Engineering* **40**(7), 2154–2166.