EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

# Tecnológico Nacional de México

## Centro Nacional de Investigación y Desarrollo Tecnológico

# Tesis de Doctorado

## Control de formación y coordinación basado en sistemas multiagentes distribuidos desempeñando tareas

presentada por

**MC. Jesus Avelino Vazquez Trejo**

como requisito para la obtención del grado de

**Doctor en Ciencias en Ingeniería Electrónica**

Director de tesis

**Dr. Manuel Adam Medina**

Codirector de tesis

**Dr. Guillermo Valencia Palomo**

**Cuernavaca, Morelos, México. Enero de 2026.**

cenidet®
Centro Nacional de Investigación y Desarrollo Tecnológico

| cenidet® Centro Nacional de Investigación y Desarrollo Tecnológico | ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL | Código: CENIDET-AC-006-D20 |
|---|---|---|
| | | Revisión: O |
| | Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4 | Página 1 de 1 |

Cuernavaca, Mor., a 22 de enero de 2026

**Dr. Carlos Manuel Astorga Zaragoza**
**Subdiretor Académico**
**P r e s e n t e**

**At'n: Dr. Juan Reyes Reyes**
**Presidente del Claustro Doctoral**
**del Departamento De Ing. Electrónica**

Los abajo firmantes, miembros del Comité Tutorial del estudiante Jesús Avelino Vázquez Trejo manifiestan que después de haber revisado el documento de tesis titulado "CONTROL DE FORMACIÓN Y COORDINACIÓN BASADO EN SISTEMAS MULTI-AGENTES DISTRIBUIDOS DESEMPEÑANDO TAREAS", realizado bajo la dirección del Dr. Manuel Adam Medina y la codirección del Dr. Guillermo Valencia Palomo, el trabajo se ACEPTA para proceder a su impresión.

A T E N T A M E N T E

Dr. Jarniel García Morales
CENIDET

Dr. Gerardo Vicente Guerrero Ramírez
CENIDET

Dr. Manuel Adam Medina
CENIDET

Dr. José Fermi Guerrero Castellanos
CENIDET

Dr. Guillermo Valencia Palomo
CENIDET

EDUCACIÓN
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SERVICIOS ESCOLARES
RECIBIDO

c.c.p: Mtra. Verónica Sotelo Boyas/ jefa del Departamento de Servicios Escolares.
c.c.p: Dr. Jarniel García Morales / jefe del Departamento de Ingeniería Electrónica.
c.c.p: Expediente.

SEP TecNM CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
RECIBIDO
22 ENE 2026
SUBDIRECCIÓN ACADÉMICA

# Agradecimientos

En primer lugar, me gustaría agradecer a mis directores de tesis, Manuel Adam-Medina, Guillermo Valencia-Palomo y Jean-Christophe Ponsart, quienes siempre han sido un apoyo inquebrantable. Aprecio cada minuto de su tiempo, la motivación y los consejos que llevaré conmigo siempre. También me gustaría agradecer a los miembros de los comités de revisión de la tesis. En México, quiero expresar mi gratitud a Gerardo Vicente Guerrero Ramírez, Jarniel García Morales y Gabriela Hernández Luna. En Francia, deseo agradecer a los relatores, Pedro Castillo y José Fermi Guerrero Castellanos, así como a los examinadores, Flor Lizeth Torres Ortiz, Mohammed Chadli y Christophe Aubrun, por el tiempo y el esfuerzo que dedicaron a evaluar mi trabajo. Sus críticas reflexivas y su orientación fueron inmensamente apreciadas.

También me gustaría agradecer al CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías) por la beca que me permitió realizar mi doctorado; también agradezco al CENIDET (Centro Nacional de Investigación y Desarrollo Tecnológico) por permitirme usar sus instalaciones y por aceptarme en sus programas de maestría y doctorado en el Departamento de Ingeniería Electrónica (DIE). En Francia, extiendo mi gratitud al CRAN (Centre de Recherche en Automatique de Nancy) y a la Université de Lorraine por trabajar conmigo en el programa de cotutela y por permitirme trabajar en sus instalaciones.

A mis amigos del CENIDET, César Augusto Peregrino de la Cruz, Héctor Martín Cortés Campos, Luis Felipe Avalos Ruiz, con quienes he compartido tantas anécdotas personales y laborales, siempre apreciaré su amistad incondicional. Alexis Alonso Bastida, Christian ("El profe"), Jorge Yusef Colín Castillo, Dulce Alejandra Serrano Cruz, Arturo Daniel Espinosa Pedroza, Dryden Gilberto Estrada Téllez, Jesús Reyes ("Le tocaye"), Alfonso de Jesús Morales, Laura Adriana Galicia Galicia, y a toda la familia del CENIDET con la que me he cruzado a lo largo de estos años, gracias por cada conversación, broma y momentos que ahora y para siempre son inolvidables para mí.

Me gustaría agradecer a mi hermano mayor, a mi mamá y a mis abuelos, ustedes son mi apoyo, y les agradezco su amor incondicional en los buenos y malos momentos, por ser siempre mi guía y un ejemplo para mí. Al resto de mi familia materna, mis tíos, primos y sobrinos, gracias por todos esos domingos familiares que compartimos, y espero que sigamos compartiendo por mucho tiempo.

Por último, me gustaría hacer una mención especial a mi mejor amigo en la vida, Agustín Erasmo Juárez Martínez ("El Guty"), a quien considero más que un amigo, es mi hermano de otra madre. Gracias por estar siempre ahí.

# Acknowledgment

First of all, I would like to thank my thesis supervisors, Manuel Adam-Medina, Guillermo Valencia-Palomo, and Jean-Christophe Ponsart, who have always been an unwavering support. I appreciate every minute of their time, the motivation, and the advice that I will carry with me always. I would also like to thank the members of the thesis review committees. In Mexico, I would like to express my gratitude to Gerardo Vicente Guerrero Ramírez, Jarniel García Morales, and Gabriela Hernández Luna. In France, I wish to thank the rapporteurs, Pedro Castillo and José Fermi Guerrero Castellanos, as well as the examiners, Flor Lizeth Torres Ortiz, Mohammed Chadli, and Christophe Aubrun, for the time and effort they devoted to evaluating my work. Their thoughtful critiques and guidance were immensely appreciated.

I would also like to thank CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías) for the scholarship that allowed me to pursue my doctoral degree; I also thank CENIDET (Centro Nacional de Investigacion y Desarrollo Tecnologico) for allowing me to use its facilities and for accepting me into their master's and doctoral programs in the Department of Electronic Engineering (DIE). In France, I extend my gratitude to CRAN (Centre de Recherche en Automatique de Nancy) and l'Université de Lorraine for working with me in the co-tutelle program and for allowing me to work in their facilities.

To my friends at CENIDET, César Augusto Peregrino de la Cruz, Héctor Martín Cortés Campos, Luis Felipe Avalos Ruiz, with whom I have shared so many personal and work-related anecdotes, I will always appreciate your unconditional friendship. Alexis Alonso Bastida, Christian ("El profe"), Jorge Yusef Colín Castillo, Dulce Alejandra Serrano Cruz, Arturo Daniel Espinosa Pedroza, Dryden Gilberto Estrada Téllez, Jesús Reyes ("Le tocaye"), Alfonso de Jesús Morales, Laura Adriana Galicia Galicia, and the entire CENIDET family with whom I have crossed paths over these years, thank you for every conversation, jokes, and moments that are now and forever unforgettable for me.

I would like to thank my older brother, my mom, and my grandparents you are my support, and I thank you for your unconditional love in good and bad times, for always being my guide and an example for me. To the rest of my maternal family, my uncles, cousins, and nephews, thank you for all those family Sundays we shared, and I hope we continue to share for a long time.

Lastly, I would like to make a special mention to my best friend in life, Agustín Erasmo Juárez Martínez ("El Guty"), who I consider more than a friend he is my brother from another mother. Thank you for always being there.

*This thesis is dedicated to all my Family, thank you for all the love and wisdom.*

# Resumen

Esta tesis explora estrategias de control cooperativo para sistemas multiagente con parámetros variables en el tiempo (LPV). Los sistemas multiagente (MAS) tienen un potencial significativo en diversas aplicaciones, incluyendo sistemas eléctricos, misiones de vigilancia y control de redes de vehículos. Esta investigación desarrolla métodos de control de consenso basados en observadores para sistemas multiagente LPV. El objetivo principal es lograr formaciones deseadas y mantener la estabilidad del sistema. Además, el diseño de control de consenso líder-seguidor basado en observador se centra en el control cooperativo entre agentes bajo fallos de actuadores. Se considera la aplicación de Desigualdades Matriciales Lineales (LMIs) para garantizar la estabilidad en todos los submodelos utilizando el teorema de Polya de la teoría de control difuso. Las estrategias de control propuestas se validan mediante simulaciones, demostrando su eficacia para mantener la formación y lograr consenso incluso en presencia de fallos. La principal contribución del trabajo propuesto es desarrollar una estrategia de control de consenso tolerante a fallos del tipo líder-seguidor basada en observador para MASs LPV. Se derivan condiciones en forma de LMIs para garantizar la existencia de las ganancias del controlador LPV, del observador LPV y del actuador virtual LPV. Si bien se reduce el conservadurismo, el cálculo simultáneo de estas ganancias incrementa las restricciones computacionales.

# Abstract

This thesis explores cooperative control strategies for Linear Parameter-Varying (LPV) multi-agent systems. Multi-agent systems (MAS) have significant potential in diverse applications, including electrical systems, surveillance missions, and vehicle network control. This research develops observer-based consensus control methods for LPV multi-agent systems. The primary objective is to achieve desired formations and maintain system stability.

Further, the observer-based leader-following consensus control design focuses on cooperative control among agents under actuator faults. The application of Linear Matrix Inequalities (LMIs) to ensure stability across all submodels using Polya's theorem from fuzzy control theory is considered. The proposed control strategies are validated through simulations, demonstrating their efficacy in maintaining formation and achieving consensus even in the presence of faults.

The main contribution of the proposed work is to develop an observer-based leader-following fault-tolerant consensus control strategy for LPV MASs. LMIs conditions are derived to ensure the existence of LPV controller, LPV observer, and LPV virtual actuator gains. While reducing conservatism, computing these gains simultaneously increases computational constraints.

# Résumé

Cette thèse examine les stratégies de contrôle coopératif pour les systèmes multi-agents à paramètres linéaires variables (LPV). Les systèmes multi-agents (MAS) offrent un potentiel considérable dans diverses applications, telles que les systèmes électriques, les missions de surveillance et le contrôle des réseaux de véhicules. Cette recherche propose des méthodes de contrôle par consensus basées sur des observateurs pour les systèmes multi-agents LPV, avec pour objectif principal d'obtenir des formations spécifiques tout en maintenant la stabilité du système.

En outre, la conception du contrôle par consensus basé sur des observateurs pour le suivi de leader se concentre sur le contrôle coopératif entre les agents en présence de défaillances des actionneurs. L'application des inégalités matricielles linéaires (LMI) pour garantir la stabilité de tous les sous-modèles, en s'appuyant sur le théorème de Polya issu de la théorie du contrôle flou, est étudiée. Les stratégies de contrôle proposées sont validées par des simulations, démontrant leur efficacité à maintenir la formation et à atteindre le consensus même en cas de défaillances.

La principale contribution de ce travail consiste à développer une stratégie de contrôle par consensus tolérante aux pannes, basée sur des observateurs, pour les MAS LPV en mode suivi de leader. Les conditions LMI sont dérivées pour garantir l'existence des gains du contrôleur LPV, de l'observateur LPV et de l'actionneur virtuel LPV. Bien que cette approche réduise le conservatisme, le calcul simultané de ces gains augmente les contraintes computationnelles.

# List of Symbols

## General System Matrices and Vectors

| | |
|---|---|
| $A, B, C, D$ | System matrices for Linear Time-Invariant (LTI) systems. |
| $A(\zeta(t)), B(\zeta(t)), C(\zeta(t))$ | Parameter-dependent system matrices for Linear Parameter-Varying (LPV) systems. |
| $A_h, B_h, C_h$ | Vertex matrices of the $h$-th submodel in the LPV polytopic representation. |
| $x_i(t)$ | State vector of the $i$-th agent. |
| $u_i(t)$ | Control input vector of the $i$-th agent. |
| $y_i(t)$ | Output vector of the $i$-th agent. |
| $d_{u_i}(t)$ | External disturbance vector acting on the $i$-th agent. |
| $d_{y_i}(t)$ | Sensor noise vector affecting the $i$-th agent. |
| $\hat{x}_i(t)$ | Estimated state vector of the $i$-th agent. |
| $\hat{y}_i(t)$ | Estimated output vector of the $i$-th agent. |
| $e_i(t)$ | State estimation error vector of the $i$-th agent, $e_i(t) = x_i(t) - \hat{x}_i(t)$. |
| $L$ | Observer gain matrix (LTI case). |
| $L(\zeta(t))$ | Parameter-dependent observer gain matrix (LPV case). |
| $L_h$ | Vertex observer gain for the $h$-th submodel. |
| $K$ | Controller gain matrix (LTI case). |
| $K(\zeta(t))$ | Parameter-dependent controller gain matrix (LPV case). |
| $K_h$ | Vertex controller gain for the $h$-th submodel. |

## Multi-Agent System Coordination

| | |
|---|---|
| $N$ | Total number of follower agents. |
| $\mathcal{G}$ | Communication graph. |
| $\mathcal{V}$ | Set of nodes (agents) in the graph, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$. |
| $\mathcal{E}$ | Set of edges (communication links) in the graph. |
| $\mathcal{A} = [a_{ij}]$ | Adjacency matrix of the graph. |
| $a_{ij}$ | Element of the adjacency matrix; $a_{ij} > 0$ if agents $i$ and $j$ are connected. |
| $\mathcal{L} = [l_{ij}]$ | Laplacian matrix of the communication graph. |
| $\mathcal{N}_i$ | Set of neighbors of the $i$-th agent. |
| $x_l(t)$ | State vector of the virtual leader. |
| $\delta_i(t)$ | Synchronization error between the $i$-th agent and the leader, $\delta_i(t) = x_i(t) - x_l(t)$. |
| $\alpha_i$ | Leader-follower connection weight; $\alpha_i > 0$ if agent $i$ is connected to the leader. |
| $\Lambda$ | Diagonal matrix of leader-follower connection weights, $\Lambda = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$. |
| $\overline{\mathcal{L}}$ | Matrix representing the overall leader-following communication, $\overline{\mathcal{L}} = \mathcal{L} + \Lambda$. |
| $\lambda_j$ | $j$-th eigenvalue of the matrix $\overline{\mathcal{L}}$. |
| $H$ | Matrix containing the desired formation vectors $h_i$ for all agents. |
| $h_i$ | Desired relative position vector for the $i$-th agent in the formation. |
| $D_{ij}, d_{ij}$ | Distance error between agents $i$ and $j$ relative to the desired formation. |
| $\Delta_i, \delta_i$ | Synchronization error magnitude between agent $i$ and the leader relative to a desired distance (e.g., formation radius). |

## LPV System and Stability Analysis

| | |
|---|---|
| $\zeta(t), \beta(t)$ | Scheduling parameter vector for the LPV system. |
| $\zeta_1, \zeta_2, \zeta_3$ | Components of the scheduling parameter vector (e.g., $\zeta_1 = \cos(\theta)\cos(\phi)$, $\zeta_2 = \dot{\phi}$, $\zeta_3 = \dot{\theta}$). |
| $\rho_h(\zeta(t)), \Theta_h(\beta(t))$ | $h$-th weighting (or scheduling) function for the LPV polytope. |
| $S$ | Number of vertices in the LPV polytopic representation. |
| $P, P_1, P_2, P_3$ | Positive definite matrices used in Lyapunov functions. |
| $\bar{P}_1, \bar{P}_2$ | Inverses of Lyapunov matrices ($\bar{P}_1 = P_1^{-1}$). |
| $\gamma$ | $\mathcal{H}_\infty$ performance level (attenuation level from disturbances to errors). |
| $\mu, \mu_1, \mu_2$ | Tuning scalar variables used in LMI derivations (Young's relation). |
| $N_c, N_h, M_o, M_h, \mathcal{M}_h, \mathcal{N}_h, \Omega_h$ | Auxiliary matrices defined for LMI synthesis ($N_h = K_h\bar{P}_1$, $M_h = P_2L_h$, etc.). |
| $s$ | Complexity parameter for Polya's theorem. |
| $\otimes$ | Kronecker product operator. |
| $\mathrm{He}\{X\}$ | Hermitian operator, $\mathrm{He}\{X\} = X + X^T$. |
| $I_N$ | Identity matrix of size $N \times N$. |
| $\mathrm{diag}(\cdot)$ | Diagonal matrix operator. |

## Fault-Tolerant Control

| | |
|---|---|
| $B_f(\beta(t), \vartheta_i(t))$ | Faulty input matrix under multiplicative actuator faults. |
| $\vartheta_i(t)$ | Diagonal matrix of multiplicative actuator fault effectiveness factors for the $i$-th agent, $\vartheta_i(t) = \mathrm{diag}(\kappa_{i_1}(t), \ldots, \kappa_{i_r}(t))$. |
| $\kappa_{i_\varepsilon}(t)$ | Effectiveness factor of the $\varepsilon$-th actuator of the $i$-th agent (0: total failure, 1: healthy). |
| $f_{u_i}(t)$ | Additive actuator fault vector for the $i$-th agent. |
| $\hat{\vartheta}_i(t), \hat{f}_{u_i}(t)$ | Estimated multiplicative and additive actuator faults, respectively. |
| $u_{v_i}(t)$ | Virtual actuator output (reconfigured control input) for the $i$-th agent. |
| $y_{v_i}(t)$ | Virtual actuator output vector. |
| $x_{v_i}(t)$ | State vector of the dynamic virtual actuator for the $i$-th agent. |
| $N_v(\beta(t), \hat{\vartheta}_i(t))$ | Virtual actuator input redistribution matrix. |
| $M_v(\beta(t))$ | Parameter-dependent virtual actuator gain matrix. |
| $M_{v_h}$ | Vertex virtual actuator gain for the $h$-th submodel. |
| $B^*(\beta(t))$ | Reconstituted input matrix after virtual actuator compensation. |
| $L_P(\beta(t))$ | Parameter-dependent observer gain used in the FTC scheme. |
| $L_{P_h}$ | Vertex observer gain for the $h$-th submodel in the FTC scheme. |

## UAV Model and Control

| | |
|---|---|
| $\xi = [p_x, p_y, p_z]^T$ | Position vector of the UAV in the inertial frame. |
| $p_x$ | Position coordinate along the x-axis in the inertial frame. |
| $p_y$ | Position coordinate along the y-axis in the inertial frame. |
| $p_z$ | Position coordinate along the z-axis in the inertial frame (altitude). |
| $v = [v_x, v_y, v_z]^T$ | Linear velocity vector of the UAV in the inertial frame. |
| $v_x$ | Linear velocity component along the x-axis in the inertial frame. |
| $v_y$ | Linear velocity component along the y-axis in the inertial frame. |
| $v_z$ | Linear velocity component along the z-axis in the inertial frame. |
| $\Omega = [p, q, r]^T$ | Angular velocity vector in the body-fixed frame. |
| $p$ | Angular velocity component around the x-axis of the body frame (roll rate). |
| $q$ | Angular velocity component around the y-axis of the body frame (pitch rate). |
| $r$ | Angular velocity component around the z-axis of the body frame (yaw rate). |
| $[\phi, \theta, \psi]^T$ | Euler angles vector. |
| $\phi$ | Roll angle (rotation around the x-axis of the body frame). |
| $\theta$ | Pitch angle (rotation around the y-axis of the body frame). |
| $\psi$ | Yaw angle (rotation around the z-axis of the body frame). |
| $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ | Euler angle rates vector. |
| $\dot{\phi}$ | Roll rate (derivative of roll angle). |
| $\dot{\theta}$ | Pitch rate (derivative of pitch angle). |
| $\dot{\psi}$ | Yaw rate (derivative of yaw angle). |
| $R$ | Rotation matrix from the body frame to the inertial frame. |
| $m$ | Total mass of the UAV. |
| $J = \text{diag}(J_x, J_y, J_z)$ | Moment of inertia matrix. |
| $J_x$ | Moment of inertia about the x-axis of the body frame. |
| $J_y$ | Moment of inertia about the y-axis of the body frame. |
| $J_z$ | Moment of inertia about the z-axis of the body frame. |
| $F$ | Total thrust force produced by the motors. |
| $\tau$ | Total torque vector produced by the motors. |
| $u = [u_z, u_\phi, u_\theta, u_\psi]^T$ | Control input vector. |
| $u_z$ | Total thrust force (along the z-axis of the body frame). |
| $u_\phi$ | Torque around the x-axis of the body frame (roll torque). |
| $u_\theta$ | Torque around the y-axis of the body frame (pitch torque). |
| $u_\psi$ | Torque around the z-axis of the body frame (yaw torque). |
| $f_e$ | Thrust force generated by the $\varepsilon$-th propeller. |
| $f_1, f_2, f_3, f_4$ | Thrust forces generated by propellers 1, 2, 3, and 4 respectively. |
| $\Gamma$ | Control effectiveness matrix relating rotor speeds to control inputs. |
| $\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$ | Angular velocities vector of the four rotors. |
| $b$ | Thrust coefficient of the propellers. |
| $d$ | Drag torque coefficient of the propellers. |
| $d_f$ | Drag force coefficient. |
| $d_t$ | Drag torque coefficient. |
| $l$ | Arm length of the quadcopter. |
| $g$ | Acceleration due to gravity. |
| $\mathcal{I}$ | Inertial reference frame. |
| $\mathcal{B}$ | Body-fixed reference frame. |

# Contents

# Chapter 1

# General Introduction

## Contents



## 1.1 Introduction

In recent years, multi-agent systems have gained significant attention due to their superior potential compared to single-agent systems across various applications [1]. These systems consist of multiple autonomous entities, known as agents, which in control theory are defined as autonomous dynamical systems [2]. Multi-agent systems are increasingly being employed in diverse fields, including electrical systems [3], surveillance and reconnaissance missions [4], robotic arm coordination [5], and vehicle network control [6], among others. The design and implementation of multi-agent systems depend on their configuration, which can be categorized into two primary approaches: centralized and distributed. In a centralized approach, a central hub collects information from and transmits control signals to all agents [7]. In contrast, distributed systems rely on local interactions, where agents exchange information only with their neighbors [8]. Additionally, multi-agent systems can be classified as homogeneous or heterogeneous. Homogeneous systems consist of agents with similar characteristics, capabilities, and dynamics, while heterogeneous systems comprise agents with differing attributes [2]. The choice between these configurations depends on the specific requirements and objectives of the application. Regardless of the configuration, a critical challenge in the operation of multi-agent systems is the occurrence of faults, which can manifest in actuators, sensors, or the system dynamics itself [9]. Unlike complete failures that result in total loss of functionality, faults represent partial degradation that progressively impair system performance while allowing continued operation [10]. Faults can be classified according to their origin and characteristics.

Actuator faults manifest as deviations from nominal behavior, including partial effectiveness loss, output biases, or unintended oscillations [11]. Sensor faults typically appear as measurement inaccuracies through biases, signal drift, or noise amplification [12]. System faults, arising from parameter variations, structural damage, or unmodeled disturbances, present particular detection challenges due to their complex interaction with nominal dynamics. Mathematically, faults are categorized as either additive or multiplicative. Additive faults [13] introduce independent deviations to system states or outputs, enabling relatively straightforward detection. Multiplicative faults [14], however, modify system parameters or gains through scaling effects, requiring more sophisticated identification techniques due to their coupled nature with system dynamics. Given the diverse nature and potential impact of faults, addressing these challenges requires robust fault-tolerant control (FTC) techniques. These techniques can be broadly divided into passive and active approaches. Passive techniques treat faults as disturbances and rely on the control system's inherent robustness to compensate for them [15]. Active techniques, on the other hand, adjust the control law based on fault information obtained through detection and isolation methods [16]. One prominent tool in active FTC is the virtual actuator, which simulates the behavior of physical actuators to compensate for faults by adjusting control inputs or system dynamics [17]. For instance, adaptive virtual actuators for linear systems have been proposed to handle uncertainties and disturbances [18]. Similarly, FTC strategies for discrete-time descriptor systems employ virtual actuators and sensors to mitigate faults while addressing challenges such as algebraic loops and control input dependencies [19]. Other applications include FTC for turbofan engines using virtual actuators and gain-scheduled control theory [20], and Linear Parameter-Varying (LPV) systems that reconfigure virtual actuators and sensors to ensure stability under faults and operating point changes [21].

Multi-agent systems have emerged as a powerful paradigm for addressing complex tasks across diverse applications, offering advantages over single-agent systems in terms of scalability, flexibility, and robustness. However, the operation of these systems is often challenged by the occurrence of faults in actuators, sensors, or system dynamics, which can degrade performance and compromise safety. To mitigate these challenges, FTC techniques—ranging from passive robustness to active reconfiguration strategies—have been developed. Among these, virtual actuators and sensors have proven particularly effective in compensating for faults and maintaining system stability. This thesis explores advanced FTC strategies for multi-agent systems, aiming to enhance their reliability and performance in the presence of uncertainties and faults.

## 1.2 Motivation

Multi-agent systems have become indispensable in modern applications due to their ability to achieve complex objectives through collaboration. However, their reliance on interconnected agents makes them vulnerable to disruptions, particularly actuator faults. Such faults can destabilize individual agents and, due to the interconnected nature of the system, propagate through the network, threatening the entire system's ability to achieve consensus. For instance, in a leader-following consensus scenario, a single faulty agent can disrupt the collective behavior. Figure 1.1 depicts a leader-following consensus control scheme for a multi-agent system operating under actuator faults. In this scenario, the absence of a FTC mechanism means that a faulty agent cannot follow the leader or coordinate with neighboring agents. Figure 1.1 uses two gray boxes to visually separate the independent control systems: one for the virtual leader and one for the followers. This same convention is used in the subsequent diagrams. It is important to note that while the leader's control is designed independently, the followers have access to its control law.

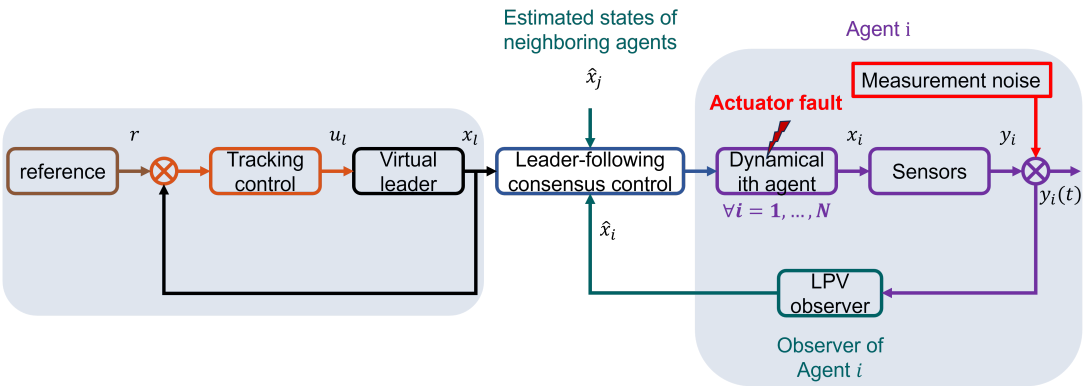


Figure 1.1: Leader-following consensus control for multi-agent systems under actuator faults.

The implementation of FTC in multi-agent systems is critical to ensure their stability and reliability in the presence of actuator faults. Without FTC, a fault in one agent can cascade through the network, leading to widespread performance degradation or even system failure. FTC enables affected agents to dynamically adapt their control strategies, mitigating the impact of faults and preserving the system's ability to achieve consensus. This capability is especially vital in safety-critical applications, such as autonomous flying, autonomous transportation or distributed energy systems, where actuator faults can have severe consequences.

## 1.3 Objectives of the Thesis

The aim of this thesis is to develop an observer-based fault-tolerant consensus control for LPV multi-agent systems. This approach is designed to enable affected agents to compensate for both additive and multiplicative actuator faults, ensuring robust performance under faulty conditions. The main objective of this research is to design a consensus control law for distributed LPV multi-agent systems to achieve desired formations under faulty scenarios. Below, the specific objectives of the research are outlined.

### Specific objectives

- To design a consensus strategy to coordinate LPV multi-agent systems.
- To establish how the agents are affected by the faults.

- To propose a strategy to solve the distributed consensus problem in teams of LPV multi-agent under faulty scenarios.

## 1.4 Contributions and publications

The main contribution is the design of a distributed observer-based fault-tolerant leader-following consensus control law based on estimated states of the neighboring agents to coordinated LPV multi-agent systems in faulty scenarios, where the vertex FTC and observer gains are computed simultaneously based on LMIs.

Related works were published in international journals and conferences, presenting this research's main contributions.

- Vazquez Trejo J. A., Ponsart J-C., Adam-Medina M., Valencia-Palomo G., Vazquez Trejo J. A., and Theilliol D., "Distributed Observer-based Fault-tolerant Leader-following Control of Multi-agent Systems," 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS: Pafos, Cyprus, 8-10 June 2022, vol. 55, no. 6, pp. 691-697.

- Vazquez Trejo J. A., Adam Medina M., Vázquez Trejo J. A., García Beltrán C. D. and García Morales J., "Robust Observer-based Leader-Following Consensus for Multi-agent Systems," in *IEEE Latin America Transactions*, vol. 19, no. 11, pp. 1949-1958, Nov. 2021.

- Vazquez Trejo, J.A., Ponsart JC., Adam-Medina, M., Valencia-Palomo, G., Vazquez Trejo, J.A. (2023). "Distributed Observer-Based Leader-Following Consensus Control Robust to External Disturbance and Measurement Sensor Noise for LTI Multi-agent Systems," in: Theilliol, D., Korbicz J., Kacprzyk J. (eds) *Recent Developments in Model-Based and Data-Driven Methods for Advanced Control and Diagnosis*. ACD 2022. Studies in Systems, Decision and Control, vol 467. Springer, Cham.

- Vazquez Trejo, J. A., Ponsart JC., Adam-Medina M., Valencia-Palomo G. and Theilliol D., "Distributed Observer-based Leader-following Consensus Control for LPV Multi-agent Systems: Application to multiple VTOL-UAVs Formation Control," 2023 International Conference on Unmanned Aircraft Systems (ICUAS), Warsaw, Poland, 2023, pp. 1316-1323.

- Vazquez Trejo, J. A., Ponsart JC., Adam-Medina M., & Valencia-Palomo G. (2024). "Fault-tolerant observer-based leader-following consensus control for LPV multi-agent systems using virtual actuators," *International Journal of Systems Science*, **56**(8), 1816–1833.

## 1.5 Content of the Thesis

The content of the thesis is organized as follows:

- Chapter 2 reviews the state-of-the-art in multi-agent systems, including graph-based theory, general applications of multi-agent systems, and FTC.
- Chapter 3 addresses the design of robust observer-based leader-following consensus control for linear multi-agent systems, focusing on rejecting the effects of external disturbances and measurement sensor noise.
- Chapter 4 presents an observer-based leader-following consensus control strategy for LPV multi-agent systems, with an emphasis on ensuring system stability through the application of fuzzy theory in solving the proposed LMIs.
- Chapter 5 develops a fault-tolerant observer-based leader-following consensus control for LPV multi-agent systems, aiming to mitigate the effects of additive and multiplicative actuator faults using an LPV virtual actuator to reconfigure the control law.
- In Chapter 6, the main conclusions and perspectives are presented.

# Chapter 2

# State-of-the-Art in Multi-Agent Systems

## Contents

## 2.1 Introduction

Multi-agent systems consists of multiple autonomous entities, known as agents, which interact within a shared environment to achieve individual or collective goals [22]. Each agent in a multi-agent system is capable of perceiving its environment, making decisions, and executing actions based on its local knowledge and interactions with other agents. The design of multi-agent system emphasizes decentralization, with agents operating independently and often collaboratively to solve complex problems that are beyond the capabilities of a single agent [23]. This approach is employed in various domains, including robotics [24], distributed sensing [25], network management [26], and automated trading [27], considering the collective intelligence and adaptive capabilities of the agents to enhance system performance and resilience [28].

## 2.2 Introduction to the Cooperative Control of Multi-agent Systems

Multi-agent systems can be mainly designed based on agent uniformity: homogeneous [29] and heterogeneous [30] systems. Each type presents distinct characteristics, advantages, and challenges.

### 2.2.1 Homogeneous Multi-Agent Systems

Homogeneous multi-agent systems, composed of identical agents, offer simplicity and scalability advantages that contrast with the challenges of heterogeneous systems [31]. Their uniform architecture enables easier coordination through shared behaviors and reduces development complexity. While heterogeneous systems must manage coordination challenges from agent diversity, homogeneous designs allow straightforward scaling since additional agents require no architectural changes [32]. The predictability of homogeneous systems ensures consistent performance across all agents. However, this uniformity brings inherent limitations. Such systems demonstrate reduced adaptability when facing tasks requiring specialized capabilities [33]. More critically, the identical nature of agents creates systemic vulnerabilities - any design flaw, security weakness, or environmental condition that affects one agent will likely compromise the entire system. This single point of failure risk is particularly concerning for safety-critical applications, as the system lacks the natural redundancy of diverse architectures where different agent types might remain operational despite certain failures [34]. In conclusion, homogeneous multi-agent systems offer simplicity and reliability through their uniform architecture, making them ideal for predictable, large-scale applications. However, their identical nature limits adaptability and creates potential single points of failure, restricting their use in dynamic environments requiring specialized capabilities.

### 2.2.2 Heterogeneous Multi-Agent Systems

Heterogeneous multi-agent systems integrate diverse agents with distinct capabilities, objectives, and architectures (both hardware and software) to address complex problems. This compositional diversity enables the system to combine specialized expertise while compensating for individual limitations, resulting in superior adaptability and resource efficiency compared to homogeneous systems [35]. Heterogeneous architectures increase flexibility to handle a wider range of tasks through complementary capabilities, enhanced robustness via functional redundancy and specialization, and optimized performance through strategic task allocation based on agent strengths [36]. However, the advantages of heterogeneity come with significant implementation challenges. The integration of dissimilar agents creates complex interdependencies that require careful management, increasing overall system complexity [37]. Additionally, differences in agent objectives and capabilities can lead to conflicts that require advanced coordination strategies to maintain coherent system behavior [38]. While homogeneous systems benefit from simpler coordination through uniformity, heterogeneous systems offer greater versatility at the cost of increased management overhead. The choice between these approaches depends on specific application requirements, with heterogeneous architectures being particularly suited for complex, dynamic environments where their adaptability justifies the additional coordination effort.

### 2.2.3 Representation Frameworks for Multi-Agent Systems

As previously noted, multi-agent systems can be found based on their heterogeneity, distinguishing between homogeneous systems, where all agents share identical characteristics, and heterogeneous systems, where agents exhibit differences in their properties or dynamics. Beyond this point of view, multi-agent systems can also be differentiated by their mathematical representation. In particular, there exist linear time-invariant systems [39], whose dynamics are described by linear models with constant parameters. Furthermore, linear multi-agent systems are typically represented in three different ways. The first corresponds to first-order or single-integrator systems [40], where consensus is limited to coordinating agent positions. The second encompasses second-order or double integrator systems [41], where consensus incorporates both positions and velocities. Finally, in higher-order systems [42], consensus extends to a broader set of states, increasing the complexity of the coordination problem. Another important variant is multi-agent systems with time-varying parameters [43], whose modeling and control present greater challenges due to the need to employ multilinear control techniques, such as those based on fuzzy theory [44]. Furthermore, nonlinear multi-agent systems can be considered [45], while this approach introduces greater difficulties in the analysis and design of coordination strategies, it enables a more accurate representation of complex systems, thereby facilitating a more realistic approximation for various practical applications.

Multi-agent systems exhibit significant diversity in both structural composition and dynamical behavior, ranging from homogeneous to heterogeneous configurations and linear to nonlinear modeling frameworks. This cataloging underscores critical challenges in control design, particularly in achieving an optimal balance between analytical tractability and practical implementation in real-world applications.

### 2.2.4 Consensus Mechanisms in Multi-Agent Coordination

Consensus algorithms serve as the fundamental coordination mechanism for multi-agent systems, enabling distributed agreement through localized interactions. The average consensus algorithms [46] provides a foundational approach where networked agents asymptotically converge to the mean of their initial states through iterative local information exchange. These systems operate under two principal paradigms: leaderless consensus [47] that achieve emergent coordination through decentralized interactions, and leader-follower architectures [48], that incorporate external reference tracking while maintaining prescribed formation constraints. The performance and robustness of consensus algorithms are governed by the following factors. The network topology, whether directed, undirected, or time-varying, fundamentally determines the convergence properties and information propagation characteristics. Communication constraints including time delays [49] and bandwidth limitations necessitate advanced techniques such as event-triggered control [50] to maintain stability while optimizing resource utilization. Robust techniques capable of rejecting the effect of external disturbances [51]. Finally, security considerations against cyber/physical threats [52] demand resilient architectures incorporating FTC mechanisms to ensure reliable operation under adverse conditions. Modern consensus implementations increasingly address these challenges through adaptive control strategies, communication protocols, and distributed security frameworks. The interplay between these factors ultimately determines the achievable performance and operational reliability in multi-agent systems.

### 2.2.5 Applications of Multi-agent Systems

Multi-agent systems have demonstrated diverse potential across different domains by consider the coordinated behaviors at multiple scales. In robotics, swarm systems [53] exemplify homogeneous coordination, enabling scalable solutions for search-and-rescue operations and precision agriculture through emergent self-organization. Similarly, distributed sensor networks [54] achieve robust environmental monitoring by fusing decentralized data streams into coherent situational awareness. The transportation sector benefits from autonomous vehicle platooning [55], where coordinated control strategies optimize traffic flow and reduce energy consumption through aerodynamic coupling. Energy infrastructure likewise employs multi-agent principles in smart grids [56], which dynamically balance heterogeneous generation sources and consumption demands while maintaining grid resilience. For complex, time-critical missions such as disaster response [57], multi-agent systems demonstrate unique advantages by integrating specialized teams of aerial drones, ground robots, and human operators. These applications rely on advanced task allocation strategies [58] that dynamically match mission requirements with agent capabilities using optimization techniques or market-based mechanisms. The resulting systems exhibit both the flexibility to handle unpredictable scenarios and the robustness to operate under constrained conditions.

### 2.2.6 Conclusion

Multi-agent systems present fundamental trade-offs between homogeneity and heterogeneity. Homogeneous architectures offer simplicity and scalability for uniform tasks, while heterogeneous systems enable specialized problem-solving at increased coordination complexity. The mathematical models, algorithms, and applications discussed demonstrate how these approaches address real-world challenges across domains. Emerging research directions include adaptive architectures that dynamically reconfigure agent roles, balancing the benefits of both paradigms for complex, uncertain environments.

## 2.3 Consensus and Coordination of Multi-Agent Systems

In the context of the cooperative control problem, the primary objective is to develop suitable controllers aimed at achieving a specific coordination objective [2]. Strategies for coordinating multi-agent systems include consensus-based approaches [59], which aim to reach an agreement on a common goal, such as selecting a particular action, value, or state. Consensus algorithms can be leaderless consensus and leader-following consensus. Leaderless consensus distributes the responsibility for decision-making across all agents in the system [60]. This is exemplified in traffic management systems, where leaderless multi-agent system can optimize traffic flow by coordinating autonomous vehicles without the need for a central leader. In such systems, agents adjust their speeds based on local information to reduce congestion [61]. On the other hand, leader-following consensus [62] is a method where agents organize themselves into two distinct groups: leaders and followers. Leaders, which can be virtual or another autonomous system, play a prominent role in guiding the consensus process, while followers adjust their behavior to converge towards the decisions made by the leaders. In aerial surveillance missions, for instance, the leader guides the followers, which adjust their positions to maintain formation and surveillance coverage [63]. Formation control strategies [64] organize agents into virtual structures or formations to achieve coordinated behavior and accomplish specific tasks. These strategies use virtual structures to facilitate communication, collaboration, and synchronization among agents without the need for explicit centralized control. Behavior-based strategies [65] focus on designing individual agent behaviors to get a collective behavior of the multi-agent systems. Instead of relying on explicit communication, agents exhibit autonomous behaviors that interact with those of other agents and the environment to achieve coordination. Artificial potentials [66], where the interaction between agents and their environment are modeled as potential fields. These strategies are inspired by physics and use the concept of artificial forces to control agents, avoid collisions, and monitor trajectories and targets. Graph-theory methodologies, such as those discussed in [67], are frequently employed to model the communication networks among agents and ensure the stability of the multi-agent systems. It considers the mathematical framework of graph-theory to model the relationships and interactions between agents. These strategies use graph properties, algorithms, and structures to facilitate agents' coordination, communication, and decision-making. Further, common objectives of the coordination of multi-agent systems are: flocking [68], which describes collective behavior where agents align their velocities while maintaining proximity to neighbors and avoiding collisions, resulting in emergent self-organized motion patterns; formation control [69], where agents achieve and maintain a predefined geometric configuration; and formation tracking [70], an advanced form of formation control where the entire formation must additionally track a reference trajectory while preserving the desired spatial relationships among agents.

Coordination and control refer to the systematic approaches for governing interactions and behaviors among autonomous agents to achieve collective objectives. Coordination specifically involves the alignment of agent actions, information sharing, and collaborative execution of tasks toward common or complementary goals. Effective coordination harmonizes agent activities to minimize conflicts and redundancies while optimizing system-wide performance. Key enabling techniques include negotiation protocols, consensus algorithms, and distributed task allocation methods, which facilitate cooperation among heterogeneous agents with diverse capabilities and objectives [71].

Control in multi-agent system refers to the methodologies used to guide and regulate the actions of individual agents and the system as a whole. This includes designing control laws and protocols that dictate how agents respond to changes in the environment, interact with other agents, and adjust their behavior to meet collective objectives. Control strategies in multi-agent system can be centralized, decentralized, or distributed approaches [72]. Centralized control involves a single entity making decisions for all agents [73], whereas decentralized control allows each agent to make its own decisions based on local information [74]. Distributed control combines elements of both, enabling agents to coordinate through local interactions without a central authority, thus enhancing scalability and robustness [75].

### 2.3.1 Centralized Multi-Agent Systems

In centralized approaches, a single central authority manages and coordinates the actions and decisions of all agents within the system [2]. This central controller maintains a global view of the environment and the state of each agent, enabling it to make informed, system-wide decisions. By consolidating control, this approach simplifies coordination, as the central entity can efficiently allocate tasks, resolve conflicts, and optimize overall performance [74]. However, centralized systems may face scalability challenges and represent a single point of failure, making them less robust than decentralized or distributed approaches particularly in dynamic or large-scale environments [76]. Figure 2.1 shows a centralized multi-agent system, where the arrows represent the communication between the UAVs and the central computer.

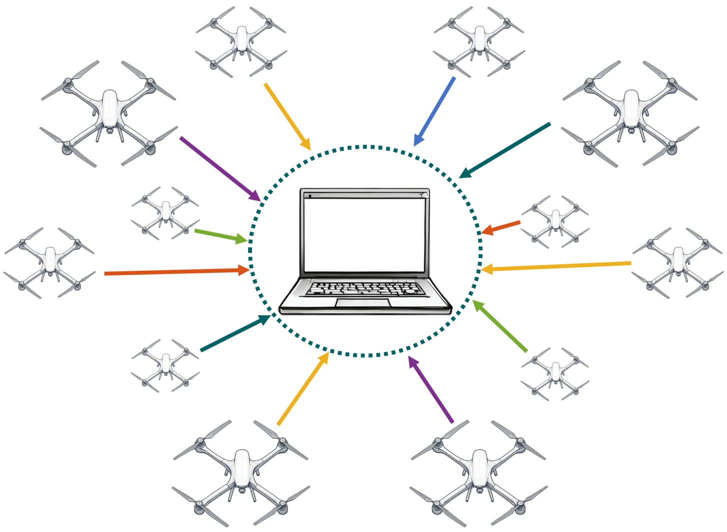

Figure 2.1: Example of a centralized UAV network.

Centralized multi-agent systems simplifies coordination and decision-making by relying on a single authority to manages and directly all agents behavior, ensuring coherent and unified operations [77]. With its comprehensive perspective, the central entity optimizes resource allocation and resolves conflicts, enhancing system efficiency. Additionally, its global awareness of agent states and environmental conditions allows for more strategic decision-making, further improving performance [78].

### 2.3.2 Decentralized Multi-Agent Systems

In decentralized multi-agent systems, each agent operates independently, making decisions based on local information and interactions with neighboring agents, eliminating the need for a central authority [79]. This approach improves scalability and robustness since the system lacks a single point of control and can remain operational even if individual agents fail. Coordination is achieved through algorithms and protocols that enable cooperation, task allocation, and information sharing among agents [80]. As a result, the system can adapt to dynamic environments and diverse tasks by consider the collective intelligence of all participating agents. Figure 2.2 shows a decentralized multi-agent system, where it is shown that control does not depend on a single central unit, but rather there are multiple units that calculate the control signals and these units must be in communication with each other.

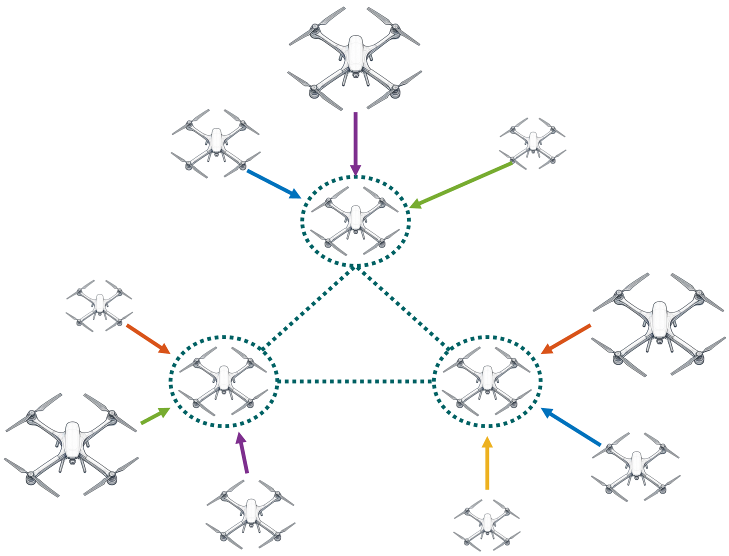

Figure 2.2: Example of a decentralized UAV network.

The decentralized approach offers several advantages. First, it enhances scalability because each agent operates independently, allowing seamless addition or removal of agents without disrupting overall performance. Second, it improves robustness and fault tolerance since the system does not rely on a central authority; faults in individual agents do not compromise the entire system [81]. Third, decentralized control provides greater flexibility and adaptability, enabling agents to respond quickly to local changes and dynamic environments. Finally, these systems reduce communication overhead as agents only need to exchange information with nearby neighbors, leading to more efficient resource usage and lower decision-making latency [82].

### 2.3.3 Distributed Multi-Agent Systems

A distributed multi-agent system is a computational framework where control and decision-making are shared among multiple agents without central oversight [83]. In this architecture, each agent operates autonomously using local information while cooperating with others through structured communication protocols to achieve common objectives. This approach combines decentralized operation with enhanced coordination mechanisms, enabling agents to achieve consensus, allocate tasks efficiently, and synchronize their actions systematically [84]. Distributed systems are particularly characterized by their scalability, fault tolerance, and dynamic adaptability, making them ideal for complex environments requiring collaborative resource management and coordinated responses to changing conditions [85]. Figure 2.3 shows a distributed multi-agent system, where the calculation of the control signals necessary for the multi-agent team is performed in a distributed manner among the agents themselves, the colored lines represent a possible information flow between the agents.

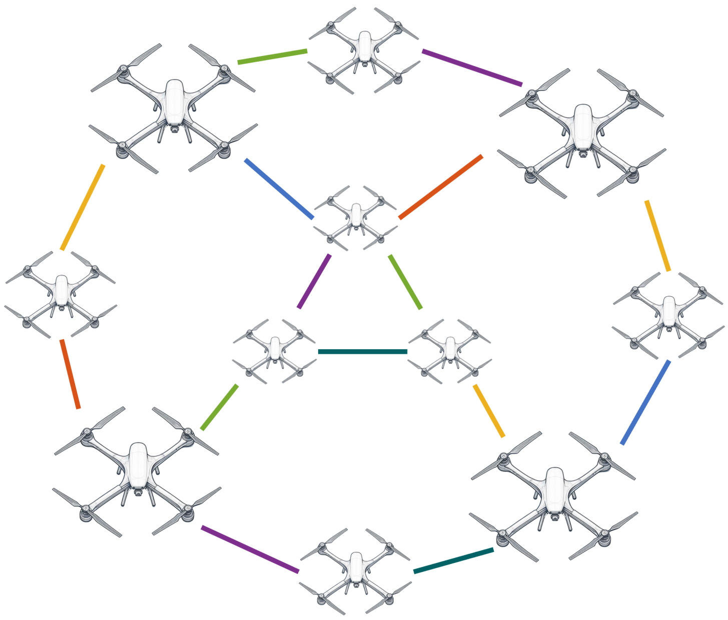

Figure 2.3: Example of a distributed UAV network.

While both decentralized and distributed multi-agent systems eliminate centralized control, they differ fundamentally in their coordination mechanisms. In decentralized systems, agents act independently based solely on local information and neighbor interactions, with system-wide coordination emerging organically from these local interactions [86]. Distributed systems, conversely, employ explicit cooperation strategies where agents actively share decision-making responsibilities through communication protocols [87]. The main difference between distributed and decentralized systems lies in their coordination mechanisms: distributed systems rely on direct cooperation protocols, while decentralized systems depend on emergent coordination through local rules.

### 2.3.4 Conclusion

Consensus control and coordination are fundamental to the effective operation of multi-agent systems. The selection of an appropriate approach, such leaderless or leader-following consensus, centralized, decentralized, or distributed coordination, depends on the system's objectives, environmental demands, and operational constraints. Leaderless consensus enables distributed decision-making, promoting flexibility and robustness through emergent coordination. In contrast, leader-following consensus provides structured guidance via hierarchical control, enhancing predictability in mission-critical tasks. Centralized systems offer globally optimized decision-making but face limitations in scalability and single-point vulnerability. Decentralized and distributed architectures, however, improve fault tolerance and adaptability by distributing control across agents, making them suitable for dynamic, large-scale environments. Effective coordination relies on diverse methodologies, including behavior-based algorithms, formation control techniques, and graph-theory frameworks, to ensure system synchronization and operational efficiency.

## 2.4 Application of Select Graph Theory Concepts in Multi-Agent Systems

In multi-agent systems, graph theory provides a mathematical framework to model and analyze the relationships and interactions between agents [88]. The two primary types of graphs used are directed and undirected graphs, each serving distinct purposes in modeling agent relationships. A directed graph represents one-way relationships where influence or communication flows in a specific direction. For example, if Agent A sends data to Agent B but not vice versa, this is shown as a directed edge from A to B. These graphs are useful for modeling hierarchical systems (like leader-follower setups) or networks where information transfer is unidirectional [89]. An undirected graph, in contrast, represents mutual or bidirectional interactions where communication is two-way. If Agent A and Agent B share data equally, their connection is an undirected edge. These graphs are common in cooperative systems where all agents work together symmetrically.

To model the communications between agents consider a directed graph $\mathcal{G}$ with nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ representing the agents. The set of edges, $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}\} \subseteq \mathcal{V} \times \mathcal{V}$, defines the relationships between nodes. The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ is characterized by $a_{ii} = 0$ and $a_{ij} > 0$ if there is an connection between nodes $i$ and $j$ otherwise $a_{ij} = 0$. For undirected graphs, symmetry is enforced where $a_{ij} = a_{ji}$ for all $i \neq j$, ensuring that the adjacency matrix is symmetrical $\mathcal{A} = \mathcal{A}^T$.

The Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N \times N}$ represents the information exchange between agents and is defined for directed graphs as $l_{ii} = \sum_{j \neq i} a_{ij}$ and $l_{ij} = -a_{ij}$. For an undirected graph, $l_{ij} = -a_{ij} = -a_{ji}$, ensuring symmetry. The set of neighbors associated with the $i$-th agent is denoted as $\mathcal{N}_i$.

In this framework, the concept of a leader is introduced. The leader serves as a reference point for the other agents, guiding them towards achieving consensus. This leader can be virtual or another agent. The parameter $\Lambda = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N) \in \mathbb{R}^{N \times N}$, where $\alpha_i$ indicates the communication exchange between the leader and the $i$-th follower. Specifically, $\alpha_i = 1$ signifies a link between the $i$-th follower and the leader, while $\alpha_i = 0$ indicates the absence of such a link. The communication in the leader-follower multi-agent system is expressed as $\bar{\mathcal{L}} = \mathcal{L} + \Lambda$, which is positive definite. The following figures present examples of six-node graphs to illustrate their key characteristics [2]. In graph theory, a directed edge is typically represented by a line with a single arrowhead, while an undirected edge is depicted as a line with either bidirectional arrows or no arrows at all. Figure 2.4 shows an undirected graph where all edges are bidirectional. This implies that if an agent transmits information to another, the connection allows reciprocal communication.

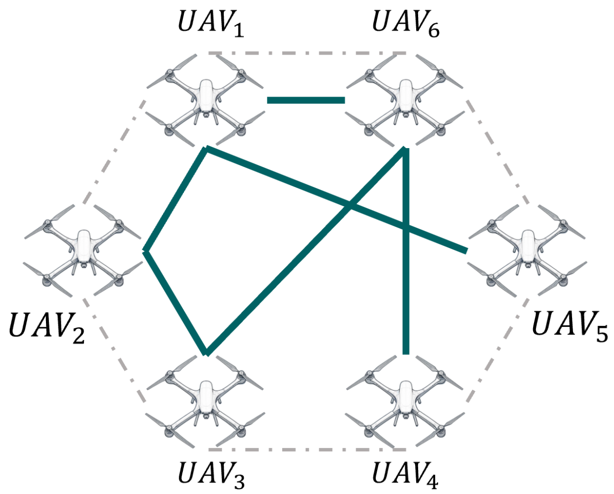


Figure 2.4: Undirected connected graph.

A directed graph has edges with specific directions, while a strongly connected graph is a special type of directed graph where there is a directed path from every node to every other node, ensuring mutual reachability in both directions. Figure 2.5 shows a strongly connected directed graph. Note that for undirected graphs, this property is simply called connectedness. A directed graph is considered complete if it contains edges from every node to all other nodes.

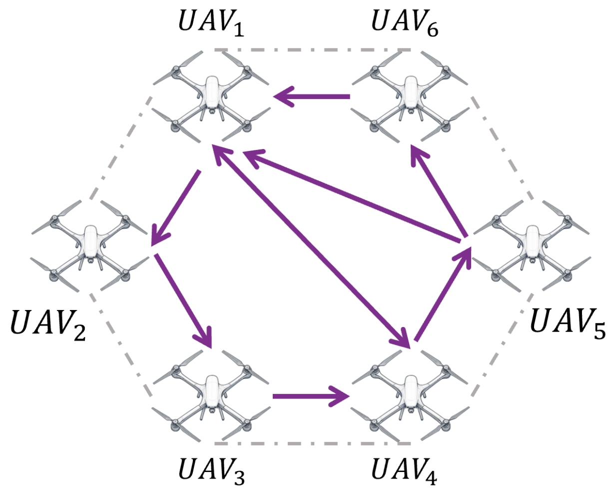


Figure 2.5: Strongly connected graph.

Figure 2.6 displays a balanced, strongly connected graph. A graph is balanced when each node has equal numbers of incoming and outgoing edges.

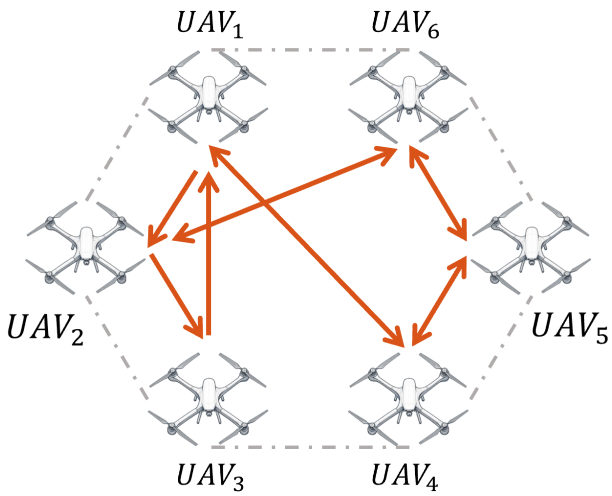


Figure 2.6: Balanced and strongly connected graph.

Figure 2.7 shows a directed spanning tree. A rooted directed tree is a directed graph where every node has exactly one parent except for the root node, which has no parent and contains directed paths to all other nodes. A directed spanning tree is a tree that includes all nodes of the original graph.

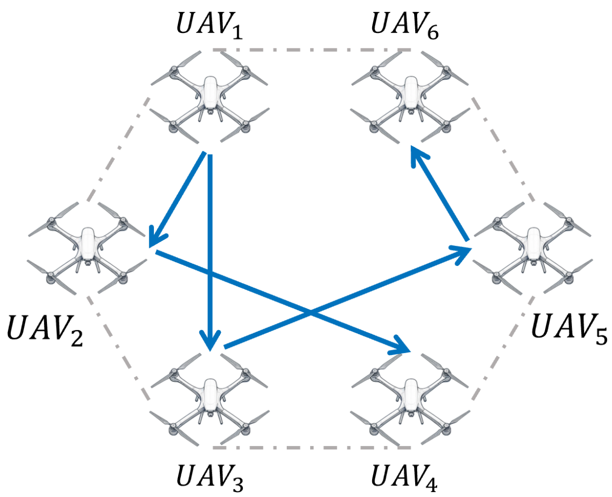


Figure 2.7: Directed spanning tree.

Graph theory offers a mathematical framework for modeling and analyzing interactions in multi-agent systems. The presented graph types, directed, undirected, strongly connected, balanced, and spanning trees, capture essential structural properties that influence system behavior. These concepts not only formalize agent relationships but also provide the tools necessary to design and analyze multi-agent networks.

## 2.5 Control of Multi-Agent Systems: LTI and LPV Approaches

Linear multi-agent systems are widely used to model and control the coordinated behavior of multiple autonomous agents. These systems allow for the design of distributed control strategies, ensuring effective communication and synchronization among agents. In a linear multi-agent system, interactions between agents can be represented using linear state-space models with fixed coefficients describing the system dynamics. This framework enables the application of established linear control techniques, including consensus algorithms and observer-based control, to maintain network stability, coordination, and performance. By use these linear models, effective solutions can be developed for challenges such as agent coordination, formation control, and fault tolerance. For example, [90] introduces a novel flocking control algorithm for multi-agent systems that utilizes historical information to improve convergence rates. Theoretical analysis and simulations confirm the algorithm's advantages over conventional methods, demonstrating faster convergence and improved coordination among agents. Expanding the discussion to event-triggered mechanisms, [91] presents an adaptive event-triggered control for LTI multi-agent systems, suitable for strongly connected directed graphs and general directed graphs containing a spanning tree. This method handles leader-follower consensus by tracking a leader with bounded control inputs. The protocol combines adaptive control with event-triggered communication, enabling fully distributed implementation that requires only relative neighbor information at discrete event times, eliminating the need for global Laplacian matrix computations. Further enhancing fault tolerance and communication efficiency, [92] proposes a distributed adaptive event-triggering mechanism that compensates for actuator faults and non-zero leader inputs. The design ensures the asymptotic convergence of tracking errors to zero while significantly reducing communication frequency, making the approach highly efficient in resource-constrained environments. Furthermore, [93] focuses on the design of distributed observers for estimating states and detecting sensor faults in linear multi-agent systems. Based on relative output estimation errors, the adaptive fault-tolerant output regulation ensures system stability and convergence of tracking errors to zero. By solving the regulator equation, the proposed control strategy achieves consensus under sensor faults and directed topology. Simulations confirm the effectiveness of this approach, showing that all tracking errors converge to zero, even in the presence of sensor faults. The use of UAVs has emerged as a highly promising platform for autonomous navigation. These systems have been applied across various fields, including agriculture assessment [94], search and rescue missions [95], marine exploration [96], and fire monitoring [97]. Within the realm of multi-agent systems involving UAVs, research has explored numerous approaches, with particular attention to leader-following consensus for LTI systems. Key areas of focus include second-order multi-agent systems with irregular discrete sampling times [98], particle swarm optimization [99], event-triggered mechanisms [100], and strategies for addressing communication faults [23]. However, multi-agent systems are inherently vulnerable to external disturbances and measurement noise, which can disrupt the synchronization and coordination of agents. To mitigate these challenges, robust control strategies are essential. By incorporating exogenous inputs into the system's dynamic model, such strategies enable the development of control and observation schemes that effectively counteract disturbances and sensor noise, thus preserving system performance [101]. Several robust control strategies have been proposed in the literature to address these challenges. For instance, [102] introduces a finite-time control method designed to reject external disturbances in second-order multi-agent systems. Similarly, [103] proposes a sliding mode leader-following control strategy for UAVs, emphasizing resilience to disturbances. Further contributions include [104], which presents a sliding mode formation control approach for underactuated surface vehicles that accounts for model uncertainties and external disturbances. Additionally, [105] employs an $H_2$ performance controller to minimize errors and disturbances in systems with communication and input time delays. Other notable approaches include reinforcement learning-based optimal formation control for heterogeneous multi-agent systems under external disturbances [106], and an adaptive bipartite consensus scheme for systems with connected switching topologies under input saturation [107].

LPV systems offer a more effective representation of nonlinear systems compared to traditional models. These systems enable the design of gain-scheduling controllers with time-varying parameters, resulting in superior control performance. An LPV system can be viewed as a specialized linear system, where the coefficient matrix in the state-space model changes in response to variations in scheduling parameters. These variations, driven by time-varying parameters, allow the use of linear system techniques and theories to enhance control strategies [108]. In the context of LPV multi-agent systems, each agent's dynamics evolve based on parameters that change over time or space. These systems are modeled using linear parameter-varying frameworks, which enable adaptive control to ensure stability and coordination across the network. For example, [109] introduced an observer-based consensus control approach for handling actuator faults in LPV multi-agent systems. Building on this, [110] investigated a gain-scheduled observer-based consensus, where the controller and observer gains are functions of a time-varying parameter vector that is measured in real time. Polya's theorem [111] was applied in this work to reduce conservatism compared to [109].

## 2.6 Fault-Tolerant Control for Multi-Agent Systems

In multi-agent systems, control objectives critically depend on coordination based on information exchange between agents. This interdependence means that faults in a single agent's actuators or sensors can propagate to its neighbors, degrading collective performance. This highlights the critical need to implement FTC schemes capable of fault detection and isolation, dynamic fault compensation through reconfiguration of the affected agent, and local containment of fault effects to prevent system-wide propagation.

To address these challenges, various FTC methods have been developed. A significant body of work focuses on LTI multi-agent systems. For instance, actuator fault estimation and FTC are proposed in [114]. Furthermore, [117] presents a fault-tolerant leader-following consensus control using dynamic virtual actuators, an approach extended in [118] for event-triggered leader-following formation control under communication faults in UAV fleets.

Concurrently, advancements have been made for more complex models. An active FTC approach using a distributed observer-based fault detection and compensation mechanism is presented in [115]. This method, grounded in Lyapunov's stability theory and $\mathcal{H}\infty$ control, adjusts control gains in real-time to ensure stability. Other contributions include fault detection and isolation filters for LPV heterogeneous multi-agent systems [116], and an observer-based consensus control to compensate for actuator faults in LPV systems using robust $\mathcal{H}\infty$ criteria [109]. A comprehensive FTC strategy incorporating virtual actuators and sensors for both LTI and LPV systems is proposed in [119].

While significant progress has been made for LTI systems, challenges remain in the LPV case, particularly in fault estimation. Recent research continues to address these open problems. For example, a distributed fault detection and isolation filter design for heterogeneous LPV systems using an unknown input observer structure is proposed in [134]. Similarly, [135] presents a scheme designing Unknown Input Observers using LMIs for robustness and fault sensitivity in a finite frequency domain.

Further developments include a leader-following consensus protocol for systems subject to bounded disturbances based on the quadratic boundedness concept [136], and a resilient formation control strategy for UAVs under false data injection cyber-attacks [137]. The development of a digital twin for cooperative load transportation using multiple multi-rotor drones is shown in [138].

Recent schemes also aim to reduce communication costs. An event-triggered fault-tolerant control scheme employing a unified virtual actuator and sensor approach is presented in [139]. Furthermore, [140] introduces a distributed fault estimation framework for nonlinear multi-agent systems using a Takagi-Sugeno model and a dynamic event-triggered protocol. Lastly, [141] addresses the simultaneous fault estimation and fault-tolerant consensus control problem for nonlinear multi-agent systems subject to multiple time-varying delays and disturbances.

## 2.7 Analysis of the State-of-the-art

Multi-agent systems range from homogeneous designs, which prioritize simplicity and scalability, to heterogeneous configurations, which enhance adaptability and robustness at the cost of increased coordination complexity. The mathematical representation of agent dynamics is primarily accomplished through LTI models for their analytical tractability, or LPV frameworks to more accurately capture complex, nonlinear behaviors.

Consensus algorithms form the core coordination mechanism, enabling agreement through either emergent, decentralized interactions or structured, hierarchical control. These strategies are implemented within centralized, decentralized, or distributed architectures, each offering distinct trade-offs between global optimization and system resilience. Graph theory provides the essential formalism for analyzing the underlying communication topologies.

Consequently, FTC has emerged to ensure reliability against actuator and sensor faults, with significant advances concentrated on LTI systems. While LTI models provide a tractable foundation, their inability to fully capture real-world nonlinearities is a significant limitation. LPV frameworks offer a more powerful alternative but introduce substantial design and computational challenges, creating a persistent tension between model fidelity and practical implementation.

Robust FTC for LTI systems is well-established, yet its extension to LPV systems remains markedly underdeveloped. This gap in reliably diagnosing and compensating for faults within parameter-varying environments constitutes a major obstacle for deploying multi-agent systems in safety-critical applications. The integration of resource-efficient, event-triggered communication with these advanced FTC schemes presents a further compelling challenge.

| Article | Leaderless/Leader-following | LTI/LPV/Nonlinear MAS | Homogeneous/Heterogeneous | FDI | FTC | Actuator Faults | Sensor Faults |
|---|---|---|---|---|---|---|---|
| Guerrero-Castellanos et al. (2017) | Leaderless | Nonlinear | Homogeneous | ✗ | ✗ | ✗ | ✗ |
| Jianliang Chen et al. (2017) | Leaderless | LPV | Homogeneous | ✗ | Passive | ✓ | ✗ |
| Chadli et al. (2017) | Leaderless | LPV | Heterogeneous | ✓ | ✗ | ✗ | ✓ |
| Susbielle et al. (2019) | Leader-following | Nonlinear | Homogeneous | ✗ | ✗ | ✗ | ✗ |
| Davoodi et al. (2019) | Leaderless | LPV | Heterogeneous | ✓ | ✗ | ✓ | ✓ |
| Juan Vazquez-Trejo, et al. (2022) | Leader-following | LTI | Homogeneous | ✗ | ✗ | ✗ | ✗ |
| Rotondo et al. (2022) | Leaderless | LPV | Homogeneous | ✗ | ✗ | ✗ | ✗ |
| Juan Vazquez-Trejo, et al. (2023) | Leader-following | LTI | Homogeneous | ✗ | ✗ | ✗ | ✗ |
| Rotondo et al. (2024) | Leaderless | Both | Homogeneous | ✗ | ✓ | ✓ | ✓ |
| Wang & Chadli (2024) | Leader-following | LTI | Homogeneous | ✗ | ✓ | ✓ | ✓ |
| Wang & Chadli (2025) | Leader-following | LPV | Homogeneous | ✓ | ✗ | ✓ | ✓ |
| Yimin Liu et al. (2025) | Leaderless | Nonlinear | Homogeneous | ✓ | ✓ | ✓ | ✗ |

Figure 2.8: Summarize of the state-of-art.

In Fig. 2.8, the most important articles consulted for the state-of-the-art review are shown, which allowed us to visualize our contribution. We want to highlight that our contribution is the novel design of an observer-based leader-following fault-tolerant consensus control using virtual actuators for LPV multi-agent systems, validated through an application to a fleet of unmanned aerial vehicles. Furthermore, an opportunity still exists to solve the fault-tolerant control problem by considering a complete estimation and compensation scheme for both types of faults, in actuators and sensors.

## 2.8 Conclusions

In this chapter, a comprehensive review of the current state-of-the-art in control strategies for multi-agent systems, with a particular focus on LTI and LPV frameworks, has been presented. Additionally, the challenges associated with achieving robustness and fault tolerance in dynamic and uncertain environments were discussed, along with the strategies developed to address these problems.

The analysis of existing literature has revealed significant progress in the field, particularly in the design and implementation of control strategies that enhance the stability and performance of multi-agent systems under varying operational conditions. However, it also identified several areas where further research is needed, including the integration of more sophisticated fault diagnosis mechanisms and the adaptation of these control methods to increasingly complex multi-agent configurations.

This chapter has laid the groundwork for the subsequent development of novel control strategies proposed in this thesis. By critically examining the strengths and limitations of current approaches, it has provided a clear understanding of the existing gaps and opportunities, thereby setting the stage for the innovative contributions that follow in the succeeding chapters.

# Chapter 3

# Observer-Based Leader-Following Consensus Control for Linear Multi-Agent Systems

## Contents



## 3.1 Introduction

This chapter presents the design of a distributed leader-following control law for coordinated linear multi-agent systems subject to external disturbances and sensor noise. Figure 3.1 shows the proposed distributed leader-following observer-based robust control scheme. Consensus among agents is achieved through inter-agent communication. Each agent is modeled as an autonomous dynamic system capable of processing, transmitting, and receiving information. The exchanged data consists of estimated state vectors, which may include positions, velocities, accelerations, or other variables depending on the system dynamics. As depicted in Figure 3.1, agent $i$ can compute its own consensus control law and estimate its state vector for cases when the complete state vector is unavailable. It then shares this information with neighboring agents $j$. The robust control and observer gains are determined simultaneously using LMIs, taking into account external disturbances and measurement noise. The primary objective is to ensure that the agents follow the trajectories defined by a virtual leader while maintaining consensus among the followers, despite the presence of external disturbances and sensor noise.

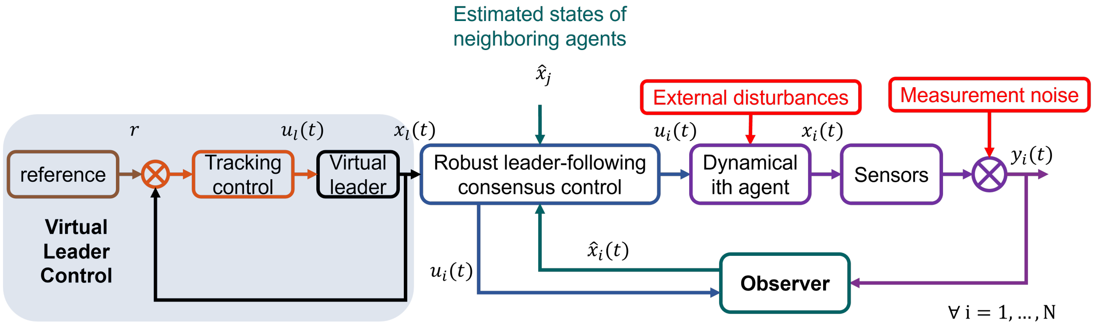


Figure 3.1: Diagram of the proposed robust control of LTI multi-agent systems.

## 3.2 Robust Observer-Based Leader-Following Consensus Control

### 3.2.1 Problem Formulation and System Description

Consider a linear homogeneous multi-agent system under external disturbances and sensor noise as follows:

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + Bu_i(t) + D_u d_{u_i}(t), \\ y_i(t) &= Cx_i(t) + D_y d_{y_i}(t), \end{aligned} \tag{3.1}$$

where $i = 1, 2, \dots, N$; $N$ is the total number of agents; $x_i(t) \in \mathbb{R}^n$ is the state vector, $u_i(t) \in \mathbb{R}^m$ is the input vector, $y_i(t) \in \mathbb{R}^p$ is the output vector, $d_{u_i}(t) \in \mathbb{R}^m$ is the external disturbance vector, and $d_{y_i}(t) \in \mathbb{R}^p$ is the sensor noise vector; $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D_u \in \mathbb{R}^{n\times m}$, and $D_y \in \mathbb{R}^{p\times n}$ are constant matrices of the system. The following assumptions are held in this chapter:

**Assumption 1.** *The pair $(A, B)$ is stabilizable.*

**Assumption 2.** *The pair $(A, C)$ is observable.*

**Assumption 3.** *The graph $\mathcal{G}$ is undirected.*

The main objective of this chapter is the design of a robust leader-following control for multi-agent systems consensus-based, where the virtual leader dynamics is presented as follows:

$$\dot{x}_l(t) = Ax_l(t), \tag{3.2}$$

where $x_l(t) \in \mathbb{R}^n$ is the state vector of the virtual leader. Let $\delta_i = x_i - x_l$, then, the dynamics of the synchronization error between each agent $i$ and the leader is:

$$\dot{\delta}_i(t) = A\delta_i(t) + Bu_i(t) + D_u d_{u_i}(t). \tag{3.3}$$

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

$$\begin{aligned} \dot{\hat{x}}_i(t) &= A\hat{x}_i(t) + Bu_i(t) + L(y_i(t) - C\hat{x}_i(t)), \\ \hat{y}_i(t) &= C\hat{x}_i(t), \end{aligned} \tag{3.4}$$

where $\hat{x}_i(t) \in \mathbb{R}^n$ is the estimated state vector, $L \in \mathbb{R}^{n\times p}$ is the observer gain to be designed, and $\hat{y}_i(t) \in \mathbb{R}^p$ is the estimated output vector. The dynamics of the estimation error $e_i = x_i - \hat{x}_i$ is computed as follows:

$$\dot{e}_i(t) = (A - LC)e_i(t) + D_u d_{u_i}(t) - LD_y d_{y_i}(t). \tag{3.5}$$

The proposed LMI formulation integrates an $\mathcal{H}_\infty$ performance criterion to minimize the $\mathcal{L}_2$-gain from external disturbances $d_{u_i}(t)$ and measurement noise $d_{y_i}(t)$ to both state estimation and trajectory tracking errors [120]. This approach simultaneously ensures robust attenuation of $d_{u_i}(t)$ through the controller and effective noise rejection $d_{y_i}(t)$ via the observer, thereby establishing comprehensive closed-loop robustness. The resulting design guarantees stability against bounded perturbations while preserving system performance. The considered observer-based leader-following consensus control law is [100]:

$$u_i(t) = K\left[\sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i(t) - \hat{x}_j(t)) + \alpha_i(\hat{x}_i(t) - x_l(t))\right], \tag{3.6}$$

where $K \in \mathbb{R}^{n\times p}$ is the control gain matrix to be designed, $a_{ij}$ denotes the elements of the adjacency matrix, $\hat{x}_i(t)$ represents the estimated state vector of the $i$-th agent, and $\hat{x}_j(t)$ corresponds to the estimated state vectors of its neighboring agents. The term $\alpha_i$ quantifies the leader-follower communication strength, where $\alpha_i > 0$ indicates a directed edge from the leader to the $i$-th agent. A larger value of $\alpha_i$, implies a stronger influence of the leader on agent $i$, while $\alpha_i = 0$ signifies no connection. The considered problem is the design of a robust control gain $K$ and a robust observer gain $L$ to steer the multi-agent system (3.1) subject to external disturbance and sensor noise, to follow the virtual leader's trajectories.

### 3.2.2 $\mathcal{H}_\infty$ Robust Control Design for Disturbances and Noise Rejection

In this section, the LMI conditions to guarantee the existence of the robust control and observer gains based on the Lyapunov stability analysis are presented. Considering the following new variables $e(t) = [e_1(t)^T, e_2(t)^T, \ldots, e_N(t)^T]^T$, the disturbance vector $d_u(t) = [d_{u_1}(t)^T, d_{u_2}(t)^T, \ldots, d_{u_N}(t)^T]^T$, and noise vector $d_y(t) = [d_{y_1}(t)^T, d_{y_2}(t)^T, \ldots, d_{y_N}(t)^T]^T$, then, using the Kronecker product $\otimes$ the dynamic estimation error can be expressed as:

$$\dot{e}(t) = (I_N \otimes (A - LC))e(t) + (I_N \otimes D_u)d_u(t) - (I_N \otimes LD_y)d_y(t). \tag{3.7}$$

Substituting $u_i(t)$ from (3.6) into (3.3) and defining $\delta(t) = \left[\delta_1(t)^T, \delta_2(t)^T, \ldots, \delta_N(t)^T\right]^T$, and $\overline{\mathcal{L}} = \mathcal{L} + \Lambda$ where $\mathcal{L}$ is the Laplacian matrix and $\Lambda = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$ represents the leader-follower communication weights, the synchronization error dynamics of the multi-agent system can be rewritten as:

$$\dot{\delta}(t) = (I_N \otimes A + \overline{\mathcal{L}} \otimes BK)\delta(t) - (\overline{\mathcal{L}} \otimes BK)e(t) + (I_N \otimes D_u)d_u(t). \tag{3.8}$$

Let $z(t) = [\delta(t)^T, e(t)^T]^T$. Then, the estimation error (3.7) and synchronization error (3.8) can be expressed as the following closed-loop multi-agent system (3.9):

$$\dot{z}(t) = \widetilde{A}z(t) + \widetilde{D}_u d_u(t) - \widetilde{D}_y d_y(t), \tag{3.9}$$

where

$$\widetilde{A}=\begin{bmatrix} I_N \otimes A + \overline{\mathcal{L}} \otimes BK & -\overline{\mathcal{L}} \otimes BK \\ 0 & I_N \otimes (A-LC) \end{bmatrix}, \widetilde{D}_u=\begin{bmatrix} I_N \otimes D_u \\ I_N \otimes D_u \end{bmatrix}, \widetilde{D}_y=\begin{bmatrix} 0 \\ I_N \otimes LD_y \end{bmatrix}. \tag{3.10}$$

The distributed $\mathcal{H}_\infty$ criterion is considered as in [109], in order to guarantee the existence of robust control and observer gains able to reject the effect of the external disturbances and sensor noise, then, the following theorem is proposed.

**Theorem 1.** *Consider the closed-loop system (3.9). Given the eigenvalues $\lambda_j(\bar{\mathcal{L}})$, $j = 1, 2, \ldots, N$; if there exist symmetric matrices $\bar{P}_1 > 0 \in \mathbb{R}^{n\times n}$ and $P_2 > 0 \in \mathbb{R}^{n\times n}$, the tuning scalar variable $\mu > 0$, and minimizing $\gamma$ by the $\mathcal{H}_\infty$ criterion satisfying the following condition:*

$$\begin{bmatrix} \eta_{1_j} & 0 & D_u & 0 & -\lambda_j BN_c & 0 \\ * & \eta_2 & P_2 D_u & -M_o D_y & 0 & I \\ * & * & -\gamma^2 I_N & 0 & 0 & 0 \\ * & * & * & -\gamma^2 I_N & 0 & 0 \\ * & * & * & * & -\mu^{-1}\overline{P}_1 & 0 \\ * & * & * & * & * & -\mu\overline{P}_1 \end{bmatrix} < 0, \tag{3.11}$$

*the synchronization error (3.8) and estimation error (3.7) are stable; where $\eta_{1_j} = He\left\{A\overline{P}_1 + \lambda_j BN_c\right\} + I$, and $\eta_2 = He\left\{P_2A - M_oC\right\} + I$, then, the robust control law can be computed with $K = N_c\overline{P}_1^{-1}$, and the observer gain with $L = P_2^{-1}M_o$.*

*Proof.* Consider the candidate Lyapunov function as follows:

$$V(t) = z(t)^T \begin{bmatrix} I_N \otimes P_1 & 0 \\ 0 & I_N \otimes P_2 \end{bmatrix} z(t), \tag{3.12}$$

where, $P_1 \in \mathbb{R}^{n\times n} = P_1^T > 0$, and $P_2 \in \mathbb{R}^{n\times n} = P_2^T > 0$. The derivative of this candidate Lyapunov function along the trajectories of (3.9) is given by:

$$\begin{aligned} \dot{V}(t) =& 2\delta(t)^T(I_N \otimes P_1A + \overline{\mathcal{L}} \otimes P_1BK)\delta(t) - 2\delta(t)^T(\overline{\mathcal{L}} \otimes P_1BK)e(t) \\ &+ 2e(t)^T(I_N \otimes P_2(A-LC))e(t) + 2\delta(t)^T(I_N \otimes P_1D_u)d_u(t) \\ &+ 2e(t)^T(I_N \otimes P_2D_u)d_u(t) - 2e(t)^T(I_N \otimes P_2LD_y)d_y(t). \end{aligned} \tag{3.13}$$

Let us perform a spectral decomposition of the matrix $\overline{\mathcal{L}}$, such that $\overline{\mathcal{L}} = TJT^{-1}$ with an invertible matrix $T \in \mathbb{R}^{N\times N}$ and a diagonal matrix $J = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N) \in \mathbb{R}^{N\times N}$. Defining the change of coordinates as $\xi = (T^{-1} \otimes I_N)\delta$, $\epsilon = (T^{-1} \otimes I_N)e$, $\omega_u = (T^{-1} \otimes I_N)d_u$, and $\omega_y = (T^{-1} \otimes I_N)d_y$. For reasons of space, the notation of the time dependence is omitted. Replacing the new coordinates in (3.13) leads to:

$$\begin{aligned} \dot{V} =& 2\xi^T(I_N \otimes P_1A + J \otimes P_1BK)\xi - 2\xi^T(J \otimes P_1BK)\epsilon + 2\epsilon^T(I_N \otimes P_2(A-LC))\epsilon \\ &+ 2\xi^T(I_N \otimes P_1D_u)\omega_u + 2\epsilon^T(I_N \otimes P_2D_u)\omega_u - 2\epsilon^T(I_N \otimes P_2LD_y)\omega_y. \end{aligned} \tag{3.14}$$

Since Assumption 3 specifies an undirected communication topology, (3.14) can be reformulated as follows:

$$\begin{aligned}\dot{V} =& \sum_{j=1}^{N} \xi_j^T He\left\{P_1 A + \lambda_j P_1 B K\right\} \xi_j - 2\sum_{j=1}^{N} \xi_j^T \lambda_j P_1 B K \epsilon_j + \sum_{j=1}^{N} \epsilon_j^T He\left\{P_2 A - P_2 L C\right\} \epsilon_j \\ &+ 2\sum_{j=1}^{N} \xi_j^T He\left\{P_1 D_u\right\} \omega_{u_j} + 2\sum_{j=1}^{N} \epsilon_j^T He\left\{P_2 D_u\right\} \omega_{u_j} + 2\sum_{j=1}^{N} \epsilon_j^T He\left\{P_2 L D_y\right\} \omega_{y_j},\end{aligned} \tag{3.15}$$

then, considering (3.15) with $\mathcal{H}_\infty$ criterion, $\varphi_j(t) = [\xi_j(t)^T, \epsilon_j(t)^T, \omega_{u_j}(t)^T,\ \omega_{y_j}^T]^T$, and $J_T \leq \sum_{j=1}^{N} \varphi(t)_j^T \Omega_j \varphi(t)_j$, where $\Omega_j$ is defined as follows:

$$\Omega_j = \begin{bmatrix} Q_{1_j} + I & -\lambda_j P_1 B K & P_1 D_u & 0 \\ * & Q_2 + I & P_2 D_u & -P_2 L D_y \\ * & * & -\gamma^2 I & 0 \\ * & * & * & -\gamma^2 I \end{bmatrix} < 0, \tag{3.16}$$

where $Q_{1_j} = He\left\{P_1 A + \lambda_j P_1 B K\right\}$, and $Q_2 = He\left\{P_2 A - P_2 L C\right\}$. Then, (3.16) is pre and post multiplied by $\overline{P}_1$, where $\overline{P}_1 = \overline{P}_1^T = P_1^{-1} > 0$

$$\begin{bmatrix} \Upsilon_{1_j} & 0 & D_u & 0 \\ * & \Upsilon_2 & P_2 D_u & -P_2 L D_y \\ * & * & -\gamma^2 I & 0 \\ * & * & * & -\gamma^2 I \end{bmatrix} + He\left\{ \begin{bmatrix} -\lambda_j B K \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & I & 0 & 0 \end{bmatrix} \right\} < 0, \tag{3.17}$$

where $\Upsilon_{1_j} = He\left\{A\overline{P}_1 + \lambda_j B K \overline{P}_1\right\} + \overline{P}_1 \overline{P}_1$, and $\Upsilon_2 = He\left\{P_2 A - \ P_2 L C\right\} + I$. Applying the Young relation [113], the following inequality is obtained:

$$\begin{aligned}&\begin{bmatrix} \Upsilon_{1_j} & 0 & D_u & 0 \\ * & \Upsilon_2 & P_2 D_u & -P_2 L D_y \\ * & * & -\gamma^2 I & 0 \\ * & * & * & -\gamma^2 I \end{bmatrix} + \mu \begin{bmatrix} -\lambda_j B K \\ 0 \\ 0 \\ 0 \end{bmatrix} \overline{P}_1^{-1} \begin{bmatrix} (-\lambda_j B K)^T & 0 & 0 & 0 \end{bmatrix} \\ &+ \mu^{-1} \begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}^T \overline{P}_1^{-1} \begin{bmatrix} 0 & I & 0 & 0 \end{bmatrix} < 0,\end{aligned} \tag{3.18}$$

where $\mu > 0$. Using the Schur complement in (3.18), choosing $N_c = K\overline{P}_1$, and $M_o = P_2 L$, the proof is complete. □

## 3.3 Simulation Results: Application to a Fleet of UAVs

This section presents the simulation results for our leader-following robust consensus control strategy applied to a team of homogeneous UAVs. To demonstrate the proposed method, we consider a multi-agent system composed of quadcopter UAVs. First, we introduce the nonlinear dynamic model of an individual quadcopter, based on the framework described in [121]. Figure 3.2 shows the scheme of the quadcopter representing the main forces acting on the vehicle.

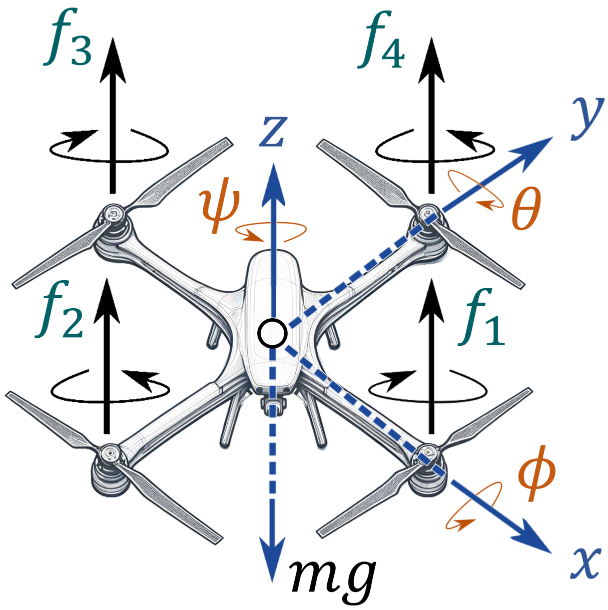


Figure 3.2: Diagram representing the primary forces acting on the quadcopter vehicle.

To formulate the dynamic model of the UAVs, the following assumptions are stated [122]:

**Assumption 4.** *The quadcopter structure is rigid and symmetrical,*

**Assumption 5.** *The center of mass and the origin of the body frame $O_B$ coincide.*

The quadcopter is composed of four independent motors with four propellers that produce torques and thrusts in the direction of the propellers axis of rotation. Propellers 1 and 3 turn in the clockwise direction, while propellers 2 and 4 turn in the counterclockwise direction. From Figure 3.2, $\mathcal{I} = \{O_I, p_{x_I}, p_{y_I}, p_{z_I}\}$ denotes an inertial frame, and $\mathcal{B} = \{O_B, p_{x_B}, p_{y_B}, p_{z_B}\}$ denotes a rigid frame attached to the center of mass of the vehicle.

The dynamics of the quadcopter can be defined as follows [123]:

$$\begin{aligned} \dot{\xi} &= v, & m_s\dot{v} &= RF, \\ \dot{R} &= R\hat{\Omega}, & J\dot{\Omega} &= -\Omega \times J\Omega + \tau, \end{aligned} \tag{3.19}$$

where $\xi = [p_x, p_y, p_z]^T \in \mathbb{R}^3$ and $v = [v_x, v_y, v_z]^T \in \mathbb{R}^3$ denotes the position and velocity of the vehicle with respect to the frame $\mathcal{I}$, respectively. The angular velocity is defined by $\Omega = [p, q, r]^T \in \mathbb{R}^3$ in the body-fixed frame $\mathcal{B}$, and $m$ is the total mass of the vehicle. The Euler angles are expressed by roll $\phi$, pitch $\theta$, and yaw $\psi$. The moment of inertia is denoted by $J = \text{diag}(J_x, J_y, J_z) \in \mathbb{R}^{3\times3}$ defined in $\mathcal{B}$, $\tau$ expresses the moments in $\mathcal{B}$, $\hat{\Omega}$ introduces the skew-symmetric matrix of the vector $\Omega$, $R$ means the rotation matrix from $\mathcal{B}$ to $\mathcal{I}$, and $F$ are the forces produced by the motors. By considering Assumption

4, Assumption 5, and using the Newton–Euler formalism, the nonlinear model of an UAV can be expressed as follows [121]:

$$
\begin{aligned}
\dot{p}_x &= v_x,\\
\dot{p}_y &= v_y,\\
\dot{p}_z &= v_z,\\
\dot{v}_x &= \big(\cos(\psi)\sin(\theta)\cos(\phi)+\sin(\psi)\sin(\phi)\big)\frac{1}{m}u_z-\frac{d_f}{m}v_x,\\
\dot{v}_y &= \big(\sin(\psi)\sin(\theta)\cos(\phi)-\cos(\psi)\sin(\phi)\big)\frac{1}{m}u_z-\frac{d_f}{m}v_y,\\
\dot{v}_z &= -g+\big(\cos(\theta)\cos(\phi)\big)\frac{1}{m}u_z-\frac{d_f}{m}v_z,\\
\dot{\phi} &= p+\big(\sin(\phi)\tan(\theta)\big)q+\cos(\phi)\tan(\theta)\big)r,\\
\dot{\theta} &= \cos(\phi)q-\sin(\phi)r,\\
\dot{\psi} &= \big(\sin(\phi)q+\cos(\phi)r\big)\frac{1}{\cos(\theta)},\\
\dot{p} &= \big(qr(J_y-J_z)+u_\phi\big)\frac{1}{J_x}-\frac{d_t}{J_x}p,\\
\dot{q} &= \big(pr(J_z-J_x)+u_\theta\big)\frac{1}{J_y}-\frac{d_t}{J_y}q,\\
\dot{r} &= \big(pq(J_x-J_y)+u_\psi\big)\frac{1}{J_z}-\frac{d_t}{J_z}r.
\end{aligned}
\tag{3.20}
$$

The translational and attitude dynamics of the system are influenced by inputs denoted as $u_z$ the thrust, $u_\phi$, $u_\theta$, $u_\psi$ the respective torques of roll, pitch and yaw. The inputs $u=[u_z,u_\phi,u_\theta,u_\psi]^T$ allow to generate the real forces applied by each propeller $f_\varepsilon$, defined as $f_1$, $f_2$, $f_3$, and $f_4$. (see Figure 3.2). The main objective of the multi-agent system, where each UAV is associated to one agent, is to follow the trajectories of a virtual leader and maintain a desired shape, where $H=[h_1^T,h_2^T,\dots,h_N^T]^T$ contains the desired distance column vectors $h_i$ of every agent.

The double integrator model [124] is widely employed in UAV control due to its simplicity and effectiveness in capturing the fundamental dynamics of motion. In three-dimensional space, this model describes the system dynamics as

$$\ddot{\mathbf{p}}=\mathbf{a}, \tag{3.21}$$

where $\mathbf{p}=[x,y,z]^T$ represents the position vector and $\mathbf{a}=[a_x,a_y,a_z]^T$ is the acceleration input acting as the control variable. This linear approximation assumes that the UAV's dynamics are dominated by inertial effects, neglecting higher-order nonlinearities such as aerodynamic drag and cross-coupling under moderate operating conditions. The decoupled structure of the double integrator permits straightforward controller synthesis using linear techniques while providing a tractable foundation for trajectory planning and stability analysis. The double integrator model is used to find a solution for the LMIs presented in Theorem 1. However, it is noted that for all simulations, the complete nonlinear

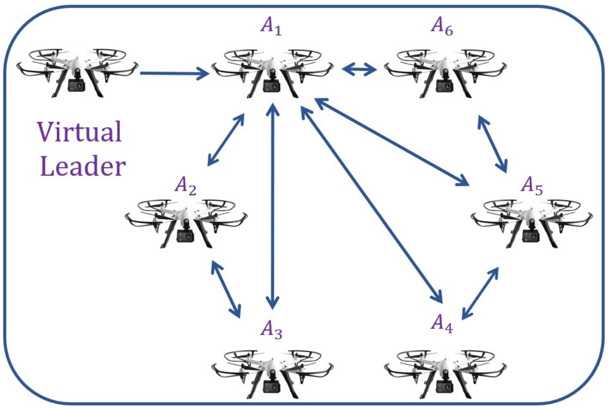


Figure 3.3: Directions of the communication links between agents.

UAV (3.20) is used. The LMI-based design, derived from the double-integrator model, ensures local stability and robustness; its performance is then validated on the nonlinear system.

Considering $p_i(t)$, $v_i(t)$, $u_i(t)$ the position, velocity and acceleration respectively of agent $i$, $\hat{\bar{s}}_i(t) = [\hat{p}_i(t)^T - h_i^T, \hat{v}_i(t)^T]^T$ the estimated states of agent $i$, $\bar{s}_l(t) = [p_l(t)^T, v_l(t)^T]^T$ the virtual leader states, $h_i$ is a vector of matrix $H$ of the respective agent, then, the control law (3.6) is rewritten as:

$$u_i(t) = K \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{\bar{s}}_i(t) - \hat{\bar{s}}_j(t)) + \alpha_i K(\hat{\bar{s}}_i(t) - \bar{s}_l(t)), \tag{3.22}$$

where $K$ is the control gain to be designed. Considering $\hat{\bar{\delta}}_i(t) = \hat{\bar{s}}_i(t) - \bar{s}_l(t)$ equation (3.22) is rewritten as follows:

$$u_i(t) = K \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{\bar{\delta}}_i(t) - \hat{\bar{\delta}}_j(t)) + \alpha_i K \hat{\bar{\delta}}_i(t). \tag{3.23}$$

Since the dynamic of each UAV is considered as a particle, the desired Euler angles $\theta_{di}$, $\phi_{di}$, and $\psi_{di}$ are computed by (3.24) as in [125], to control the position of the UAVs, with the consensus protocol (3.23).

$$\phi_{di} = \arctan\left(\frac{-u_{2i}}{\sqrt{u_{1i}^2 + (u_{3i}+g)^2}}\right), \quad \theta_{di} = \arctan\left(\frac{u_{1i}}{u_{3i}+g}\right), \quad \psi_{di} = 0. \tag{3.24}$$

The thrust of the UAVs rotors is computed by $T_i = m_i(u_{1i}^2 + u_{2i}^2 + (u_{3i}+g)^2)^{1/2}$. The communication topology of the multi-agent system is depicted in Figure 3.3 (where a UAV represents an agent and the arrows the flow of information), and the Laplacian matrix $\mathcal{L}$:

$$\mathcal{L} = \begin{bmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 & 0 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}. \tag{3.25}$$

The matrix $H$ represents the desired formation. Its values were selected to position the UAVs at the vertices of a hexagon, where all agents must maintain the same altitude while preserving their relative x,y-plane distances to achieve the hexagonal shape.

$$H = \begin{bmatrix} 0 & 4 & 6 & 4 & 0 & -2 \\ 0 & 0 & 2\sqrt{3} & 4\sqrt{3} & 4\sqrt{3} & 2\sqrt{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.26}$$

The matrix $\Lambda$ encodes the communication topology between the leader UAV and the follower UAVs. It is designed such that the leader communicates only with UAV 1. When the positions of the leader and UAV 1 converge, the leader-following objective is achieved.

$$\Lambda = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.27}$$

The first simulation result in Figure **??**, shows that the formation control problem is solved and the multi-agent system follows the trajectories of the virtual leader. All the UAVs reach the trajectories of the virtual leader while maintaining the rigid formation of a hexagon defined by $H$ matrix. This first result does not include external disturbance or measurement sensor noise. The nonlinear dynamic model of the virtual leader and followers is considered as (3.20), and the parameters of the UAVs are described in Table 3.1.

Table 3.1: Parameters of the VTOL-UAV.

| Parameter | Value | Unit |
|---|---|---|
| Mass of the vehicle, $m$ | 0.33 | Kg |
| Acceleration due to gravity, $g$ | 9.81 | m/s$^2$ |
| Moment of inertia about $x$, $J_x$ | $3.1 \times 10^{-5}$ | Kgm$^2$ |
| Moment of inertia about $y$, $J_y$ | $2.98 \times 10^{-5}$ | Kgm$^2$ |
| Moment of inertia about $z$, $J_z$ | $6.1 \times 10^{-5}$ | Kgm$^2$ |

The initial conditions of the quadcopters and their respective observers are presented in Table 3.2.

Table 3.2: Initial conditions.

| Quadcopter | $p_{x_i}$ | $p_{y_i}$ | $p_{z_i}$ | $\hat{p}_{x_i}$ | $\hat{p}_{y_i}$ | $\hat{p}_{z_i}$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0 | 0 | 0 | 0 |
| 2 | 2 | 1 | 0 | 0 | 0 | 0 |
| 3 | 3 | 4 | 0 | 0 | 0 | 0 |
| 4 | 4 | 3 | 0 | 0 | 0 | 0 |
| 5 | 2 | 5 | 0 | 0 | 0 | 0 |
| 6 | 3 | 1 | 0 | 0 | 0 | 0 |

The initial conditions of the linear velocities, the angles, and the angular velocities are zero for all UAVs and observers. All the initial states of the virtual leader are zero. Figure 3.4 shows that the control objectives are achieved, the UAVs are able to maintain their desired positions at the vertices of the hexagon. Likewise, the agents are in the desired position with respect to the virtual leader.

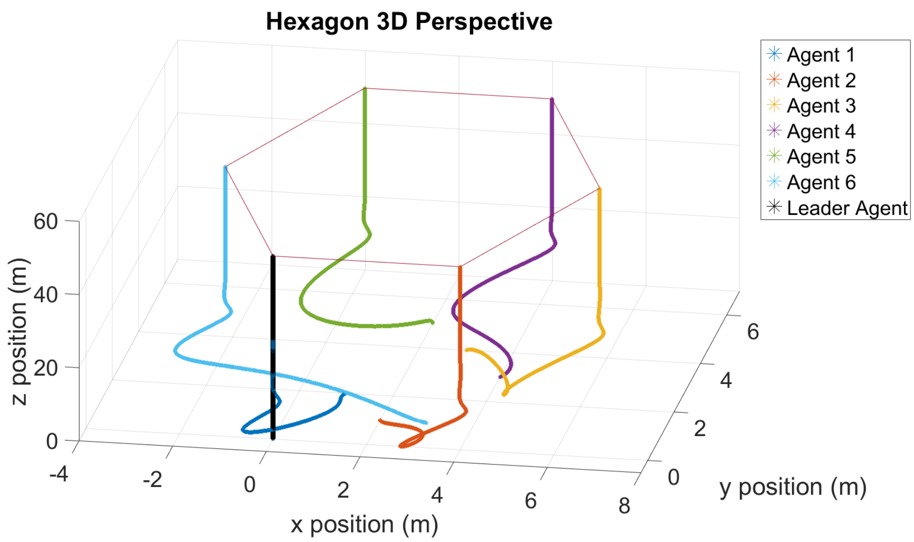


Figure 3.4: Formation without disturbance or sensor noise and distances between followers.

When the distances $D_{ij} = \|([x_i, y_i, z_i]^T - [x_j, y_j, z_j]^T) - (h_i - h_j)\|$ are $D_{ij} = 0$, it is because the formation control objective is fulfilled, meaning the agents are in their desired positions in space. In Figure 3.5 the change in distances throughout the simulation is shown.

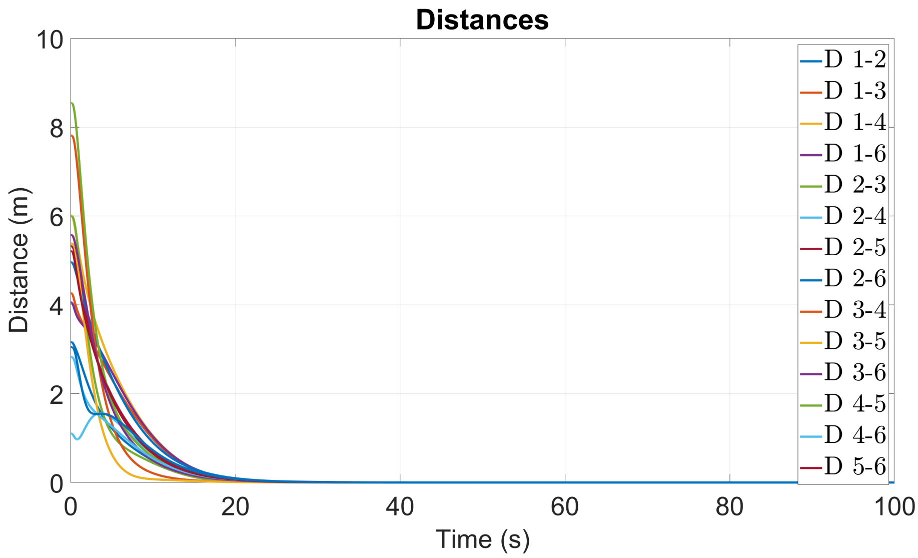


Figure 3.5: Formation achieved: distances between UAVs drop to zero as settle into desired position.

The considered external disturbance is modeled as the Dryden wind turbulence presented in [126], and the sensor noise was added as a random function with normal distribution, mean value of zero, and standard deviation of 0.8 m/s, in order to verify the robustness of the control law (3.23). If the measurement noise or the external disturbance magnitudes increase, the control law (3.23) is not able to maintain the desired formation or to follow the virtual leader. To overcome the disturbance and noise problem, the LMI presented in (3.11) is computed with the Sedumi Toolbox [127], where the LMI variable $\gamma$ value obtained with $\mu = 0.1$ is $\gamma = 1$. The computed control and observer gains are:

$$K = \begin{bmatrix} -0.756 & 0 & 0 & -2.651 & 0 & 0 \\ 0 & -0.756 & 0 & 0 & -2.651 & 0 \\ 0 & 0 & -0.756 & 0 & 0 & -2.651 \end{bmatrix}, \tag{3.28}$$

$$L = \begin{bmatrix} 3.208 & 0 & 0 \\ 0 & 3.208 & 0 \\ 0 & 0 & 3.208 \\ 0.836 & 0 & 0 \\ 0 & 0.836 & 0 \\ 0 & 0 & 0.836 \end{bmatrix}. \tag{3.29}$$

The robustness of the distributed control and estimation implemented in the team of UAVs has been verified by solving the leader-follower formation control problem. The effectiveness of the proposed work is shown in Figure 3.6, where each UAV achieves its desired position in the rigid formation of a hexagon while following the trajectory of the virtual leader because the external disturbance and the measurement sensor noise are rejected.

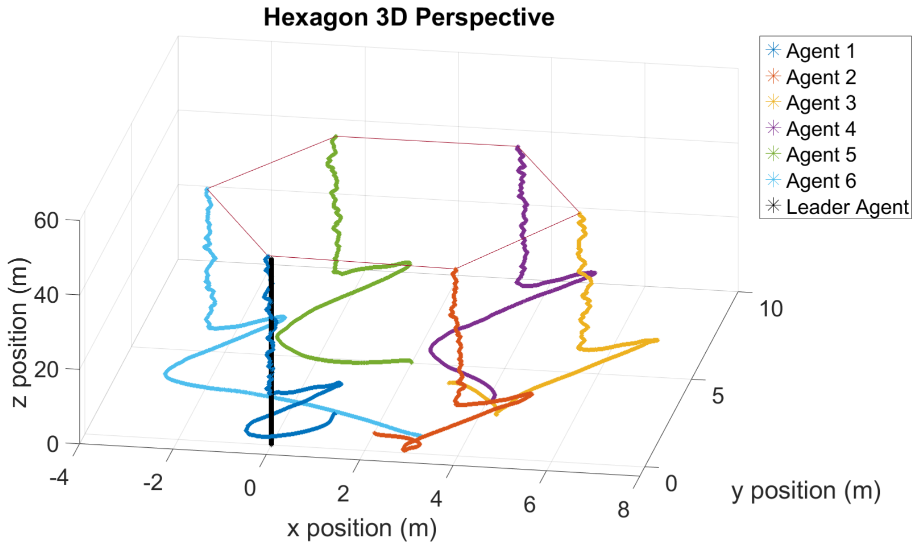


Figure 3.6: Formation with disturbance and sensor noise using the control law (3.23) and observer (3.4) with gains $K$, and $L$ (3.29) computed using Theorem 1.

Figure 3.6 demonstrates the effectiveness of our approach, as the UAVs successfully maintain their desired positions despite the introduction of external disturbances. It is important to highlight that the simulation in Figure 3.5 is neither robust to disturbances nor to sensor noise. While the results in Figure 3.6 not fully recover the ideal dynamics of Figure 3.5, the multi-agent system maintains the control objectives: the followers successfully track the virtual leader while remaining in close proximity to their designated vertices of the hexagonal formation. This disturbance rejection capability is achieved through the $\mathcal{H}_\infty$ control criterion.

3.7 3.5

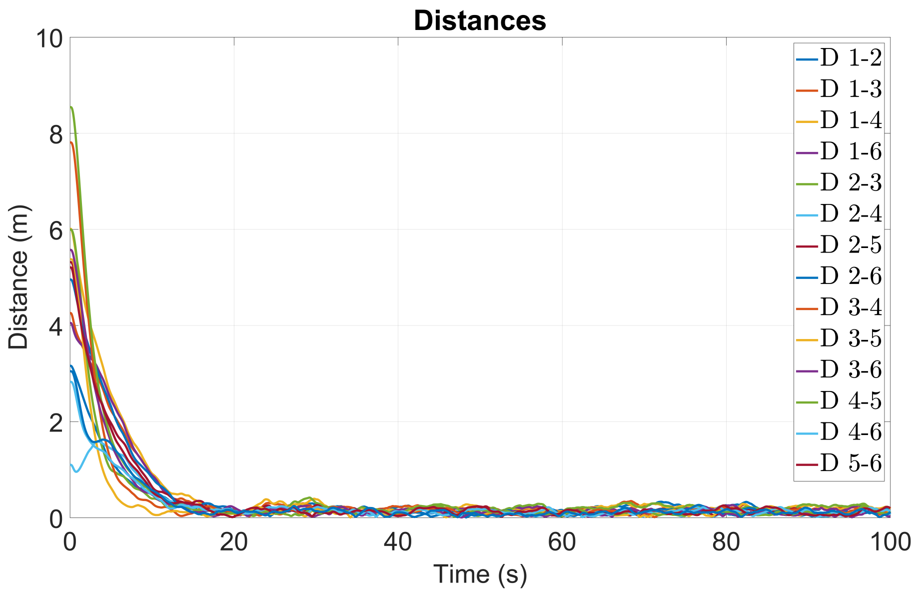


Figure 3.7: Formation achieved: distances between UAVs drop close to zero despite disturbance and sensor noise.

The distances shown in Figure 3.7 are $D_{ij} \approx 0$, which shows that the control technique rejects the effects of external inputs.

## 3.4 Conclusions

A distributed robust leader-following consensus control observer-based for LTI multi-agent systems was presented. The objective of the multi-agent system is to follow the trajectories described by a virtual leader's dynamics and to maintain a desired rigid formation between followers. This was exemplified with the simulation of a team of UAVs where the agents were able to reject external disturbance and measurement sensor noise.

While LTI models provide a simplified framework for analysis and design, many real-world systems exhibit parameter-dependent dynamics that cannot be adequately captured by time-invariant representations. For this reason, the next chapter extends this work to LPV systems, which explicitly account for time-varying parameters—offering greater flexibility and accuracy in modeling complex. This generalization will enhance the applicability to scenarios where system dynamics change with respect to measurable scheduling variables.

# Chapter 4

# Observer-Based Leader-Following Consensus Control for LPV Multi-Agent Systems

## Contents



## 4.1 Introduction

This chapter extends the distributed leader-following consensus control strategy from LTI multi-agent systems to LPV multi-agent systems, where system dynamics depend on time-varying parameters. LPV systems improve upon LTI models by incorporating time-varying parameters enabling more accurate representation of real-world dynamic conditions while maintaining the computational advantages of linear control methodologies. While Chapter 3 employed LMIs with an $\mathcal{H}_\infty$ criterion for robustness, this work adapts the framework to LPV systems, where the main constraint is to gurantee the feasibility of the LMIs considering the permutations and combinations of the LPV vertex systems, omitting $\mathcal{H}_\infty$ constraints as a first step in LPV multi-agent systems control. The proposed observer-based controller, designed using fuzzy control theory, ensures consensus across the virtual leader and the followers. Numerical simulations with UAVs validate the approach, demonstrating its effectiveness under time-varying conditions. In Figure 4.1, the control scheme for the distributed leader-following LPV consensus control is presented.

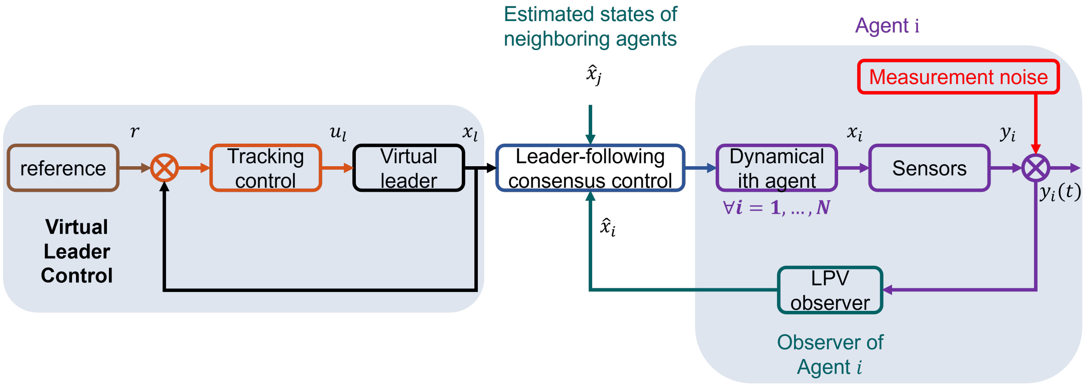


Figure 4.1: Control diagram for LPV multi-agent systems.

## 4.2 LPV Observer-Based Leader-Following Consensus Control: Application to a Team of UAVs

### 4.2.1 Problem Formulation

Let us consider the following LPV homogeneous multi-agent system

$$\begin{aligned} \dot{x}_i(t) &= A(\zeta_i(t))x_i(t) + B(\zeta_i(t))u_i(t), \\ y_i(t) &= C(\zeta_i(t))x_i(t), \end{aligned} \tag{4.1}$$

composed of $N$ identical systems. For each one, $i$ is the agent number ($i = 1, \ldots, N$); $x_i(t) \in \mathbb{R}^n$, $u_i(t) \in \mathbb{R}^r$, $y_i(t) \in \mathbb{R}^p$, are the state, input, and output, vectors respectively; $A(\zeta_i(t)) \in \mathbb{R}^{n\times n}$, $B(\zeta_i(t)) \in \mathbb{R}^{n\times r}$, and $C(\zeta_i(t)) \in \mathbb{R}^{p\times n}$ are the matrix functions scheduled by the vector $\zeta_i(t) \in \mathbb{R}^{n_\zeta}$, which is considered as known and available for every $i$-th agent, and is assumed to vary in a closed set $\mathcal{R}$. The *bounding box* method [128] is used in order to express $A(\zeta_i(t)), B(\zeta_i(t))$, and $C(\zeta_i(t))$ as a convex combination of $S$ vertex matrices, such that

$$\begin{aligned} &\begin{pmatrix} A(\zeta_i(t)) \\ B(\zeta_i(t)) \\ C(\zeta_i(t)) \end{pmatrix} = \sum_{h=1}^{S} \rho_h(\zeta_i(t)) \begin{pmatrix} A_h \\ B_h \\ C_h \end{pmatrix}, \\ &\sum_{h=1}^{S} \rho_h(\zeta_i(t)) = 1, \quad \rho_h(\zeta_i(t)) \geq 0 \quad \forall \zeta_i(t) \in \mathcal{R} \end{aligned} \tag{4.2}$$

where each agent is considered as an autonomous dynamical system able to share the information of their states with the neighboring agents.

The following assumptions are held in this work ($\forall h = 1, \ldots, S$):

**Assumption 6.** *The pair $(A_h, B_h)$ is stabilizable.*

**Assumption 7.** *The pair $(A_h, C_h)$ is observable.*

**Assumption 8.** *The graph $\mathcal{G}$ is undirected.*

The main objective of this chapter is the design of a leader-following control for LPV multi-agent systems consensus-based. The LPV virtual leader dynamic is defined by:

$$\dot{x}_l(t) = A(\zeta_l(t))x_l(t) + B(\zeta_l(t))u_l(t), \tag{4.3}$$

where $x_l(t) \in \mathbb{R}^n$, $u_l(t) \in \mathbb{R}^r$ are the state and input vectors of the virtual leader. Note that, unlike the previously presented leader (3.2), the newly proposed leader (4.3) includes an input signal. Nevertheless, this does not affect the system formulation since the leader serves solely as a reference for the follower agents. Consensus between the leader and the followers can be achieved only if $\zeta_1(t) = \zeta_2(t) = \ldots = \zeta_N(t) = \zeta_l(t) = \zeta(t)$ because homogeneous multi-agent systems are considered. Let $\delta_i(t) = x_i(t) - x_l(t)$, then, the dynamics of the synchronization error between each agent $i$ and the leader is:

$$\dot{\delta}_i(t) = A(\zeta(t))\delta_i(t) + B(\zeta(t))\big(u_i(t) - u_l(t)\big). \tag{4.4}$$

Let us propose the following observer-based consensus protocol:

$$u_i(t) = K(\zeta(t))\left[\sum_{j\in\mathcal{N}_i} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) + \alpha_i\big(\hat{x}_i(t) - x_l(t)\big)\right] + u_l(t), \tag{4.5}$$

where $K(\zeta(t)) \in \mathbb{R}^{r\times n}$ is the LPV control gain to be designed, $\hat{x}_j(t) \in \mathbb{R}^n$ are the estimated states of the neighboring agents, and $a_{ij}$ represent the entries of the adjacency matrix. Due to the estimated states of agent $i$ and agent $j$ are considered in the consensus protocol (4.29), the estimated state vector $\hat{x}_i(t) \in \mathbb{R}^n$ is given by the following distributed LPV observer:

$$\begin{aligned} \dot{\hat{x}}_i(t) &= A(\zeta(t))\hat{x}_i(t) + B(\zeta(t))u_i(t) + L(\zeta(t))\Big(y_i(t) - \hat{y}_i(t)\Big), \\ \hat{y}_i(t) &= C(\zeta(t))\hat{x}_i(t), \end{aligned} \tag{4.6}$$

where $L(\zeta(t)) \in \mathbb{R}^{n\times p}$ is the LPV observer gain to be designed and $\hat{y}_i(t) \in \mathbb{R}^p$ is the estimated output vector. The polytopic representation of the control gain $K(\zeta(t))$ and the observer gain $L(\zeta(t))$ are considered as follows:

$$\begin{pmatrix} K(\zeta(t)) \\ L(\zeta(t)) \end{pmatrix} = \sum_{h=1}^{S} \rho_h(\zeta(t)) \begin{pmatrix} K_h \\ L_h \end{pmatrix}. \tag{4.7}$$

Defining the estimation error $e_i(t) = x_i(t) - \hat{x}_i(t)$, then, the dynamics of the estimation error are:

$$\dot{e}_i(t) = [A(\zeta(t)) - L(\zeta(t))C(\zeta(t))]e_i(t). \tag{4.8}$$

The key challenge is the simultaneous design of the vertex control and observer gains, $K_h$ and $L_h$, respectively, through the formulation and solution of a structured LMI problem.

### 4.2.2 LPV Observer-Based Leader-Following Consensus Controller Design

The LMI conditions to guarantee the existence of the control (4.29) and observer (4.6) gains are presented in this section using the Lyapunov stability analysis. Let us consider the time-varying parameters vector $\zeta(t) = [\zeta_1(t)^T, \zeta_2(t)^T, \ldots, \zeta_N(t)^T]^T$, defining the matrix:

$$\rho_{ij}(\zeta(t)) = \mathrm{diag}(\rho_{i\star j}(\zeta_1(t)), \ldots, \rho_{i\star j}(\zeta_N(t))) \tag{4.9}$$

with $\rho_{i\star j}(\zeta_g(t)) \triangleq \rho_i(\zeta_g(t))\rho_j(\zeta_g(t))$, $g = 1, \ldots, N$. Considering $e(t) = [e_1(t)^T, e_2(t)^T, \ldots, e_N(t)^T]^T$, and $\delta(t) = [\delta_1(t)^T, \delta_2(t)^T, \ldots, \delta_N(t)^T]^T$, the estimation error and the synchronization error are expressed using the Kronecker product as follows:

$$\dot{e}(t) = \sum_{h=1}^{S}\sum_{g=1}^{S} \left[\rho_{hg}(\zeta(t)) \otimes (A_h - L_h C_g)\right] e(t), \tag{4.10}$$

$$\begin{aligned} \dot{\delta}(t) = \sum_{h=1}^{S}\sum_{g=1}^{S} \rho_{hg}(\zeta(t)) \left[I_N \otimes A_h + \bar{\mathcal{L}} \otimes B_g K_h\right] \delta(t) - \\ \left[\rho_{hg}(\zeta(t))\bar{\mathcal{L}} \otimes B_g K_h\right] e(t) \end{aligned} \tag{4.11}$$

where $\bar{\mathcal{L}} = \mathcal{L} + \Lambda$ is the sum of the Laplacian matrix $\mathcal{L}$ and $\Lambda = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$, which define the communication exchange between the leader and each follower. Let $z(t) = [\delta(t)^T, e(t)^T]^T$, then the closed-loop multi-agent system is expressed as follows (for easy notation, the time dependency is omitted):

$$\dot{z} = \sum_{h=1}^{S}\sum_{g=1}^{S} \rho_h(\zeta)\rho_g(\zeta) \times \begin{bmatrix} I_N \otimes A_h + \overline{\mathcal{L}} \otimes B_g K_h & -\overline{\mathcal{L}} \otimes B_g K_h \\ 0 & I_N \otimes (A_h - L_h C_g) \end{bmatrix} z. \tag{4.12}$$

Let us choose a quadratic candidate Lyapunov function as follows:

$$V(z) = \begin{bmatrix} \delta^T & e^T \end{bmatrix} \begin{bmatrix} I_N \otimes P_1 & 0 \\ 0 & I_N \otimes P_2 \end{bmatrix} \begin{bmatrix} \delta \\ e \end{bmatrix}. \tag{4.13}$$

The following theorem provides LMI-based conditions that guarantee the existence of the LPV control and LPV observer gains based on the Lyapunov stability analysis.

**Theorem 2.** *Consider the closed-loop LPV multi-agent system (4.12). The estimation (4.10) and synchronization (4.11) errors are proven to be exponentially stable by (4.13) for any $s \in \mathbb{N}$, with $s \geq 2$, given the eigenvalues $\lambda_j(\bar{\mathcal{L}})$, $\forall j = 1, 2, \ldots, N$; if there exist symmetric matrices $\bar{P}_1 > 0 \in \mathbb{R}^{n\times n}$ and $P_2 > 0 \in \mathbb{R}^{n\times n}$, matrices $N_1, N_2, \ldots, N_S$, and $M_1, M_2, \ldots, M_S$; the tuning scalar variable $\mu > 0$, such that*

$$\sum_{\vec{g}\in\mathcal{D}(\vec{d})} \begin{bmatrix} \mathcal{Q}_{j11} & 0 & \mathcal{Q}_{j13} & 0 \\ * & \mathcal{Q}_{22} & 0 & I_N \\ * & * & -\mu^{-1}\bar{P}_1 & 0 \\ * & * & * & -\mu\bar{P}_1 \end{bmatrix} < 0 \tag{4.14}$$

*holds* $\forall \vec{d} \in \mathbb{D}_s^+$, *where* $\mathcal{Q}_{j_{11}} = He\{A_{\vec{\varrho}1}\bar{P}_1 + \lambda_j B_{\vec{\varrho}2} N_{\vec{\varrho}1}\}, \mathcal{Q}_{22} = He\{P_2 A_{\vec{\varrho}1} - M_{\vec{\varrho}1} C_{\vec{\varrho}2}\}$, *and* $\mathcal{Q}_{j_{13}} = -\lambda_j B_{\vec{\varrho}2} N_{\vec{\varrho}1}$, *then, the control vertex gain can be computed with* $K_h = N_h \bar{P}_1^{-1}$ *and the observer vertex gain with* $L_h = P_2^{-1} M_h$.

*Proof.* The derivative of (4.13) along the trajectories of (4.12) is given by:

$$\begin{aligned}\dot{V}(z) = &\sum_{h=1}^{S}\sum_{g=1}^{S}\rho_h(\zeta)\rho_g(\zeta)\Big\{2\delta^T(I_N \otimes P_1 A_h + \\ &\bar{\mathcal{L}} \otimes P_1 B_g K_h)\delta - 2\delta^T(\bar{\mathcal{L}} \otimes P_1 B_g K_h)e \\ &+ 2e^T(I_N \otimes P_2(A_h - L_h C_g))e\Big\}.\end{aligned} \tag{4.15}$$

Let us perform a spectral decomposition of the matrix $\bar{\mathcal{L}}$, such that $\bar{\mathcal{L}} = TJT^{-1}$ with an invertible matrix $T \in \mathbb{R}^{N\times N}$ and a diagonal matrix $J = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N) \in \mathbb{R}^{N\times N}$ of which $\lambda_j(\bar{\mathcal{L}})$, $j = 1, 2, \ldots, N$ are the eigenvalues of $\bar{\mathcal{L}}$. Defining the change of coordinates as $\varphi = (T^{-1} \otimes I_N)\delta$ and $\phi = (T^{-1} \otimes I_N)e$, (4.15) can be rewritten as:

$$\begin{aligned}\dot{V}(z) = &\sum_{h=1}^{S}\sum_{g=1}^{S}\rho_h(\zeta)\rho_g(\zeta)\Big\{\sum_{j=1}^{N}\varphi^T \mathrm{He}\{P_1 A_h \\ &+ \lambda_j P_1 B_g K_h\}\varphi_j - 2\sum_{j=1}^{N}\varphi^T \lambda_j P_1 B_g K_h \phi_j \\ &+ \sum_{j=1}^{N}\phi^T \mathrm{He}\{P_2 A_h - P_2 L_h C_g\}\phi_j\Big\}.\end{aligned} \tag{4.16}$$

Considering the vector $[\varphi_j^T, \phi_j^T]^T$, (4.16) can be rewritten as follows:

$$\dot{V}(z) = \sum_{h=1}^{S}\sum_{g=1}^{S}\rho_h(\zeta)\rho_g(\zeta)\left\{\begin{bmatrix}\varphi_j^T & \phi_j^T\end{bmatrix}\mathfrak{B}_{j_{hg}}\begin{bmatrix}\varphi_j \\ \phi_j\end{bmatrix}\right\}, \tag{4.17}$$

$$\mathfrak{B}_{j_{hg}} = \begin{bmatrix}\mathrm{He}\{P_1 A_h + \lambda_j P_1 B_g K_h\} & -\lambda_j P_1 B_g K_h \\ * & \mathrm{He}\{P_2 A_h - P_2 L_h C_g\}\end{bmatrix} \tag{4.18}$$

which corresponds to the problem of verifying the negativity of double polytopic sums. We define $\bar{P}_1 = P_1^{-1}$

$$\beta_{j_{hg}} = \begin{bmatrix}\bar{P}_1 & 0 \\ 0 & I_N\end{bmatrix}\mathfrak{B}_{j_{hg}}\begin{bmatrix}\bar{P}_1 & 0 \\ 0 & I_N\end{bmatrix} < 0 \tag{4.19}$$

$$\beta_{j_{hg}} = \begin{bmatrix}\mathrm{He}\{A_h\bar{P}_1 + \lambda_j B_g K_h \bar{P}_1\} & -\lambda_j B_g K_h \\ * & \mathrm{He}\{P_2 A_h - P_2 L_h C_g\}\end{bmatrix} < 0, \tag{4.20}$$

then, (4.20) can be rewritten as follows:

$$\begin{bmatrix}\beta_{j_{hg11}} & 0 \\ * & \beta_{j_{hg22}}\end{bmatrix} + \mathrm{He}\left\{\begin{bmatrix}\beta_{j_{hg12}} \\ 0\end{bmatrix}\begin{bmatrix}0 & I_N\end{bmatrix}\right\} < 0 \tag{4.21}$$

where $\beta_{j_{hg_{11}}} = \mathrm{He}\{A_h\bar{P}_1 + \lambda_j B_g K_h \bar{P}_1\}$, $\beta_{j_{hg_{22}}} = \mathrm{He}\{P_2 A_h - P_2 L_h C_g\}$, and $\beta_{j_{hg_{12}}} = -\lambda_j B_g K_h$. Applying the Young relation [113], the following inequality is obtained:

$$\begin{aligned} &\begin{bmatrix} \beta_{j_{hg_{11}}} & 0 \\ * & \beta_{j_{hg_{22}}} \end{bmatrix} + \mu \begin{bmatrix} \beta_{j_{hg_{12}}} \\ 0 \end{bmatrix} \bar{P}_1 \begin{bmatrix} \beta^T_{j_{hg_{12}}} & 0 \end{bmatrix} \\ &\qquad + \mu^{-1} \begin{bmatrix} 0 \\ I_N \end{bmatrix} \bar{P}_1^{-1} \begin{bmatrix} 0 & I_N \end{bmatrix} < 0 \end{aligned} \tag{4.22}$$

where $\mu > 0$. Using the Schur complement [112], considering $N_h = K_h\bar{P}_1$, and $M_h = P_2 L_h$ and by applying the Polya's theorem on definite quadratic forms in (4.17) as in [110], (4.14) is obtained, and the proof is completed. □

## 4.3 Simulation Results: Application to a Fleet of UAVs

In this section, the proposed leader-following consensus formation control based on LPV multi-agent systems is applied to a fleet of five identical quadcopters. The two objectives of the multi-agent system are to follow the trajectories described by a virtual leader and maintain a desired shape. An LPV system is characterized by linear dynamics that depend on a parameter vector $\theta(t)$, where $\theta(t)$ is exogenous, meaning it is independent of the system's states $x(t)$ or inputs $u(t)$. In contrast, a quasi-LPV system (or state-dependent LPV) generalizes this framework by allowing $\theta(t)$ to be endogenous, i.e., explicitly dependent on $x(t)$ or $u(t)$. This enables the representation of nonlinear systems as LPV models, where nonlinearities are embedded into the parameter variation. Some considerations are made for the full nonlinear System (3.20) in order to obtain a simpler model for control design purposes. When the vehicle is near to hover, the rotational dynamics can be simplified by considering that $\left(\dot{\phi}, \dot{\theta}, \dot{\psi}\right) \approx \left(p, q, r\right)$ [129]. Also, if the hover condition is established during the entire flight period $u_z \approx m_s g$. Drag force and drag torque can be considered as null when the vehicle velocity is low. Therefore, the nonlinear equation for the quadcopter vehicle (3.20) is rewritten in an quasi-LPV state-space form as follows:

$$A(\zeta) = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&g&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-g&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&Q_1\zeta_3 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&Q_2\zeta_2 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&Q_3\zeta_3&0&0&0&0 \end{bmatrix}, B(\zeta) = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ \zeta_1/m&0&0&0 \\ 0&0&0&0 \\ 0&1/J_x&0&0 \\ 0&0&0&0 \\ 0&0&1/J_y&0 \\ 0&0&0&0 \\ 0&0&0&1/J_z \end{bmatrix} \tag{4.23}$$

where $\zeta_1 = \cos(\theta)\cos(\phi)$, $\zeta_2 = \dot{\phi}$, $\zeta_3 = \dot{\theta}$, $Q_1 = (J_y - J_z)/J_x$, $Q_2 = (J_z - J_x)/J_y$, and $Q_3 = (J_x - J_y)/J_z$. The state vector for one quadcopter vehicle is defined as $x =$

$[p_x, v_x, p_y, v_y, p_z, v_z, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ and $C$ matrix is chosen in order that the output vector is $y = [p_x, p_y, p_z, \phi, \theta, \psi]$.

Based on the *bounding box* method [128], the number of local linear models is directly related to the number of nonlinear terms. For each nonlinear term, two sub-models are obtained such that for $\bar{f}$ nonlinear terms, the global model is composed of $m = 2^{\bar{f}}$ sub-models. The scheduling variables $\zeta = [\zeta_1, \zeta_2, \zeta_3]^T$ are the non-constant elements in (4.23), where $\zeta_1 \in [0.5, 1]$(rad), $\zeta_2 \in [-\pi/2, \pi/2]$(rad/s), and $\zeta_3 \in [-\pi/2, \pi/2]$(rad/s), these constraints are acceptable when assuming slow dynamic reference displacements. The scheduling vector is defined as $\zeta \in [\zeta_f^0, \zeta_f^1]$; where $\zeta_f^0$ and $\zeta_f^1$ represents the minimum and maximum scalar values of $\zeta_f$, respectively, with $f = 1, \ldots, 3$. For each scheduling variable, two weighting functions are computed as follows:

$$\mu_f^0 = \frac{\zeta_f^1 - \zeta_f}{\zeta_f^1 - \zeta_f^0}, \quad \mu_f^1 = 1 - \mu_f^0, \tag{4.24}$$

Thus, applying (4.24) to the quadcopter vehicle, the weighting functions are expressed as:

$$\begin{aligned} \mu_1^0 &= \frac{1 - \boldsymbol{c}\theta \boldsymbol{c}\phi}{0.5}, & \mu_1^1 &= 1 - \mu_1^0, \\ \mu_2^0 &= \frac{\pi - 2\dot{\phi}}{2\pi}, & \mu_2^1 &= 1 - \mu_2^0, \\ \mu_3^0 &= \frac{\pi - 2\dot{\theta}}{2\pi}, & \mu_3^1 &= 1 - \mu_3^0, \end{aligned} \tag{4.25}$$

Therefore, for $\bar{f} = 3$, $m = 8$ scheduling functions are computed as the product of the weighting functions that correspond to each local model, as:

$$\begin{aligned} \rho_1(\zeta) &= \mu_1^0\mu_2^0\mu_3^0, & \rho_2(\zeta) &= \mu_1^0\mu_2^0\mu_3^1, \\ \rho_3(\zeta) &= \mu_1^0\mu_2^1\mu_3^0, & \rho_4(\zeta) &= \mu_1^0\mu_2^1\mu_3^1, \\ \rho_5(\zeta) &= \mu_1^1\mu_2^0\mu_3^0, & \rho_6(\zeta) &= \mu_1^1\mu_2^0\mu_3^1, \\ \rho_7(\zeta) &= \mu_1^1\mu_2^1\mu_3^0, & \rho_8(\zeta) &= \mu_1^1\mu_2^1\mu_3^1, \end{aligned} \tag{4.26}$$

satisfying the convex set sum property in (4.2). Note that (4.26) can be rewritten, as $\rho_h(\zeta) = \mu_{h_1}^1 \mu_{h_2}^2 \mu_{h_3}^3$, where $[h_1, h_2, h_3]$ is the 3-digit binary representation of $(h-1)$. The known matrices $A_h$ and $B_h$, with $h = 1, \ldots, 8$ (defining the 8 sub-models) are computed by replacing the scheduling variables $\zeta_f^0$ or $\zeta_f^1$, with $f = 1, 2$ and 3, to the matrices $A(\zeta)$ and $B(\zeta)$ in (4.23), such that:

$$A_h = A\big(\zeta_1^{h_1}, \zeta_2^{h_2}, \zeta_3^{h_3}\big), \qquad B_h = B\big(\zeta_1^{h_1}, \zeta_2^{h_2}, \zeta_3^{h_3}\big), \tag{4.27}$$

where $\zeta_f$ indicate which portion of the $f$–th scheduling variable is involved in the $h$–th sub-model. Consequently, by using the scheduling functions given by (4.26), the nonlinear system (3.20) is exactly represented as the following quasi-LPV model:

$$\begin{aligned} \dot{x} &= \sum_{h=1}^{8} \rho_h(\zeta)\big(A_h x + B_h u\big), \\ y &= Cx. \end{aligned} \tag{4.28}$$

To compute the vertex control gains $K_h$ and the vertex observer gains $L_h$, the following quasi-LPV representation of the UAVs (4.28) is used to find a solution for the LMIs presented in Theorem 2 with $\mu = 0.001$ and the Matlab software with the toolbox SDPT3 [130]. When finding the solution to Theorem 2 with the quasi-LPV model of the UAV (4.28), the total number of vertices is $S = 8$, and for the combinations of these 8 submodels, at least $s = 2$ vertices are combined at a time. We can increase the value of $s$, for example, $s = 3$, and according to [111], the higher the value of $s$, the more the conservatism is reduced. However, the computational complexity to find a solution to Theorem 2, increases. The nonlinear dynamic model of the virtual leader and followers is considered as (3.20), and the parameters of the UAVs are described in Table 4.1.

Table 4.1: Parameters of the VTOL-UAV.

| Parameter | Value | Unit |
|---|---|---|
| Mass of the vehicle, $m$ | 1.4 | Kg |
| Acceleration due to gravity, $g$ | 9.81 | m/s$^2$ |
| Moment of inertia about $x$, $J_x$ | 0.02 | Kgm$^2$ |
| Moment of inertia about $y$, $J_y$ | 0.03 | Kgm$^2$ |
| Moment of inertia about $z$, $J_z$ | 0.04 | Kgm$^2$ |

The initial conditions of the quadcopters and their respective observers are presented in Table 4.2.

Table 4.2: Initial conditions

| | Agent $i$ | | | Observer $i$ | | |
|---|---|---|---|---|---|---|
| Quadcopter$_i$ [131] | $p_{x_i}$ [m] | $p_{y_i}$ [m] | $p_{z_i}$ [m] | $\hat{p}_{x_i}$ [m] | $\hat{p}_{y_i}$ [m] | $\hat{p}_{z_i}$ [m] |
| 1 | 6 | -5 | 0 | 8 | -3.5 | 0 |
| 2 | 3 | -5 | 0 | 4 | -7 | 0 |
| 3 | 0 | -5 | 0 | -1.5 | -6 | 0 |
| 4 | -3 | -5 | 0 | -4 | -4 | 0 |
| 5 | -6 | -5 | 0 | -8 | -3 | 0 |

The initial states of the linear velocities, angles, and angular velocities are uniformly set to zero for both UAVs and observers. Furthermore, all initial conditions of the UAV leader are initialized to zero. To address the leader-following formation control problem based on consensus, the control law (4.29) is reformulated, taking into account the positions $p_i = [p_{x_i}, p_{y_i}, p_{z_i}]$, linear velocities $v_i = [v_{x_i}, v_{y_i}, v_{z_i}]$, angular positions $\omega_i = [\phi_i, \theta_i, \psi_i]$, angular velocities $\dot{\omega}_i = [\dot{\phi}_i, \dot{\theta}_i, \dot{\psi}_i]$, and $u_l$ the control law of the virtual leader, as follows:

$$u_i = \sum_{h=1}^{8} \Theta_h(\beta) K_h \left[ \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{\bar{X}}_i - \hat{\bar{X}}_j) + \alpha_i(\hat{\bar{X}}_i - \bar{X}_l) \right] + u_l, \ \forall i = 1, \dots, N, \tag{4.29}$$

where $\hat{\bar{X}}_i(t) = [\hat{p}_i^T - h_i^T, \hat{v}_i^T, \hat{\omega}_i^T, \dot{\hat{\omega}}_i^T]^T$, represents the estimated state vector of the $i$-th UAV and $\bar{X}_l = [p_l^T, v_l^T, \omega_l^T, \dot{\omega}_l^T]^T$ the state vector of the virtual UAV leader. Additionally, $H = [h_1^T, h_2^T, \ldots, h_N^T]^T$ contains the desired distance column vectors $h_i$ of every agent. In this example, five agents ($N = 5$) are considered and the matrix $H$ is configured to form a pentagon, with its components $h_i$ representing the desired positions in terms of the $x$, $y$, and $z$ axes:

$$H = r_a \begin{bmatrix} \sin(\frac{1}{5}\pi) & \sin(\frac{3}{5}\pi) & \sin(\pi) & \sin(\frac{7}{5}\pi) & \sin(\frac{9}{5}\pi) \\ -\cos(\frac{1}{5}\pi) & -\cos(\frac{3}{5}\pi) & -\cos(\pi) & -\cos(\frac{7}{5}\pi) & -\cos(\frac{9}{5}\pi) \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.30}$$

where $r_a = 4\,\mathrm{m}$ is the radius of the pentagon. The considered undirected communication graph topology is represented in Figure 4.2.

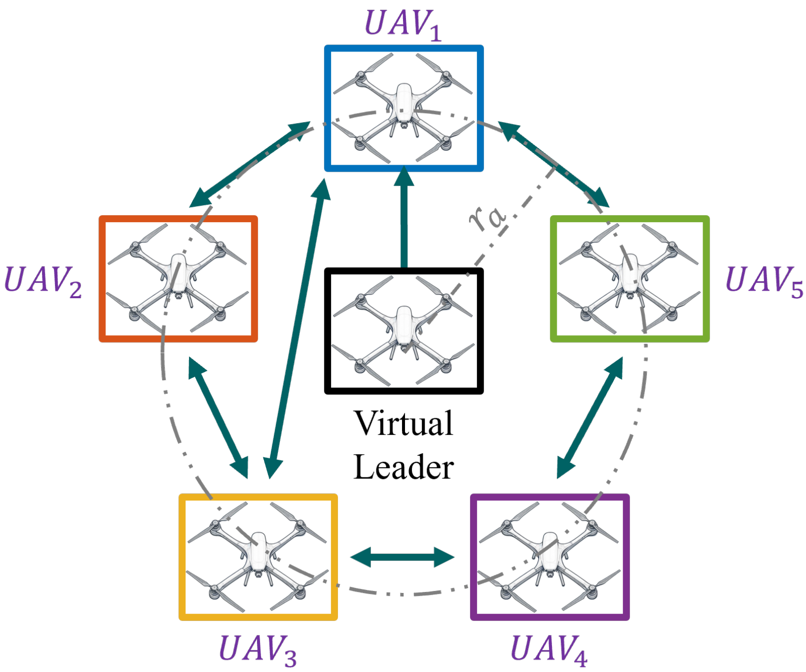


Figure 4.2: Topology of the undirected leader-following communication graph.

An LPV proportional-integral (LPV-PI) controller was implemented for trajectory tracking to stabilize the virtual leader and utilize it as a reference model for the followers. Consequently, stability of the followers is achieved through tracking the leader. Referring to Figure 4.2, the communication with the UAV leader is represented by the matrix $\Lambda = \mathrm{diag}(1, 0, 0, 0, 0)$, alongside the Laplacian matrix $\mathcal{L}$:

$$\mathcal{L} = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}. \tag{4.31}$$

To prove the performance of the LPV observer (4.6), the estimation error $e_i = y_i - \hat{y}_i$ is shown in Figure 4.3, where the convergence of all the estimated states of all agents is guaranteed. Sensor noise was added as a random function with normal distribution, a mean value of zero, and a standard deviation of 0.003.

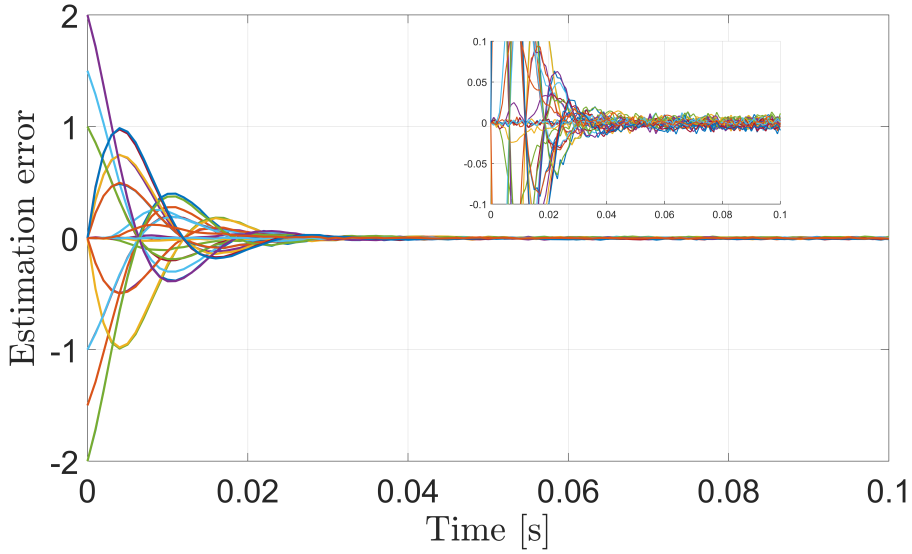


Figure 4.3: Observer convergence.

To ensure smooth trajectory tracking, we employ the following trajectory planning equation (4.32) to adjust the altitude of the virtual leader:

$$z_{ref} = \begin{cases} z(0) + \frac{10}{T^3}(z(T) - z(0))t^3 - \frac{15}{T^4}(z(T) - z(0))t^4 + \frac{6}{T^5}(z(T) - z(0))t^5, & 0 \leq t \leq T \\ 20, & t > T \end{cases} \tag{4.32}$$

The initial altitude of the virtual leader is $z(0) = 0\,\text{m}$, while the final altitude is $z(T) = 20\,\text{m}$, with a time duration of $T = 5\,\text{s}$ between the initial and final altitudes. To formulate (4.32), the initial and final conditions of the velocity $\dot{z}_{ref}$ and the acceleration $\ddot{z}_{ref}$ are taken into account, both set to zero to ensure a smooth trajectory. For clarity, it is important to note that the references for the $x$ and $y$ axes are always zero, as trajectory tracking is only considered for the $z$-axis. Figure 4.4 illustrates the altitude changes of the reference, virtual leader, and followers throughout the simulation that has a duration of 20 s.

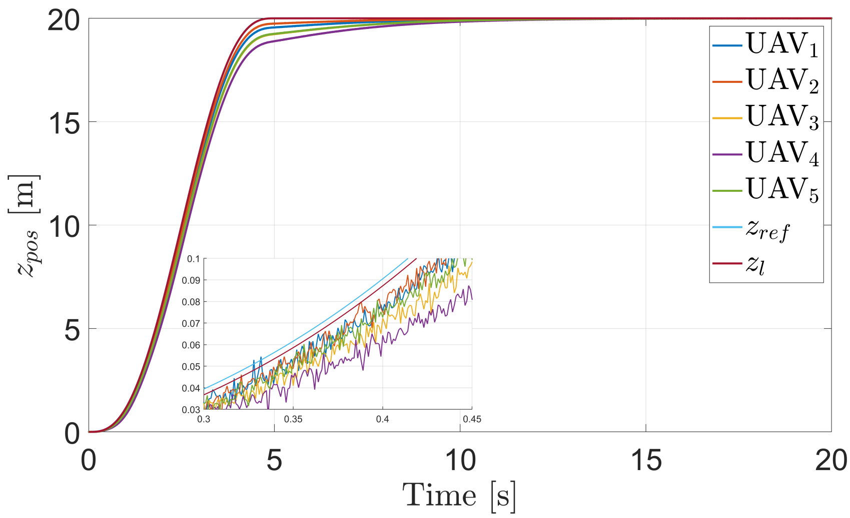


Figure 4.4: Reference altitude, UAV leader, and followers' altitude.

The velocity error during the transient between the follower UAVs and the leader in Figure 4.4 arises from the absence of an integral gain in the consensus algorithm (4.29), which would otherwise enhance trajectory tracking. Figure 4.5 shows the obtained thrust $u_z$ for the virtual leader and the five quadcopters when (4.32) is considered for the virtual leader.

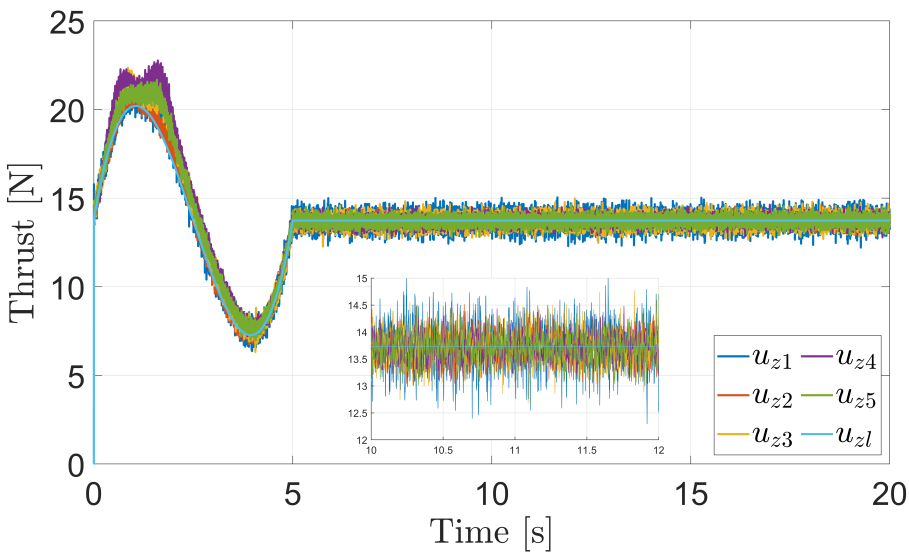


Figure 4.5: Thrust of the leader and the followers.

Figure 4.6 shows the performance of the quadcopters' trajectories. The leader quadcopter is in the middle of all the followers in black color at the desired altitude. The followers reach their desired position in the vertex of the pentagon highlighted in red color.

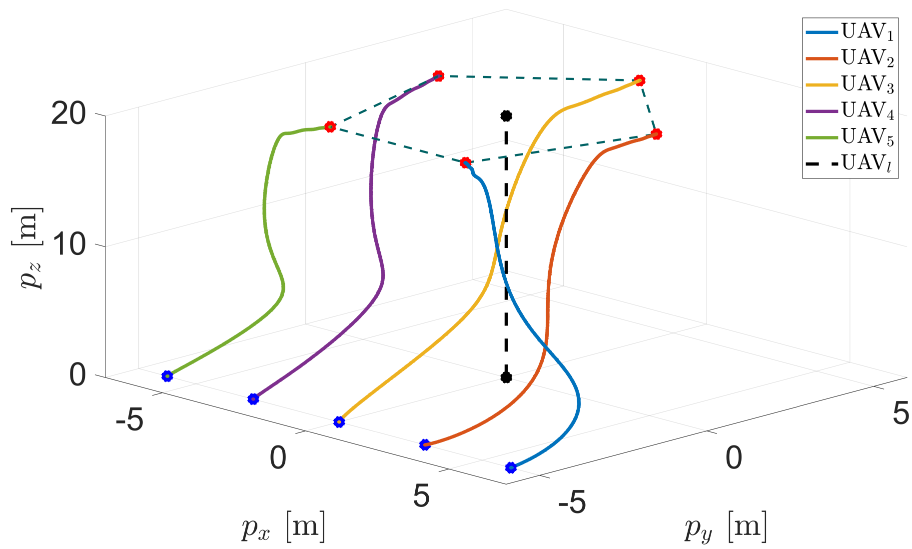


Figure 4.6: Trajectories of the quadcopters.

To measure the synchronization error in this example, let us define $\Delta_i = \|x_i - x_l\| - r$. Figure 4.7 represents the evolution of $\Delta_i$. The quadcopters follow the leader's trajectories in their desired pentagon vertex when $\Delta_i = 0$.

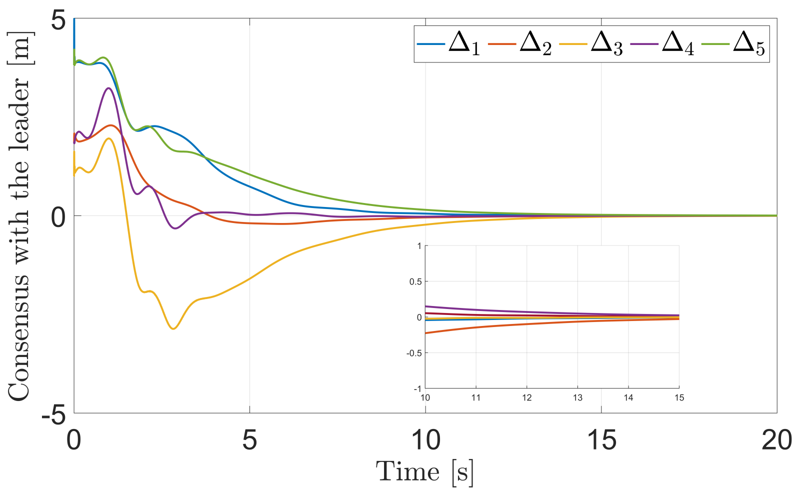


Figure 4.7: Synchronization error.

Figure 4.7 shows that at approximately when $t = 12$s, all followers reach consensus with the virtual leader. This means they are positioned at the vertices of the pentagon, maintaining the desired distance from the leader and the same altitude, as all errors converge to zero.

Let us define $d_{ij} = \|([x_i, y_i, z_i]^T - [x_j, y_j, z_j]^T) - (h_i - h_j)\|$ if $d_{ij} = 0$, then, the desired formation is reached. Figure 4.8 illustrates the relative distances performance between the quadcopters to reach the desired formation $H$.

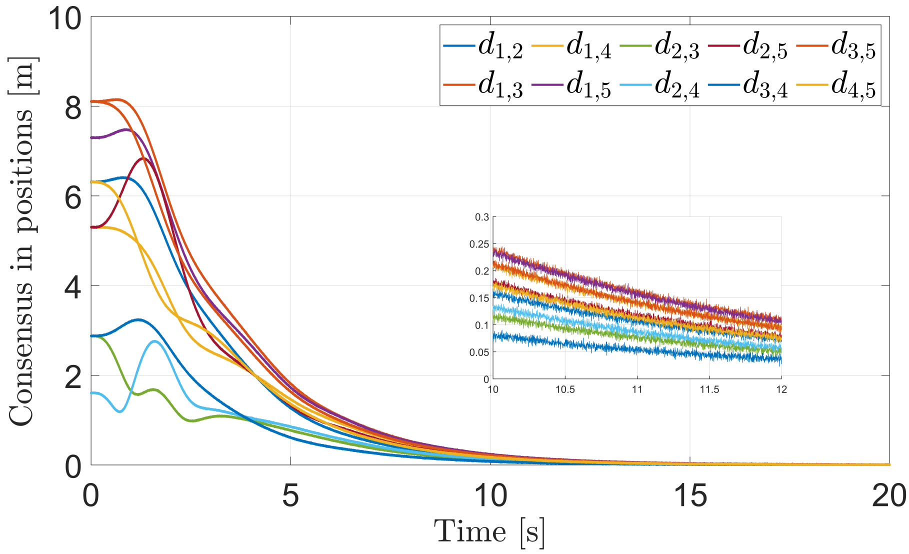


Figure 4.8: Distances between followers to reach the formation.

Figure 4.8 illustrates consensus without considering the virtual leader. When all UAVs reach their desired positions relative to the other followers, they achieve consensus and fulfill the formation control objective. This is why, around $t = 12$s the distances between them match the desired values and consequently converge to zero.

## 4.4 Conclusions

A distributed leader-following consensus control for LPV multi-agent systems was presented. The LMIs conditions to compute the control and observer gains were derived using the less conservative Polya's theorem. The simulation results demonstrate that a team of UAVs successfully followed the trajectories of a virtual leader while maintaining the desired formation under the proposed approach. Although a direct comparison between the results in Chapter 3 and this chapter is not feasible (due to the absence of the $\mathcal{H}_\infty$ criterion), it is worth noting that shifting from an LTI to an LPV design increases the complexity of solving the LMIs and raises computational demands. This arises because the LPV approach must account for submodel combinations and permutations, but the added value lies in its ability to better capture the nonlinear dynamics and time-varying parameters of real-world systems, improving robustness and performance compared to LTI methods. Because an agent can be affected by actuator or sensor faults, compromising the global objective of the team, an LMI-based FTC for LPV multi-agent systems is proposed in the next chapter.

# Chapter 5

# Fault-Tolerant Leader-Following Consensus Control for LPV Multi-Agent Systems

## Contents



## 5.1 Introduction

Building upon the observer-based leader-following consensus control strategy developed in Chapter 4 for LPV multi-agent systems, this chapter extends the framework to incorporate fault tolerance in the presence of actuator faults. While Chapter 4 addressed stability and consensus under nominal conditions using an observer-based approach, the current work introduces a virtual actuator mechanism to actively compensate for actuator faults, ensuring the system's resilience and performance preservation.

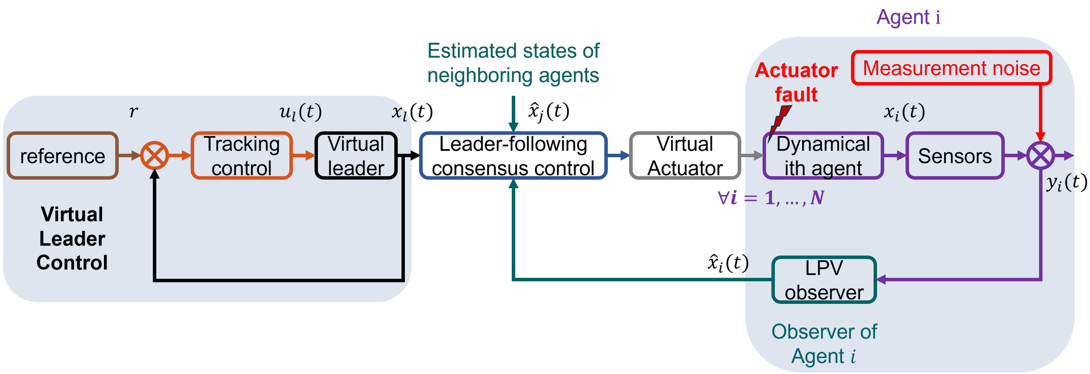


Figure 5.1: Diagram illustrating the FTC mechanism for LPV multi-agent systems.

As illustrated in Figure 5.1, the proposed FTC design integrates the observer-based consensus methodology with a reconfigurable virtual actuator, enabling the multi-agent system to maintain formation control and leader-tracking objectives despite actuator failures. This diagram extends the nominal control scheme presented in Figure 4.1, where no fault compensation was considered. The virtual actuator dynamically adjusts the control signals to mitigate fault effects, ensuring stability without requiring controller redesign, unlike the baseline approach in Chapter 4, which assumes fully functional actuators. Simulation results demonstrate the effectiveness of this FTC strategy in preserving consensus under faulty conditions.

## 5.2 Problem Formulation

Let us consider the following homogeneous LPV continuous multi-agent system with actuator faults

$$\begin{aligned} \dot{x}_i(t) &= A\big(\beta_i(t)\big)x_i(t) + B_f\Big(\beta_i(t), \vartheta_i(t)\Big)\Big(u_{v_i}(t) + f_{u_i}(t)\Big), \\ y_i(t) &= C\big(\beta_i(t)\big)x_i(t), \end{aligned} \tag{5.1}$$

composed of $N$ identical systems. For each one, $i$ is the agent number ($i = 1, \ldots, N$); $x_i(t) \in \mathbb{R}^n$ is the state vector, $u_{v_i}(t) \in \mathbb{R}^r$ is the input vector, $f_{u_i}(t) \in \mathbb{R}^r$ is the additive actuator fault, $y_i(t) \in \mathbb{R}^p$ is the output vector, $A\big(\beta_i(t)\big) \in \mathbb{R}^{n\times n}$, $B_f\big(\beta_i(t)\big) \in \mathbb{R}^{n\times r}$, and $C\big(\beta_i(t)\big) \in \mathbb{R}^{p\times n}$ are the matrix functions scheduled by the vector $\beta_i(t) \in \mathbb{R}^{n_\beta}$, which is considered known and available for every $i$-th agent, and is assumed to vary in a closed set $\mathcal{R}$. $\vartheta_i(t) \in \mathbb{R}^{n_\vartheta}$ represents the multiplicative actuator faults as follows:

$$B_f\big(\beta_i(t), \vartheta_i(t)\big) = B\Big(\beta_i(t)\Big)\vartheta_i(t)\Big(\kappa_i(t)\Big), \tag{5.2}$$

where

$$\vartheta_i(t)\big(\kappa_i(t)\big) = \mathrm{diag}(\kappa_{i_1}(t), \kappa_{i_2}(t), \ldots, \kappa_{i_r}(t)),\ 0 \leq \kappa_{i_\varepsilon}(t) \leq 1,\ 1 \leq \varepsilon \leq r \tag{5.3}$$

$B\big(\beta_i(t)\big) \in \mathbb{R}^{n\times r}$ and $B_f\big(\beta_i(t)\big)$ are the faultless and faulty input matrices respectively. $\kappa_{i_\varepsilon}(t)$ denotes the effectiveness of the $\varepsilon$-th actuator of each $i$-th agent, such that $\kappa_{i_\varepsilon}(t) = 0$

is a total failure of the $\varepsilon$-th actuator and $\kappa_{i_\varepsilon}(t) = 1$ represents a healthy $\varepsilon$-th actuator. The *bounding box* method [128] is employed to represent the system matrices, $A(\beta(t))$, $B(\beta(t))$, and $C(\beta(t))$, as a convex combination of $S$ vertex matrices, ensuring that

$$\begin{aligned} &\begin{pmatrix} A(\beta_i(t)) \\ B(\beta_i(t)) \\ C(\beta_i(t)) \end{pmatrix} = \sum_{h=1}^{S} \Theta_h(\beta_i(t)) \begin{pmatrix} A_h \\ B_h \\ C_h \end{pmatrix}, \\ &\sum_{h=1}^{S} \Theta_h(\beta_i(t)) = 1, \quad 0 \leq \Theta_h(\beta_i(t)) \leq 1 \quad \forall \beta_i(t) \in \mathcal{R} \end{aligned} \tag{5.4}$$

Consensus between the leader and the followers can be achieved only if $\beta_1(t) \in \mathcal{R}$, $\beta_2(t) \in \mathcal{R}, \ldots$, $\beta_N(t) \in \mathcal{R}$, and $\beta_l(t) \in \mathcal{R}$, indicating synchronization between the followers and the virtual leader since the scheduling matrices vary within the same set $\mathcal{R}$ [119]. The dependencies of $\beta_i(t)$ and $\beta_l(t)$ are considered as $\beta(t)$ for simplification. The synchronization error between the followers and the virtual leader is denoted by:

$$\delta_i(t) = x_i(t) - x_l(t), \tag{5.5}$$

then, the dynamics of the synchronization error between each agent $i$ and the leader is defined by:

$$\dot{\delta}_i(t) = A(\beta(t))\delta_i(t) + B_f\big(\beta(t), \vartheta_i(t)\big)\big(u_{v_i}(t) + f_{u_i}(t)\big). \tag{5.6}$$

Let us propose the nominal observer-based leader-following consensus control law $u_i(t)$

$$u_i(t) = K\big(\beta(t)\big)\Bigg[\sum_{j \in \mathcal{N}_i} a_{ij}\big(\hat{x}_i(t) - \hat{x}_j(t)\big) + \alpha_i\big(\hat{x}_i(t) - x_l(t)\big)\Bigg], \tag{5.7}$$

where the LPV control gain to be designed is denoted by $K(\beta(t)) \in \mathbb{R}^{r \times n}$, and $\hat{x}_j(t) \in \mathbb{R}^n$ represents the estimated state vector of neighboring agents. This protocol considers the estimated states of agent $i$ and their neighbors $j$. Consequently, the estimated state vector of each agent, denoted as $\hat{x}_i(t) \in \mathbb{R}^n$, is determined by the following distributed LPV observer [117]. Here, $\hat{x}_i(t)$ represents the estimation of $x_i(t)$ regardless of whether an actuator fault occurs:

$$\begin{aligned} \dot{\hat{x}}_i(t) &= A(\beta(t))\hat{x}_i(t) + B(\beta(t))u_i(t) + L_P(\beta(t))\Big(y_i(t) + y_{v_i}(t) - \hat{y}_i(t)\Big), \\ \hat{y}_i(t) &= C(\beta(t))\hat{x}_i(t), \end{aligned} \tag{5.8}$$

where $L_P(\beta(t)) \in \mathbb{R}^{n \times p}$ is the LPV observer gain to be designed, $y_{v_i}(t) \in \mathbb{R}^p$ is the virtual actuator output vector, and $\hat{y}_i(t) \in \mathbb{R}^p$ is the estimated output vector.

To ensure the stability of the multi-agent system (5.1) in the presence of actuator faults, a virtual actuator is considered. The virtual actuator acts as a reconfiguration block that compensates for faults by redistributing control efforts to healthy actuators while isolating the controller from the fault effects [19]. The structure of the virtual actuator can be static or dynamic depending on whether the following condition is satisfied:

$$\operatorname{rank}\Big(B_f\big(\beta(t), \vartheta_i(t)\big)\Big) = \operatorname{rank}\Big(B\big(\beta(t)\big)\Big) \neq 0 \; \forall \beta(t) \in \mathcal{R}, \; i = 1, \ldots, N. \tag{5.9}$$

When (5.9) is satisfied, the performance degradation due to the faulty actuator can be represented as a linear combination of the remaining actuators. In such cases, fault tolerance can be achieved through the redistribution of control inputs. The virtual actuator reconfiguration in this case is static, as it does not involve additional dynamics. The virtual actuator reconfiguration structure is defined as follows [119]:

$$u_{v_i}(t) = N_v(\beta(t), \hat{\vartheta}_i(t))u_i(t) - \hat{f}_{u_i}(t), \tag{5.10}$$

where $\hat{\vartheta}_i(t)$ and $\hat{f}_{u_i}(t)$ are the estimates of $\vartheta_i(t)$ and $f_{u_i}(t)$ respectively, $u_i(t)$ is the fault-free consensus control law (5.7) and $N_v(\beta(t), \hat{\vartheta}_i(t))$ is given by

$$N_v(\beta(t), \hat{\vartheta}_i(t)) = B_f\big(\beta(t), \hat{\vartheta}_i(t)\big)^{\dagger} B(\beta(t)), \tag{5.11}$$

with $B_f\big(\beta(t), \hat{\vartheta}_i(t)\big)^{\dagger}$ being the pseudo-inverse of $B_f\big(\beta(t), \hat{\vartheta}_i(t)\big)$.

When (5.9) is not satisfied, the control input redistribution (5.10) becomes insufficient. In these cases, fault tolerance requires a dynamic virtual actuator [21] determined by:

$$B^*\big(\beta(t)\big) = B_f\big(\beta(t), \hat{\vartheta}_i(t)\big) N_v\big(\beta(t), \hat{\vartheta}_i(t)\big). \tag{5.12}$$

Notably, $B^*\big(\beta(t)\big)$ is independent of $\vartheta_i(t)$ since $N_v\big(\beta(t), \hat{\vartheta}_i(t)\big)$ compensates for the actuator fault. The corresponding control law with virtual actuator becomes:

$$u_{v_i}(t) = N_v\big(\beta(t), \hat{\vartheta}_i(t)\big)\big(u_i(t) - M_v(\beta(t))x_{v_i}(t)\big) - \hat{f}_{u_i}(t), \tag{5.13}$$

where $M_v(\beta(t))$ is the LPV virtual actuator gain. The virtual actuator state $x_{v_i}(t) \in \mathbb{R}^n$ is given by:

$$\begin{aligned} \dot{x}_{v_i}(t) &= \Big(A\big(\beta(t)\big) + B^*\big(\beta(t)\big)M_v\big(\beta(t)\big)\Big)x_{v_i}(t) + \Big(B\big(\beta(t)\big) - B^*\big(\beta(t)\big)\Big)u_i(t), \\ y_{v_i}(t) &= C\big(\beta(t)\big)x_{v_i}(t). \end{aligned} \tag{5.14}$$

The polytopic control gains $K_h$, the polytopic observer gains $L_{P_h}$, and the polytopic virtual actuator gains $M_{v_h}$ are written considering the convex combination as follows

$$\begin{aligned} &\begin{pmatrix} K(\beta(t)) \\ L_P(\beta(t)) \\ M_v(\beta(t)) \end{pmatrix} = \sum_{h=1}^{S} \Theta_h(\beta(t)) \begin{pmatrix} K_h \\ L_{P_h} \\ M_{v_h} \end{pmatrix}. \\ &\sum_{h=1}^{S} \Theta_h(\beta(t)) = 1, \quad 0 \le \Theta_h(\beta(t)) \le 1 \quad \forall \beta(t) \in \mathcal{R} \end{aligned} \tag{5.15}$$

Let us consider the time-varying parameters vector $\beta(t) = [\beta_1(t)^T, \beta_2(t)^T, \ldots, \beta_N(t)^T]^T$, defining the matrix:

$$\Theta_{ij}(\beta(t)) = \mathrm{diag}(\Theta_{ij}(\beta_1(t)), \ldots, \Theta_{ij}(\beta_N(t))) \tag{5.16}$$

with $\Theta_{ij}(\beta_g(t)) \triangleq \Theta_i(\beta_g(t))\Theta_j(\beta_g(t))$, $g = 1, \ldots, N$. Consider $\delta(t) = [\delta_1^T(t), \delta_2^T(t), \ldots, \delta_N^T(t)]^T$, $x_v(t) = [x_{v_1}^T(t), x_{v_2}^T(t), \ldots, x_{v_N}^T(t)]^T$, and $\hat{x}(t) = [\hat{x}_1^T(t), \hat{x}_2^T(t), \ldots, \hat{x}_N^T(t)]^T$. Thus, (5.6), (5.8), and (5.14) can be rewritten using the Kronecker product as follows:

$$\dot{x}_v(t) = \sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_{hg}\big(\beta(t)\big)\Big[\big(I_N \otimes (A_h + B_g^* M_{v_h})\big)x_v(t) + \big(\bar{\mathcal{L}} \otimes (B_g - B_g^*)K_h\big)\hat{\delta}(t)\Big] \quad (5.17)$$

$$\dot{\delta}(t) = \sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_{hg}\big(\beta(t)\big)\Big[(I_N \otimes A_h)\delta(t) + (\bar{\mathcal{L}} \otimes B_g^* K_h)\hat{\delta}(t) - (I_N \otimes B_g^* M_{v_h})x_v(t)\Big] \quad (5.18)$$

$$\dot{\hat{x}}(t) = \sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_{hg}\big(\beta(t)\big)\Big[(I_N \otimes A_h)\hat{x}(t) + (\bar{\mathcal{L}} \otimes B_g K_h)\hat{\delta}(t) + (I_N \otimes L_{P_h} C_g)\big(x(t) + x_v(t) - \hat{x}(t)\big)\Big] \quad (5.19)$$

Considering $z_1(t) = x_v(t)$, $z_2(t) = \delta(t) + x_v(t)$, $z_3(t) = \delta(t) + x_v(t) - \hat{\delta}(t) = x(t) + x_v(t) - \hat{x}(t)$, the closed-loop multi-agent system is expressed as follows (to simplify the notation, time dependency is omitted):

$$\dot{z} = \sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_{hg}(\beta)\times \begin{bmatrix} I_N \otimes (A_h + B_g^* M_{v_h}) & \bar{\mathcal{L}} \otimes (B_g - B_g^*)K_h & -\bar{\mathcal{L}} \otimes (B_g - B_g^*)K_h \\ \underline{0} & I_N \otimes A_h + \bar{\mathcal{L}} \otimes B_g K_h & -\bar{\mathcal{L}} \otimes B_g K_h \\ \underline{0} & \underline{0} & I_N \otimes (A_h - L_h C_g) \end{bmatrix} z \quad (5.20)$$

where $\underline{0}$ represents a matrix of zeros with proper dimensions.

The primary objective of this work is to guarantee the existence of the LPV consensus control gain (5.7), the LPV observer gain (5.8), and the LPV virtual actuator gain (5.14) for coordinating the multi-agent system (5.1). This is achieved using LMIs and ensuring that these gains are computed simultaneously, aiming to reduce conservatism.

**Remark 1.** *Notice that $\hat{\vartheta}_i$ and $\hat{f}_{u_i}$ represent unknown parameters requiring detection, isolation and estimation through fault diagnosis methodologies. For this, numerous efficient techniques documented in the literature facilitate precise estimation of both additive and multiplicative faults, e.g. [132]. Therefore, for the sake of simplicity, $\hat{\vartheta}_i$ and $\hat{f}_{u_i}$ are assumed to be known to focus the presentation on the core contribution of this work which is analyzing and developing the theoretical framework of virtual actuators in multi-agent systems.*

**Remark 2.** *While there are various approaches, such as Finsler's lemma, that yield strong results in robust optimization and LMI calculations, Polya's theorem, originating from fuzzy control theory, provides a more structured way to perform the necessary permutations and combinations for calculating gains in the submodels of the LPV system. This results in less conservative solutions compared to other methods of combining and permuting the vertices, as demonstrated in [111].*

## 5.3 Fault-Tolerant Observer-Based Leader-Following Consensus Control for LPV MASs

In this section, the LMI-based conditions ensuring the existence of LPV control, LPV observer, and LPV virtual actuator gains are presented in the following theorem. These conditions are established through Lyapunov stability analysis for a vertex combination of the LPV multi-agent system (5.1). The implementation of Polya's theorem on definite quadratic forms is applied to manage the various sets of possible permutations among the considered subsystems in the LPV representation.

**Theorem 3.** *Consider the closed-loop LPV multi-agent system* (5.20)*. The virtual actuator* (5.17)*, synchronization error* (5.18)*, and estimation* (5.19) *are proven to be exponentially stable for any $s \in \mathbb{N}$, with $s \geq 2$, given the eigenvalues $\lambda_j(\bar{\mathcal{L}})$, $\forall j = 1, 2, \ldots, N$; if there exist symmetric matrices $\bar{P}_1 > 0 \in \mathbb{R}^{n\times n}$, $\bar{P}_2 > 0 \in \mathbb{R}^{n\times n}$, $\bar{P}_3 > 0 \in \mathbb{R}^{n\times n}$, matrices $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_S$, $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_S$, and $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_S$; the tuning scalar variables $\mu_1 > 0$ and $\mu_2 > 0$ such that*

$$\sum_{\vec{\sigma}\in\mathcal{D}(\vec{\sigma})}\begin{bmatrix} \mathcal{Q}_{11} & \lambda_j\big(B_{\sigma_2}-B^*_{\sigma_2}\big)\mathcal{N}_{\sigma_1} & \underline{0} & \mathcal{Q}_{j14} & \underline{0} & \underline{0} & \underline{0} \\ * & \mathcal{Q}_{j22} & \underline{0} & \underline{0} & \underline{0} & \mathcal{Q}_{j26} & \underline{0} \\ * & * & \mathcal{Q}_{33} & \underline{0} & I & \underline{0} & I \\ * & * & * & -\mu_1\bar{P}_2 & \underline{0} & \underline{0} & \underline{0} \\ * & * & * & * & -\mu_1^{-1}\bar{P}_2 & \underline{0} & \underline{0} \\ * & * & * & * & * & -\mu_2\bar{P}_2 & \underline{0} \\ * & * & * & * & * & * & -\mu_2^{-1}\bar{P}_2 \end{bmatrix} < 0 \quad (5.21)$$

*where $\mathcal{Q}_{11} = He\{A_{\sigma_1}\bar{P}_1 + B^*_{\sigma_2}\mathcal{M}_{\sigma_1}\}$, $\mathcal{Q}_{j22} = He\{A_{\sigma_1}\bar{P}_2 + \lambda_j B_{\sigma_2}\mathcal{N}_{\sigma_1}\}$, $\mathcal{Q}_{33} = He\{P_3A_{\sigma_1} - \mathcal{O}_{\sigma_1}C_{\sigma_2}\}$, $\mathcal{Q}_{j14} = -\lambda_j\big(B_{\sigma_2} - B^*_{\sigma_2}\big)\mathcal{N}_{\sigma_1}$, and $\mathcal{Q}_{j26} = -\lambda_j B_{\sigma_2}\mathcal{N}_{\sigma_1}$, He is used to represent the Hermitian matrices. If* (5.21) *holds $\forall\vec{\sigma} \in \mathcal{R}_s^+$, where:*

$$\mathcal{R}_s = \left\{\vec{\sigma} = [\sigma_1, \ldots, \sigma_s]^T \in \mathbb{N}^s | 1 \leq \sigma_l \leq S \ \forall l = 1, \ldots, s\right\}, \quad (5.22)$$

$$\mathcal{R}_s^+ = \{\vec{\sigma} \in \mathcal{R}_s | \sigma_k \leq \sigma_{k+1}, k = 1, \ldots, s-1\}, \quad (5.23)$$

*and the set of permutations with possible repeated elements of the multi-index $\vec{\sigma}$ is denoted by $\mathcal{D}(\vec{\sigma}) \subset \mathcal{R}_s$, then, the virtual actuator vertex gains can be computed with $M_{v_h} = \mathcal{M}_h\bar{P}_1^{-1}$ the control vertex gains with $K_h = \mathcal{N}_h\bar{P}_2^{-1}$, and the observer vertex gain with $L_h = P_3^{-1}\mathcal{O}_h$ for $1 \leq h \leq S$.*

*Proof.* Let us choose a quadratic candidate Lyapunov function as follows:

$$V(z) = z^T\begin{bmatrix} I_N\otimes P_1 & \underline{0} & \underline{0} \\ \underline{0} & I_N\otimes P_2 & \underline{0} \\ \underline{0} & \underline{0} & I_N\otimes P_3 \end{bmatrix} z \quad (5.24)$$

The derivative of (5.24) along the trajectories of (5.20) is given by:

$$
\begin{aligned}
dV(z)/dt = & \sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_{hg}(\beta)\Big\{2z_1^T\Big[I_N\otimes\big(P_1A_h+P_1B_g^*M_{v_h}\big)\Big]z_1+ \\
& 2z_1^T\Big[\bar{\mathcal{L}}\otimes P_1\big(B_g-B_g^*\big)K_h\Big]z_2-2z_1^T\Big[\bar{\mathcal{L}}\otimes P_1\big(B_g-B_g^*\big)K_h\Big]z_3+ \\
& 2z_2^T\Big[I_N\otimes P_2A_h+\bar{\mathcal{L}}\otimes P_2B_gK_h\Big]z_2-2z_2^T\Big[\bar{\mathcal{L}}\otimes P_2B_gK_h\Big]z_3+ \\
& 2z_3^T\Big[I_N\otimes\big(P_3A_h-P_3L_hC_g\big)\Big]z_3\Big\}.
\end{aligned} \tag{5.25}
$$

Let us perform a spectral decomposition of the matrix $\bar{\mathcal{L}}$, such that $\bar{\mathcal{L}}=TJT^{-1}$ with an invertible matrix $T\in\mathbb{R}^{N\times N}$ and a diagonal matrix $J=\text{diag}(\lambda_1,\lambda_2,\ldots,\lambda_N)\in\mathbb{R}^{N\times N}$ of which $\lambda_j(\bar{\mathcal{L}})$, $j=1,2,\ldots,N$ are the eigenvalues of $\bar{\mathcal{L}}$. Defining the change of coordinates as $\xi_1=(T^{-1}\otimes I_N)z_1$, $\xi_2=(T^{-1}\otimes I_N)z_2$, and $\xi_3=(T^{-1}\otimes I_N)z_3$, (5.25) can be rewritten as:

$$
\begin{aligned}
dV(z)/dt = & \sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_{hg}(\beta)\Big\{\sum_{j=1}^{N}\xi_1^T\text{He}\{P_1A_h+P_1B_g^*M_{v_h}\}\xi_1+2\sum_{j=1}^{N}\xi_1^T\big[\lambda_jP_1\big(B_g-B_g^*\big)K_h\big]\xi_2 \\
& -2\sum_{j=1}^{N}\xi_1^T\big[\lambda_jP_1\big(B_g-B_g^*\big)K_h\big]\xi_3+\sum_{j=1}^{N}\xi_2^T\text{He}\{P_2A_h+\lambda_jP_2B_gK_h\}\xi_2 \\
& -2\sum_{j=1}^{N}\xi_2^T\big[\lambda_jP_2B_gK_h\big]\xi_3+\sum_{j=1}^{N}\xi_3^T\text{He}\{P_3A_h-P_3L_hC_g\}\xi_3\Big\}.
\end{aligned} \tag{5.26}
$$

Considering the vector $[\xi_{1_j}^T,\xi_{2_j}^T,\xi_{3_j}^T]^T$, (5.26) can be rewritten as follows:

$$
dV(z)/dt=\sum_{h=1}^{S}\sum_{g=1}^{S}\Theta_h(\beta)\Theta_g(\beta)\left\{\begin{bmatrix}\xi_{1_j}^T & \xi_{2_j}^T & \xi_{3_j}^T\end{bmatrix}\mathfrak{B}_{j_{hg}}\begin{bmatrix}\xi_{1_j}\\ \xi_{2_j}\\ \xi_{3_j}\end{bmatrix}\right\}, \tag{5.27}
$$

with

$$
\mathfrak{B}_{j_{hg}}=\begin{bmatrix}\text{He}\{P_1A_h+P_1B_g^*M_{v_h}\} & \lambda_jP_1\big(B_g-B_g^*\big)K_h & -\lambda_jP_1\big(B_g-B_g^*\big)K_h \\ (-\lambda_jP_1\big(B_g-B_g^*\big)K_h)^T & \text{He}\{P_2A_h+\lambda_jP_2B_gK_h\} & -\lambda_jP_2B_gK_h \\ (-\lambda_jP_1\big(B_g-B_g^*\big)K_h)^T & (-\lambda_jP_2B_gK_h)^T & \text{He}\{P_3A_h-P_3L_hC_g\}\end{bmatrix}. \tag{5.28}
$$

We establish $\bar{P}_1=P_1^{-1}$ and $\bar{P}_2=P_2^{-1}$. Subsequently, (5.28) is both pre and post-multiplied by $\text{diag}(\bar{P}_1,\bar{P}_2,I)$, resulting in:

$$
\rho_{j_{hg}}=\begin{bmatrix}\rho_{j_{hg_{11}}} & \rho_{j_{hg_{12}}} & \rho_{j_{hg_{13}}} \\ * & \rho_{j_{hg_{22}}} & \rho_{j_{hg_{23}}} \\ * & * & \rho_{j_{hg_{33}}}\end{bmatrix}<0, \tag{5.29}
$$

where $\rho_{j_{hg_{11}}} = \text{He}\{A_h\bar{P}_1 + B_g^* M_{v_h}\bar{P}_1\}$, $\rho_{j_{hg_{12}}} = \lambda_j(B_g - B_g^*)K_h\bar{P}_2$, $\rho_{j_{hg_{13}}} = -\lambda_j(B_g - B_g^*)K_h\bar{P}_2$, $\rho_{j_{hg_{22}}} = \text{He}\{A_h\bar{P}_2 + \lambda_j B_g K_h\bar{P}_2\}$, $\rho_{j_{hg_{23}}} = -\lambda_j B_g K_h$, and $\rho_{j_{hg_{33}}} = \text{He}\{P_3A_h - P_3L_hC_g\}$, then, (5.29) can be rewritten as follows:

$$\begin{bmatrix} \rho_{j_{hg_{11}}} & \rho_{j_{hg_{12}}} & \underline{0} \\ * & \rho_{j_{hg_{22}}} & \underline{0} \\ * & * & \rho_{j_{hg_{33}}} \end{bmatrix} + \text{He}\left\{ \begin{bmatrix} \rho_{j_{hg_{13}}} \\ \underline{0} \\ \underline{0} \end{bmatrix} \begin{bmatrix} \underline{0} & \underline{0} & I_N \end{bmatrix} \right\} + \text{He}\left\{ \begin{bmatrix} \underline{0} \\ \rho_{j_{hg_{23}}} \\ \underline{0} \end{bmatrix} \begin{bmatrix} \underline{0} & \underline{0} & I_N \end{bmatrix} \right\} < 0 \tag{5.30}$$

Applying the Young relation [113], the following inequality is obtained:

$$\begin{aligned} &\begin{bmatrix} \rho_{j_{hg_{11}}} & \rho_{j_{hg_{12}}} & \underline{0} \\ * & \rho_{j_{hg_{22}}} & \underline{0} \\ * & * & \rho_{j_{hg_{33}}} \end{bmatrix} + \mu_1 \begin{bmatrix} \rho_{j_{hg_{13}}} \\ \underline{0} \\ \underline{0} \end{bmatrix} \bar{P}_2 \begin{bmatrix} \rho_{j_{hg_{13}}}^T & \underline{0} & \underline{0} \end{bmatrix} + \mu_1^{-1} \begin{bmatrix} \underline{0} \\ \underline{0} \\ I_N \end{bmatrix} \bar{P}_2 \begin{bmatrix} \underline{0} & \underline{0} & I_N \end{bmatrix} \\ &+ \mu_2 \begin{bmatrix} \underline{0} \\ \rho_{j_{hg_{23}}} \\ \underline{0} \end{bmatrix} \bar{P}_2 \begin{bmatrix} \underline{0} & \rho_{j_{hg_{23}}}^T & \underline{0} \end{bmatrix} + \mu_2^{-1} \begin{bmatrix} \underline{0} \\ \underline{0} \\ I_N \end{bmatrix} \bar{P}_2 \begin{bmatrix} \underline{0} & \underline{0} & I_N \end{bmatrix} < 0 \end{aligned} \tag{5.31}$$

Employing the Schur complement [112] and considering $\mathcal{M}_h = Mv_h\bar{P}_1$, $\mathcal{N}_h = K_h\bar{P}_2$, and $\mathcal{O}_h = P_3L_h$, where $1 \le h \le S$, and applying Polya's theorem on definite quadratic forms [111] to address the double sum problem, (5.21) is obtained. Hence, the proof is complete. □

**Remark 3.** *As extensively discussed by [111], the application of Polya's theorem provides a set of sufficient conditions to assess the definiteness of fuzzy summations. These conditions become progressively less conservative as the complexity parameter s increases. Notably, they are asymptotically exact, meaning there exists a finite value of s at which they become necessary and sufficient.*

## 5.4 Simulation Results: Application to a Fleet of UAVs

In this section, is demostrated the effectiveness of the introduced observer-based leader-following fault-tolerant consensus control, where the LPV control, observer and virtual actuator gains are computed simultaneously considering the LMIs conditions presented in(5.21). The main objectives of the multi-agent system are tracking the trajectories of a virtual leader and maintaining formation control, even in the presence of actuator faults. Figure 5.2 shows the model-based leader-following formation consensus control block diagram employed in the simulation results. Figure 5.2 provides a more detailed schematic compared to Figure 5.1. Here, we emphasize the reconfiguration of the control law using the virtual actuator, highlighted in gray. Additionally, actuator faults are considered in both additive and multiplicative forms.

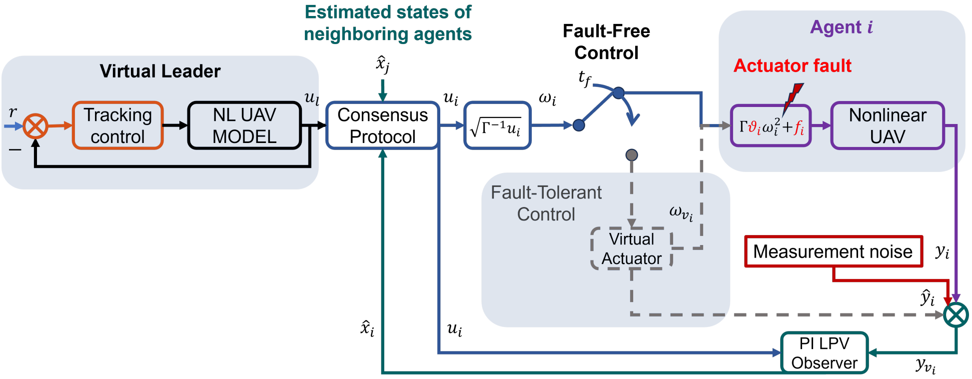


Figure 5.2: Model-based leader-following consensus control scheme for a team of UAVs.

To solve the LMI (5.21), as in Chapter 4, the quasi-LPV representation (4.28) of the nonlinear UAV (3.20) is considered, with $\mu_1 = 1$ and $\mu_2 = 70$, using MATLAB software and the SDPT3 toolbox [130].

The UAVs parameters are considered as in [131] for the quadcopter model DJI-F450. The parameters can be found in Table 5.1.

Table 5.1: Parameters of the DJI FlameWheel 450 quadcopter (Model: DJI-F450).

| Symbol | Parameter description | Value |
|---|---|---|
| $m$ | Quadcopter mass | $0.91\,\mathrm{kg}$ |
| $l$ | Arm length | $0.225\,\mathrm{m}$ |
| $g$ | Gravity | $9.81\,\mathrm{m/s^2}$ |
| $J_x$ | Inertial on X-axis | $0.0104\,\mathrm{kg\cdot m^2}$ |
| $J_y$ | Inertial on Y-axis | $0.0104\,\mathrm{kg\cdot m^2}$ |
| $J_z$ | Inertial on Z-axis | $0.0196\,\mathrm{kg\cdot m^2}$ |
| $b$ | Thrust coefficient | $8.3496\times10^{-6}\,\mathrm{N/(rad/s)^2}$ |
| $d$ | Drag coefficient | $1.52\times10^{-7}\,\mathrm{Nm/(rad/s)^2}$ |

The simulation conditions used in this section include: the initial conditions shown in Table 4.2, the consensus control law (4.29), the desired formation shown in Figure 4.2, the $H$ matrix (4.30), the matrices $\Lambda$ and $\mathcal{L}$ (4.31), and the trajectory tracking (4.32), as presented in Chapter 4. In the following subsections, three different cases are presented. First, a fault-free scenario is considered, similar to the results in Chapter 4 but incorporating the new LMI (5.21), where no actuator faults are present. Second, the faulty scenario is examined, which is essential for analyzing the impact of actuator faults on the multi-agent system. Finally, the fault-tolerant case is demonstrated, showing that the proposed approach effectively mitigates the effects of both additive and multiplicative actuator faults, compensating for the faults and restoring the multi-agent system to its desired positions.

### 5.4.1 Fault-Free Case

To demonstrate the effectiveness of the proposed control structure, considering the gains obtained from Theorem 3, the consensus control is initially evaluated under fault-free conditions. Sensor noise is introduced using a random function with a normal distribution, featuring a mean value of zero and a standard deviation of 0.0003. Figure 5.3 illustrates the trajectories of the quadcopters. The leader UAV is positioned at the center of all the follower UAVs, depicted in black, at the desired altitude. The followers successfully reach their designated positions at the vertices of the pentagon from their initial conditions defined in Table 4.2.

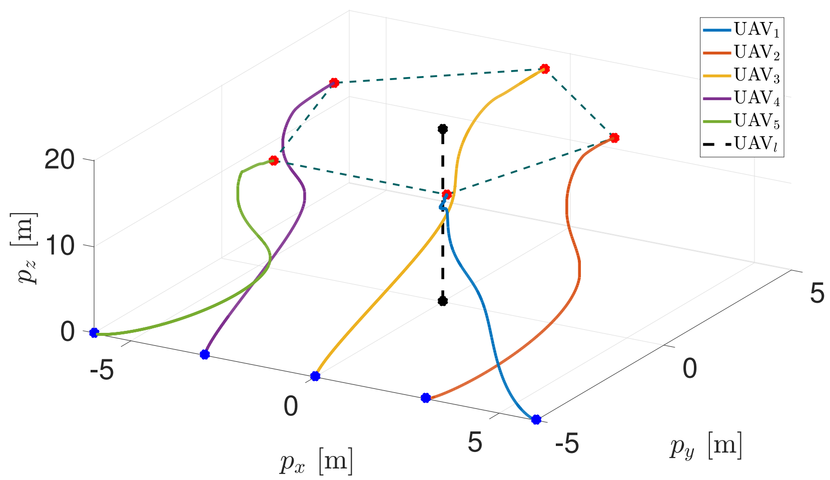


Figure 5.3: 3D trajectories of the quadcopters.

The control input of the UAV can be expressed in terms of the angular velocities of its rotors as follows:

$$u = \Gamma\Omega^2, \tag{5.32}$$

where $\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$ denotes the angular velocity vector of the rotors, with $\Omega_\varepsilon$ representing the angular velocity of the $\varepsilon$-th actuator. The relationship between $\Omega_\varepsilon$ and the control input is governed by the coefficient matrix $\Gamma$ [133], which encapsulates the drone's geometric and dynamic properties:

$$\Gamma = \begin{bmatrix} b & b & b & b \\ -bl & 0 & bl & 0 \\ 0 & -bl & 0 & bl \\ -d & d & -d & d \end{bmatrix}. \tag{5.33}$$

From (5.32), the angular velocities can be derived as:

$$\Omega = \sqrt{\Gamma^{-1}u}, \tag{5.34}$$

enabling the calculation of rotor speeds required to achieve the desired control efforts. The lift force generated by each rotor is given by $f_\varepsilon = b\Omega_\varepsilon^2$, linking the dynamics directly to the propulsion system. Although the real inputs of the UAV are voltage inputs. A current regulation allow to obtain the desired efforts necessary to follow the desired formation control and trajectory tracking objectives. Figure 5.4 illustrates the rotor's angular velocities for each UAV follower in the fault-free case. The motor velocities correspond to the effort applied, and these velocities stabilize once the UAV reaches its desired altitude.

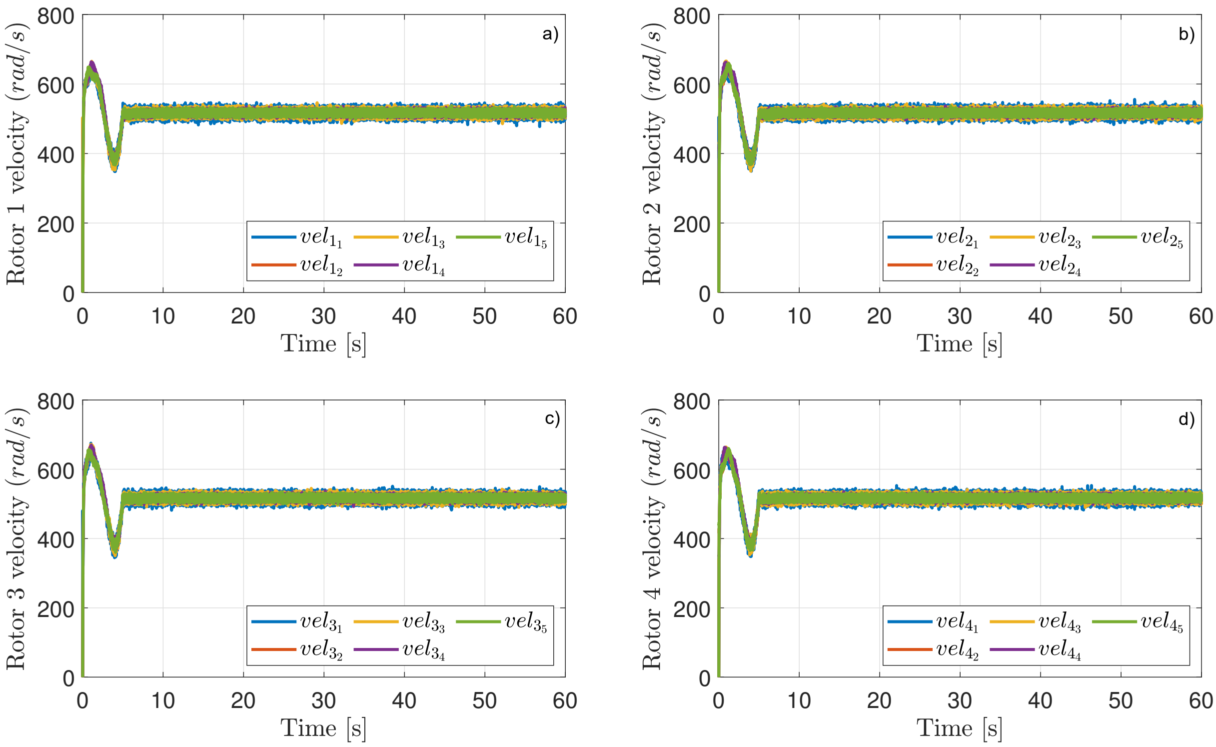


Figure 5.4: Rotor velocities of all follower UAVs.

Since the followers are in consensus, their velocities are similar. The green line, which represents UAV 5, appears more prominent because it was the last one plotted, not because its velocity is intentionally emphasized. However, a slight variation around 500rad/s is observed. This variation is due to minor adjustments made by the UAV to maintain its position in response to sensor noise, which requires the motors to constantly fine-tune their speeds. It is important to note that this approach effectively manages the level of sensor noise introduced. Let us define $\delta_i = \|x_i - x_l\| - r_a$ to measure the synchronization error between the leader UAV and each quadcopter follower. Figure 5.5 illustrates the evolution of $\delta_i$. When $\delta_i = 0$, it indicates that follower $i$ is following the virtual leader.

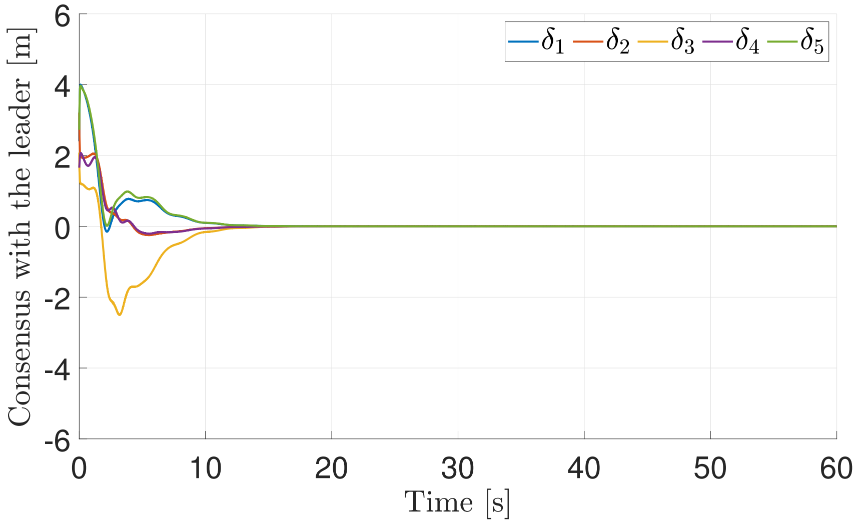


Figure 5.5: Synchronization error between the leader UAV and each quadcopter.

Let's define the relative distances as $d_{ij} = \|([x_i, y_i, z_i]^T - [x_j, y_j, z_j]^T) - (h_i - h_j)\|$. A value of $d_{ij} = 0$ indicates the desired formation has been achieved. Figure 5.6 depicts the performance of relative distances between the quadcopters in reaching the desired formation $H$.

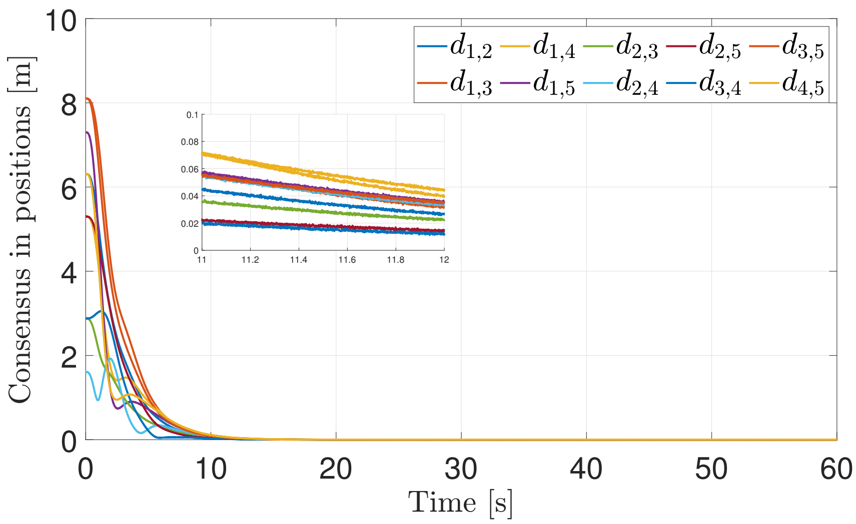


Figure 5.6: Relative distances between the quadcopters.

Based on Figures 5.5 and 5.6, it is notable that the team of UAVs is following the virtual leader at the desired pentagon vertex, achieving both control objectives, virtual leader tracking and formation control, and showing that consensus on both objectives is achieved within the first 15 s of simulation.

### 5.4.2 Faulty Case

In order to evaluate the behavior of the UAVs, with respect to the formation control objectives, in faulty situations, we consider in this subsection the following actuator faults. The first one occurs on $\mathrm{UAV}_4$ as an additive fault and is characterized by:

$$f_4(t) = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & 0 \leq t \leq 20\,\mathrm{s} \\ \begin{bmatrix} 0 \\ -100\,\mathrm{rad/s} \\ 0 \\ 0 \end{bmatrix}, & 20\,\mathrm{s} < t \leq 60\,\mathrm{s} \end{cases} \tag{5.35}$$

The second one occurs on $\mathrm{UAV}_3$ as a multiplicative fault and is such as:

$$\vartheta_3(t) = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & 0 \leq t \leq 35\,\mathrm{s} \\ \begin{bmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix}, & 35\,\mathrm{s} < t \leq 60\,\mathrm{s} \end{cases} \tag{5.36}$$

When the actuator faults (5.35) and (5.36) occur, as illustrated in Figure 5.7, it can be seen that the formation control objective is not achieved: the desired pentagon shape is not reached. In particular, the formation integrity is compromised, resulting in the follower UAVs deviating from the trajectory of the UAV leader.

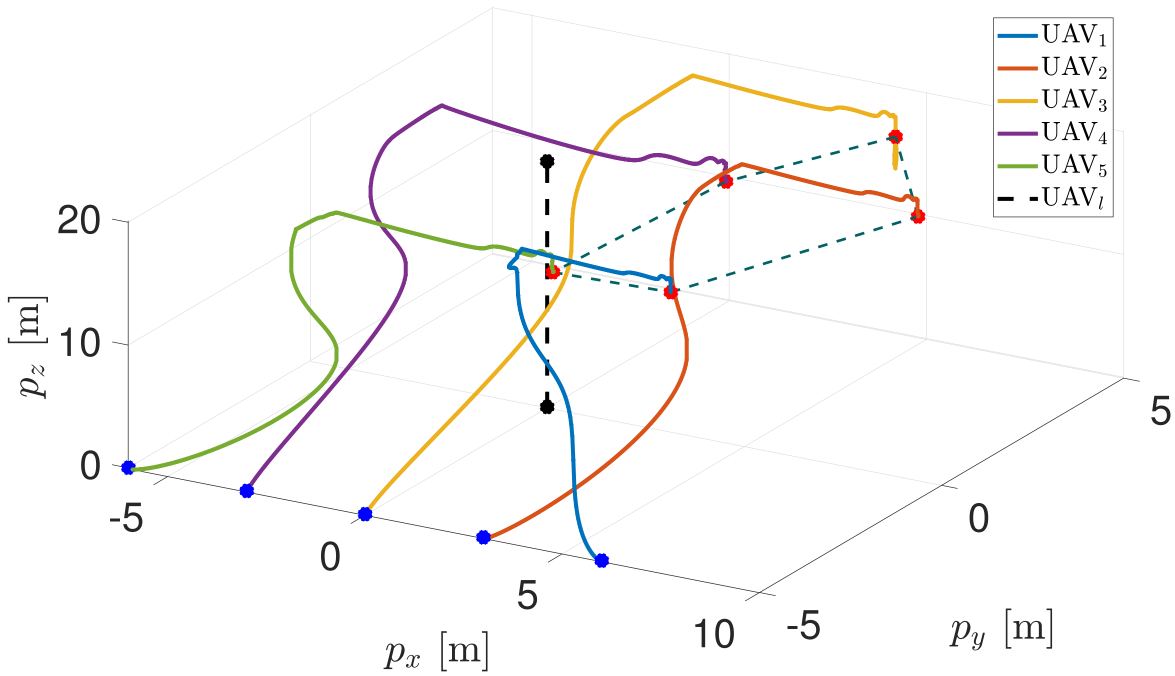


Figure 5.7: 3D trajectories of the quadcopters in faulty situation

Figure 5.8 shows the rotor velocities' behavior when faults (5.35) and (5.36) are introduced. It is evident that without a control action capable of compensating the additive and multiplicative faults, the multi-agent system diverges from its desired trajectory and loses synchronization with the virtual leader.

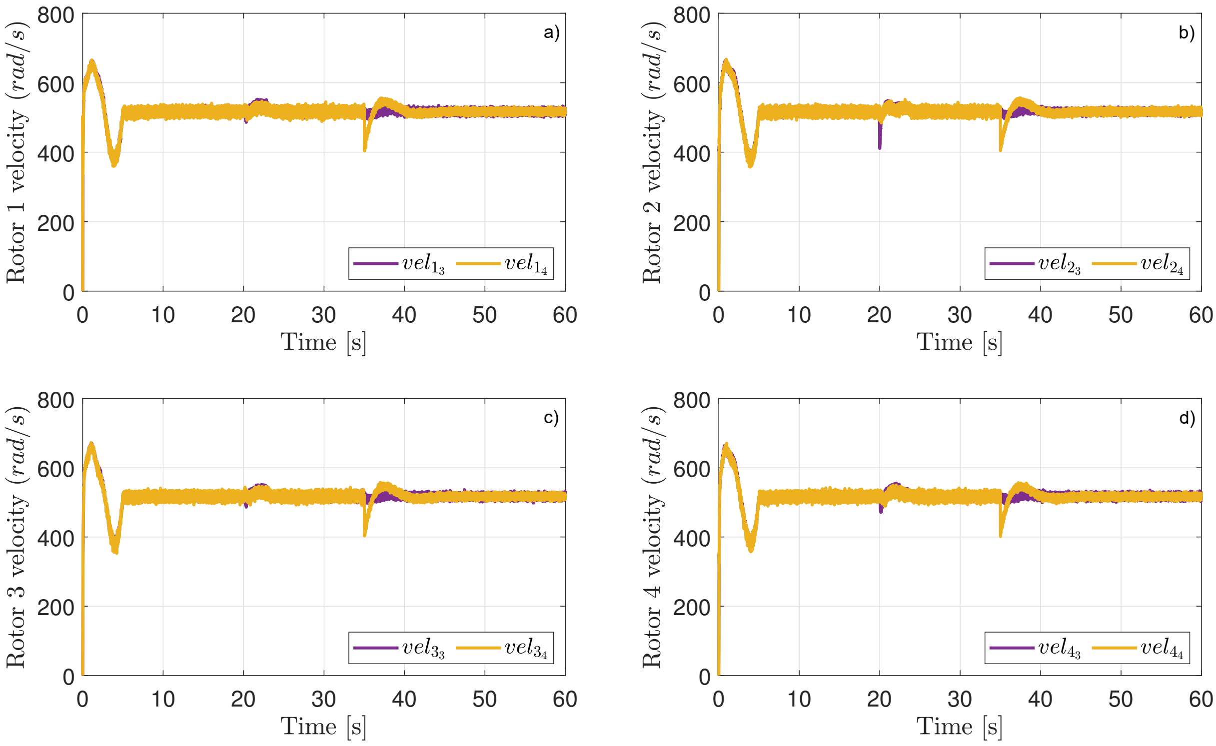


Figure 5.8: Rotor velocities of the faulty UAVs 3 & 4.

Consequently, all $\delta_i$ are different from 0 for all $i$ values after the fault occurences as shown on Figure 5.9. As a consequence of the additive fault at $t = 20\,\mathrm{s}$, it can be observed in Figure 5.7 that the follower UAVs move away from the leader, resulting in an increase in the synchronization error $\delta_i$. Subsequently, when the multiplicative fault (5.36) affects $\mathrm{UAV}_3$ at $t = 35\,\mathrm{s}$, it results in a 20% power loss in the rotors, causing the UAV to lose altitude. The impact of this fault is primarily observed in $\mathrm{UAV}_3$, which moves away from the leader. However, due to the interconnection in multi-agent systems, $\mathrm{UAV}_3$ attempts to return to the desired altitude but is unable to do so due to the presence of the fault.

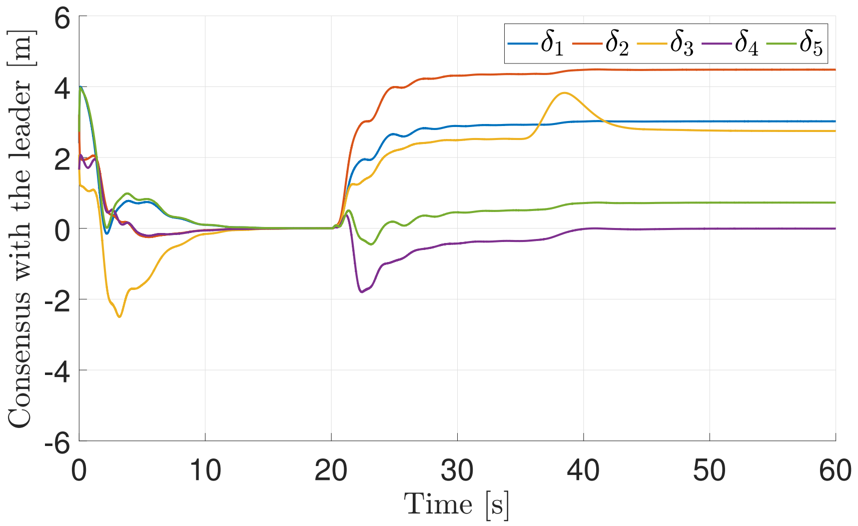


Figure 5.9: Synchronization error between the leader UAV and each quadcopter under actuator faults.

Figure 5.10 shows that the relative distances between the followers and their desired positions cannot be maintained, as indicated by $d_{ij} \neq 0$.

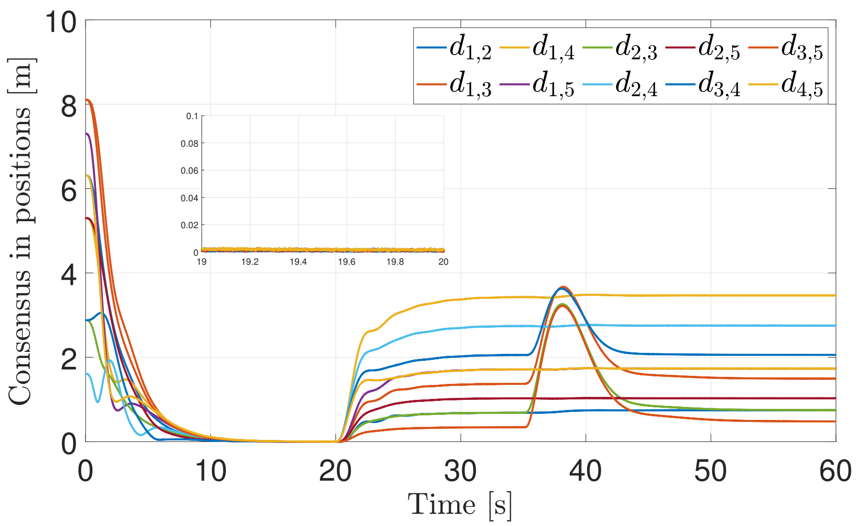


Figure 5.10: Relative distances between the quadcopters under actuator faults.

It is important to note that the fault magnitudes (5.35) and (5.36) were selected to induce faulty behavior in the follower UAVs without incurring instability. It is essential to emphasize that increasing fault magnitudes jeopardize the stability of the entire multi-agent system. The numerical example reveals that even a single faulty rotor on one UAV compromises the performance of all UAVs, preventing both formation control and leader tracking objectives.

### 5.4.3 Fault-Tolerant Control Case

As demonstrated in Section 5.4.2, the results indicate the necessity of implementing a FTC strategy, as depicted in Figure 5.2. This strategy integrates the utilization of a virtual actuator (5.14) to maintain the stability of the closed-loop control and to uphold the formation control objectives as closely as feasible. Since the fault information is assumed to be known a priori, a 5, s delay is intentionally introduced between the fault occurrences ((5.35), (5.36)) and the control reconfiguration. This delay value was chosen to clearly illustrate the fault's impact on the multi-agent system, as shorter delays might not sufficiently highlight the transient effects before the FTC strategy takes action. In real-world UAV applications, fault detection and reconfiguration times depend on sensor sampling rates, computational delays, and communication latency among agents. While some FTC systems can react faster, a 5, s delay represents a conservative yet realistic scenario where fault diagnosis, decision-making, and actuator reconfiguration require non-negligible time, especially in distributed systems with limited computational resources or intermittent communication. This choice ensures a balance between analytical clarity and practical feasibility in evaluating the FTC strategy's effectiveness. When a fault occurs in one agent, it affects the desired position of the entire multi-agent system. As a result, the whole team adjusts its movement in response to the fault, as illustrated in Figure 5.11. This disruption impacts the consensus objective, as the healthy UAVs begin to follow the affected agent due to communication exchanges and the interconnections among the followers. Consequently, both the leader-following consensus and the formation control objectives are compromised. However, the use of virtual actuators mitigates the effects of actuator faults, allowing the system to recover. The multi-agent system regains its desired formation and resumes tracking the leader quadcopter effectively.

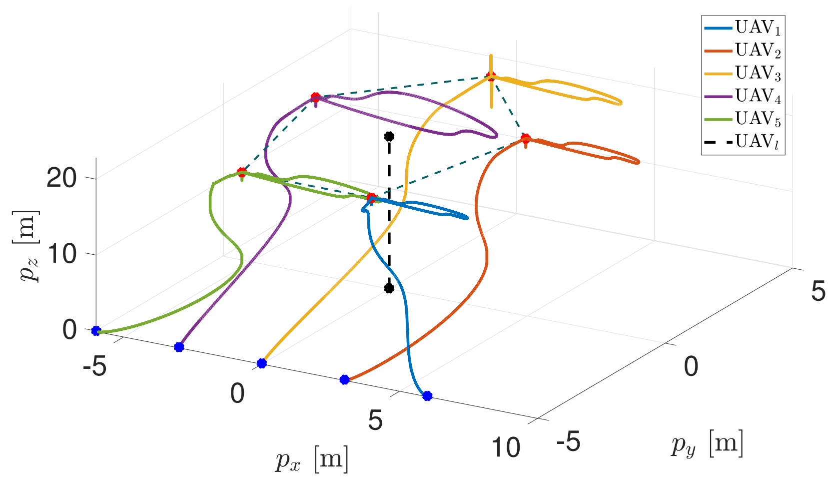


Figure 5.11: Trajectories of the quadcopters considering the FTC strategy.

As depicted in Figure 5.11, the occurrence of a fault in one agent disrupts the desired position of the entire system, prompting the team to shift accordingly. This shift happens because the healthy UAVs are influenced by the faulty agent through communication and interconnections. As a result, the consensus objective is affected, compromising both the leader-following consensus and the formation control. However, after the implementation of FTC multi-agent systems, the agents successfully reject the effects of the fault. The team returns to its desired positions, and the UAVs resume following the virtual leader. In Figure 5.12, it is possible to visualize the control action performed by the virtual actuators to compensate for the faults in agents 3 and 4.

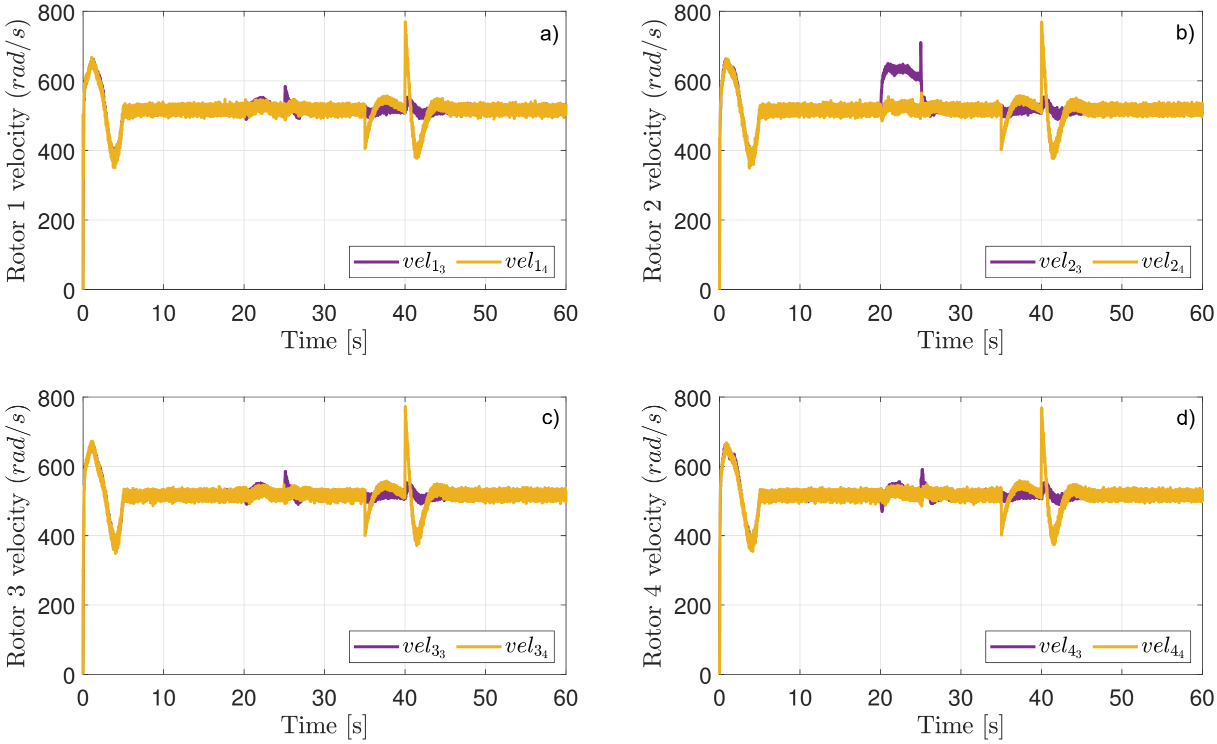


Figure 5.12: Rotor velocities of all follower UAVs considering the FTC strategy.

Figure 5.13 shows that for $t < 20\,\text{s}$, the multi-agent system operates without faults. At $t = 20\,\text{s}$, the additive fault (5.35) is introduced in $\text{UAV}_4$, causing the system to lose both control objectives: formation maintenance among followers and synchronization with the virtual leader. Subsequently, at $t = 25\,\text{s}$, the virtual actuator is implemented in $\text{UAV}_4$, which compensates for the fault effects. By $t = 32\,\text{s}$, both control objectives are recovered. At $t = 35\,\text{s}$, the second fault (5.36) is added, now affecting $\text{UAV}_3$. Notably, both errors are impacted but only with respect to $\text{UAV}_3$. Five seconds after fault occurrence ($t = 40\,\text{s}$), the virtual actuator is implemented in $\text{UAV}_3$, after the compensation action of the virtual actuators, $\delta_i$ returns to its desired value of zero for all followers, confirming that the effects of the faults were rejected.

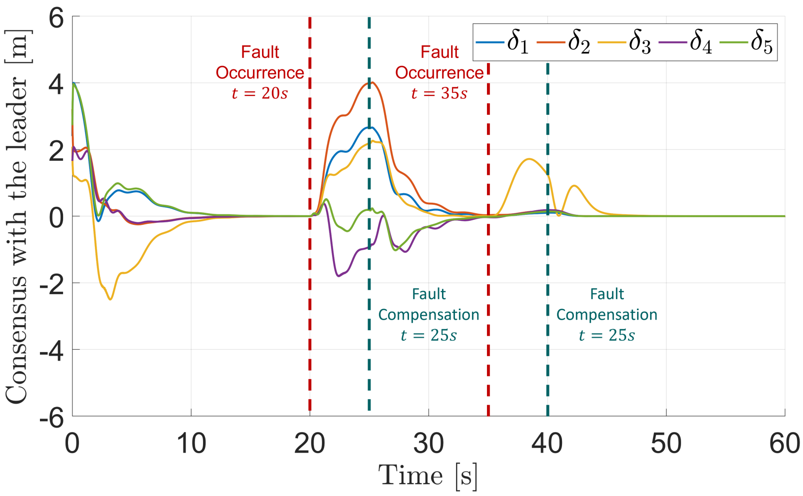


Figure 5.13: Synchronization error between the leader UAV and each quadcopter considering the FTC strategy.

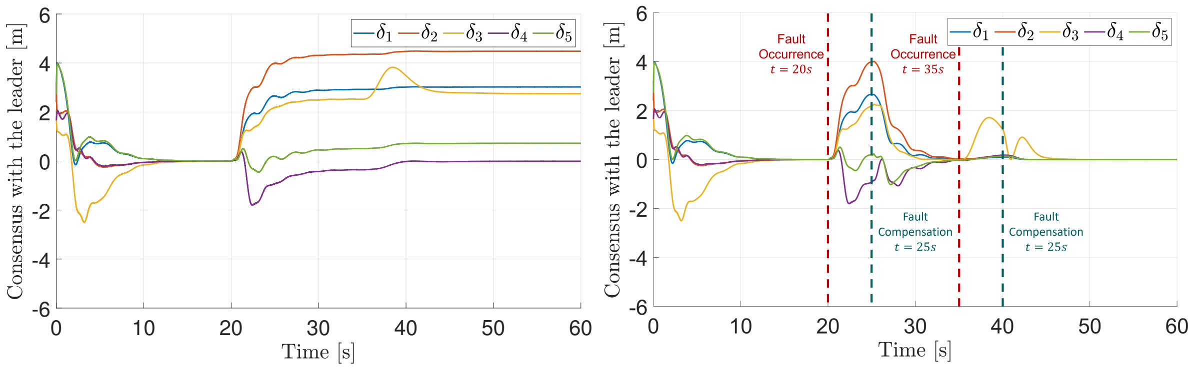


Figure 5.14: Comparison between figures 5.9 and 5.13.

The fault in one agent affects the desired position of the global multi-agent system. Then, all the team moves after the appearance of the fault. This is why the consensus objective is affected. All the healthy UAVs follow the affected one due to the communication exchange and the interconnection between the followers, compromising both objectives: the leader-following consensus and the formation control. A comparative analysis of Figures 5.9 and 5.13 is shown in Figure 5.14 demonstrates that our proposed FTC strategy effectively compensates for both fault types, as evidenced by all deviation signals $\delta$ converging to zero following fault compensation. The fault reject validates the effectiveness of our FTC approach. Figure 5.15 confirms that all quadcopters return to their desired positions at the vertices of the pentagon because $d_{ij} = 0$. The two transients in $\delta_3$ right after the virtual actuator activation ($t \approx 38$ s and $t \approx 41$ s) are due to the fact that once the multiplicative fault is compensated, the agent try to return to its desired position, but initially experiences an overshoot that increases the desired altitude, and then returns to its position.

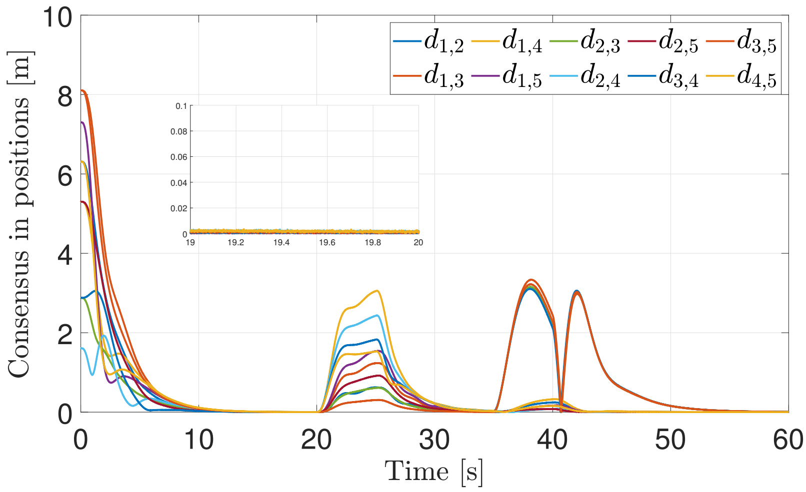


Figure 5.15: Relative distances between the quadcopters considering the FTC strategy.

Figure 5.15 demonstrates substantial performance improvement over Figure 5.10, with the proposed FTC scheme successfully achieving complete fault mitigation. All desired distance converge to zero following fault compensation, confirming the FTC effectiveness in fully restoring both the desired formation positions and accurate leader-follower tracking performance.

In this section, we have shown the performance of the virtual actuator in handling both additive and multiplicative faults, showcasing its effectiveness in rejecting both types of faults and enabling the multi-agent system to achieve both formation control and leader tracking objectives.

## 5.5 Conclusions

This chapter develops an observer-based leader-following fault-tolerant consensus control scheme for LPV multi-agent systems. Sufficient LMI conditions for the existence of admissible LPV controller, observer, and virtual actuator gains are derived using Polya's theorem on definite quadratic forms, reducing design conservatism. The effectiveness of this approach is validated through numerical simulations with a team of nonlinear quadcopters.

# Chapter 6

# General Conclusions and Perspectives

**Contents**



## 6.1 General Conclusions

This thesis has made significant contributions to the development of advanced control strategies for LPV multi-agent systems through the integration of observer-based consensus control methods. The research has systematically addressed key challenges in achieving robustness and fault tolerance, progressing from basic linear systems to more complex LPV formulations with fault compensation capabilities.

In Chapter 3 the development of a robust observer-based leader-following consensus control strategy for linear multi-agent systems was presented. This approach successfully addressed the dual challenges of external disturbances and sensor noise, ensuring reliable trajectory tracking of a virtual leader while maintaining precise consensus among follower agents.

Based on these results, Chapter 4 extend the framework to LPV multi-agent systems. This extension required novel solutions to handle parameter-varying dynamics while preserving the consensus properties. The successful implementation with UAV teams confirmed that the LPV observer-based controller could effectively manage the additional complexity introduced by time-varying parameters.

The main contribution is presented in Chapter 5, a fault-tolerant control strategy designed for LPV multi-agent systems. By incorporating a virtual actuator mechanism, the proposed solution demonstrated capability in maintaining formation control and leader-following objectives despite actuator faults. The simulation results shows that each UAV system could autonomously compensate for fault conditions without requiring complete reconfiguration.

## 6.2 Perspectives

While this thesis has advanced distributed fault-tolerant control for actuator faults in multi-agent systems, several important research directions emerge from this work. A natural extension would be to develop integrated fault diagnosis and tolerance schemes for sensor faults, which remain unaddressed in the current framework but are equally crucial for complete system reliability. The methodology could also be expanded to handle coupled fault scenarios, where simultaneous actuator and sensor faults occur, requiring new approaches to maintain cooperative control stability. Beyond fault scope expansion, practical implementation challenges warrant investigation, particularly how obstacle avoidance algorithms might interact with FTC strategies during fault conditions, ensuring safe operation in real-world environments. Communication resilience must also be examined, as packet loss, delays, or adversarial interference could significantly impact distributed FTC performance. Theoretical improvements could explore adaptive mechanisms for time-varying faults or incorporate machine learning techniques to enhance fault diagnosis accuracy. Finally, large-scale experimental validation with outdoor UAV swarms would demonstrate the approach's scalability and robustness under realistic operating conditions, bridging the gap between theoretical development and practical deployment.

# Bibliography

[1] Z. Yu, Y. Zhang, B. Jiang, J. Fu, and Y. Jin, "A review on fault-tolerant cooperative control of multiple unmanned aerial vehicles," *Chinese Journal of Aeronautics*, vol. 35, no. 1, pp. 1–18, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936121001771

[2] Z. Li and Z. Duan, *Cooperative control of multi-agent systems: a consensus region approach*. CRC press, 2017.

[3] M. Cucuzzella, S. Trip, C. De Persis, X. Cheng, A. Ferrara, and A. van der Schaft, "A robust consensus algorithm for current sharing and voltage regulation in dc microgrids," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1583–1595, 2019. [Online]. Available: https://doi.org/10.1109/TCST.2018.2834878

[4] J. M. Leon-Blanco, P. Gonzalez-R, J. L. Andrade-Pineda, D. Canca, and M. Calle, "A multi-agent approach to the truck multi-drone routing problem," *Expert Systems with Applications*, vol. 195, p. 116604, 2022. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.116604

[5] N. Joshi and A. Mehta, "Distributed discrete-time exponential sliding mode consensus protocol for discrete multi-agent system comprise of multiple robotic arms," *ISA Transactions*, 2024. [Online]. Available: https://doi.org/10.1016/j.isatra.2024.02.027

[6] L. Wang, D. Zhu, W. Pang, and Y. Zhang, "A survey of underwater search for multi-target using multi-auv: Task allocation, path planning, and formation control," *Ocean Engineering*, vol. 278, p. 114393, 2023. [Online]. Available: https://doi.org/10.1016/j.oceaneng.2023.114393

[7] B. Lindqvist, S. S. Mansouri, P. Sopasakis, and G. Nikolakopoulos, "Collision avoidance for multiple micro aerial vehicles using fast centralized nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9303–9309, 2020, 21st IFAC World Congress. [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.2384

[8] X. Zhang, T. Chen, and X. Liu, "Distributed bipartite output containment control of heterogeneous multi-agent systems with directed signed graphs," *International Journal of Systems Science*, vol. 54, no. 7, pp. 1588–1601, 2023. [Online]. Available: https://doi.org/10.1080/00207721.2023.2186193

[9] W. Chen, J. Hu, Z. Wu, and S. M. and, “A survey on fault detection for networked systems under communication constraints,” *Systems Science & Control Engineering*, vol. 13, no. 1, p. 2460434, 2025. [Online]. Available: https://doi.org/10.1080/21642583.2025.2460434

[10] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. Springer Science & Business Media, 3 2000. [Online]. Available: https://books.google.com/books/about/Issues_of_Fault_Diagnosis_for_Dynamic_Sy.html?hl=&id=7KyjIzr0XssC

[11] K. Ramachandran and J.-C. Juang, “Fault tolerant consensus of multi agent systems under actuator faults,” *IEEE Access*, vol. 10, pp. 9420–9430, 2022.

[12] S. Xiong and Z. Hou, “Data-driven formation control for unknown mimo nonlinear discrete-time multi-agent systems with sensor fault,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7728–7742, 2022.

[13] B. Farrera, F.-R. López-Estrada, M. Chadli, G. Valencia-Palomo, and S. Gómez-Peñate, “Distributed fault estimation of multi–agent systems using a proportional–integral observer: A leader–following application,” *International Journal of Applied Mathematics and Computer Science*, vol. 30, no. 3, pp. 551–560, 2020. [Online]. Available: https://doi.org/10.34768/amcs-2020-0040

[14] P. Gasga, M. Bernal, S. Gómez-Peñate, F. López-Estrada, G. Valencia-Palomo, and I. Santos-Ruiz, “Actuator multiplicative and additive simultaneous faults estimation using a qlpv proportional integral unknown input observer,” *International Journal of Adaptive Control and Signal Processing*, vol. 38, no. 8, pp. 2930–2950, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.3859

[15] S. Jadidi, H. Badihi, and Y. Zhang, “Passive fault-tolerant control strategies for power converter in a hybrid microgrid,” *Energies*, vol. 13, no. 21, 2020. [Online]. Available: 10.3390/en13215625

[16] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*. Springer Science & Business Media, 2012, vol. 3.

[17] M. M. Quadros, I. V. de Bessa, V. J. Leite, and R. M. Palhares, “Fault tolerant control for linear parameter varying systems: An improved robust virtual actuator and sensor approach,” *ISA Transactions*, vol. 104, pp. 356–369, 2020. [Online]. Available: https://doi.org/10.1016/j.isatra.2020.05.010

[18] N. M. Meysam Yadegar and A. Afshar, “Fault-tolerant control of linear systems using adaptive virtual actuator,” *International Journal of Control*, vol. 92, no. 8, pp. 1729–1741, 2019. [Online]. Available: 10.1080/00207179.2017.1408921

[19] Y. Wang, D. Rotondo, V. Puig, and G. Cembrano, “Fault-tolerant control based on virtual actuator and sensor for discrete-time descriptor systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 5316–5325, Dec 2020. [Online]. Available: 10.1109/TCSI.2020.3015887

[20] Y.-H. Ma, X. Du, X.-M. Sun, and F.-J. Zhao, “Active fault tolerant tracking control of turbofan engine based on virtual actuator,” *ISA Transactions*, vol. 122, pp. 247–259, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057821002263

[21] D. Rotondo, F. Nejjari, and V. Puig, “A virtual actuator and sensor approach for fault tolerant control of lpv systems,” *Journal of Process Control*, vol. 24, no. 3, pp. 203–222, 2014. [Online]. Available: https://doi.org/10.1016/j.jprocont.2013.12.016

[22] Y. Li and C. T. and, “A survey of the consensus for multi-agent systems,” *Systems Science & Control Engineering*, vol. 7, no. 1, pp. 468–482, 2019. [Online]. Available: https://doi.org/10.1080/21642583.2019.1695689

[23] J. A. Vazquez Trejo, D. Theilliol, M. Adam Medina, C. D. García Beltrán, and M. Witczak, “Leader-following formation control for networked multi-agent systems under communication faults/failures,” in *Advances in Diagnostics of Processes and Systems*. Cham: Springer International Publishing, 2021, pp. 45–57. [Online]. Available: https://doi.org/10.1007/978-3-030-58964-6_4

[24] J. Orr and A. Dutta, “Multi-agent deep reinforcement learning for multi-robot applications: A survey,” *Sensors*, vol. 23, no. 7, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/7/3625

[25] M. V. Muntean, “Multi-agent system for intelligent urban traffic management using wireless sensor networks data,” *Sensors*, vol. 22, no. 1, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/1/208

[26] K. Griparic, M. Polic, M. Krizmancic, and S. Bogdan, “Consensus-based distributed connectivity control in multi-agent systems,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1264–1281, 2022.

[27] A. Shavandi and M. Khedmati, “A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets,” *Expert Systems with Applications*, vol. 208, p. 118124, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422013082

[28] M. Herrera, M. Pérez-Hernández, A. Kumar Parlikad, and J. Izquierdo, “Multi-agent systems and complex networks: Review and applications in systems engineering,” *Processes*, vol. 8, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2227-9717/8/3/312

[29] J. A. V. Trejo, M. Chadli, D. Rotondo, M. A. Medina, and D. Theilliol, “Event-triggered fault-tolerant leader-following control for homogeneous multi-agent systems,” in *11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Safeprocess 2022*, 2022.

[30] G. Bao, L. Ma, and X. Yi, “Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: a survey,” *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 539–551, 2022.

[31] O. Skeik and A. Lanzon, "Robust output consensus of homogeneous multi-agent systems with negative imaginary dynamics," *Automatica*, vol. 113, p. 108799, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109819306624

[32] A. Saberi, A. A. Stoorvogel, M. Zhang, and P. Sannuti, *Synchronization of Multi-Agent Systems in the Presence of Disturbances and Delays*. Springer Nature, 2022.

[33] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[34] R. Owoputi and S. Ray, "Security of multi-agent cyber-physical systems: A survey," *IEEE Access*, vol. 10, pp. 121 465–121 479, 2022.

[35] M. Shamrooz Aslam, H. Bilal, W.-J. Chang, A. Yahya, I. Anjum Badruddin, S. Kamangar, and M. Hussien, "Formation control of heterogeneous multi-agent systems under fixed and switching hierarchies," *IEEE Access*, vol. 12, pp. 97 868–97 882, 2024.

[36] D. G. Lui, A. Petrillo, and S. Santini, "Adaptive fault-tolerant containment control for heterogeneous uncertain nonlinear multi-agent systems under actuator faults," *Journal of the Franklin Institute*, vol. 361, no. 17, p. 107317, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003224007385

[37] S. Liu, B. Jiang, Z. Mao, and Y. Ma, "Adaptive fault-tolerant formation control of heterogeneous multi-agent systems under directed communication topology," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6212

[38] J. Gong, Y. Ma, B. Jiang, and Z. Mao, "Distributed adaptive fault-tolerant formation control for heterogeneous multiagent systems under switching directed topologies," *Journal of the Franklin Institute*, vol. 359, no. 8, pp. 3366–3388, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003222002174

[39] X. Zhang, Q. Yang, F. Xiao, H. Fang, and J. Chen, "Linear formation control of multi-agent systems," *Automatica*, vol. 171, p. 111935, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109824004291

[40] L. Martinović, Žarko Zečević, and B. Krstajić, "Cooperative tracking control of single-integrator multi-agent systems with multiple leaders," *European Journal of Control*, vol. 63, pp. 232–239, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0947358021001308

[41] W. Hu, Y. Cheng, and Z. Chen, "Reset control for consensus of double-integrator multi-agent systems," *Automatica*, vol. 136, p. 110057, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109821005859

[42] Z. Deng and T. Chen, "Distributed algorithm design for constrained resource allocation problems with high-order multi-agent systems," *Automatica*, vol. 144, p. 110492, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000510982200351X

[43] P. S. P. Pessim, P. H. S. Coutinho, M. J. Lacerda, V. Puig, and R. M. Palhares, "A distributed gain-scheduled observer-based approach for the practical consensus of lpv multi-agent systems," *Automatica*, vol. 177, p. 112353, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109825002468

[44] Y. Hu and C. Lin, "Adaptive event-triggered dynamic output feedback $\mathcal{H}_\infty$ control for networked t-s fuzzy systems," *Systems Science & Control Engineering*, vol. 9, no. sup2, pp. 38–47, 2021. [Online]. Available: https://doi.org/10.1080/21642583.2020.1836527

[45] T. Guo, Y. Liu, and H.-X. Hu, "Adaptive connectivity-preserving formation control with a dynamic event-triggering mechanism," *Journal of the Franklin Institute*, vol. 362, no. 1, p. 107331, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001600322400752X

[46] R. K. Mishra and H. Ishii, "Event-triggered control for discrete-time multi-agent average consensus," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 1, pp. 159–176, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.5815

[47] F. Gulzar, N. Muhammad, Y. Awais, and A. Iqbal, "Constraint-oriented formation control of multi-robot system in leaderless consensus under confined conditions," *Systems Science & Control Engineering*, vol. 12, no. 1, p. 2436666, 2024. [Online]. Available: https://doi.org/10.1080/21642583.2024.2436666

[48] J. Yang and B. G. Lee, "Distributed adaptive tracking control of hidden leader-follower multi-agent systems with unknown parameters," *Mathematics*, vol. 12, no. 7, 2024. [Online]. Available: https://www.mdpi.com/2227-7390/12/7/1013

[49] L.-A. Galicia-Galicia, O. Hernández-González, C. D. Garcia-Beltran, G. Valencia-Palomo, and M.-E. Guerrero-Sánchez, "Distributed observer for linear systems with multirate sampled outputs involving multiple delays," *Mathematics*, vol. 12, no. 18, 2024. [Online]. Available: https://www.mdpi.com/2227-7390/12/18/2943

[50] M. Ban, J. Li, H. Deng, Y. Wang, and J. Cheng, "Dynamic event-triggered $\mathcal{H}_\infty$ leader-following consensus for nonlinear stochastic multi-agent systems with semi-markov switching topologies and application," *Journal of the Franklin Institute*, vol. 362, no. 6, p. 107596, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003225000900

[51] J.-N. Li, X. Liu, X.-F. Ru, and X. Xu, "Disturbance rejection adaptive fault-tolerant constrained consensus for multi-agent systems with failures," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3302–3306, 2020.

[52] L. Zhao and G.-H. Yang, "Cooperative adaptive fault-tolerant control for multi-agent systems with deception attacks," *Journal of the Franklin Institute*, vol. 357, no. 6, pp. 3419–3433, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003219309457

[53] G. M. Atınç, D. M. Stipanović, and P. G. Voulgaris, "A swarm-based approach to dynamic coverage control of multi-agent systems," *Automatica*, vol. 112, p. 108637, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109819304984

[54] T. Liang, Y. Lin, L. Shi, J. Li, Y. Zhang, and Y. Qian, "Distributed vehicle tracking in wireless sensor network: A fully decentralized multiagent reinforcement learning approach," *IEEE Sensors Letters*, vol. 5, no. 1, pp. 1–4, 2021.

[55] S. Dai, S. Li, H. Tang, X. Ning, F. Fang, Y. Fu, Q. Wang, and L. Cheng, "Marp: A cooperative multi-agent drl system for connected autonomous vehicle platooning," *IEEE Internet of Things Journal*, pp. 1–1, 2024.

[56] O. P. Mahela, M. Khosravy, N. Gupta, B. Khan, H. H. Alhelou, R. Mahla, N. Patel, and P. Siano, "Comprehensive overview of multi-agent systems for controlling smart grids," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 115–131, 2022.

[57] Y. Guan, S. Zou, H. Peng, W. Ni, Y. Sun, and H. Gao, "Cooperative uav trajectory design for disaster area emergency communications: A multiagent ppo method," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8848–8859, 2024.

[58] S. Wang, N. Niu, Z. Wang, Y. Lv, and J. Zhang, "Heterogeneous multi-agent task planning method in complex marine environment," *IEEE Access*, pp. 1–1, 2025.

[59] M. Rehan, C. K. Ahn, and M. Chadli, "Consensus of one-sided lipschitz multi-agents under input saturation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 4, pp. 745–749, 2019.

[60] R. Mu, A. Wei, H. Li, and Z.-M. Wang, "Distributed adaptive fault-tolerant consensus control for multi-agent systems with event-triggered communication," *International Journal of Systems Scie11nce*, vol. 54, no. 4, pp. 880–894, 2023. [Online]. Available: https://doi.org/10.1080/00207721.2022.2148494

[61] G. Difilippo, M. P. Fanti, and A. M. Mangini, "Maximizing convergence speed for second order consensus in leaderless multi-agent systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 259–269, February 2022. [Online]. Available: https://doi.org/10.1109/JAS.2021.1004320

[62] D. Wang, Y. Shen, and Z. Fang, "Leader–follower consensus for nonlinear multi-agent systems with unknown measurement sensitivities," *International Journal of Systems Science*, vol. 54, no. 7, pp. 1575–1587, 2023. [Online]. Available: https://doi.org/10.1080/00207721.2023.2186192

[63] P. Zhang, Q. Wang, Y. Wang, J. Tang, and D. Duan, "Agent-based finite-time leader-following consensus for earth-observation tasks on multiple airships," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1924–1931, May 2022. [Online]. Available: https://doi.org/10.1109/TNSE.2022.3154829

[64] X. Yan, D. Jiang, R. Miao, and Y. Li, "Formation control and obstacle avoidance algorithm of a multi-usv system based on virtual structure and artificial potential field," *Journal of Marine Science and Engineering*, vol. 9, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/2/161

[65] J. Huang, M. Cao, N. Zhou, Q. Yang, and X. Bai, "Distributed behavioral control for second-order nonlinear multi-agent systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2445–2450, 2017, 20th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317307565

[66] X. Liu, S. S. Ge, and C.-H. Goh, "Formation potential field for trajectory tracking control of multi-agents in constrained space," *International Journal of Control*, vol. 90, no. 10, pp. 2137–2151, 2017.

[67] D. Li, S. S. Ge, W. He, G. Ma, and L. Xie, "Multilayer formation control of multi-agent systems," *Automatica*, vol. 109, p. 108558, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109819304194

[68] S. Yazdani and M. Haeri, "Flocking of multi-agent systems with multiple second-order uncoupled linear dynamics and virtual leader," *IET Control Theory & Applications*, vol. 10, no. 8, pp. 853–860, 2016.

[69] J. A. Vazquez Trejo, M. Adam-Medina, C. D. Garcia-Beltran, G. V. G. Ramírez, B. Y. López-Zapata, E.-M. Sanchez-Coronado, and D. Theilliol, "Robust formation control based on leader-following consensus in multi-agent systems with faults in the information exchange: Application in a fleet of unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 104 940–104 949, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3098303

[70] J. A. V. Trejo, J.-C. Ponsart, M. Adam-Medina, G. Valencia-Palomo, and D. Theilliol, "Distributed observer-based leader-following consensus control for lpv multi-agent systems: Application to multiple vtol-uavs formation control," *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1316–1323, June 2023. [Online]. Available: https://doi.org/10.1109/ICUAS57906.2023.10156012

[71] M. Wooldridge, *An introduction to multiagent systems.* John wiley & sons, 2009.

[72] D. Maldonado, E. Cruz, J. Abad Torres, P. J. Cruz, and S. d. P. Gamboa Benitez, "Multi-agent systems: A survey about its components, framework and workflow," *IEEE Access*, vol. 12, pp. 80 950–80 975, 2024.

[73] A. Mostaani, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Task-effective compression of observations for the centralized control of a multiagent system over bit-budgeted channels," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6131–6143, Feb 2024.

[74] D. F. S. De Sá and J. V. D. F. Neto, "Multi-agent collision avoidance system based on centralization and decentralization control for uav applications," *IEEE Access*, vol. 11, pp. 7031–7042, 2023.

[75] W. Wu, L. Fu, and H. Li, "Distributed adaptive control of nonlinear multi-agent systems-based controller dynamic linearization method," *IEEE Access*, vol. 13, pp. 64 796–64 809, 2025.

[76] J. Shi, X. Chen, S. Xing, A. Liu, and C. Chen, "Robust cooperative fault-tolerant control for uncertain multi-agent systems subject to actuator faults," *Sensors*, vol. 24, no. 8, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/8/2651

[77] M.-M. Duan, C.-L. Liu, and F. Liu, "Event-triggered consensus seeking of heterogeneous first-order agents with input delay," *IEEE Access*, vol. 5, pp. 5215–5223, 2017.

[78] W. Zou and Z. Xiang, "Event-triggered leader–following consensus of non-linear multi-agent systems with switched dynamics," *IET Control Theory & Applications*, vol. 13, no. 9, pp. 1222–1228, 2019. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cta.2018.5126

[79] Z. Yu, Y. Zhang, B. Jiang, J. Fu, Y. Jin, and T. Chai, "Composite adaptive disturbance observer-based decentralized fractional-order fault-tolerant control of networked uavs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 799–813, 2022.

[80] F. Ho, R. Geraldes, A. Gonçalves, B. Rigault, B. Sportich, D. Kubo, M. Cavazza, and H. Prendinger, "Decentralized multi-agent path finding for uav traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 997–1008, 2022.

[81] S. Khodabandeh, H. Kharrati, and F. Hashemzadeh, "Control for leader–follower consensus of multi-agent systems with actuator faults using decentralized robust fault-tolerant control," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, pp. 529–541, 2021.

[82] Z. Yu, Y. Zhang, B. Jiang, C.-Y. Su, J. Fu, Y. Jin, and T. Chai, "Decentralized fractional-order backstepping fault-tolerant control of multi-uavs against actuator faults and wind effects," *Aerospace Science and Technology*, vol. 104, p. 105939, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963820306210

[83] Q. Zhang, K.-X. Yan, B. Xiao, and T. Han, "Distributed observer-based adaptive formation-containment tracking for multi-agent systems on directed graphs," *Mathematics*, vol. 13, no. 4, 2025. [Online]. Available: https://www.mdpi.com/2227-7390/13/4/558

[84] K. Chen, L. Xiong, H. Yan, K. Shi, and H. Zhang, "Distributed consensus control based on double event-triggered mechanism for multi-agent systems with irregular switching topologies under malicious attacks," *Journal of the Franklin Institute*, vol. 361, no. 15, p. 107075, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003224004964

[85] X. Li, Y. Tang, and H. R. Karimi, "Consensus of multi-agent systems via fully distributed event-triggered control," *Automatica*, vol. 116, p. 108898, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109820300960

[86] B. Kada, M. Khalid, and M. S. Shaikh, "Distributed cooperative control of autonomous multi-agent uav systems using smooth control," *Journal of Systems Engineering and Electronics*, vol. 31, no. 6, pp. 1297–1307, 2020.

[87] Z. Wang and M. Chadli, "Distributed event-triggered fault estimation for takagi–sugeno multi-agent systems with unmeasurable decision variables," *Journal of the Franklin Institute*, vol. 362, no. 8, p. 107689, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003225001826

[88] M. Herrera, M. Pérez-Hernández, A. Kumar Parlikad, and J. Izquierdo, "Multi-agent systems and complex networks: Review and applications in systems engineering," *Processes*, vol. 8, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2227-9717/8/3/312

[89] Z. Li, M. Z. Chen, and Z. Ding, "Distributed adaptive controllers for cooperative output regulation of heterogeneous agents over directed graphs," *Automatica*, vol. 68, pp. 179–183, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109816300231

[90] H. Chen, M. He, W. Han, S. Liu, and C. Wei, "A discrete-time fractional-order flocking control algorithm of multi-agent systems," *Fractal and Fractional*, vol. 8, no. 2, 2024. [Online]. Available: https://www.mdpi.com/2504-3110/8/2/85

[91] X. Li, Z. Sun, Y. Tang, and H. R. Karimi, "Adaptive event-triggered consensus of multiagent systems on directed graphs," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1670–1685, 2021.

[92] L.-B. W. Guo-Fei Cui and M. Wu, "Adaptive event-triggered fault-tolerant control for leader-following consensus of multi-agent systems," *International Journal of Systems Science*, vol. 55, no. 15, pp. 3291–3303, 2024. [Online]. Available: https://doi.org/10.1080/00207721.2024.2367074

[93] Y. Sun, Y. Xia, J. Zhang, and D.-W. Ding, "Adaptive fault-tolerant output regulation of linear multi-agent systems with sensor faults," *IEEE Access*, vol. 8, pp. 159 440–159 448, 2020.

[94] T. Elmokadem, "Distributed coverage control of quadrotor multi-uav systems for precision agriculture," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 251–256, 2019, 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319324498

[95] D. Paez, J. P. Romero, B. Noriega, G. A. Cardona, and J. M. Calderon, "Distributed particle swarm optimization for multi-robot system in search and rescue operations," *IFAC-PapersOnLine*, vol. 54, no. 4, pp. 1–6, 2021, 4th IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control CESCIT 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896321014038

[96] G. Wang, F. Wei, Y. Jiang, M. Zhao, K. Wang, and H. Qi, "A multi-auv maritime target search method for moving and invisible objects based on multi-agent deep reinforcement learning," *Sensors*, vol. 22, no. 21, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8562

[97] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Fault-tolerant cooperative navigation of networked uav swarms for forest fire monitoring," *Aerospace Science and Technology*, vol. 123, p. 107494, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963822001687

[98] J. Wang, L. Han, X. Li, X. Dong, Q. Li, and Z. Ren, "Time-varying formation of second-order discrete-time multi-agent systems under non-uniform communication delays and switching topology with application to uav formation flying," *IET Control Theory & Applications*, vol. 14, no. 14, pp. 1947–1956, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cta.2020.0183

[99] A. Belkadi, L. Ciarletta, and D. Theilliol, "Particle swarm optimization method for the control of a fleet of unmanned aerial vehicles." *Journal of Physics: Conference Series*, vol. 659, p. 012015, 2015.

[100] J. A. V. Trejo, D. Rotondo, M. Adam-Medina, and D. Theilliol, "Observer-based event-triggered model reference control for multi-agent systems," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 421–428.

[101] J. A. V. Trejo, M. Adam-Medina, C. D. Garcia-Beltran, G. V. G. Ramírez, B. yolanda lópez Zapata, E.-M. Sanchez-Coronado, and D. Theilliol, "Robust formation control based on leader-following consensus in multi-agent systems with faults in the information exchange: Application in a fleet of unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 104 940–104 949, 2021.

[102] X. Tian, H. Liu, and H. Liu, "Robust finite-time consensus control for multi-agent systems with disturbances and unknown velocities," *ISA transactions*, vol. 80, pp. 73–80, 2018.

[103] D. V. Redrovan and D. Kim, "Multiple quadrotors flight formation control based on sliding mode control and trajectory tracking," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2018, pp. 1–6.

[104] Z. Sun, G. Zhang, Y. Lu, and W. Zhang, "Leader-follower formation control of underactuated surface vehicles based on sliding mode control and parameter estimation," *ISA transactions*, vol. 72, pp. 15–24, 2018.

[105] Z. Ahmed, M. M. Khan, M. A. Saeed, and W. Zhang, "Consensus control of multi-agent systems with input and communication delay: A frequency domain perspective," *ISA transactions*, vol. 101, pp. 69–77, 2020.

[106] W. Lin, W. Zhao, and H. Liu, "Robust optimal formation control of heterogeneous multi-agent system via reinforcement learning," *IEEE Access*, vol. 8, pp. 218 424–218 432, 2020.

[107] H. Xu, H. Su, Q. Wang, and C. Xu, "Semi-global adaptive bipartite output consensus of multi-agent systems subject to input saturation and external disturbance under switching network," *International Journal of Control, Automation and Systems*, vol. 19, no. 9, pp. 3037–3048, 2021.

[108] F. Zhu and C. Tan, "Consensus control of linear parameter-varying multi-agent systems with unknown inputs," *Sensors*, vol. 23, no. 11, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/11/5125

[109] J. Chen, W. Zhang, Y.-Y. Cao, and H. Chu, "Observer-based consensus control against actuator faults for linear parameter-varying multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1336–1347, July 2017. [Online]. Available: https://doi.org/10.1109/TSMC.2016.2587300

[110] D. Rotondo, J.-C. Ponsart, and D. Theilliol, "Gain-scheduled observer-based consensus for linear parameter varying multi-agent systems," *Automatica*, vol. 135, p. 109979, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109821005057

[111] A. Sala and C. Ariño, "Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications of polya's theorem," *Fuzzy Sets and Systems*, vol. 158, no. 24, pp. 2671–2686, 2007, theme: Modelling and Control. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165011407003284

[112] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[113] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory.* SIAM, 1994.

[114] B. Zhou, W. Wang, and H. Ye, "Cooperative control for consensus of multi-agent systems with actuator faults," *Computers & Electrical Engineering*, vol. 40, no. 7, pp. 2154–2166, 2014. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2014.04.015

[115] J. Wang and J. Zhang, "Active fault-tolerant control research of multi-agent system," in *Journal of Physics: Conference Series*, vol. 1754, no. 1. IOP Publishing, 2021, p. 012088.

[116] M. Chadli, M. Davoodi, and N. Meskin, "Distributed fault detection and isolation filter design for heterogeneous multi-agent lpv systems," *2017 American Control Conference (ACC)*, pp. 1610–1615, 2017. [Online]. Available: https://doi.org/10.23919/ACC.2017.7963183

[117] J. A. V. Trejo, D. Rotondo, M. A. Medina, and D. Theilliol, "Observer-based fault-tolerant leader-following control for multi-agent systems," *2021 European Control Conference (ECC)*, pp. 2285–2290, June 2021. [Online]. Available: https://doi.org/10.23919/ECC54610.2021.9655148

[118] J. A. Vazquez Trejo, A. Guenard, M. Adam-Medina, J.-C. Ponsart, L. Ciarletta, D. Rotondo, and D. Theilliol, "Event-triggered leader-following formation control for multi-agent systems under communication faults: Application to a fleet of unmanned aerial vehicles," *Journal of Systems Engineering and Electronics*, vol. 32, no. 5, pp. 1014–1022, Oct 2021. [Online]. Available: https://doi.org/10.23919/JSEE.2021.000086

[119] D. Rotondo, D. Theilliol, and J.-C. Ponsart, "Virtual actuator and sensor fault tolerant consensus for homogeneous linear multi-agent systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 3, pp. 1324–1334, March 2024. [Online]. Available: https://doi.org/10.1109/TCSI.2023.3340341

[120] M. H. Asemani and V. J. Majd, "A robust $\mathcal{H}_\infty$ observer-based controller design for uncertain t–s fuzzy systems with unknown premise variables via lmi," *Fuzzy Sets and Systems*, vol. 212, pp. 21–40, 2013, theme: Control Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165011412003089

[121] I. Sadeghzadeh, A. Chamseddine, D. Theilliol, and Y. Zhang, "Linear parameter varying control synthesis: State feedback versus $h_\infty$ technique with application to quadrotor ua v," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 1099–1104. [Online]. Available: 10.1109/ICUAS.2014.6842362

[122] G. Ortiz-Torres, P. Castillo, F. D. J. Sorcia-Vázquez, J. Y. Rumbo-Morales, J. A. Brizuela-Mendoza, J. De La Cruz-Soto, and M. Martínez-García, "Fault estimation and fault tolerant control strategies applied to vtol aerial vehicles with soft and

aggressive actuator faults," *IEEE Access*, vol. 8, pp. 10 649–10 661, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2963693

[123] G. Ortiz-Torres, P. Castillo, and J. Reyes-Reyes, "An actuator fault tolerant control for vtol vehicles using fault estimation observers: Practical validation," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1054–1062.

[124] H. Yang, Z. Zhang, and S. Zhang, "Consensus of second-order multi-agent systems with exogenous disturbances," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 9, pp. 945–956, 2011.

[125] J.-F. Guerrero-Castellanos, A. Vega-Alonzo, N. Marchand, S. Durand, J. Linares-Flores, and G. Mino-Aguilar, "Real-time event-based formation control of a group of vtol-uavs," in *2017 3rd International Conference on Event-Based Control, Communication and Signal Processing (EBCCSP)*. IEEE, 2017, pp. 1–8.

[126] A. Rodríguez-Mata, G. Flores, A. Martínez-Vásquez, Z. Mora-Felix, R. Castro-Linares, and L. Amabilis-Sosa, "Discontinuous high-gain observer in a robust control uav quadrotor: Real-time application for watershed monitoring," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[127] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[128] X.-D. Sun and I. Postlethwaite, "Affine lpv modelling and its use in gain-scheduled helicopter control," in *UKACC International Conference on Control '98 (Conf. Publ. No. 455)*, 1998, pp. 1504–1509 vol.2.

[129] H. Zaki, M. Unel, and Y. Yildiz, "Trajectory control of a quadrotor using a control allocation approach," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 533–539.

[130] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "Sdpt3—a matlab software package for semidefinite programming, version 1.3," *Optimization methods and software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[131] C. Toapanta, J. Villafuerte, and P. J. Cruz, "3dof multi-rotor experimental testbed for teaching control systems," *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–6, Oct 2018. [Online]. Available: https://doi.org/10.1109/ETCM.2018.8580337

[132] P. Gasga, M. Bernal, S. Gómez-Peñate, F. López-Estrada, G. Valencia-Palomo, and I. Santos-Ruiz, "Actuator multiplicative and additive simultaneous faults estimation using a qlpv proportional integral unknown input observer," *International Journal of Adaptive Control and Signal Processing*, vol. 38, no. 8, pp. 2930–2950, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.3859

[133] W. Jun, Y. Xiong-Dong, and T. Yu-Yang, "Fault-tolerant control design of quadrotor uav based on cpso," Aug 2018, pp. 279–283. [Online]. Available: https://doi.org/10.1109/CCSSE.2018.8724803

[134] M. Chadli, M. Davoodi, and N. Meskin, "Distributed fault detection and isolation filter design for heterogeneous multi-agent LPV systems," in *2017 American Control Conference (ACC)*, 2017, pp. 1610–1615.

[135] M. Davoodi, M. Chadli, N. Meskin, and J. M. Velni, "Finite Frequency Fault Diagnosis for Heterogeneous Multi-agent LPV Systems," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 2019, pp. 1018–1023.

[136] J. A. V. Trejo, D. Rotondo, D. Theilliol, and M. A. Medina, "Observer-based quadratic boundedness leader-following control for multi-agent systems," *International Journal of Control*, vol. 96, no. 5, pp. 1314–1323, 2023.

[137] J. A. V. Trejo, A. Guenard, M. Adam-Medina, L. Ciarletta, J.-C. Ponsart, and D. Theilliol, "Resilient leader-following formation control for a fleet of unmanned aerial vehicles under cyber-attacks," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 731–737.

[138] P. Susbielle, A. Namigtle-Jimenez, J. F. Guerrero-Castellanos, N. Marchand, N. Mendoza-Robles, S. Durand, J. Díaz-Téllez, and R. Enríquez-Caldera, "Towards a digital twin for cooperative load transportation involving multiple multirotor drones," in *2024 Congreso Nacional de Control Automático*, vol. 7, no. 1, pp. 30–36, 2024.

[139] Z. Wang and M. Chadli, "A Virtual Actuator and Sensor Approach for Event-Triggered Fault-Tolerant Control of Multi-Agent Systems," *IFAC-PapersOnLine*, vol. 58, no. 4, pp. 282–287, 2024.

[140] Z. Wang and M. Chadli, "Distributed event-triggered fault estimation for Takagi–Sugeno multi-agent systems with unmeasurable decision variables," *Journal of the Franklin Institute*, vol. 362, no. 8, p. 107689, 2025.

[141] Y. Liu, J. Chen, Y. Yang, and H. Fu, "Simultaneous Fault Estimation and Fault-Tolerant Consensus Control for Multi-Agent Systems Against Multiple Time Delays," *IEEE Access*, vol. 13, pp. 93551–93565, 2025.

# Appendix A

# Résumé Détaillé de la Thèse

L'automatisation distribuée connaît une évolution majeure avec l'essor des systèmes multi-agents (SMA), où plusieurs entités autonomes interagissent pour accomplir des tâches collectives. Ces systèmes présentent des avantages décisifs en termes de flexibilité, de redondance et d'évolutivité par rapport aux architectures centralisées. Parmi leurs applications les plus prometteuses figure le contrôle coordonné de flottes de véhicules aériens sans pilote (UAV), dédiées à des missions de surveillance, d'inspection ou de logistique.

Cependant, la robustesse de ces systèmes coopératifs face à des défaillances reste un défi scientifique et technique majeur. En effet, dans un SMA, la défaillance d'un seul actionneur peut, via les interconnexions, déstabiliser l'ensemble du réseau et compromettre l'objectif commun (consensus, formation). Il est donc impératif de doter ces systèmes de capacités autonomes de tolérance aux défauts (Fault-Tolerant Control - FTC), leur permettant de maintenir un niveau de performance acceptable malgré la survenue de pannes.

Cette thèse s'attaque précisément à ce problème en proposant un cadre méthodologique complet pour la synthèse de lois de commande **distribuées, basées sur des observateurs et tolérantes aux défauts**. Pour modéliser avec réalisme la dynamique souvent non linéaire des agents (comme les UAV), nous adoptons le formalisme des systèmes **Linéaires à Paramètres Variants (LPV)**. Ce cadre permet de prendre en compte les variations paramétriques mesurables (vitesse, angle, etc.) tout en conservant les outils puissants de l'automatique linéaire. L'objectif ultime est de concevoir une architecture de contrôle qui garantisse à la fois le suivi d'un leader de référence et le maintien d'une formation géométrique, même lorsque certains actionneurs présentent des défaillances partielles ou totales.

## Problématique et Modélisation

Le système considéré est un ensemble homogène de $N$ agents dont la dynamique nominale est décrite par un modèle LPV continu. Pour chaque agent $i$, nous avons :

$$\begin{aligned} \dot{x}_i(t) &= A(\zeta_i(t))x_i(t) + B(\zeta_i(t))u_i(t), \\ y_i(t) &= C(\zeta_i(t))x_i(t), \end{aligned} \tag{A.1}$$

où $x_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^r$ et $y_i \in \mathbb{R}^p$ sont respectivement le vecteur d'état, le vecteur de commande et le vecteur de sortie. Les matrices $A(\zeta_i(t))$, $B(\zeta_i(t))$ et $C(\zeta_i(t))$ dépendent du vecteur de paramètres de *scheduling* $\zeta_i(t) \in \mathbb{R}^{n_\zeta}$. Ce vecteur, supposé mesurable en temps réel, varie dans un ensemble compact $\mathcal{R}$ et reflète les conditions de fonctionnement de l'agent. Pour permettre l'analyse et la synthèse avec des outils linéaires, le modèle LPV est converti en une représentation polytopique via la méthode du *bounding box* :

$$\begin{pmatrix} A(\zeta_i(t)) \\ B(\zeta_i(t)) \\ C(\zeta_i(t)) \end{pmatrix} = \sum_{h=1}^{S} \rho_h(\zeta_i(t)) \begin{pmatrix} A_h \\ B_h \\ C_h \end{pmatrix}, \quad \sum_{h=1}^{S} \rho_h(\zeta_i(t)) = 1, \quad \rho_h(\zeta_i(t)) \geq 0 \quad \forall \zeta_i(t) \in \mathcal{R}. \tag{A.2}$$

Pour prendre en compte l'apparition de défaillances au niveau des actionneurs, le modèle est étendu afin d'intégrer à la fois des défauts multiplicatifs (perte d'efficacité) et additifs (biais) :

$$\begin{aligned} \dot{x}_i(t) &= A(\beta_i(t))x_i(t) + B_f\big(\beta_i(t), \vartheta_i(t)\big)\big(u_{v_i}(t) + f_{u_i}(t)\big), \\ y_i(t) &= C(\beta_i(t))x_i(t), \end{aligned} \tag{A.3}$$

avec $B_f\big(\beta_i(t), \vartheta_i(t)\big) = B(\beta_i(t))\vartheta_i(t)$. La matrice diagonale $\vartheta_i(t) = \text{diag}(\kappa_{i_1}(t), \ldots, \kappa_{i_r}(t))$ modélise l'efficacité de chaque actionneur ($0 \leq \kappa \leq 1$), et $f_{u_i}(t)$ représente un défaut additif indépendant. Cette modélisation permet de décrire une large gamme de scénarios de défaillance réalistes.

Les objectifs de contrôle sont doubles et doivent être atteints de manière robuste et tolérante aux défauts :

1. **Suivi de leader :** Assurer la convergence de l'état de chaque agent suiveur vers celui d'un leader virtuel de référence $x_l(t)$, dont la dynamique est également LPV. Cela implique que l'erreur de synchronisation $\delta_i(t) = x_i(t) - x_l(t)$ tende vers zéro.

2. **Contrôle de formation :** Maintenir une configuration géométrique spécifique (par exemple, un hexagone ou un pentagone) entre tous les agents. Cette formation est définie par un ensemble de vecteurs de décalage $h_i$ fixés préalablement.

# Architecture de Contrôle et Contributions Principales

Pour relever ces défis, la thèse développe de manière progressive une architecture de contrôle de plus en plus sophistiquée. Le cheminement suit une logique claire : partir d'un cas linéaire simple et robuste, généraliser au cadre paramétrique plus réaliste (LPV), puis intégrer la capacité de reconfiguration active face aux pannes.

## 1. Commande Robuste par Consensus Basée sur Observateur pour SMA Linéaires (LTI) – Chapitre 3

Cette première étape pose les fondations en considérant le cas simplifié, mais fondamental, des SMA linéaires et invariants dans le temps (LTI). Elle vise à établir une stratégie de base robuste aux incertitudes de modélisation et aux perturbations externes.

- **Problème :** Concevoir un contrôleur distribué garantissant le consensus leader-suiveur et la formation, tout en atténuant les effets de **perturbations externes bornées** (telles que le vent) et du **bruit de mesure** des capteurs.
- **Solution :** Nous proposons une loi de commande reposant sur un observateur de Luenberger distribué. Chaque agent $i$ calcule sa commande à partir de l'estimation de son propre état et des états estimés de ses voisins :

$$u_i(t) = K\left[\sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i(t) - \hat{x}_j(t)) + \alpha_i(\hat{x}_i(t) - x_l(t))\right]. \tag{A.4}$$

  L'observateur local reconstruit l'état $\hat{x}_i(t)$ à partir des mesures bruitées.
- **Synthèse :** L'innovation réside dans le calcul **simultané** des gains de retour d'état $K$ et de l'observateur $L$ via la résolution d'un problème d'**Inégalités Matricielles Linéaires (LMI)**. Ce problème est formulé sous un critère de performance $\mathcal{H}_\infty$ qui minimise le gain $\mathcal{L}_2$ des perturbations vers les erreurs de consensus et d'estimation, garantissant ainsi une robustesse intrinsèque.
- **Validation :** La méthode est validée par simulation numérique sur une flotte non linéaire de 6 quadrirotors. Les résultats mettent en évidence la capacité du système à conserver une formation hexagonale et à suivre une trajectoire de référence, malgré l'injection de turbulences atmosphériques (modèle de Dryden) et un bruit de mesure significatif.

## 2. Généralisation au Cas LPV et Réduction du Conservatisme – Chapitre 4

La deuxième étape étend le cadre au réalisme des SMA-LPV, permettant de capturer les variations dynamiques des agents (comme les UAV) en fonction de paramètres opérationnels mesurables.

- **Problème :** Adapter la stratégie de consensus à un modèle LPV, où les matrices du système et, par conséquent, les gains du contrôleur et de l'observateur doivent s'ajuster en ligne en fonction du vecteur de paramètres $\beta(t)$.
- **Solution :** Nous concevons un **contrôleur LPV** $K(\beta(t))$ et un **observateur LPV** $L(\beta(t))$, tous deux représentés sous forme polytopique. La loi de commande intègre également la commande du leader $u_l(t)$ pour un suivi plus précis.

- **Synthèse Avancée :** La synthèse de gains pour un système polytopique peut être très conservative si l'on ne considère que les conditions aux sommets. Pour y remédier, nous introduisons l'**application du théorème de Polya**, issu de la théorie du contrôle flou. Ce théorème permet d'exprimer des conditions de stabilité qui considèrent des produits d'ordre supérieur des fonctions de pondération $\rho_h(\beta)$, conduisant à des conditions LMI **asymptotiquement nécessaires et suffisantes**, et donc beaucoup moins conservatives.

- **Validation :** Une modélisation quasi-LPV précise d'un quadrirotor DJI F450 est élaborée, avec 3 paramètres de scheduling générant 8 sommets polytopiques. Appliquée au contrôle d'une formation pentagonale de 5 UAVs, l'approche démontre son efficacité à maintenir un consensus stable et un suivi de trajectoire, validant ainsi la pertinence du cadre LPV et de l'outil de synthèse.

## 3. Commande Tolérante aux Défauts via Actionneurs Virtuels LPV – Chapitre 5

Cette troisième étape constitue le cœur de la contribution : doter l'architecture de contrôle d'une capacité active de reconfiguration face aux pannes d'actionneurs, sans remettre en cause la stabilité du réseau.

- **Problème :** Maintenir les objectifs de consensus et de formation lorsque des agents subissent des **défauts simultanés, à la fois additifs et multiplicatifs**, sur leurs actionneurs.

- **Solution :** L'idée maîtresse est l'introduction d'un **actionneur virtuel LPV** dynamique pour chaque agent. Cet actionneur virtuel s'intercale entre le contrôleur nominal (conçu pour un système sain) et le système physique potentiellement défaillant. Son rôle est triple :

  1. **Isolement :** Masquer les effets du défaut au contrôleur nominal, qui continue de fonctionner comme si de rien n'était.
  2. **Reconfiguration :** Recalculer en ligne la commande $u_{v_i}$ à appliquer au système défaillant pour que son comportement global se rapproche le plus possible du comportement nominal souhaité.
  3. **Compensation d'estimation :** Injecter un signal $y_{v_i}$ dans l'observateur pour corriger l'estimation d'état, laquelle est faussée par le défaut.

  La dynamique de cet actionneur virtuel est elle-même LPV :

$$\begin{aligned} \dot{x}_{v_i}(t) &= \big(A(\beta(t)) + B^*(\beta(t))M_v(\beta(t))\big)x_{v_i}(t) + \big(B(\beta(t)) - B^*(\beta(t))\big)u_i(t), \\ y_{v_i}(t) &= C(\beta(t))x_{v_i}(t), \end{aligned} \tag{A.5}$$

  où $M_v(\beta(t))$ est le gain LPV de l'actionneur virtuel à synthétiser.

- **Synthèse Unifiée et Novatrice :** La contribution principale réside dans la dérivation d'une condition LMI unique et structurée (Théorème 3). Cette condition permet le calcul **simultané et garanti** des trois ensembles de gains polytopiques : ceux du **contrôleur consensus** ($K_h$), de l'**observateur** ($L_h$) et de l'**actionneur virtuel** ($M_{v_h}$). L'application du **théorème de Polya** à ce problème de synthèse conjointe complexe permet de minimiser le conservatisme du résultat, aboutissant à un design globalement cohérent et performant.

- **Validation Expérimentale par Simulation :** Un scénario complet et réaliste est simulé avec une formation pentagonale de 5 UAVs.

  1. **Cas nominal :** La stratégie de contrôle, utilisant les gains issus du Th. 3, maintient parfaitement la formation et le suivi.
  2. **Cas défaillant (sans FTC) :** Un défaut additif sévère est injecté sur un rotor d'un UAV à $t = 20\,s$, suivi d'un défaut multiplicatif (20% de perte d'efficacité) sur tous les rotors d'un autre UAV à $t = 35\,s$. Sans mécanisme de tolérance, la formation se dissout et le consensus est perdu, illustrant la vulnérabilité du système.
  3. **Cas FTC :** Après un délai réaliste de 5 secondes (simulant le temps de détection et de décision), les actionneurs virtuels correspondants sont activés. Les résultats sont concluants : les agents défaillants sont reconfigurés, et **l'ensemble de la flotte retrouve sa formation et son synchronisme avec le leader**. Ceci démontre sans équivoque la **résilience** apportée par l'approche et sa capacité à **contenir localement l'impact des défauts**.

# Conclusions Générales et Perspectives

Cette thèse a apporté des contributions méthodologiques substantielles au domaine du contrôle coopératif des systèmes multi-agents. En suivant une démarche incrémentale et rigoureuse – allant du contrôle robuste LTI au contrôle tolérant aux défauts LPV – nous avons démontré comment unifier des techniques avancées d'observation, de théorie LPV, d'optimisation par LMI et de reconfiguration dynamique au sein d'une architecture de contrôle distribuée. La **synthèse simultanée** des gains, rendue possible par l'exploitation du théorème de Polya, constitue une avancée notable, offrant une solution à la fois moins conservative, théoriquement garantie et directement applicable à des systèmes non linéaires modélisés sous forme LPV.

Les travaux futurs issus de cette thèse pourront s'orienter selon plusieurs axes de recherche prometteurs :

- **Diagnostic intégré :** Développer et intégrer des algorithmes de diagnostic en ligne performants pour l'estimation en temps réel des défauts ($\hat{\vartheta}_i(t)$, $\hat{f}_{u_i}(t)$), levant ainsi l'hypothèse actuelle de défauts parfaitement connus.

- **Tolérance aux défauts de capteurs :** Étendre l'architecture pour inclure également la tolérance aux défaillances des capteurs, par exemple via le concept dual de **capteur virtuel**.

- **Robustesse réseau :** Étudier la robustesse des stratégies proposées face aux imperfections des réseaux de communication (délais variables, pertes de paquets, bande passante limitée) voire à des cyber-attaques.

- **Interaction avec la planification de trajectoire :** Combiner la couche de contrôle tolérant aux défauts avec des algorithmes de planification de trajectoire et d'évitement d'obstacles dynamiques, pour assurer la sécurité opérationnelle dans des environnements complexes et incertains.

- **Validation expérimentale à grande échelle :** Implémenter et valider les algorithmes sur une plateforme physique d'essai impliquant une flotte réelle d'UAVs, afin de confronter les performances théoriques aux contraintes du monde réel (bruits, incertitudes de modélisation, limitations matérielles).

En conclusion, ce travail fournit un socle théorique et pratique robuste pour le développement de systèmes multi-agents autonomes d'une nouvelle génération : des systèmes non seulement coopératifs et performants, mais aussi intrinsèquement **robustes, résilients et dignes de confiance**. De telles propriétés sont indispensables pour leur déploiement dans des applications critiques où la sécurité et la fiabilité sont non négociables.