Synthesis of Observers and Robust Control Laws for Swarm or Formation Flight of Unmanned Aerial Vehicles (UAVs)

Quang Truc Dam

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité AUTOMATIQUE, SIGNAL, PRODUCTIQUE, ROBOTIQUE

Préparée au sein de l'Université de Rouen Normandie

Synthèse d'observateurs et de lois de commande robustes pour robots mobiles aériens (UAVs) navigant en formation ou en essaim

Présentée et soutenue par
QUANG TRUC DAM

Thèse soutenue le 05/07/2023 devant le jury composé de		
M. NICOLAS MARCHAND	DIRECTEUR DE RECHERCHE, CNRS GIPSA Lab Grenoble INP	Rapporteur du jury
M. JULIEN MARZAT	DIRECTEUR DE RECHERCHE, Office National d'Etudes&recherches Aéro	Rapporteur du jury
M. SOFIANE AHMED ALI	MAITRE DE CONFERENCES HDR, UNIVERSITE EVRY VAL D ESSONNE	Membre du jury
MME SAMIA AINOUZ	PROFESSEUR DES UNIVERSITES, Institut National des Sciences Appliquées Rouen Normandie	Membre du jury
M. RIHAB HAJRI ELHOUDA	, ESIGELEC	Membre du jury
MME CRISTINA STOICA-MANIU	PROFESSEUR DES UNIVERSITES, CENTRALESUPELEC GIF SUR YVETTE	Membre du jury
M. YACINE AZZOUZ	, ESIGELEC	Directeur de thèse
M. FRANCOIS GUERIN	MAITRE DE CONFERENCES HDR, Université de Rouen Normandie	Co-directeur de thèse

Thèse dirigée par YACINE AZZOUZ (INSTITUT DE RECHERCHE EN SYSTEMES ELECTRONIQUES EMBARQUES) et FRANCOIS GUERIN (Groupe de Recherche en Electrotechnique et Automatique du Havre)

UNIVERSITÉ DE ROUEN NORMANDIE

ESIGELEC

MIIS

irseem UR 4353

Normandie Université

THESIS

To obtain a doctoral degree

Speciality AUTOMATIC

Prepared at University of Rouen Normandy

Synthesis of Observers and Robust Control Laws for Swarm or Formation Flight of Unmanned Aerial Vehicles (UAVs)

Presented and defended by

Quang Truc DAM

Publicly defended July 05, 2023
in front of the dissertation committee

M. JULIEN MARZAT	Research Director, ONERA, France	Rapporteur
M. NICOLAS MARCHAND	Research Director, CNRS, GIPSA Lab, Grenoble INP, France	Rapporteur
Mme. CHRISTINA MANIU	Professor, CentraleSupélec , France	Examiner
Mme. SAMIA AINOUZ	Professeur, INSA Rouen, France	Examiner
M. SOFIANE AHMED ALI	Associate Professor, Université d'Evry-Val d'Essonne, France	Examiner
M. YACINE AZZOUZ	Lecturer-researcher, ESIGELEC, France	Supervisor
M. FRANÇOIS GUERIN	Associate Professor, Université du Havre, France	Supervisor
Mme. RIHAB EL HOUDA THABET	Lecturer-researcher, ESIGELEC, France	Co-Supervisor

Supervised by YACINE AZZOUZ et FRANÇOIS GUÉRIN and Co-Supervised by RIHAB EL HOUDA THABET, IRSEEM (Rouen)

UNIVERSITÉ DE ROUEN NORMANDIE
ED 590 MIIS Mathématiques, Information, Ingénierie des Systèmes
ESIGELEC irseem
UNIVERSITÉ LE HAVRE NORMANDIE
GREAH
RÉGION NORMANDIE

Résumé

Cette thèse se concentre sur la conception de contrôleurs pour des systèmes d'aéronefs sans pilote (UAV) opérant en essaim et confrontés à des retards de communication. L'étude se concentre sur deux aspects principaux : la conception d'observateur et le contrôle de suivi de position.

La première partie de la thèse aborde la conception de l'observateur pour des systèmes non linéaires sous des mesures retardées et son application aux quadrirotors UAV. La conception de l'observateur tient compte à la fois des mesures retardées et bruyantes, et l'étude étend la conception pour traiter le cas où les mesures sont à la fois retardées et bruyantes. La méthode proposée pour la conception de l'observateur est de pointe, en utilisant la technique de l'inégalité de matrice linéaire (LMI) pour calculer les gains d'observateur appropriés garantissant la stabilité pour n'importe quelle valeur de retard.

La seconde partie de la thèse explore le contrôle de suivi de position pour les quadrirotors UAV. Deux algorithmes de contrôle adaptatif sont proposés, tous deux conçus pour améliorer la précision du système de contrôle. Le premier algorithme utilise un réseau neuronal à fonction de base de rayon et la logique floue pour concevoir un contrôleur à mode glissant adaptatif, tandis que le second algorithme utilise un réseau neuronal de propagation et une technique de mode glissant pour la conception de contrôle.

S'appuyant sur la réussite de la conception d'observateur et de contrôleur pour un seul quadrirotor UAV, la thèse propose un contrôleur de suivi de formation pour l'essaim. Le contrôleur proposé tient compte du délai de communication qui existe entre les agents de l'essaim, garantissant ainsi que la formation reste stable et précise même en présence de retards. L'étude souligne l'importance de prendre en compte les retards de communication dans la conception des systèmes de contrôle UAV et démontre la faisabilité des algorithmes proposés pour la conception d'observateur et de contrôle dans un environnement d'essaim.

Cette thèse apporte des éléments de réflexion précieux sur la conception de contrôleurs pour des systèmes UAV opérant en essaim sous des retards de communication. L'étude contribue au développement de systèmes de contrôle UAV plus avancés et fiables pouvant naviguer efficacement dans des environnements difficiles.

Abstract

This thesis delves into the design of controllers for unmanned aerial vehicle (UAV) systems operating in a swarm and facing communication delays. The study focuses on two main aspects: observer design and position tracking control.

The first part of the thesis addresses the observer design for non-linear systems under delayed measurements and its application to quadrotor UAVs. The design of the observer takes into account both delayed and noisy measurements, and the study extends the design to handle the case where measurements are both delayed and noisy. The proposed method for observer design is state-of-the-art, using the linear matrix inequality (LMI) technique to calculate the appropriate observer gains that guarantee stability for any given delay value.

The second part of the thesis explores position tracking control for quadrotor UAVs. Two adaptive control algorithms are proposed, both designed to improve the accuracy of the control system. The first algorithm employs a radius basis function neural network and fuzzy logic to design an adaptive sliding mode controller, while the second algorithm uses a propagation neural network and sliding mode technique for control design.

Building on the successful observer design and controller design for a single quadrotor UAV, the thesis goes on to propose a formation tracking controller for the swarm. The proposed controller takes into account the communication delay that exists between agents in the swarm, ensuring that the formation remains stable and accurate even in the presence of delays. The study emphasizes the importance of taking communication delays into consideration in the design of UAV control systems, and demonstrates the feasibility of the proposed observer design and control algorithms in a swarm environment.

This thesis provides valuable insights into the design of controllers for UAV systems operating in a swarm under communication delays. The study contributes to the development of more advanced and reliable UAV control systems that can effectively navigate in challenging environments.

Acknowledgements

This thesis has been realized in Écoles d'ingénieur-es généralistes (ESIGELEC) and funded by University of Rouen Normandy.

First and foremost, I would like to express my sincere gratitude to my thesis advisor, Dr Sofiane Ahmed Ali, Dr François Guérin, and my additional supervisor, Dr Rihab El Houda Thabet for their guidance, encouragement, and support throughout my graduate studies. Their expertise in automation control, their invaluable feedback on my work, and their constant availability to discuss my progress and ideas have been instrumental in the successful completion of this thesis. With their enormous helps, I have gained a lot of experiences working in the field of research, and lead me to many success scientific publications throughout the work for this thesis.

Dr Sofiane Ahmed Ali, Dr François Guérin and Dr Rihab El Houda Thabet have been an invaluable mentor, not only in the research and writing of this thesis, but also in guiding me through the process of preparing my work for publication. Their insight, guidance, and support have been instrumental in helping me to publish my work in prestigious journal and conference papers.

I am truly grateful for my supervisor's dedication and commitment to my success, and I feel honored to have had the opportunity to work under their supervision. I would not be where I am today without their mentorship, and I am forever grateful for all that they have done for me

I would also like to thank Julien Marzat, and Nicolas Marchand, for their valuable feedback, insights, and time spent on my thesis.

I would like to extend my appreciation to Dr. Nicolas Langlois and Mme. Isabelle RIGUIDEL for their support and for providing me with the opportunity to work in IRSEEM, where I meet many friendly colleagues, who always help me during my working time.

I would like to extend my deepest appreciation to the trainees who assisted me in the laboratory. Antoine Hugo, Belin Gaëtan, Dimitri Cauffiez, Jean Lagrue and Yuxuan Tang. Their hard work, dedication and enthusiasm were instrumental in the success of this research. Their expertise in C++ programming language was invaluable in the development of the experiments that form the basis of this thesis. I am extremely grateful for their invaluable contributions and insights, which greatly enhanced the quality of the research. I would also like to thank them for their patience and support throughout the duration of the project.

I would like to express my deepest gratitude to my friends for their unwavering love, support, and understanding throughout my PhD journey. Their encouragement and belief in me have been a constant source of motivation and inspiration. Their constant support and encouragement, for always being there to listen and offer valuable advice, and for being a constant source of laughter and joy.

I am especially grateful to my family, for their love and support throughout the years. Their constant encouragement and belief in me have been instrumental in my ability to achieve this goal. I would like to thank my parents, for their unwavering support and for always being there for me.

I could not have completed this journey without the love and support of my friends and family. I am forever grateful for all that they have done for me.

Additionally, I would like to acknowledge the funding received from ARTISMO project that made many conference publications in this research possible.

Finally, I would like to express my sincere gratitude to all participants who took time out of their busy schedule to contribute to this research.

Thank you all.

List of Abbreviations

$\{\boldsymbol{B}\}$	Body reference frame
$\{\boldsymbol{E}\}$	Earth reference frame
2D	Two dimensional
3D	Three dimensional
BPNN	**B**ack **P**ropagation **N**eural **N**etwork
CFD	**C**omputational **F**luid **D**ynamics
CoM	**C**entre **o**f **M**ass
EKF	**E**xtended **K**alman **F**ilter
ESC	**E**lectronic **S**peed **C**ontroller
FHGO	**F**iltered **H**igh **G**ain **O**observer
GPS	**G**lobal **P**osition **S**system
HGO	**H**igh **G**ain **O**observer
HITL	**H**ardware **I**in **T**he **L**oop
HOSMO	**H**ight **O**rder **S**liding **M**ode **O**bserver
IMU	**I**nertial **M**easurement **U**nit
KLO	**K**alman **L**ike **O**bserver
LMI	**L**inear **M**atrix **I**neuqality
LPV	**L**inear **P**arameter **V**ariying
LQR	**L**inear **Q**uadratic **R**egulator
LQG	**L**inear **Q**uadratic **G**aussian
MAVTD	**M**aximum **A**llowable **V**alue of **T**ime **D**elay
PID	**P**roportional **I**ntegral **D**erivative
PWM	**P**ulse **W**idth **M**odulation
QUEST	**QU**aternion **EST**imator
RBFNN	**R**adial **B**asic **F**unctions **N**eural **N**etwork
SHGO	**S**ampled **H**igh **G**ain **O**bserver
SMC	**S**liding **M**ode **C**ontroller
SPD	**S**ymmetric **P**ositive **D**efinite
SITL	**S**oftware **I**n **T**he **L**oop
TCP	**T**ransmission **C**ontrol **P**rotocol
TRIAD	**TRI**-axial **A**ttitude **D**etermination
UAV	**U**nmanned **A**erial **V**ehicle
VOTL	**V**ertical **T**ake- **O**ff and **L**anding

List of Symbols

$\mathbb{R}$	The real space
$\mathbb{H}$	The quaternion space
$\mathbb{I}$	The imaginary space
$\mathbb{R}^{n\times m}$	The set of real matrices of n rows and m columns
$\mathbb{R}^{+}$	The set of non-negative numbers
$tr(P)$	trace of matrix P
$eig(P)$	The set of eigenvalues of square matrix $P \in \mathbb{R}^{n\times n}$
P^T	The transpose of matrix P
$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$	The norm of vector $x \in \mathbb{R}^{n\times 1}$
$diag(x)$	The diagonal matrix representation of vector $x \in \mathbb{R}^{n\times 1}$
$\|P\| = \sqrt{tr(P^T P)}$	The norm of matrix $P \in \mathbb{R}^{n\times n}$
$\lambda_{\max}(P)$, $\lambda_{\min}(P)$	The maximum and minimum eigenvalues of $P \in \mathbb{R}^{n\times n}$
$\|.\|$	The absolute value
$L_f h(w(t))$	The Lie derivative of the function h with respect to the vector f
$\frac{\partial f}{\partial x}$	The partial derivative of function f with respect to the vector x.
$\frac{df}{dt}$	The derivative of function f with respect to the time.
$x^{(n)}$	The n^{th} derivative of x with respect to the time.
$\bigotimes$	The quaternion product.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In this chapter, the reader will find a brief but comprehensive review of Unmanned Aerial Vehicles (UAVs) and the underlying reasons behind selecting a specific UAV platform for use in the present thesis. The chapter further delves into the various research directions in the UAV study area, elucidating the different approaches that researchers have taken to address the challenges associated with UAVs, such as controller design, observer design. Moreover, the objectives of the thesis are clearly stated, giving the reader a clear sense of what the study aims to accomplish. Finally, the structure of the thesis document is outlined, providing a roadmap for the reader to navigate through the study's various sections and chapters.

1.1 Unmanned aerial vehicles (UAVs)

In the last decade, UAV have been leading trends in robotic research. Due to their great advantages in terms of speed and maneuverability, the UAVs have been employed in many applications from the military aspect such as battlefield monitoring, carrying weapon, air-to-ground and air-to- air attacks, to the industrial, commercial applications, such as photography, agriculture, disaster monitoring, environmental surveillance and electric power inspection. Attracted by its huge potential, many company, research organization and laboratories, R&D centers, universities and also governments have been participating in the race to develop UAV system. Due to the different needs for each application, we find different types of UAVs, which can be categorized as follows:

1. Fixed-wing UAVs: they provide a wide range of working with long endurance. However, in order to operate a fixed-wing UAVs in a mission, a strong and long-distance communication such as satellites system is required. Therefore this type of UAVs are often used in military application with large scale, Figure 1.1 shows some examples of fixed-wing UAVs used in military applications.

(a) Global hawk RQ-4 UAV (b) Scaneagle UAV (c) RCExplorer Bicopter UAV

Figure 1.1: Examples of miniature fixed-wing UAVs

Fixed-wing UAVs with small scale are also used in some commercial applications (Figure 1.2). However, they have some limitations during the execution of missions in urban environments. Vertical Take-Off and Landing UAVs (VTOL UAVs): VTOL UAVs have several advantages compared to fixed-wing UAVs. Due to their ability to take off and land vertically, they require less space for launch and recovery, as they do not need to use a runway like fixed-wing UAVs. Therefore, VTOL

(a) HEE WING T-1 Ranger UAV (b) Opterra 2m UAV (c) Hubsan H301S Spy Hawk UAV

Figure 1.2: Examples of commercial fixed-wing UAVs

UAVs are well-suited for applications such as inspection and monitoring, where the UAVs need to maintain a fixed position for a period of time. However, VTOL UAVs are not known for their wide range of working and long endurance, which is a strength of fixed-wing UAVs. As a result, VTOL UAVs are primarily used in civil and commercial applications. VTOL UAVs can be further categorized into two smaller groups as follows:

- Helicopter UAVs: they operate on the same flight principles as manned helicopters. The most common design of helicopter is the use of one rotor to generate lift and thrust, plus a tail rotor that provides counter-torque. Figure. 1.3 presents some examples of this type of UAVs.

(a) Vapor helicopter UAV (b) Neo S300 helicopter UAV (c) Black Hornet UAV

Figure 1.3: Examples of single lift rotor helicopter UAVs

Beside this design, there exists also coaxial rotors helicopter, this design use a pair of rotors mounted one above the other on concentric shafts. They rotate on the same axis of rotation, but turning in opposite directions to control the yaw rotation. We can see some examples of this type of UAVs in Figure. 1.4. Due to the simply motor configuration, the helicopter UAVs can be designed in small, medium and large scale for different purposes. However, one of the most disadvantages of helicopter UAVs is that they have to use complex variable rotor for roll and pitch control.

(a) Ingenuity UAV (b) Modlab UAV (c) Koax X-240 UAV

Figure 1.4: Examples of coaxial rotors helicopter UAVs.

- Bi-copter UAVs or Dual-copter UAVs: They have two lift rotors, they rotate on different axis of rotation one clockwise and the other counterclockwise. Lifts rotors which will generate the thrust force for altitude control and rotation force for roll and yaw control. However, for pitch control, complex variable pitch rotors are used (two fixed horizontal motors to adjust the rotation axis of each lift rotors). The most disadvantage of using Bi-copter UAV is the balance control complexities when the UAV generates two thrust forces which respond differently to

disturbances across the arm direction. However, the configuration of Bi-copter UAV allows the UAV the adjustment of the rotation axis for each lift rotor, which is more flexible than helicopter. Hence thee vertical translation speed of this UAV can be higher. Figure 1.5 shows some examples of Bi-copter UAVs.

(a) RCExplorer Bicopter UAV (b) V-Copter Falcon UAV (c) RC Bicopter UAV

Figure 1.5: Examples of Bi-copter UAVs

- Multi-copter UAVs: They have more than two lift motors. Unlike bi-copter UAV or Helicopter UAV, Multi-copter UAV does not need complex variable rotors whose pitch varies as the blade rotates for flight control. Multi-copter UAV often use fixed rotors. The motion control is achieved by varying the relative speed of each rotor to produce the desired thrust force and rotation torques for flight control. There are four popular Multi-copter UAVs, namely tricopter (trirotor) UAV, quadcopter (quadrotor) UAV, hexacopter (hexarotor) UAV and octocopter (octocrotor) UAV, which have respectively three, four, six and eight lift rotors. Figure. 1.6 shows the rotor configuration of four different Multi-copter UAVs.

(a) Tri-copter UAV (b) Quad-copter UAV

(c) Hexa-copter UAV (d) Octo-copter UAV

Figure 1.6: Examples of tri-copter UAV, quad-copter UAV, hexa-copter UAV and octo-copter UAV .

Beside these mentioned designs, there exists also some variations of hexacopter (hexarotor)

UAV and octocopter (octocrotor) UAV, namely hex-tri UAV and octo-quad UAV. Hexa-tri UAV has the same configuration as the tricopter UAV, except it uses six lift rotors instead of three. Similarly, octo-quad UAV also has similar configuration to the quadracopter, but it uses eight lift rotors to double the lift force. The rotor's configuration of these UAV is shown in Figure. 1.7

(a) Hex-tri UAV (b) Octo-quad UAV

Figure 1.7: Rotor's configuration of hex-tri UAV and octo-quad UAV

2. VTOL fixed-wing UAVs: to overcome the disadvantages of fixed-wing UAV and VTOL UAV, a new design, namely VTOL Fixed-Wing UAV (hybrid UAV) is born. This design combine the vertical lift rotor to a Fixed-wing UAV, which allows the UAV to take of and landing vertically and flight in a long distance with higher speed. This configuration creates a highly flexible UAV. However, this design does not bring an optimal aerodynamic effects to the UAV when it work as fixed-wing UAV or VTOL UAV. Figure. 1.8 shows some examples of VTOL fixed-wing UAVs.

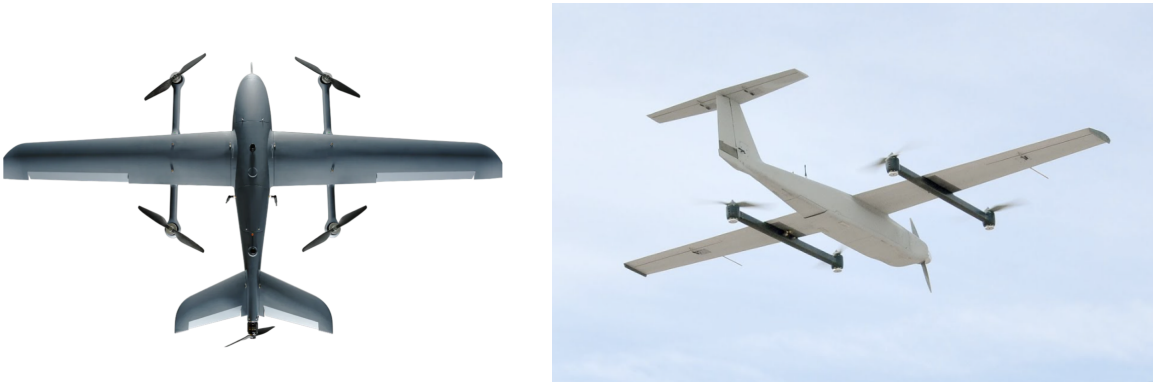


Figure 1.8: Examples of VTOL Fixed-Wing UAVs

Among these configurations, quad-copter (quadrotor) UAV is the most popular in research, commercial and entertainment because of low cost, low inertial force and simple flight control system. Quadrotor UAV is the most popular commercialized UAV which has been successfully designed for research purposes. As shown by figure .1.9, some popular platforms of quadrotor UAVs are usually used in research namely:

- DJI F450 (Figure. 1.9a): This is one of the most popular quadrotor UAV platform used in research. It is designed by DJI Technology. This is Do It Yourself (DIY) platform requires developer to assemble all the component of UAV. Hence, this will allows the developer to modify the hardware of UAV such as main control board, camera and sensors based on their research purpose.
- S500 (Figure. 1.9b): Similar to DJI F450, S500 is also DIY and developed by Holybro. DJI F450 and S500 normally come with Pixhawk control board. This is an open-source control board, which

allows the developer to modify the control program. Hence, those type of UAV are very suitable for some research such as controller and observer design.

(a) DJI F450

(b) S500

(c) AR Parrot

(d) Draganflyer X4

Figure 1.9: Quadrotor platform UAV used in research

In this thesis, the S500 quadrotor UAV will be used to validate the proposed algorithms.

1.2 UAV control system

Due to the variety of configurations and use cases in many domains, the study of UAVs has gained much interest around the world. However, from an academic perspective, the control design of UAV systems remains challenging due to their nonlinear dynamic behavior, external unknown disturbances, and unknown parameters, such as inertia and mass. These challenges have attracted much attention from the automatic control community in recent years and have also motivated the study in this thesis. In the following sections, the global control structure and the flight control module will be presented.

1.2.1 Global control structure

The control structure of a UAV normally consists of four mains part, namely: guidance, core controller, actuator calculator and actuator controller as shown in Figure. 1.10. Each component has its own function as follows:

- Guidance: The guidance module has input as the navigation signal from ground control computer or radio controller. From these signal, this module will generate the corresponding reference trajectories as desired angles, positions and velocities. This desired trajectory will be sent to the flight control module (core controller). The guidance module can perform two main tasks, namely: mission planning and trajectory generation. Mission planning will generate flight plans, timing for the

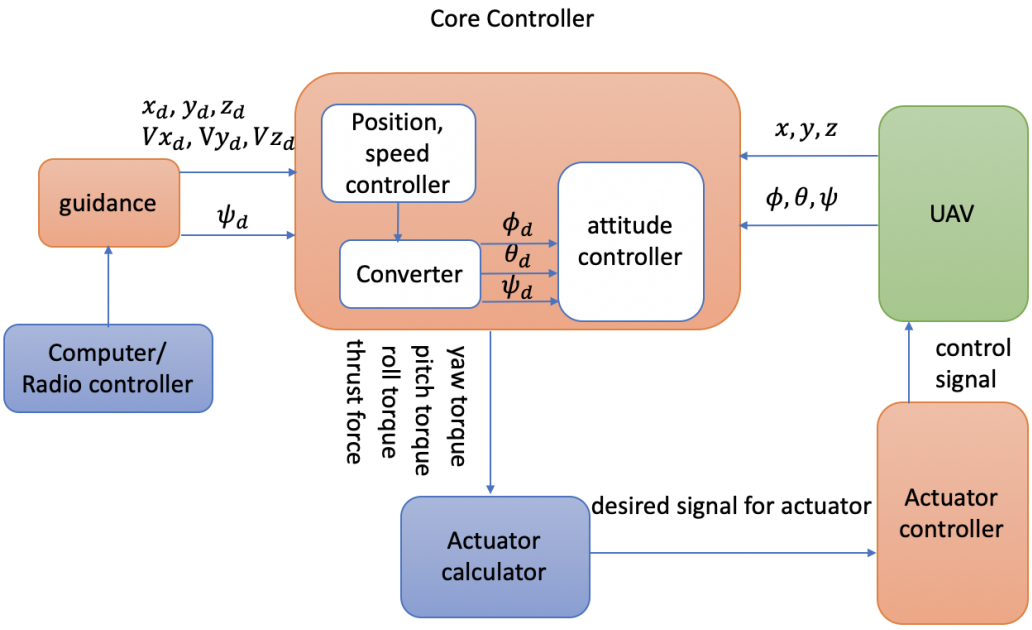


Figure 1.10: Control structure of a UAV

vehicle, route coordination and tactical goals, in real-time. This task is done in the autonomous flight mode, and given by a human pilot or by the software system (for example, the Qgroundcontrol software). Trajectory generation task will compute and provide real time reference trajectories for the flight control. For more intelligent UAV system, the Guidance module might also perform the Path-planning and obstacle avoidance task.

- Core controller (flight control module): this is the main part of the control structure for any UAV system, this module will calculate the appropriate control signal to achieve the desired reference given by the guidance module. Based on the signal coming from guidance module, the light control module will perform the different tasks such as altitude control, position control, velocity control, etc. To achieve these tasks, the core controller consists of two sub control systems, namely: position controller (outer controller) and attitude controller (inner controller). This control block is in charged of calculating the desired thrust force, rotation torques in order to track the desired position, velocity and steering angle given by the guidance block.

- Actuator calculator: from the desired thrust force and rotation torques, the actuator calculator will calculate the desired actuator signal for UAV. This calculator is different according to the used the UAV system. For example, for the quadrotor UAV system the actuator calculator will calculate the desired rotation velocity of four rotors.

- Actuator controller: This block will control the actuator of UAV based on the desired signal given by the actuator calculator block. For example, in the quadrotor UAV, this block will contain four motor controllers in order to control four lift motors of quadrotor UAV tracking the desired velocity given by the actuator calculator block.

After many decades of developing, the control boards for UAV system are equipped nowadays with a relatively perfect actuator calculator and actuator controller. The guidance block, which has to generate the desired position, velocity and steering angle for UAV, is also well developed. Therefore, most of the recent researches for UAV control are focusing on the core controller design.

1.2.2 Flight control module

The flight control module calculates the necessary moments and forces to control the UAV, to achieve the tracking or navigational task, as well as to compensate the internal and external disturbances. It can perform the following tasks:

- Position control: in this task, the controller controls the UAV in order to track a desired trajectory. This task is achieved by combing both control loops namely inner and outer controllers.
- Altitude control: It is for controlling the height of the UAV to a desired value. "Take off" is also included in this task. This task can be considered as a small task of position control where the height is the only desired input.
- Attitude control: this task concerns only tracking the desired attitude and maintaining the UAV's height simultaneously. This task is usually deployed in manual control mode, where the UAV is controlled by radio controller.
- Yaw or heading control: this task focuses on orienting the UAV to the desired orientation.
- Velocity control: it will deploy the position controller to calculate the desired attitude for the inner controller and the thrust force, but rather give the desired position as a reference. The desired velocity will be the reference in this case.

Beside the problem of controller design for UAV system. The case of multi UAVs will be also studied.

1.3 Swarm of UAVs

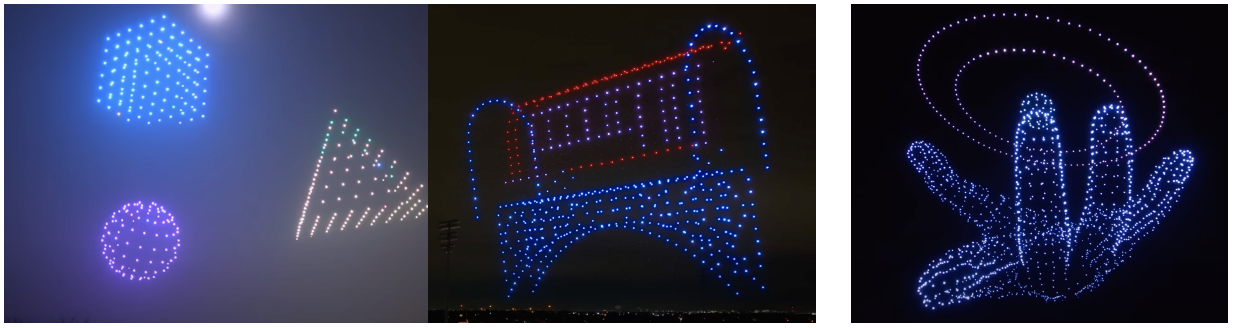
Since the study of UAVs has gained great interest, a power-full controller for a UAV has been provided, and many application have been illustrated. However, UAVs exhibits limitations due to its size, its payload capacity. For example, in many applications such as enhancing agriculture practices, fighting fires, transportation search and rescue, it is obviously these task are not suitable for a single UAV due to the mentioned limitation. Therefore, the study of control of multiple UAVs or so called Swarm of UAV is very attractive.

1.3.1 Application of Swarm UAVs

Due to the ability of cooperating in large teams, the swarm of UAVs has overcome many limitations of using individual UAV such as working rang, battery limitation, etc...

In the following, one shall present some of the most popular application of swarm UAVs over individual UAV:

- Entertainment: Light shows can be considered as one of the most successful entertainment application of swarm UAVs. In this application, each UAV will be equipped with LEDs to perform a colorfully show. Firefly Drone Shows company provide a light show service using 100, 300 and 500 UAVs (Figure. 1.11a). Sky Elements Drone Shows company made a drone show ever seen in Texas with 1001 drones (Figure. 1.11b). In Bussan, Korea, they present the colourful lights show based on different themes like 'Magic' or 'Racing Drone' using more than 300 drones Figure. 1.11c.



(a) Firefly Drone Shows (b) Sky Elements Drone Shows (c) Busan Gwangalli Drone Show

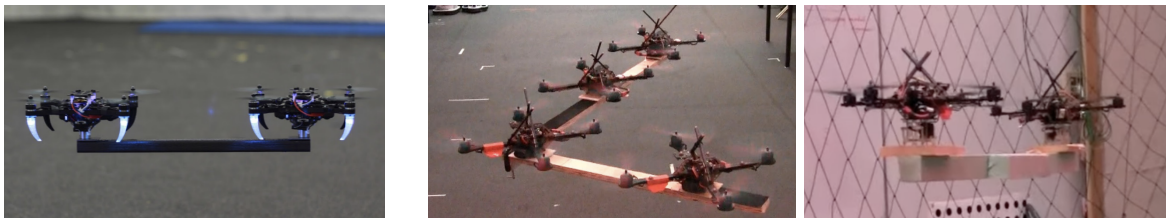
Figure 1.11: UAVs used in light shows for entertainment

- Security and Surveillance: Compare to fixed CCTV cameras, one UAV itself provides several advantages due to its mobility. For example, the monitored areas will be vastly extended. However, using single UAV for surveillance purpose comes with big drawback, which is the working time or fly time of UAV. Due to the limitation of the battery capacity, the surveillance. Inherit this

advantage, using swarm of UAVs in surveillance can cover much larger areas in shorter times. Due to high number of drones using in swarm, then the time limitation problem can be solved. The applications of multi-UAVs in security and surveillance can bee found in (Ahmadzadeh et al., 2006; Petrlík, Vonásek, and Saska, 2019; Ding, Rahmani, and Egerstedt, 2010; Nigam et al., 2012; Scherer and Rinner, 2020)

- Environmental Monitoring: similar to security and surveillance applications, swarm of UAVs can be used for environmental monitoring applications by equipping with some special sensors. For example, the swarm of UAV is used for flood monitoring in (Abdelkader et al., 2014).

- Collaborative Transportation: transportation and goods delivery are one of the most potential wide used applications of UAV, especially for package delivery in the logistic sectors. However, using single UAV for transportation presents a huge disadvantage due to the pay load capacity. Hence, deploying multi UAVs for collaborative transportation task can extend the play load of the object. Figure. 1.12 demonstrates some example of using multi UAV to execute a collaborative transportation task .



(a) Two quadrotors UAV during a collaborative transportation task in (Loianno and Kumar, 2018).

(b) Four quadrotors UAV and two quadrotors UAV for collaborative transportation task carrying a line-shape object (right) and EL-shape object (left) in (Mellinger et al., 2013)

Figure 1.12: Example of using multi UAV in collaborative transportation application

To ensure all the mentioned tasks, an efficient controller for the swarm of UAVs is needed.

1.3.2 Control of a Swarm UAV

To control a swarm of UAVs the called multi agent control is used. The main problems of multi agent study can be categorised as follows:

- Consensus problem: This is one of the most fundamental problems in multi-agent control. The consensus problem concerns the cooperative behavior of multi-agents, which involves reaching an agreement between the agents in the swarm to achieve a common goal. The study of consensus focuses on representing the relationships among agents in the swarm through mathematical models. For example, the communication topology of agents can be represented by a set of matrices using Graph Theory.

- Formation planing: It is similar to path planing in control problem, which aims to design the desired trajectory to achieve the control goal such as tracking, obstacle avoidance , etc. Formation planing focus on providing the desired formation for the system to achieve the desired purpose.

- Formation control: It is the most active areas in multi-agent research. The formation control focus on controlling the agents to move cohesively to perform cooperative task. This task is done by controlling the agent so that some constraints, such as the relative distance, angle and velocity between agents are achieved to obtain the desired formation. Then based on these relative relation, each agent will build its own desired trajectory to fit with the desired formation for the multi-agent system. In some study, lower controller such as position tracking for each agent is considered perfect. Hence the formation control can be done by just giving the desired trajectory for each agent.

Beside the problem of control design for highly nonlinear dynamic of UAV ans swarm of UAVs. Knowing the full state vector of UAV is also necessary as it can be contribute to ensure better controller performance for a UAV as well as a swarm of UAVs.

1.4 State estimation problem

The state vector of UAV is not fully measurable. For example, the attitude of UAV is not directly measured, but can be estimated from measurable values such as magnetic field, gyroscope by using TRIAD or QUEST algorithms (M.D.shuster and S.D.Oh, 1981). As a results, its derivative which is needed to build the controller can not be directly obtained. These estimations are also affected by the quality of the measured data provided by sensors. Nowadays, many low-cost IMUs have been developed to provide good measurements with a data fusion module embedded inside. However, these available IMUs still can not solve the problem of providing the full state vector of the UAV. Therefore observer design for state estimation become crucial.

In this context, sampling data, poor sensor synchronization, operational restrictions on the sensors and time delay measurements constitute major drawbacks for the purpose of the state estimation design. Despite its performance-degrading effect and the interest of this problem, in most of the existing state estimation methods, time delay and sampled-data measurements has been widely neglected.

Furthermore, in the context of multi agents system, its control structure lies on the communication between the agents. Therefore, the data transmission between agents is crucial. Indeed, due to the time taken for transmission signal, the latency in signal rounding between agents is inevitable. Hence, the control performance might be degraded, and in the extreme situations, they delay may cause the loss of stability. In particular, the consensus time delay occurs, when the protocols depend on the relative state information transmitted over the network, which connects the agents together. Therefore, the study of delay effects during the communication between the agents for control design is necessary. In (Wang et al., 2020; Zhang et al., 2019; Hua, Li, and Guan, 2019; Wang et al., 2017; Wang et al., 2019) the problem of delay effect for multi agent control design is investigated.

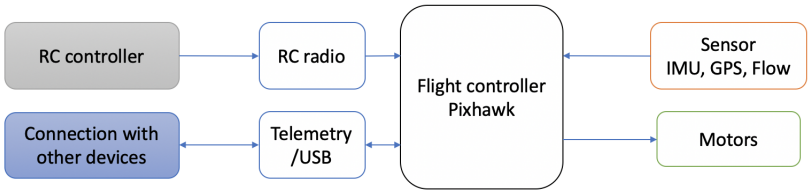


Figure 1.13: Diagram of using Pixhawk flight controller

The Figure. 1.13 shows a simple diagram of using Pixhawk flight controller for the research propose, as can be seen from this figure, the control board deploys some sensor such as IMU, GPS, Flow in order to obtained the state vector for the sake of controller design and these measurement are affected by noise, delay as reported by Px4 Firmware developers.

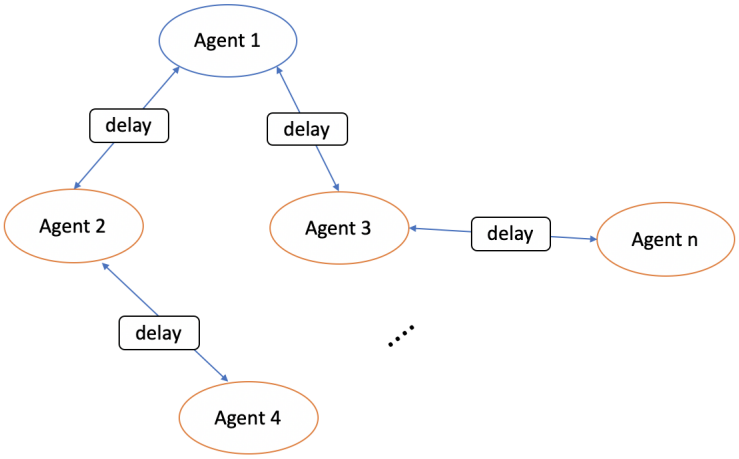


Figure 1.14: Illustrate example of delay effect in communication between agents

Figure. 1.14 illustrates an example of multi agents system with delay communication.

1.5 Contribution of the Thesis

Motivated by the interesting problems mentioned above, this thesis aims to address the control design problems for quadrotor UAVs and multi-agent systems. This work focuses on studying several crucial and related problems regarding control design, including observer design for nonlinear systems subjected to delay and sampled data measurements, adaptive control design for quadrotor UAV systems, and formation control for multi-agent systems subject to delayed communication between the agents in the swarm. The main contributions of this thesis are summarized as follows:

- To solve the problem of state estimation for nonlinear system subjected to delay, we propose an innovative observer structure under very restrictive conditions namely: noisy, sampled and delayed measurement. The proposed observer allows the reduction of noise effect and the compensation of the delay effect.
- An adaptive controller for quadrotor UAV is proposed for trajectory tracking task. In this study, we assume that the model and the external effect such as wind and aerodynamic drag, are unknown. Then the adaptive controllers will be developed in this thesis using adaptive neural network and fuzzy logic. The proposed controllers are well performed for trajectory tracking task, it is validated through simulation and experimental results.
- Finally, based on the successes of the controller design and observer design, the last contribution of this thesis is to develop a controller for controlling a swarm of UAV under communication delay between UAVs. This study assume that there exits a communication delay between the agents. Hence the transmissions between agents are always affected by a constant delay. Due to this lagging, the formation tracking control can not converge with tracking error to zero. In this study, under some specific conditions we proposed a controller for the formation tracking control so that the tracking error can converge to zero.

1.6 Organization of the Thesis

The structure of this thesis is summarized as follows:

- In Chapter 1, some applications of UAV are summarised and the motivation of this thesis is illustrated.
- In Chapter 2, the modeling method of a quadrotor UAV system is presented. The attitude presentation of UAV is firstly presented. Then the forces, torques and dynamic model of quadrotor UAV system will be studied. The chapter ended with a brief presentation of quadrotor UAV platform and the software which will be used in this thesis.
- In Chapter 3, the innovative methods for state estimation dealing with sampled, delay and noisy affect to output measurement are presented. The chapter firstly starts with a brief review of the state of the art for state estimation as well as dealing with sampled, delayed and noisy measurement. Then a novel structure of a observer, namely the Filtered High Gain Observer (FHGO) is firstly designed where the output measurements are only affected by the delay. Then, the proposed FHGO is extended to the case of sampled-delayed outputs. Using a LMI design approach, the observer parameters are then designed. The application for state estimation is then studied through the simulation and experimental results. To highlight the effectiveness and the superior of the proposed observers in this thesis, a comparison with Kalman-like observer and high gain observer is analysed.
- In Chapter 4: the design of the adaptive controllers for quadrotor UAV is presented. The uncertainties such that unknown inertia, mass and external disturbances, nonlinear aerodynamic forces are commonly challenging to design a model-based control scheme. The tracking process will be illustrated in this chapter. The proposed controller in this study is designed based on Sliding Mode technique. The different learning methods such as neural network and Fuzzy logic will be then presented and applied to improve the adaptive capacity of SMC controller. The control design method will be derived from the stability analysis by means of Lyapunov functional. The simulation and experimental are carried out to verify the performance and effectiveness of the proposed methods in comparison with the current existing methods.
- In Chapter 5: In this chapter, the problem of formation flight for a swarm of quadrotor UAVs with communication delays between the UAVs is analysed. The study in this chapter will focus

on overcome the problem of delay effect into the formation control performance for achieving the formation tracking task. Based on the controller and observer designed in the previous chapters, the study of designing a controller for position and speed control of one quadrotor UAVs has been well established. Then the problem of formation follow and maintenance under delayed communication is proposed using the decentralized leader-follower strategy. To illustrate the effective of the proposed work, a 3D simulation will be carried out.

CHAPTER 2

Quadrotor Platform

2.1 Mathematical Background and Quadrotor Modeling

UAVs have revolutionized the way we think about aerial operations, offering unprecedented capabilities and flexibility in a wide range of applications. One of the most versatile and agile types of UAVs is the quadrotor, a device that uses four rotors to generate lift and control its motion. Quadrotors have become increasingly popular in recent years, due to their ability to perform a wide range of tasks, from aerial photography and surveillance to search and rescue operations. To achieve these capabilities, quadrotors must be precisely controlled, using advanced algorithms and sensors to stabilize and guide their motion. This requires a deep understanding of the quadrotor's motion and behavior, which can be captured using mathematical models.

Mathematical modeling is a crucial aspect of engineering and design, allowing engineers to capture and analyze complex systems in a systematic and quantitative manner. In the case of quadrotors, mathematical models can help us to understand and predict the drone's motion and behavior, and to design control algorithms that can stabilize and control the quadrotor in flight. These models can capture the complex interactions between the drone's motors, propellers, and sensors, and can provide a powerful tool for analyzing and optimizing the quadrotor's performance.

The dynamics of quadrotors are inherently complex, involving a wide range of physical phenomena and interactions. To capture these dynamics, researchers have developed a range of mathematical models that describe the quadrotor's motion and behavior. These models typically consist of sets of differential equations that describe the motion of the drone in response to external forces and torques. The equations are derived from basic principles of physics, such as Newton's laws of motion and the conservation of energy, and can be solved numerically using simulation software.

One of the most widely used models of quadrotor dynamics is the rigid body model, which treats the drone as a rigid body with six degrees of freedom. This model captures the drone's motion and attitude using a set of six equations, which describe the drone's position and orientation in three-dimensional space. The equations are based on the forces and torques acting on the drone, which are generated by the rotors and other external factors such as wind and gravity. By solving these equations, we can predict the drone's motion and behavior under different conditions, and design control algorithms that can stabilize and guide the drone in flight.

Another important aspect of quadrotor modeling is the aerodynamics of the rotors and propellers. The behavior of the rotors and propellers is critical to the drone's flight, as they generate the lift and thrust necessary for flight. To model the aerodynamics of the rotors and propellers, researchers use a range of techniques, including computational fluid dynamics (CFD) and empirical models based on experimental data. These models capture the complex interactions between the rotor blades and the surrounding air, and can provide accurate predictions of the drone's lift and thrust characteristics.

In addition to the rigid body and aerodynamic models, quadrotor modeling also involves the development of control algorithms that can stabilize and guide the drone in flight. Control algorithms use feedback from sensors such as accelerometers, gyroscopes, and GPS to adjust the drone's motion and attitude in real time. These algorithms can be based on a range of techniques, including classical control theory, modern control theory, and machine learning. The choice of control algorithm depends on the

specific application and requirements of the quadrotor, and can have a significant impact on the drone's performance.

Unlike other ground mobile robots, quadrotor UAVs are highly nonlinear and coupled systems, operating with volatility and uncertainties caused by the external environment and aerodynamic phenomena. Moreover, the motions of quadrotor UAV systems include two main components: rotation and translation, which make the modeling task for quadrotor UAV systems more challenging.

In most of the work, attention is paid to the mathematical model in terms of rotation due to its complexity, while the model for translation can be easily obtained using Newton's second law. Hence, the state representation of a quadrotor UAV in terms of rotation is crucial. In this chapter, we will introduce the concept of reference frame presentation and attitude presentation, which will be used to present the state and study the dynamic model of a quadrotor UAV in terms of rotation.

In this chapter, we will give an overview of the dynamic model describing the flying quadrotor. Firstly, we will introduce the reference frame and the attitude presentation, which are the basic elements needed to define the state vector of the quadrotor system. Secondly, we will introduce the concepts of forces and moments applied to the quadrotor system, as these concepts are crucial for analyzing any dynamic system that includes a quadrotor system. Finally, we will present the dynamic model of the quadrotor UAV based on different methods.

2.1.1 Reference frame representation

a) Body-frame: The Body-frame $\{\boldsymbol{B}\}$ reference is associated to the vehicle and usually origins at the Center Of Mass (COM) as shown in Figure. 2.1. The axes of body frame are given as follows:

- x-axis: this axis directs along the longitudinal axis oriented from the rear towards the front.
- y-axis: this axis directs along the transverse axis oriented from left to right.
- z-axis: this axis completes the direct Cartesian coordinate following the rule of the right hand.

Figure 2.1: UAV body frame

b) Earth-frame:
The earth frame $\{\boldsymbol{E}\}$ define a fix coordinate at a predefined fixed point in the earth. This frame will be used to determine the global position and attitude of UAV. The axes of earth frame is usually given as follows:

- x-axis: this axis directs along the longitude axis oriented from south to north.
- y-axis: this axis directs along the latitude axis oriented from west to east.
- z-axis: this axis completes the Direct Cartesian coordinate following the rule of the right hand.

2.1.2 Attitude representation

The orientation of UAV in space is crucial especially in aerospace applications such as flight control design. In this section, an overview of different attitude presentation methods will be given.

a) Euler Angle

The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body with respect to a fixed coordinate system. They can also represent the orientation of a mobile frame of reference in physics or the orientation of a general basis in 3-dimensional linear algebra.

Euler angles can be defined by elemental geometry or by composition of rotations. The geometrical definition demonstrates that three composed elemental rotations (rotations about the axes of a coordinate system) are always sufficient to reach any target frame.

The three elemental rotations may be extrinsic (rotations around the axes xyz w.r.t $\{\boldsymbol{E}\}$, which is assumed to remain motionless), or intrinsic (rotations around the axes of the rotating coordinate system XYZ w.r.t $\{\boldsymbol{B}\}$, solidary with the moving body, which changes its orientation after each elemental rotation).

Considering a quadrotor as a solid body evolving in a three dimensional space and subject to the main vertical thrust and three torques. We will follow some conventions: the reference associates to the vehicle,

with its origin at the CoM of the quadrotor, will be the body frame $\{\boldsymbol{B}\}$, the reference which follows the rotation of the earth origin of this reference is the center of the earth, will be the earth frame $\{\boldsymbol{E}\}$. x, y, z will stands for the coordinate of the quadrotor w.r.t $\{\boldsymbol{E}\}$. In other words, x, y, z are the position of the quadrotor. $\dot{x}$, $\dot{y}$, $\dot{x}$, will stands for translation velocities of quadrotor w.r.t $\{\boldsymbol{E}\}$. $i^E = \begin{bmatrix} i_x^E & i_y^E & i_x^E \end{bmatrix}$ represents the unit vector of three coordinate axes respect to $\{\boldsymbol{E}\}$, $i^B = \begin{bmatrix} i_x^E & i_y^E & i_x^E \end{bmatrix}$ represent the unit vector of three coordinate axes w.r.t $\{\boldsymbol{B}\}$. The relation between i^E and i^B is expressed as

$$i^E = Ri^B \tag{2.1}$$

Where R is the rotation matrix which expresses the transformation from the body frame to the earth frame. let define $i_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$, $i_y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$, $i_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$

The rotation matrix R is derived from the combination of rotations around axes. In what follows we will give the standard definition of the rotations about the three principle axes:

The rotation around x axis with rotation angle φ, namely *Roll* rotation is defined such that:

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{2.2}$$

The rotation around y axis with rotation angle θ, namely $Pitch$ rotation is defined such that:

$$R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{2.3}$$

The rotation around z axis with rotation angle ψ, namely Yaw rotation is defined such that:

$$R_z(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The rotation matrix R then can be thought of a sequence of three rotations, one about each principle axis, without considering the possibility of using two different conventions for the definition of the rotation axes (intrinsic or extrinsic) Table.2.1. There exist twelve possible sequences of rotation axes, divided in two groups:

Proper Euler angles (z-x-z, x-y-x, y-z-y, z-y-z, x-z-x, y-x-y).

Tait–Bryan angles (x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z)

The Table.2.1 shows us the rotation matrix calculated by different sequences.

Proper Euler angles		Tait–Bryan angles	
XZX=	$\begin{bmatrix} c_2 & -c_3s_2 & s_2s_3 \\ c_1s_2 & c_1c_2c_3 - s_1s_3 & -c_3s_1 - c_1c_2s_3 \\ s_1s_2 & c_1s_3 + s_1c_2c_3 & c_1c_3 - s_1c_2c_3 \end{bmatrix}$	XZY=	$\begin{bmatrix} c_2c_3 & -s_2 & c_2s_3 \\ s_1s_3 + c_1s_2c_3 & c_1c_2 & c_1s_2s_3 - s_1c_3 \\ s_1s_2c_3 - c_1s_3 & c_2s_1 & c_1c_3 + s_1s_2s_3 \end{bmatrix}$
XYX=	$\begin{bmatrix} c_2 & s_2s_3 & s_2c_3 \\ s_1s_2 & c_1c_3 - s_1c_2c_3 & -c_1s_3 - s_1c_2c_3 \\ -c_1c_2 & s_1c_3 + c_1c_2s_3 & c_1c_2c_3 - s_1s_3 \end{bmatrix}$	XYZ=	$\begin{bmatrix} c_2c_3 & -c_2c_3 & s_2 \\ c_1s_3 + s_1s_2c_3 & c_1c_3 - s_1s_2s_3 & -c_2s_1 \\ s_1s_3 - c_1s_2c_3 & s_1c_3 + c_1s_2s_3 & c_1c_2 \end{bmatrix}$
YXY=	$\begin{bmatrix} c_1c_3 - s_1c_2c_3 & s_1s_2 & c_1s_3 + s_1c_2c_3 \\ s_2s_3 & c_2 & -s_2c_3 \\ -s_1c_3 - s_1c_2c_3 & c_1s_2 & c_1c_2c_3 - s_1s_3 \end{bmatrix}$	YXZ=	$\begin{bmatrix} c_1c_3 + s_1s_2s_3 & s_1s_2c_3 & s_1c_2 \\ c_2s_3 & c_2c_3 & -s_2 \\ c_1s_2s_3 - s_1c_3 & c_1s_2c_3 + s_1s_3 & c_1c_2 \end{bmatrix}$
YZY=	$\begin{bmatrix} c_1c_2c_3 - s_1s_3 & -c_1s_2 & s_1c_3 + c_1c_2s_3 \\ s_2c_3 & c_2 & s_2s_3 \\ -c_1s_3 - s_1c_2c_3 & s_1s_2 & c_1c_3 - s_1c_2s_3 \end{bmatrix}$	YZX=	$\begin{bmatrix} c_1c_2 & s_1s_3 - c_1s_2c_3 & s_1c_3 + c_1s_2s_3 \\ s_2 & c_2c_3 & -c_2s_3 \\ -s_1c_2 & c_1s_3 + s_1s_2c_3 & c_1c_3 - s_1s_2s_3 \end{bmatrix}$
ZYZ=	$\begin{bmatrix} c_1c_3 - s_1c_2s_3 & -c_1c_3 - c_1c_2s_3 & c_1s_2 \\ c_1s_3 + s_1c_2c_3 & c_1c_3 - s_1c_2s_3 & s_1s_2 \\ -s_2c_3 & s_2s_3 & c_2 \end{bmatrix}$	ZYX=	$\begin{bmatrix} c_1c_2 & c_1s_2s_3 - s_1c_3 & s_1s_3 + c_1s_2c_3 \\ s_1c_2 & c_1c_3 + s_1s_2s_3 & s_1s_2c_3 - c_1s_3 \\ -s_2 & c_2s_3 & c_2c_3 \end{bmatrix}$
ZXZ=	$\begin{bmatrix} c_1c_3 - s_1c_2s_3 & -c_1s_3 - s_1c_2c_3 & s_1s_2 \\ s_1c_3 + c_1c_2s_3 & c_1c_2c_3 - s_1s_3 & -c_1s_2 \\ s_2s_3 & s_2c_3 & c_2 \end{bmatrix}$	ZXY=	$\begin{bmatrix} c_1c_3 - s_1s_2s_3 & -s_1c_2 & c_1s_3 + s_1s_2c_3 \\ s_1c_3 + c_1s_2s_3 & c_1c_2 & s_1s_3 - c_1s_2c_3 \\ -c_2s_3 & s_2 & c_2c_3 \end{bmatrix}$

Table 2.1: Rotation matrix in different sequences of rotation axes

Tait–Bryan convention is widely used in engineering with different purposes. There are several axes conventions in practice for choosing the mobile and fixed axes, and these conventions determine the signs of the angles. For an aircraft, they can be obtained with three rotations around its principal axes if done in the proper order. A *yaw* will obtain the bearing, a *pitch* will yield the elevation and a *roll* gives the bank angle. Therefore, in aerospace they are sometimes called *yaw*, *pitch* and *roll*. To prevent cumbersome, we use XYZ to represents the rotation matrix $R = R_x R_y R_z$, the rotation angle will be 1, 2 and 3 will represent the first, second and thirst rotation angles and 1, 2 and 3 stand for φ, θ and ψ respectively. For example, rotation XYZ and XYX can be computed as follows:

$$XYZ: \ R_z(1)R_y(2)R_z(3) = R_x(\varphi)R_y(\theta)R_z(\psi); \ XYZ: \ R_x(1)R_y(2)R_x(3) = R_x(\varphi)R_y(\theta)R_z(\psi)$$

b) Axis Angle

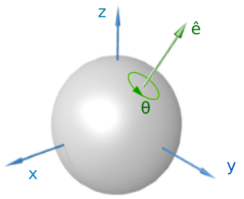


Figure 2.2: A rotation represented by Axis Angle

By Rodrigues' rotation formula, the angle and axis determine the transformation that rotates three-dimensional vectors. The rotation occurs in the sense prescribed by the right-hand rule. The rotation axis is sometimes called the Euler axis. It is one of many rotation formalisms in three dimensions.

The axis-angle representation is a way of representing rotations in three-dimensional space using an axis and an angle. The axis represents the direction of the rotation, and the angle represents the magnitude of the rotation around that axis.

The rotation can then be expressed as a combination of a rotation matrix and a translation vector. The rotation matrix is constructed using the chosen axis and angle, and the translation vector is typically set to zero.

One advantage of the axis-angle representation is that it is very intuitive and easy to visualize. For example, you can imagine spinning a top around its axis, and the axis and angle of rotation would be the same as those used in the axis-angle representation.

Another advantage of the axis-angle representation is that it is easy to convert to other representations of rotations, such as quaternions or Euler angles. Additionally, it is often used in robotics and computer graphics applications, where it is necessary to represent rotations in a way that is easy to manipulate and understand.

The axis–angle representation is equivalent to the more concise rotation vector, also called the Euler vector. In this case, both the rotation axis and the angle are represented by a vector $\hat{e}$ with the rotation axis whose length is the rotation angle θ (see Figure .2.2). Therefore a rotation k can be represented as

$$k = \hat{e}\theta \tag{2.5}$$

The axis–angle representation is convenient when dealing with rigid body dynamics. It is useful to both characterize rotations, and also for converting between different representations of rigid body motion, such as homogeneous transformations and twists.

c) Quaternion

Quaternions are "hypercomplex" numbers, which means that they have three imaginary parts i, j, k contrary to the traditional representation of complex number.

Quaternions are commonly used in computer graphics, robotics, and physics for their ability to represent rotations and orientations in three-dimensional space. They offer advantages over other methods of representing rotations, such as Euler angles, because they avoid the problem of gimbal lock.

One of the key features of quaternions is that they have a well-defined multiplication operation, which allows for easy composition of rotations. Another useful property is that they can be used to interpolate between two rotations, which is useful in animation and other applications.

A quaternion is a four tuple that belongs to the quaternion space and can be seen as a number that contains four parts one real part and three imaginary parts corresponding to imaginary units i, j, k.

$$\begin{aligned} &Q = Q_0 + Q_1 i + Q_2 j + Q_3 k; Q \in \mathbb{H} \\ &Q_0, Q_1, Q_2, Q_3 \in \mathbb{R} \\ &i, j, k \in \mathbb{I} \end{aligned} \tag{2.6}$$

In order to simplify the expression of quaternions in above equation, we express three different imaginary parts as a vector in $\mathbb{R}^3$. Hence $\mathbb{R}^3$ can be seen as a subspace of quaternions space. Accordingly, a

presentation is obtained as:

$$Q = \begin{bmatrix} Q_0 \\ \bar{Q} \end{bmatrix}; \bar{Q} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \tag{2.7}$$

where Q stands for quaternion, q_0 is the scalar part of Q, $\bar{q}$ are the complex vectorial part of Q

The most used operation in quaternion algebra in this research is the quaternion product and inverse quaternion. We assume P and Q is two quaternion number, then quaternion product of $P \in \mathbb{H}$ and $Q \in \mathbb{H}$ is defined as:

$$Q \bigotimes P = \begin{bmatrix} Q_0 P_0 - \bar{Q}^T \bar{P} \\ P_0 \bar{Q} + Q_0 \bar{P} + \bar{P} \times \bar{Q} \end{bmatrix} \tag{2.8}$$

The quaternion conjugate of Q is defined as Q^*, then the quaternion inverse Q^{-1} will be obtained from its conjugate as follows:

$$Q^* = \begin{bmatrix} Q_0 \\ -\bar{Q} \end{bmatrix}; Q^{-1} = \frac{Q^*}{\|Q\|} \tag{2.9}$$

where $\|Q\|$ is the quaternion norm, which is defined as:

$$\|Q\| = \sqrt{Q_0^2 + Q_1^2 + Q_2^2 + Q_3^2} \tag{2.10}$$

As a result, if Q is unit quaternion, which means $\|Q\| = 1$, then $Q^{-1} = Q^*$

The rotation of a $3D$ vector from one body frame to inertial can be obtained by quaternion product. Let us consider vector $b_i \in \mathbb{R}^3$ w.r.t body frame and its quaternion is defined with the scalar part is 0 and the complex part is b_i. Then the representation of r_i in inertial frame then be obtained such that:

$$\begin{bmatrix} 0 \\ r_i \end{bmatrix} = Q \otimes \begin{bmatrix} 0 \\ b_i \end{bmatrix} \otimes Q^{-1} \tag{2.11}$$

Base on the quaternion product properties mentioned above, we can deduce:

$$\begin{aligned} r_i &= Q_0^2 b_i + Q_0 \bar{Q} x b_i + \bar{Q}\bar{Q}^T b_i + Q_0 \bar{Q} \times b_i - \bar{Q} \times b_i \times \bar{Q} \\ &= (Q_0^2 + 2Q_0 S_{(\bar{Q})} + \bar{Q}\bar{Q}^T + S_{(\bar{Q})}^2) b_i \\ &= (I_d + 2Q_0 S_{(\bar{Q})} + 2S_{(\bar{Q})}^2) b_i \end{aligned} \tag{2.12}$$

Where $S_{(x)}$ is skew matrix, which is defined as:

$$S_{(x)} = \begin{bmatrix} 0 & -x_z & x_y \\ x_z & 0 & -x_x \\ -x_y & x_x & 0 \end{bmatrix}; x = \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix} \tag{2.13}$$

A rotation matrix R can be calculated based on quaternion as follows:

$$R(Q) = I_d + 2Q_0 S_{(\bar{Q})} + 2S_{(\bar{Q})}^2 \tag{2.14}$$

It is easy to verify that $R(Q) = R(-Q)$. Therefore R defines two convering map from $\mathbb{H}$ to $SO(3)$, then $R(Q)$ admits two rotation solutions by Q and $-Q$.

2.1.3 Force and moment applied to the quadrotor UAVs

The motion of a quadrotor is determined by the forces and moments acting on the quadrotor UAV, which are generated by the four rotors and other external factors such as wind and gravity. Understanding the dynamics of these forces and moments is crucial for developing accurate mathematical models of quadrotor motion and for designing control algorithms that can stabilize and guide the drone in flight.

To understand the dynamics of these forces and moments, we can use the principles of mechanics and physics. Newton's laws of motion, for example, tell us that the forces acting on an object determine its motion and acceleration. In the case of a quadrotor, we can use these laws to describe the forces acting on the drone and how they affect its translational motion. Similarly, the principles of angular momentum and torque can be used to describe the forces and moments acting on the quadrotor and how they affect its rotational motion.

In this section, we will give the explicit presentation of forces and moment applied on a quadrotor UAV.

a) Forces

The forces acting on a quadrotor can be broken down into three categories: Gravity forces, lift/thrust and hub forces and external force. Gravity forces are those that act in a straight line toward to the center of the earth. Lift/thrust and hub , on the other hand, are generated by the rotors. These forces determine the quadrotor's translational motion, including its speed and direction of travel. External forces are generated by aerodynamic drag and uncertainties.

Consider that the quadrotor UAV flying in a gravity environment in thesis context. Hence, there will be three main forces acting on the UAV as follow:

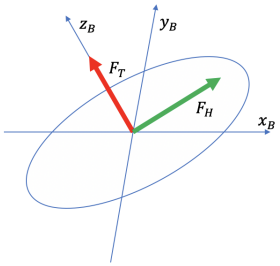


Figure 2.3: Thrust and Hub force acting of the UAV

• **Gravity Force:** Gravity force is along side with the z-axis (z_E) in Earth frame $\{\boldsymbol{E}\}$ and is given as:

$$F_G = -mg \tag{2.15}$$

where m is the total mass of the quadrotor UAV and g is the gravity constant.

• **Thrust and Hub force:** The thrust force and the hub force are generated by a rotor as shown in Figure 2.3, and are depended on the rotation velocity Ω_i of the rotor. The thrust force F_{Ti} is along side with the z-axis (z_B) in Body frame $\{\boldsymbol{B}\}$, and the rest of the force is in the plan x_B, y_B. The hub force F_{Hi} which is on the negative direction of the horizontal velocity, can be decomposed into two components, namely, F_{Hix} and F_{Hiy}. The explicit presentations of thrust force and hub force generated by one rotor are:

$$F_{Ti} = c_T \rho A (\Omega_i r_r)^2; \; F_{Hi} = c_H \rho A (\Omega_i r_r)^2 \tag{2.16}$$

where c_T and c_H are aerodynamic coefficients, ρ is the air density, A is the effective propeller disk area and r_r is its radius.

Equivalently, the total thrust force and hub force applied to a quadrotor UAV are given as:

$$F_T = \sum_{i=1}^{4} F_{Ti}; \; F_{Hx} = -\sum_{i=1}^{4} F_{Hix}; \; F_{Hy} = -\sum_{i=1}^{4} F_{Hiy} \tag{2.17}$$

• **External force:** The quadrotor UAV in some specific condition such as flying outdoor will be affected by the environment. The total external force acting on the quadrotor UAV can be considered as unknown elements and affect the UAV in three axes as following:

$$F_\delta = \begin{bmatrix} F_{\delta x} & F_{\delta y} & F_{\delta z} \end{bmatrix}^T \tag{2.18}$$

Consequently, the total force vector on a quadrotor UAV in the Earth frame $\{\boldsymbol{E}\}$ can be given as:

$$\sum F = R \begin{bmatrix} F_{Hx} \\ F_{Hy} \\ F_T \end{bmatrix} + F_\delta - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.19}$$

where R is a rotation matrix.

b) Moments

Together with forces, moments acting on the UAV generate a rotation movement, which is one of the two main movement of UAV, namely, translation and rotation. They contribute to the control of the vehicle. In this part, we will present the main moment applied on UAV and their direction in Body-frame $\{\boldsymbol{B}\}$.

• **Roll moment:** Roll moment τ_ϕ generates the rotation movement around x-axis in $\{\boldsymbol{B}\}$. This moment is achieved by the difference in the combined thrust forces acting on the UAV. The direction of roll moment is decided according to the right hand rule. It is defined as follows:

$$\tau_\phi = l_1 \sin(\alpha_1) F_{T1} - l_2 \sin(\alpha_2) F_{T2} + l_3 \sin(\alpha_3) F_{T3} - l_4 \sin(\alpha_4) F_{T4} \tag{2.20}$$

In the ideal case, where the quadrotor UAV is supper symmetric, one gets $l_1 = l_2 = l_3 = l_4 = l$ and $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha$, hence the roll moment will be rewritten as follow:

$$\tau_\phi = l \sin(\alpha) \left(F_{T1} - F_{T2} + F_{T3} - F_{T4}\right) \tag{2.21}$$

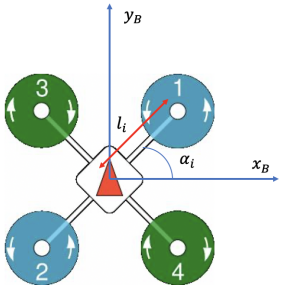


Figure 2.4: Rotation description of a quadrotor UAV

• **Pitch moment:** Pitch moment τ_θ generates the rotation movement around y-axis in $\{\boldsymbol{B}\}$. Similar to roll moment, the direction of pitch moment is decided according to the right hand rule. It is defined as follows:

$$\tau_\theta = -l_1 \cos(\alpha_1) F_{T1} + l_2 \cos(\alpha_2) F_{T2} + l_3 \cos(\alpha_3) F_{T3} - l_4 \cos(\alpha_4) F_{T4} \tag{2.22}$$

in the ideal symmetrical case, the pitch moment will be rewritten as follow:

$$\tau_\theta = l \cos(\alpha) \left(-F_{T1} + F_{T2} + F_{T3} - F_{T4}\right) \tag{2.23}$$

• **Yaw moment:** Yaw moment τ_θ generates the rotation movement around z-axis in $\{\boldsymbol{B}\}$. Different to roll and pitch moments, the yaw moment is achieved by the aerodynamic forces acting on the blade elements rather than the thrust force. All rotors induce the rotation of quadrotor UAV around z-axis and provide the yaw moment. Each rotor generates a rotation moment as follow:

$$\tau_{\psi i} = c_\psi \rho A (\Omega_i r_r)^2 \tag{2.24}$$

where c_ψ is the aerodynamic coefficients.

Based on the movement of each rotor described in Figure .2.4, the total yaw moment acting of the quadrotor UAV is derived as follow:

$$\tau_\psi = \tau_{\psi 1} + \tau_{\psi 2} - \tau_{\psi 3} - \tau_{\psi 4} \tag{2.25}$$

• **Hub moment:** As previously analysed, the hub force acting on the UAV in x and y axes, will logically generate a hub moment in z-axis. Whose expression is given as:

$$\tau_H = \sum_{i=1}^{4} l_i \sin(\alpha_i) F_{Hxi} - \sum_{i=1}^{4} l_i \cos(\alpha_i) F_{Hyi} \tag{2.26}$$

• **Blade flapping moment:** During the flight mode, when the quadrotor UAV replaces from one point to the other, the advancing blade experiences a higher velocity relative to the air than the retreating blade. This causes the advancing blade has more lift than the retreating blade. This difference of lift leads to a gyroscope response and tilts the rotor disk back. Hence the blade flapping moment is perpendicular to the plane formed by the rotor shaft and the forward velocity of the UAV. The blade flapping moment of each rotor can be expressed as follows:

$$\tau_{bi} = c_b \rho A (\Omega_i r_r)^2 r_r \tag{2.27}$$

where c_b is the aerodynamic coefficients.

The total blade flapping moment can be presented in x and y axis as following:

$$\tau_{bx} = \tau_{bx1} + \tau_{bx2} - \tau_{bx3} - \tau_{bx4};\ \tau_{by} = \tau_{by1} + \tau_{by2} - \tau_{by3} - \tau_{by4} \tag{2.28}$$

• **Gyroscopic moment:** The gyroscopic effect exists when there are two rotations in two axes. These rotation cause a third rotation, which is perpendicular to the plane formed by the two former axes. The motor rotation itself is fixed in z-axis in body-frame. Hence together with the quadrotor rotation, there will be two gyroscopic moment affecting the x and y axis. These moments are given as follows:

$$\tau_{gx} = J_r \omega_y \left(\Omega_1 + \Omega_2 - \Omega_3 - \Omega_4\right);\ \tau_{gy} = -J_r \omega_x \left(\Omega_1 + \Omega_2 - \Omega_3 - \Omega_4\right) \tag{2.29}$$

where J_r is the inertia moment of a rotor.

• **External moment:** As previously mentioned, when the quadrotor UAV is in some specific condition such as flying outdoor, these conditions will have several effects. External forces acting on the quadrotor UAV can be considered as unknown elements, these forces respectively generate external moments in three axes as following:

$$\tau_\delta = \begin{bmatrix} \tau_{\delta x} & \tau_{\delta y} & \tau_{\delta z} \end{bmatrix}^T \tag{2.30}$$

The total moment acting on a quadrotor UAV has the following expression :

$$\sum \tau = \begin{bmatrix} \tau_\phi + \tau_{bx} + \tau_{gx} + \tau_{\delta x} \\ \tau_\theta + \tau_{by} + \tau_{gy} + \tau_{\delta y} \\ \tau_p si + \tau_H + \tau_{\delta z} \end{bmatrix} \tag{2.31}$$

2.1.4 Dynamic model of a quadrotor UAVs

The Dynamic model of a quadrotor UAV is a mathematical representation of its motion and behavior. This model considers various factors such as the drone's weight, thrust, and moments of inertia, and uses them to predict the drone's position, velocity, and attitude. The dynamic model allows engineers and researchers to better understand the behavior of quadrotor UAVs, and to design and optimize their control systems. By using this model, we can simulate the quadrotor's flight in a virtual environment, predict its response to different inputs, and evaluate its performance. Overall, the dynamic model of a quadrotor UAV is a critical tool for developing and improving the capabilities of these aerial vehicles.

The first approaches is used to build the mathematical model that describe the relationship between the force and the motion of UAVs is the Euler-Lagrange method due to its simplicity and easy to implement. In (González-Hernández et al., 2017) using this model, the author design an nonlinear controller for a quadrotor UAV system. Newton-Euler is also widely used alternative method to build the dynamic model for quadrotor UAV system. In (Safeer Ullah and Iqbal, 2020),(Chen et al., 2016) this model is exploited to develop the controller, in (Dam et al., 2021), (Dam et al., 2022) the authors studied the nonlinear observer with application to quadrotor UAV using this model. The common point of these methods is that they lied on the classical Euler angles to present the model of UAV. The simplicity and easy to implement of this method are obvious, however using Euler angles presents some problems, such as singularities, and gimbal locks.

Quaternion dynamic is an alternative solution for representing the rotation motion of quadrotor, this method provides great advantages compared to using Euler angles such as lack of singularities, and gimbal locks. However, quaternion product has a different property compared to vector product, which is used to describe the translation, hence the calculation when multiple rotations and translations is more complicated.

In this section, one shall give a brief introduction about these methods.

a) Euler-Lagrange

The movement of a quadrotor system in three-dimensional space is governed by the main thrust force and three torques. In this study, we examine how a quadrotor is represented in 3D space using the vehicle's coordinates denoted by $p = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$, which indicates the position of the center of mass relative to the earth frame $\boldsymbol{E}$, and the vehicle's attitude represented by Euler angles $\eta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in \mathbb{R}^3$, indicating the roll, pitch, and yaw angles relative to the earth frame $\boldsymbol{E}$. Therefore, the Lagrangian equation can be expressed as follows:

$$L(p, \eta) = K - P \tag{2.32}$$

where $K = K_t + K_r$ represents the total kinetic energy, which can be further divided into translational kinetic energy, $K_t = \frac{m}{2}\dot{p}^T\dot{p}$, and rotational kinetic energy, $K_r = \frac{1}{2}\omega^T J\omega$. The potential energy is represented by $P = mgi_z{}^T p$, where m denotes the total mass of the quadrotor, and g is the acceleration due to gravity. The angular velocity of the quadrotor is denoted by $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T \in \mathbb{R}^3$ with respect to the body frame $\boldsymbol{B}$, and $J = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix} \in \mathbb{R}^{3\times3}$ represents the moment of inertia of the quadrotor.

It is worth noting that the first derivative of η, namely $\dot{\eta}$, and the angular velocity ω are expressed in different frames. In order to derive the dynamic equation, it is necessary to establish a transformation between the angular velocity of the body frame, ω, and the first derivative of η, which can be expressed as follows:

$$\begin{aligned} \omega = T_{(\eta)}\dot{\eta}, \qquad & \qquad \dot{\eta} = T_{(\eta)}^{-1}\omega \\ T_{(\eta)} = \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\phi) & s(\phi)c(\theta) \\ 0 & -s(\phi) & c(\phi)c(\theta) \end{bmatrix}, \quad & T_{(\eta)}^{-1} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)/c(\theta) & c(\phi)/c(\theta) \end{bmatrix} \end{aligned} \tag{2.33}$$

where $c(.) = cos(.)$, $s(.) = sin(.)$, $t(.) = tan(.)$. Then one gets the total kinetic energy is given as:

$$K = \frac{m}{2}\dot{p}^T\dot{p} + \frac{1}{2}\dot{\eta}^T T_{(\eta)}^T J T_{(\eta)}\eta \tag{2.34}$$

Using the Euler–Lagrange formulation one gets:

$$\frac{d}{dt}\frac{\partial L}{\partial p} - \frac{\partial L}{\partial p} = \sum F - mgi_z \tag{2.35}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \eta} - \frac{\partial L}{\partial \eta} = \sum \tau \tag{2.36}$$

where $\sum F$ is the total force (include gravity) applied to the quadrotor relative to $\{\boldsymbol{E}\}$, $\sum \tau$ is the total torques acting on the UAV. These force and moment are analysed in the previous part.

By replacing (2.32) into (2.35) and (2.36) we have:

$$m\ddot{p} - mgi_z = \sum F - mgi_z \tag{2.37}$$

$$T_{(\eta)}^T J T_{(\eta)} \ddot{\eta} + \left[\frac{d}{dt}\left(T_{(\eta)}^T J T_{(\eta)}\right) - \frac{1}{2}\frac{\partial}{\partial \eta}\left(\dot{\eta}^T T_{(\eta)}^T J T_{(\eta)}\right)\right]\dot{\eta} = \sum \tau \tag{2.38}$$

Leading to

$$m\ddot{p} = \sum F \tag{2.39}$$

$$T_{(\eta)}^T J T_{(\eta)} \ddot{\eta} = \sum \tau - C_r \dot{\eta} \tag{2.40}$$

$C_r = \frac{d}{dt}\left(T_{(\eta)}^T J T_{(\eta)}\right) - \frac{1}{2}\frac{\partial}{\partial \eta}\left(\dot{\eta}^T T_{(\eta)}^T J T_{(\eta)}\right) \in \mathbb{R}^{3\times 3}$ is the Coriolis matrix, whose components are

$$\begin{cases} C_{11} & = 0 \\ C_{12} & = (J_{yy} - J_{zz})[\dot{\theta}c(\phi)s(\phi) + \dot{\psi}c(\theta)(s^2(\phi) - c^2(\phi))] - J_{xx}\dot{\psi}c(\theta) \\ C_{13} & = (J_{zz} - J_{yy})\dot{\psi}c^2(\theta)c(\phi)s(\phi) \\ C_{21} & = (J_{zz} - J_{yy})[\dot{\theta}c(\phi)s(\phi) + \dot{\psi}c(\theta)(s^2(\phi) - c^2(\phi))] - J_{xx}\dot{\psi}c(\theta) \\ C_{22} & = (J_{zz} - J_{yy})\dot{\psi}c(\phi)s(\phi) \\ C_{23} & = -J_{xx}\dot{\psi}c(\theta)s(\theta) + J_{yy}\dot{\psi}s^2(\phi)c(\theta)s(\theta) + J_{zz}\dot{\psi}c^2(\phi)c(\theta)s(\theta) \\ C_{31} & = (J_{yy} - J_{zz})\dot{\psi}c^2(\theta)c(\phi)s(\phi) - J_{xx}\dot{\theta}c(\theta) \\ C_{32} & = (J_{yy} - J_{zz})[\dot{\theta}c(\phi)s(\phi) - \dot{\phi}c(\theta)(s^2(\phi) - c^2(\phi))] - C_{23} \\ C_{33} & = (J_{yy} - J_{zz})\dot{\phi}c(\phi)s(\phi)c^2(\theta) - J_{yy}\dot{\theta}s^2(\phi)c(\theta)s(\theta) - J_{zz}\dot{\theta}c^2(\phi)c(\theta)s(\theta) \\ & \quad + J_{xx}\dot{\theta}c(\theta)s(\theta) \end{cases}$$

b) Newton-Euler

Let us consider the CoM of quadrotor placed at original of $\{\boldsymbol{B}\}$. So the dynamic of quadrotor is considered as this point, then its dynamic equation will be given using the Newton-Euler as:

$$m\ddot{p} = \sum F \tag{2.41}$$

$$J\dot{\omega} = S_{(\omega)}J\omega + \sum \tau \tag{2.42}$$

Tt is worth noting that $\sum F$ represents the total force acting on the vehicle relative to $\{\boldsymbol{E}\}$ and given by (2.19). In the experiment, the drag coefficients which is used to determine the hub force are not always available. For the sake of controller design, we will consider the hub force element as unknown elements then the total force will be rewritten as follows:

$$\sum F = \begin{bmatrix} cos(\phi)sin(\theta)cos(\psi) + sin(\phi)sin(\psi) \\ cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi) \\ -cos(\phi)cos(\theta) \end{bmatrix} F_T - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} \delta_{Fx} \\ \delta_{Fy} \\ \delta_{Fz} \end{bmatrix} \tag{2.43}$$

where δ_{Fx}, δ_{Fz}, δ_{Fz} are the combination external disturbances and the projection of hub forces into the earth frame.

c) Quaternion

The dynamic equation that represents the motion of quadrotors using quaternion is given as (Abaunza, Castillo, and Lozano, 2018):

$$m\ddot{p} = Q \bigotimes \sum F \bigotimes Q^* \tag{2.44}$$

$$\dot{Q} = \frac{1}{2} Q \bigotimes \omega \tag{2.45}$$

$$\dot{\omega} = J^{-1}S_{(\omega)}J\omega + J^{-1}\sum \tau \tag{2.46}$$

With this method, we obtain a simpler model and without singularity point exit. However, the quaternion product exhibits the main drawback while the position of the vehicle is in $\mathbb{R}^3$.

2.2 Quadrotor UAV platform

In this section, we will present the detail of hardware components, which will be used to assembly a quadrotor UAV. The aim was to build a relatively low cost and light-weight quadcopter in order to perform some experiments to validate the proposed works.

Following by the hardware introduction, the software, which will be used to study the quadrotor UAV application, will be then introduced.

2.2.1 Hardware components

(a) S500 frame with gears

(b) Pixhawk 4 board

(c) 2216/880kv brushless motors

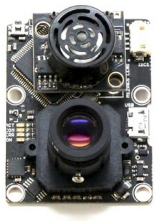
(d) 1045 propellers

(e) 40A ESC 2-6S OPTO

(f) LiPo 6S 4000mAh battery

(g) Power module PM02

(h) GPS UBLOX NEO-M8N



(i) PX4Flow

(j) 433MHz telemetry module

(k) Taranis X9-Lite

(l) Radio receiver

(m) Anti-vibration support

Figure 2.5: Hardware component to assembly the quadrotor UAV

Through out this thesis, a S500 quadrotor UAV will be used in the application for validating the proposed work in this thesis. The hardware components to assembly this platform are introduced as following:

- S500 frame: The first step is to choose a frame that can accommodate a Pixhawk 4 control card. The S500 frame (Figure 2.5a) appears to be a good option, considering the aforementioned criteria, price, and weight. The arms are made of ABS plastic, which provides robustness, while the two plates are made of fiberglass. The bottom plate includes an integrated Power Distribution (PDB). This quadrotor X airframe allows for the construction of a Do It Yourself (DIY) drone with a total flight weight ranging between 800 and 1200 grams. Additionally, landing gears are added for smooth landings and to allow attachment of a sensor beneath the bottom plate.

- Pixhawk 4 control board: Pixhawk [1] is an open source autopilot designed for autonomous systems. Originally developed by students at ETH Zurich, it ended up being a reference for autonomous systems in research and industry. It is composed of a flight controller board, a firmware and a ground control station application named QGroundControl [2]. As an open-source platform, Pixhawk has also been integrated with other tools and software, including Matlab/Simulink. In this study, we will be using the Pixhawk 4 flight controller board (Figure 2.5b), which is one of the models developed in collaboration with various manufacturers. Manufactured by Holybro, this compact case (44x84x12mm) is based on the Flight Management Unit (FMU) v5 architecture. It is equipped with the NuttX operating system, capable of running the PX4 firmware in real-time. The board features two processors: a main processor and an auxiliary processor dedicated to input/output processing. Additionally, it incorporates four built-in sensors, including two gyroscopes, a magnetometer, and a barometer. The board also provides various input/output interfaces for connecting communication modules, sensors, motor controllers, and the included power management module (PM02).

- Brushless motors: The motors recommanded for the frame are 2212 or 2216 brushless motors (i.e 22mm diameter and 12 or 16mm height). Thus, Holybro 2216/880kv brushless motors are chosen (Figure 2.5c), they consume 16.2A at full power. Four motors are requiered with two of them spinning clockwise and the two other spinning counter-clockwise.

- Propellers: The recommended dimensions of propellers are 1045 (i.e 10cm diameter and 45mm step) so two sets of 4 Holybro 1045 propellers (Figure 2.5d) has been taken for redundancy. Two can turn clockwise and two counter-clockwise and so they must be attached to the corresponding motors.

- ESCs (Electronic Speed Controllers): This component is for control the brushless motors. The two main characteristics of an ESC are the current that it can handled and the number of cells/stages in the power supply mentioned before the S letter. The ESCs must be at least of 20A calibre to support the maximum current of the motors. As the reseller was out of stock of 20A and 30A ESCs, four 40A ESC 2-6S OPTO from Readystosky were taken (Figure 2.5e). This over-calibration involves a small change in the connectors of the motors to be plugged in the ESCs. Moreover as they are 2-6S (number of cells/stages of the battery), it lets a quite large freedom. The control wires will be connected to the I/O PWM OUT port of the Pixhawk 4.

- Batteries: The batteries can now be picked up. Their capacity is given in mAh and the number of stage is mentioned before the S letter as for the ESCs. Large capacity battery (more than 3000mAh) would certainly give more autonomy. On the contrary, a low capacty battery (less than 1500mAh) would not give enough autonomy and would constrained to short flights and frequent charging. For this reason, two Tattu LiPo 6S 4000mAh (Voltage: 22.2V) batteries were used (Figure 2.5f), one to flight and one for redundancy.

- Power module PM02 (Figure 2.5g):The Pixhawk 4 board comes with the **power module** PM02 (Figure 2.5g) due to its small size. It keeps the adaptability advantage but will ask some welds to the PDB bottom plate of the frame. It powers the board by the POWER1 port.

- GPS sensor: GPS usually comes with the Pixhawk 4 as it is optimized for it, the UBLOX NEO-M8N (Figure 2.5h). It was not included in the reseller offer so it has been bought separately. It has a tricolor LED, a safety switch to allow arming the drone and a compass. As it is sensible to the electromagnetic interferences, it has to be mounted as far as possible from the motors power supply cables thanks to the provided mast. Indoor, as the coordinates are not available, only the safety switch and the LED indicator are useful. The horizontal precision is about 2m, the velocity precision is 0.05m/s and the heading precision is 0.3 degree. The GPS has a dedicated port on the board. A second GPS can be connected to the TELEM2 port in order to increase precision.

- PX4Flow (Figure 2.5i): This sensor is an optical flow smart camera coupled with a sonar that must be pointing down. Its purpose is to replace the GPS indoor to perform some stationary and autonomous flights. The camera resolution is 752×480 pixels and permit to estimate the x and y translation velocities. The sonar estimates the altitude of the drone. It has a range from 30cm to 5m and scans at 10Hz. The PX4Flow is connected via the I2C port.

- Telemetry module: A telemetry module is required to establish a wireless connection between the system and the ground control station. It allows to change some PX4 firmware parameters, send

[1] https://docs.px4.io/main/en/flight_controller/pixhawk4.html
[2] http://qgroundcontrol.com

control messages to the drone and receive sensor data on the ground control station. The 433Mhz Mini 3DR radio telemetry module (Figure 2.5j) is chosen for its small size and MAVLink [3] support. It is wired on the TELEM1 port.

- Remote controller: The FrSky Taranis X9-Lite has been taken for its low cost (Figure 2.5k). It uses the last ACCESS protocol and has 24 channels. The number of channels corresponds to the number of different controllable functions.

- Radio receiver: Together with remote controller, they are used for manually control the drone. The receiver must be adapted to the remote controller and it must have enough channels for the planned usage. For these reasons and for its reduced size, the FrSky R-XSR receiver is chosen (Figure 2.5l). It has 16 channels in SBUS protocol which will be used to connect with the Pixhawk 4 by the dedicated port.

- Anti-vibration support (Figure 2.5m): Indeed, its accelerometers are sensitive to the vibrations coming from the motors or some other external disturbances. Its size is 30.5×30.5mm and it will be mounted on the top plate of the frame.

The completed assembly of quadrotor UAV is given in Figure .2.6, the other component such as PX4Flow and PM02 are enclosed below the UAV.

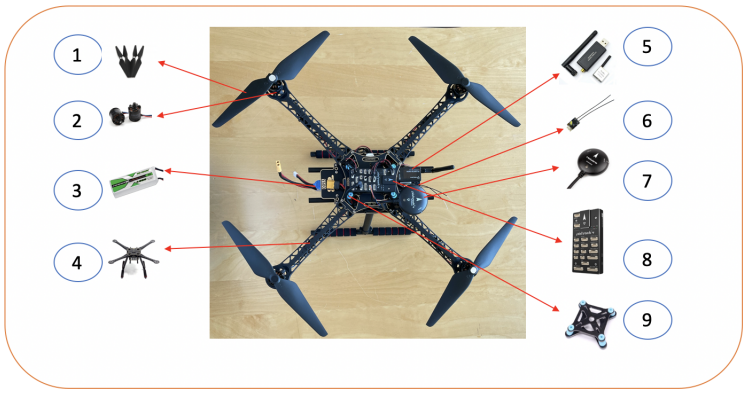


Figure 2.6: Completed quadrotor UAV platform: 1. 1045 propellers; 2. 2216 brushless motors; 3. Tattu LiPo 6S 4000mAh ; 4. The S500 frame; 5. Radio telemetry module; 6. Radio receiver; 7. GPS module; 8. Pixhawk card; 9. Anti-vibration support.

2.2.2 PX4 firmware

The PX4 firmware purpose is to **control many different vehicle types** (air, ground and underwater) while letting large possibilities in the choice of hardware components. It also provides some flexible and powerful flight modes and safety features. PX4 has been developed to run on the NuttX OS as well as on the main operating systems (macOS, Linux and Windows). As a result and thanks to simulation layer, it can be tested on a computer before being used on a flight controller board.
The firmware consists in two main stages :

- The flight stack for estimation and control. It includes three main modules. An estimator which computes the vehicle state based on combined sensor data. A controller that takes the estimated attitude to compute the attitude and thrust set-points. A mixer which translates the controller outputs into individual motor commands.

- The middleware for hardware integration and both internal and external communications. It mainly consists of device drivers for embedded sensors, an external communication module and an internal communication module. It also includes a simulation layer to perform simulations in a simulated environment.

[3] https://docs.px4.io/main/en/middleware/mavlink.html

The source code is split into modules communicating with each other through uORB [4] messaging using a publish-subscribe message bus. External communication is based on the MAVLink 2.0 protocol (for last firmware versions) which can be used on a Serial link or an UDP (User Datagram Protocol) network connection. Default UDP ports are established for MAVLink communication : 14540 for external APIs (Application Programming Interface) and 14550 for ground control stations.

PX4 supports both Software In The Loop (SITL) and Hardware In The Loop (HITL) simulations. In SITL, the flight stack runs on a computer whereas in HITL a simulation firmware is used on a flight controller board. Simulators, connected by default on TCP (Transmission Control Protocol) port 4560, allow the firmware to control a modeled vehicle in a simulated world through a ground control station application, a joystick, a remote controller or an external API . The supported simulators are Gazebo, jMAVsim, AirSim and XPlane which have their own advantages.

By its structure the firmware can be easily modified to answer a particular need. For instance, one could make some changes in a module or implement some new ones, same thing for messages. Flight control laws can be tuned or changed and so on.

2.2.3 QGroundControl

Figure 2.7: Quadground control software interface

QGroundControl is a ground control station open-source application that can be used with any MAVLink enabled drone. It is a cross plateform so it can be used on the major operating systems (Mac OS, Windows, Linux, iOS and Android) but it will be used on a laptop running Linux. Its usage is quite easy and straightforward, even for beginners, while giving some high end features for experienced users. The QGroundControl interface in present in Figure. 2.7.

The application allows to setup and configure the PX4 based autopilot vehicles. It provides some flight support with a flight map with the vehicle position, flight track, waypoints and vehicle instruments. It also supports video streaming and can deal with multiple vehicles. Moreover, it gives access to the MAVLink communication between the vehicle and the application to monitor the exchanged data. Those data can also be forwarded to be processed by another application.

In order to configure the system using QGroundControl, the board is plugged by USB to the ground station laptop. All the steps are enumerated thereafter.

1. Firmware update :
   Install the last stable firmware version in Pixhawk 4 (PX4 1.11.0 at the time of writing).

2. Airframe selection :
   The quadrotor X airframe is selected in the dedicated tab. This configures the firmware parameters

[4] https://docs.px4.io/main/en/middleware/uorb.html

to the chosen airframe.

3. Sensor orientation :
   As the flight controller board and the GPS/compass have been mounted in the default orientation (top side up with arrow pointing towards the front), there is no need to change their orientation parameters.

4. Sensor calibration :
   The compass calibration configures the magnetometers by placing the drone in a number of set orientations and rotating the vehicle about the specified axis. Then, the gyroscopes are calibrated by placing the vehicle on a flat surface and keeping it still. Next, the accelerometers calibration is achieved by placing and holding the drone in a number of set orientations. After that, the level horizon calibration is performed to compensate for small miss-alignment in controller orientation. To finish, the PX4Flow sensor is enabled by setting the parameter SENS_EN_PX4FLOW to 1 and rebooting the board. Some other parameters are filled to enable its use for Extended Kalman Filter data fusion.

5. Radio setup :
   First, the radio remote controller needs to be bound to the receiver. This step is hardware specific and easily done by following the instructions. Then the configuration is realized to assign the four first channels to the main control inputs (roll, pitch, yaw and throttle) and the others channels to the other remote inputs. Once done, flight modes and arming are assigned to switches.

6. Power setup :
   The number of battery cells, the full and empty voltages per cell are filled out (respectively 3 cells, 4.2V and 3.5V).

7. ESC calibration :
   The ESC PWM calibration is realized taking care that the propellers are removed.

8. Safety settings :
   Some failsafe actions are defined when a safety issue occurs. For instance, to land when battery is low or to return to the starting point when the remote control connection is lost.

When all the previous steps are successfully performed, the drone is ready to flight.

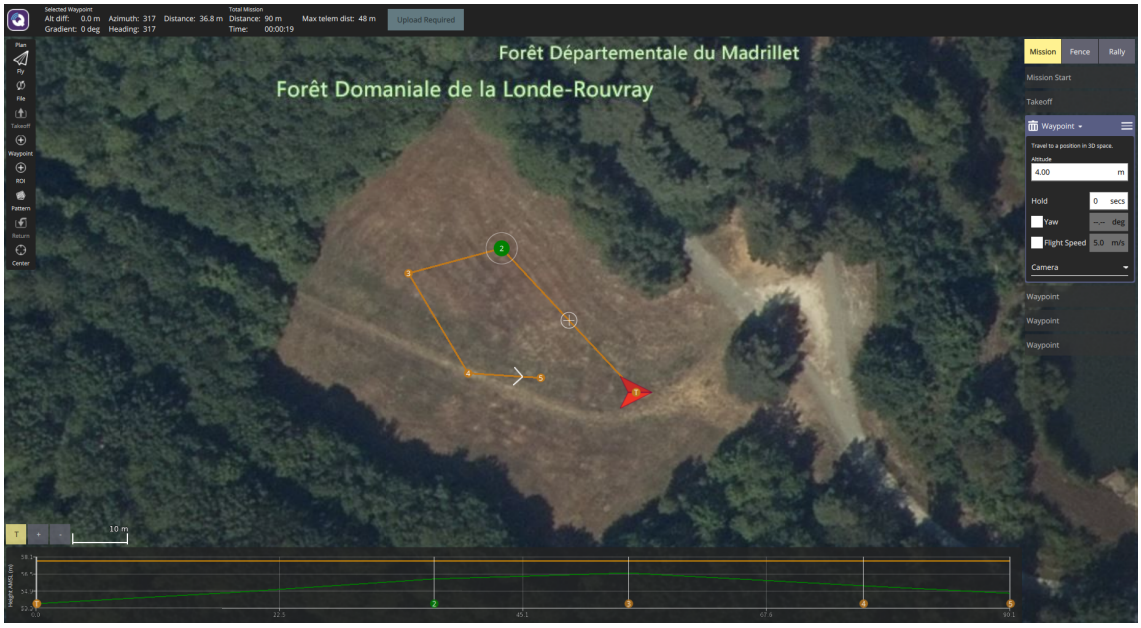


Figure 2.8: Fly plan tool in Qgroundcontrol

The Qgroundcontrol application also allows us to design the path for position tracking by using the **Fly plan tool**. Figure .2.8 shows the **Fly plan tool** in Qgroundcontrol. The path planing for position tracking is normally design by three steps, namely: take off , way point (design the trajectory in 2D XY plan) and landing . Each step allows us to design the altitude of UAV, holding time, yaw angle

and light max speed. The path planing finishes by uploading the designed fly plan to the UAV. The Qgroundcontrol will then converted the designed path to desired position in x, y, z and yaw angle in ψ and transfer to UAV flight board. The flight controller implemented in the flight board (Pixhawk) will then track these desired signals.

2.2.4 Gazebo simulator

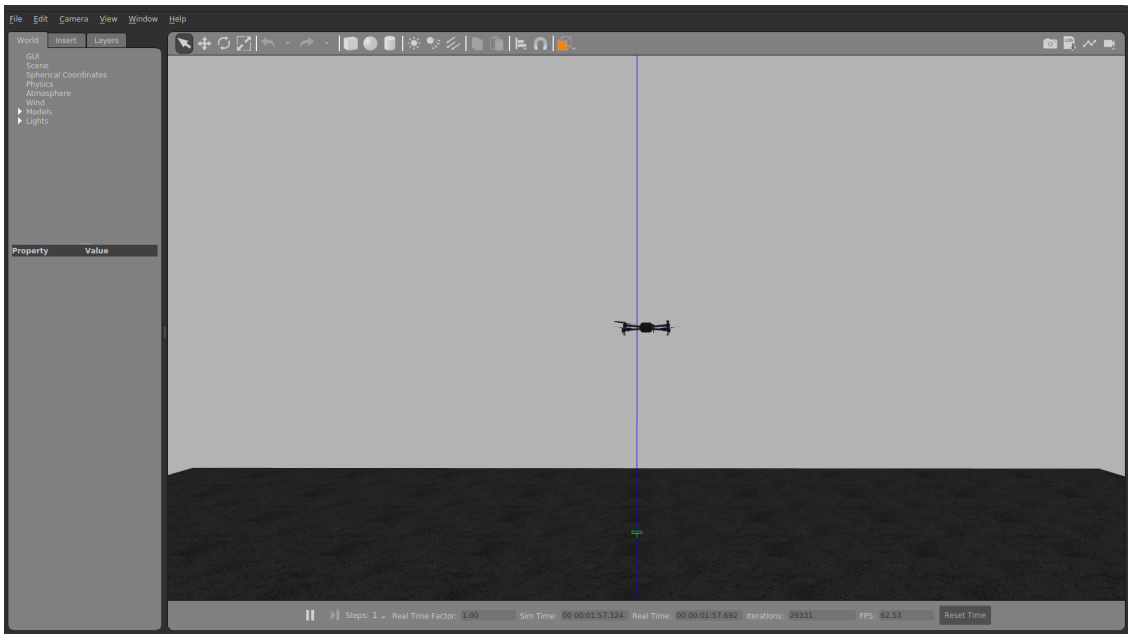


Figure 2.9: 3D model of quadrotor UAV in Gazebo simulator provided by Px4 developer

Gazebo is an open-source 3D robotics simulator, it will be used to develop a virtual environment for quadrotor simulation. The Px4 source code come together with an open source 3D model of quadrotor UAV for Gazebo simulation, the detail of Gazebo 3D model parameters can be found in this link [5]. Which will allows us to validate the proposed work in this thesis in Gazebo simulator easier. Figure .2.9 visualized the quadrotor UAV is flying in Gazebo environment for testing the controller performance.

2.2.5 Control design using Pixhawk controller

Pixhawk is an open-source autopilot [6] software for drones and other unmanned aerial vehicles (UAVs). The source code is a collection of software modules written in the C++ programming language, designed to run on micro-controller-based hardware such as the ARM Cortex-M series. The modular architecture of PX4 allows developers to customize and add new features to the autopilot software, making it highly flexible and adaptable for various UAV applications. This code-base is actively developed and maintained by a global community of contributors, making it a highly reliable and cutting-edge autopilot system.

In the provided source code, there are different modules for the different tasks. In this thesis, our studies are focus on attitude control module [7], position control module [8] and rate control module [9]. The source code allows us to implement our algorithms into the Pixhawk controller board by modifying the code inside these modules.

2.3 Conclusion

This chapter provides an in-depth understanding of the quadrotors UAV platform, as well as the software typically used to study it. Firstly, the mathematical background of a quadrotors UAV is thoroughly explored, beginning with an explanation of how the state vector, specifically the attitude of the quadrotors

[5] https://github.com/PX4/PX4-SITL_gazebo-classic/blob/main/models/iris/iris.sdf.jinja
[6] https://github.com/PX4/PX4-Autopilot
[7] https://github.com/PX4/PX4-Autopilot/tree/main/src/modules/mc_att_control
[8] https://github.com/PX4/PX4-Autopilot/tree/main/src/modules/mc_pos_control
[9] https://github.com/PX4/PX4-Autopilot/tree/main/src/modules/mc_rate_control

UAV, is presented. Additionally, the forces and moments applied to the quadrotor are analyzed in great detail, and based on this analysis, the dynamic model of the quadrotors UAV is established. This knowledge will be essential for the observer and controller developments discussed in the following chapter.

Furthermore, the mechanical platform of the quadrotors UAV, which will be utilized throughout the thesis, is introduced, including a detailed examination of the hardware and software used to validate the proposed controller and observer. This platform and software will serve as the foundation for the experimental validations discussed later in the thesis.

Overall, this chapter provides a comprehensive understanding of the quadrotors UAV system and the key mathematical concepts and principles that underpin its operation. The detailed analysis of the forces and moments acting on the quadrotor, as well as the dynamic model of the quadrotors UAV, will prove invaluable in the development of the observer and controller discussed in the following chapter. By establishing a strong foundation in the mechanical and mathematical aspects of the quadrotors UAV, this chapter sets the stage for the experimental validations to come, utilizing the platform and software introduced in this chapter.

CHAPTER 3

State estimation using continuous, discontinuous and delay measurement

3.1 Introduction

Knowing the system states is necessary to solve many control theory problems, including the control design of quadrotor UAVs. To fulfill this important task, several estimation methods have been developed in the literature, starting with the linear case (Osorio-Gordillo, Darouach, and Astorga-Zaragoza, 2015; Osorio-Gordillo et al., 2016), LPV (Pérez-Estrada et al., 2019; Pérez-Estrada et al., 2018), and nonlinear cases such as extended Luenberger observers (Huong, Huynh, and Trinh, 2019), Lyapunov-based observers (Raghavan and Hedrick, 1994; Phattanasak et al., 2016; Vaclavek and Blaha, 2009; Shin et al., 2016), sliding mode observers (Koshkouei and Zinober, 2002; Dam et al., 2023a; Abhinav and Mija, 2021; Kumar Singh, Arkdev, and Sarkar, 2018; Guezmil et al., 2015), and High-Gain Observers (HGO) (Gauthier and Kupka, 1994; Khalil and Praly, 2013; Ali and Langlois, 2017; Abraham Efraim Rodríguez-Mata and Leal, 2018; A. E. Rodríguez-Mata and Amabilis-Sosa, 2018; ZHANG and XU, 2001; Dam et al., 2021; Dam et al., 2022; Dam et al., 2023b).

In the following, we will provide a brief review of state estimation techniques, ranging from classical methods such as the Luenberger observer to intelligent observers.

The Luenberger observer (Luenberger, 1964) represents a simple and easy method to estimate the state vector of a system. However, there are several drawbacks to this technique, such as its lack of robustness and less effective performance. Therefore, various methods have been applied to improve its performance, including high gain, sliding mode, and more intelligent techniques.

High gain observers have become a widely used technique for state estimation in the nonlinear case. The popularity of this method can be attributed to its simplicity of implementation and the ability to adjust both accuracy and convergence rate through a single adjustable parameter known as the high-gain parameter (Gauthier, Hammouri, and Othman, 1992a). These features make high gain observers an attractive approach for state estimation, widely employed in the field of automation.

The idea was first described in (Doyle and Stein, 1978), and soon afterwards, the technique was further developed by Gauthier, Hammouri, and others (Gauthier, Hammouri, and Othman, 1992a; Gauthier, Hammouri, and Othman, 1992b; F. Deza and Rako-topara, 1992; Viel and Gauthier, 1995). Gauthier and others focused on deriving global results under global Lipschitz conditions. Additionally, Hassan K. Khalil made significant contributions to the development of the high gain observer in his independent work (Khalil and Praly, 2013). The high gain observer has been applied to various applications such as robot manipulators (Liaquat, Javaid, and Saad, 2017), inverted pendulums (Liu, Lv, and Wu, 2019), autonomous underwater vehicles (Zhang et al., 2016), induction motors (Msaddek, Gaaloul, and Msahli, 2015), and electro-hydraulic actuators (Ali et al., 2016a), among others. Notably, studies on quadrotor UAVs can be found in (Dam et al., 2021; Miladi et al., 2018; Dam et al., 2022; Rodriguez-Mata, Farza, and M'Saad, 2019).

However, the high gain technique is applicable when the system is in diagonal form, where the dynamic equation of the system satisfies the following form:

$$\dot{x} = A(x,u)x + \varphi(x,u) \tag{3.1}$$

where $(A)_{ij} = [\neq 0 \; if \; j = i+1; \; 0 \; if \; j \neq i+1]$.

The main advantage of this method is that the speed of the estimator can be increased by increasing the observer gain through the high-gain parameter. However, High-Gain Observers suffer from three major drawbacks in the context of state estimation, namely:

- Numerical problems arising from the high dimension of the system and the high value of the high-gain parameter.
- The well-known peaking phenomenon.
- Sensitivity to output disturbances, such as delay, sampled output, and measurement noise.

Motivated by these problems, a method was proposed in (Zemouche et al., 2018) to tune the high-gain parameter effectively, while using a much smaller value compared to the primitive method. However, by reducing the value of the high-gain parameter, the most important property of the high-gain technique, which is the ability to increase convergence speed, is implicitly compromised. Therefore, several techniques have been proposed to overcome this problem.

Another aspect of interest in high gain observers is their effectiveness in dealing with disturbances and measurement noises. An alternative solution is to filter the noise, resulting in a filtered high gain observer (FHGO). This observer is based on the idea of designing a dynamic filter that acts on the output injection term of the observer rather than the measured output. The appearance of noises is then processed by the filter along with the state estimate. This idea is discussed in (Ball and Khalil, 2008) and (BUSVELLE and GAUTHIER, 2005). The switch gain is introduced and studied in (Ball and Khalil, 2008), while ((Astolfi, Zaccarian, and Jungers, 2021), (Tréangle, Farza, and M'Saad, 2019b), (Robles-Magdaleno et al., 2020), (Astolfi et al., 2016), (Astolfi, Jungers, and Zaccarian, 2018), (Khalil and Priess, 2016)) propose FHGO for a class of uniformly observable systems involving uncertainties. In (Nicolas Boizot and Gauthier, 2015), a study of high gain observer with adaptive gain extended Kalman filter for a class of nonlinear systems is introduced. The applications of this method are studied in (Boizot N, 2007), (Sebesta and Boizot, 2013), and (Aïda Feddaoui, 2018).

In (Abraham Efraim Rodríguez-Mata and Leal, 2018), (A. E. Rodríguez-Mata and Amabilis-Sosa, 2018), (López-Gutiérrez, 2017), (ZHANG and XU, 2001), an adaptive gain is applied to estimate both the state vector and the magnitude of the disturbance. In (Ali and Langlois, 2017), a time-varying correction gain for the high-gain observer is proposed to reduce the effects of measurement noises.

The adaptive gain in this case has overcome the above-mentioned problem when working in a noisy environment. However, the main drawback of this technique is the complexity of the observer structure. On the other hand, this method increases the stability of the observer under the effects of noise rather than accurately estimating the real output signals.

It should be noted that the state estimation problem becomes more challenging when dealing with discrete measurements. In this case, the output measurements are only available at fixed sampling times. The observer design typically assumes that the outputs are continuously available. However, this is rarely the case in practice, as these outputs are usually available at sampling instants $0 \leq t_0 < t_1 < \cdots < t_k < t_{k+1} < \cdots$ with a sampling interval of $h = t_{k+1} - t_k$. This sampling rate directly affects the stability of the system, and therefore, the design of an observer will depend on the configuration of this sampling rate.

Moreover, the issue of delay is strongly involved in signal transmission. In high-speed communication networks, time delay represents a source of instability. Therefore, studying sampled-delay estimation is crucial. In the following, we will provide a brief review of the proposed methods for addressing this problem.

The problem of improving state estimation using discrete measurements has been addressed in recent works, where a Sampled-data High Gain Observer (SHGO) is proposed (Ali et al., 2016b; Karafyllis and Kravaris, 2009; Hernández-González et al., 2019; Tréangle, Farza, and M'Saad, 2019a). In the latter work, a continuous-time HGO is coupled with an output predictor for discrete measurements, allowing the prediction of unmeasurable outputs between two sampling times. The same idea has been demonstrated and applied to practical applications in (Ali et al., 2016b; Ali et al., 2014; Ali and Langlois, 2017). The condition for the maximum sampling period that ensures system stability is also detailed in these references.

The problem of unmeasurable state estimation for nonlinear systems under time delays is even more challenging compared to systems without delays. Furthermore, the problem of sampling data, caused by the configuration of sensors in specific cases, can be transformed into systems with delayed measurements.

This problem has been addressed in many research papers and has been identified as one of the major causes of performance degradation and loss of stability (Cacace, Germani, and Manes, 2010a; Fridman, 2014a; Talebi, Ataei, and Pepe, 2020).

Motivated by this problem, several methods have been developed, namely the Luenberger observer (Huong, Huynh, and Trinh, 2019), Extended Kalman Filter (EKF)-based observer (Raff and Allgower, 2006), LMI-based observer (Adil et al., 2020a; Briat, Sename, and Lafay, 2011), and High-Gain Observer (HGO) (Dam et al., 2022; Kahelras et al., 2016; Ahmed-Ali et al., 2019; Assche et al., 2011a; Adil et al., 2020a). In some works, similar to the sampling period, the time delay is also limited to ensure the stability of the observer, as described in (Kahelras et al., 2016; Ahmed-Ali et al., 2019; Assche et al., 2011a; Adil et al., 2020a). However, these methods propose limited conditions for the time delay, while in many cases, the delay value is larger. Recognizing this intriguing problem, the cascade observer method has been proposed, where the observer is decoupled into smaller steps to handle the large delay. The cascade structure can be found in (Farza et al., 2018; Sajadi, Ataei, and Ekramian, 2018; Ahmed-Ali et al., 2019; Tréangle, Farza, and M'Saad, 2019b).

In engineering applications, the problems of discrete measurements and delay measurements are usually coupled, and the assumption that the sampling data problem can be transformed into the form of delayed measurements is very restrictive and not always satisfied. Attracted by this interesting problem, some solutions have been proposed in (Tréangle, Farza, and M'Saad, 2019b; Kahelras et al., 2016; Ahmed-Ali et al., 2019; Shen, Zhang, and Xia, 2017; Assche et al., 2011a; Cuny et al., 2020).

In this chapter, motivated by these aforementioned problems, the study of observer design for nonlinear systems subject to delay and noisy measurements is carried out.

The problem is initially considered for a class of nonlinear systems with delayed output measurements. Then, a novel structure of FHGO is proposed to overcome this issue. The novelty of the designed observer lies in two main components. The first component is a linear first-order filter that acts on the output injection term of the observer. The second component is a new closed-loop output predictor term that allows for the design of an appropriate observer gain based on the delay condition. Compared to standard high-gain methodologies, this technique has surpassed the barrier of delay limitation without using a cascade structure. The convergence of the proposed observer is ensured by means of a Lyapunov-Krasovskii functional.

The extension of this result to the case of a nonlinear system with sampled-delayed output is then introduced. The convergence of the extended FHGO is proven, and the observer design technique based on the value of the delay and sampling period is provided using LMI. The application to a quadrotor UAV is considered.

Compared to the existing method , the main contributions of the observer proposed in this chapter can be summarized as follows:

- We consider firstly the case of delayed and noisy output-measurements and we propose a new structure of a FHGO. The observer design based on LMI technique where the condition of delay and sampling period are always guaranteed.
- We extend the obtained results to deal simultaneously with the triple issues namely the sampled, delayed and noisy output measurements and then, a new structure of the FHGO is proposed.
- The structure proposed in this chapter requires less computation in comparison with other method in term of dealing with high value of delay such as cascade structure.

The chapter is organized as follows: In section 3.2, the problem statement is formulated for the observer design. The new structure of the proposed observer is given in section 3.3 and 3.4 for continuous delay and sampled-delay respectively. The simulation and experimental results are presented in section 3.5.2, 3.5.3. Then, some concluding remarks are given in section 3.8 to highlight the observer effectiveness. Some remain problems will be discussed in section 3.6. Finally, section 3.7 present a method to estimate the delay when it is assumed unknown.

3.2 Problem statement

Let us recall the two main challenges to be handled in this chapter

- How to design a new FHGO, for nonlinear systems with delayed and noisy measurements?
- How to deal with sampled-delayed output measurements on the FHGO design?

The basic goal of the work in this chapter is to provide a satisfactory solution to the above two questions. Then, let us firstly focus on the following class of nonlinear systems with delayed output measurements:

$$\begin{cases} \dot{x}(t) &= Ax(t) + \varphi(u(t), x(t)) + B\xi(t) \\ z(t) &= Cx(t) \\ y(t) &= z(t-\tau) + \upsilon(t) \end{cases} \tag{3.2}$$

where $x(t) = \begin{bmatrix} x_1^T(t) & x_2^T(t) & \cdots & x_m^T(t) \end{bmatrix}^T \in \mathbb{R}^{p\times 1}$, with $x_i \in \mathbb{R}^{n\times 1}$ and $p = n.m$, denotes the state vector. $\varphi(u(t), x(t)) = \begin{bmatrix} \varphi_1^T(u(t), x_1(t)) & \cdots & \varphi_m^T(u(t), x(t)) \end{bmatrix}^T \in \mathbb{R}^{p\times 1}$ is the nonlinear function with $\varphi_i(u(t), x(t)) = \varphi_i(u(t), x_1(t), x_2(t), \cdots, x_i(t)) \in \mathbb{R}^{n\times 1}$ is triangular with respect to $x(t)$. $u(t) \in U \in \mathbb{R}^{u\times 1}$ denotes the control input. $z(t)$ and $y(t)$ are respectively the undelayed and delayed outputs. τ represents the time delay. $\xi(t)$ and $\upsilon(t)$ represent respectively the unknown but bounded disturbances and the zero-mean Gaussian white noises. The matrices A, B and C are defined as following:

$$A = \begin{bmatrix} 0_{(m-1)n\times n} & I_{(m-1)n} \\ 0_{n\times n} & 0_{n\times(m-1)n} \end{bmatrix} \in \mathbb{R}^p; B = \begin{bmatrix} 0_n & \cdots & 0_n & I_n \end{bmatrix}^T \in \mathbb{R}^{p\times n}; \tag{3.3}$$
$$C = \begin{bmatrix} I_n & 0_n & \cdots & 0_n \end{bmatrix} \in \mathbb{R}^{n\times p}$$

$\upsilon(t)$ denotes the measurement noise which can be modeled as a finite sum of sinusoids (Astolfi and Marconi, 2015):

$$\upsilon(t) = a_N sin(\omega_N t + f_N) \tag{3.4}$$

where $a_N > 0$, $\omega_N > 0$ and f_N represent respectively the amplitude, the frequency and the phase of the noise.

It should be noted that the quadrotor UAV will be considered as an application

As previously mentioned, the first objective of this work is to design a novel filtered high gain observer for the class of systems (3.2), with noisy and delayed output measurements. In order to minimize the effect of the noise and to compensate the sampling and time delay phenomenon. The proposed observer will be designed under the following assumptions, corollary and lemmas :

Assumption 3.2.1. *The disturbances $\xi(t)$ are assumed unknown but bounded with known bounds:*

$$\exists\ \epsilon_1 \in \mathbb{R}^+,\ \sup_{t\geq 0} \|\xi(t)\| \leq \epsilon_1 \tag{3.5}$$

Assumption 3.2.2. *The measurement noise $\upsilon(t)$ is assumed bounded:*

$$\exists\ \epsilon_2 \in \mathbb{R}^+,\ \sup_{t\geq 0} \|\upsilon(t)\| \leq \epsilon_2 \tag{3.6}$$

Assumption 3.2.3. *Based on (3.4) and (3.6), the first derivative of the measurement noise $\dot{\upsilon}(t)$ can be assumed bounded with known bounds:*

$$\exists\ \epsilon_3 \in \mathbb{R}^+,\ \sup_{t\geq 0} \|\dot{\upsilon}(t)\| \leq \epsilon_3 \tag{3.7}$$

Assumption 3.2.4. *It is assumed that the function $\varphi_i(u(t), x(t))$ is Lipschitz with respect to $x(t)$ and $u(t)$, $\forall i = 1, \ldots, m$. Then, $\exists\ L_\varphi \in \mathbb{R}^+$ such that $\forall u \in U$, $\forall\ x(t)$ and $\bar{x}(t) \in X$:*

$$\|\tilde{\varphi}_i(t)\| \leq L_\varphi \|\tilde{x}(t)\| \tag{3.8}$$

with $\tilde{\varphi}_i(t) = \varphi_i(u(t), x(t)) - \varphi_i(u(t), \bar{x}(t))$ and $\tilde{x}(t) = x(t) - \bar{x}(t)$

Corollary 3.2.5. *Assume that $\exists \tau \in \mathbb{R}^+$ such that the function $\varphi_i(u(t-\tau), x(t-\tau))$ is Lipschitz with respect to $x(t-\tau)$ and $u(t-\tau)$. Then,*

$$\|\tilde{\varphi}_i(t) - \tilde{\varphi}_i(t-\tau)\| \leq 2L_\varphi \|\tilde{x}(t)\| + L_\varphi \int_{t-\tau}^{t} \left\|\dot{\tilde{x}}(h)\right\| dh \tag{3.9}$$

Proof: According to Assumption 2, one gets:

$$\|\tilde{\varphi}_i(t) - \tilde{\varphi}_i(t-\tau)\| \leq L\varphi \left(\|\tilde{x}(t)\| + \|\tilde{x}(t-\tau)\|\right) \tag{3.10}$$

Then, from the fact that:

$$\|\tilde{x}(t)\| + \|\tilde{x}(t-\tau)\| \leq 2\|\tilde{x}(t)\| + \|\tilde{x}(t) - \tilde{x}(t-\tau)\| \tag{3.11}$$

and using the Leibniz integration formula, we obtain:

$$\|\tilde{x}(t)\| + \|\tilde{x}(t-\tau)\| \leq 2\|\tilde{x}(t)\| + \left\|\int_{t-\tau}^{t} \dot{\tilde{x}}(h)dh\right\| \leq 2\|\tilde{x}(t)\| + \int_{t-\tau}^{t} \left\|\dot{\tilde{x}}(h)\right\| dh \tag{3.12}$$

Based on (3.10) and (3.12), (3.9) can be derived.

Lemma 3.2.6. *(Tréangle, Farza, and M'Saad, 2019a) Let* $M = \begin{bmatrix} A & -K_1 \\ C^T C & A^T - K_2 \end{bmatrix}$. *$A$ and C are two matrices defined in (3.3). K_1 and K_2 are the block diagonal matrices given by:*

$$\begin{aligned} K_1 &= diag[k1_1 I_n, k1_2 I_n, \cdots k1_m I_n] \\ K_2 &= diag[k2_1 I_n, k2_2 I_n, \cdots k2_m I_n] \end{aligned} \tag{3.13}$$

where $k1_i,\ k2_i,\ i = 1,\ \cdots,\ m$ are non zero real constant. There exists a choice of $k1_i,\ k2_i,\ i = 1,\ \cdots,\ m$ such that the eigenvalues of matrix M can be arbitrarily assigned.

Proof : See the proof of Lemma 3.13 in (Tréangle, Farza, and M'Saad, 2019a), where A_n, K_n, C_n and D_n are replaced respectively by A, K_1, C and K_2.

Corollary 3.2.7. *Since the eigenvalues of the matrix M can be arbitrarily assigned to ensure its Hurwitz character, then, there exist a symmetric, definite and positive matrices P_x, P_η, Q_x, Q_η such that:*

$$PM + M^T P \leq -Q \tag{3.14}$$

where $P = diag\{P_x, P_\eta\}$ and $Q = diag\{Q_x, Q_\eta\}$. □

Lemma 3.2.8. *Let $f : [t-\tau,\ t] \to \mathbb{R}$ be a continuous integrable function then one gets:*

$$\int_{-\tau}^{0}\int_{t+r}^{t} \|f(s)\|^2 dsdr \leq \tau \int_{t-\tau}^{t} \|f(s)\|^2 ds \tag{3.15}$$

Proof : From (Fridman, 2014b), the equation of $\int_{-\tau}^{0}\int_{t+r}^{t} \|f(s)\|^2 dsdr$ can be rewritten as follows:

$$\int_{-\tau}^{0}\int_{t+r}^{t} \|f(s)\|^2 dsdr = \int_{t-\tau}^{t} (\tau + s - t)\|f(s)\|^2 ds \tag{3.16}$$

The function $(\tau + s - t)$ is a continuous function, $\|f(s)\|^2$ is a non-negative integrable function. Then using the means value theorem one gets there exists $c \in [t-\tau,\ t]$ such that:

$$\int_{-\tau}^{0}\int_{t+r}^{t} \|f(s)\|^2 dsdr = (\tau + c - t)\int_{t-\tau}^{t} \|f(s)\|^2 ds \tag{3.17}$$

Using the fact that in the interval $s \in [t-\tau \quad t]$, the function $(\tau + s - t)$ is a monotonically increasing then $\underset{s\in[t-\tau,\ t]}{max}(\tau + s - t) = \tau$, thus we have:

$$\int_{-\tau}^{0}\int_{t+r}^{t} \|f(s)\|^2 dsdr \leq \underset{s\in[t-\tau,\ t]}{max}(\tau + s - t)\int_{t-\tau}^{t} \|f(s)\|^2 ds = \tau\int_{t-\tau}^{t} \|f(s)\|^2 ds \tag{3.18}$$

3.3 Continuous-delay observer

3.3.1 Observer design

The proposed observer structure (depicted in Figure 3.1) consists of three cascaded subsystems. The first subsystem is devoted to predicting the state vector. The second subsystem is a filter with the same dimension as the original system, obtained by filtering the output estimation given in the first subsystem. This design is similar to the approach in (Astolfi, Zaccarian, and Jungers, 2021), but it exhibits some

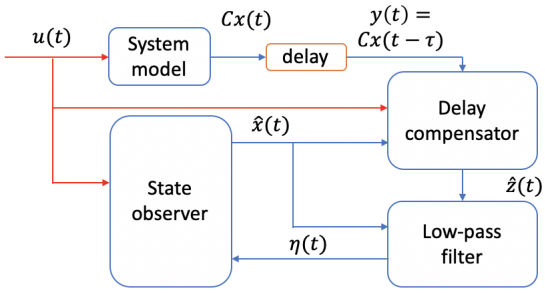


Figure 3.1: Synoptic diagram of the proposed continuous-delay observer

fundamental differences. The last subsystem is dedicated to measurement delay compensation, inspired by equation (8d) in (Cuny et al., 2020). However, in this work, a novel correction term is added, which enables a closed-loop observer and guarantees the convergence not only of the state estimation but also of the output estimation. This aspect distinguishes it from (Cuny et al., 2020) and previous works in the literature. Importantly, this correction term allows for an increased Maximum Allowable Value of Time Delay (MAVTD) compared to previous works while maintaining the simplicity of the observer structure.

The proposed observer structure is as follows:

$$\begin{cases} \dot{\hat{x}}(t) &= A\hat{x}(t) + \varphi(u(t), \hat{x}(t)) + K_1 \Lambda \Delta \eta(t) \\ \dot{\eta}(t) &= \Lambda(A^T - K_2)\eta(t) + \Lambda C^T \left[\hat{z}(t) - C\hat{x}(t)\right] \\ \hat{z}(t) &= y(t) + \int_{t-\tau}^{t} C\left[A\hat{x}(s) + \varphi(u(s), \hat{x}(s))\right] ds - \int_0^t \Lambda K_3 \left[\hat{z}(s) - C\hat{x}(s)\right] ds \end{cases} \tag{3.19}$$

where K_1, K_2 and $K_3 = k_3 I_n$ are the proposed observer gains. Λ is the high-gain parameter and Δ is the block diagonal matrix given by:

$$\Delta = diag[I_n,\ \Lambda I_n,\ \cdots,\ \Lambda^{m-1} I_n],\ \Lambda > 0 \tag{3.20}$$

where *diag* returns a diagonal matrix from its input vector.

It is worth noticing that the gain K_3, in which depends the correction term $\int_0^t \Lambda K_3 \left[\hat{z}(s) - C\hat{x}(t)\right] ds$, will be designed based on a new LMI. This LMI will be implicitly used to design the gains K_1 and K_2, under some conditions, in order to ensure the stability of the observer (3.19), as given by the following Theorem 3.3.1, Lemma 3.2.6 and Corollary 3.2.7.
Let now define the following estimation errors quantities:

$$\tilde{x}(t) = x(t) - \hat{x}(t) \tag{3.21}$$
$$\tilde{z}(t) = Cx(t) - \hat{z}(t) \tag{3.22}$$
$$\tilde{\varphi}_i(t) = \varphi_i(u(t), x(t)) - \varphi_i(u(t), \hat{x}(t)) \tag{3.23}$$
$$\tilde{\varphi}_i(t-\tau) = \varphi_i(u(t-\tau), x(t-\tau)) - \varphi_i(u(t-\tau), \hat{x}(t-\tau)) \tag{3.24}$$
$$\varepsilon_x(t) = \Delta^{-1}\tilde{x}(t) \tag{3.25}$$
$$\varepsilon_z(t) = \Lambda^{-l}\tilde{z}(t);\ l > 0 \tag{3.26}$$

Theorem 3.3.1. *Consider the system (3.2), under the assumptions 3.2.1, 3.2.3 and 3.2.4. For given positive scalars μ and μ_1, assume that there exist positive, definite and symmetric matrices Q_x, Q_η, P_x, P_η and a gain $K_3 \in \mathbb{R}^n$ such that the LMI (3.27) holds:*

$$\Phi_1 = \begin{bmatrix} -Q_x + \frac{L_\varphi P_x}{\Lambda} + \mu_1 I_p & 0_p & \frac{C^T K_3}{\Lambda^l} + 2\frac{L_\varphi}{\Lambda^l} C^T \\ * & -Q_\eta + \mu_1 I_p & -\Lambda^l P_\eta C^T \\ * & * & -2K_3 + \left(1 + \frac{L_\varphi}{\Lambda}\right)^2 \frac{I_n}{\Lambda^{l-1}} \end{bmatrix} \leq -\mu I_{2p+n} \tag{3.27}$$

then the observer (3.19) converges exponentially according to the following estimation errors bound:

$$\lim_{t\to\infty} \|\tilde{x}_i(t)\| \leq \frac{2\bar{\lambda}_{V_1}}{\Lambda^{1-i}\mu\underline{\lambda}_{V_1}} \left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\epsilon_3}{\Lambda^l}\right) \tag{3.28}$$

where

$$\underline{\lambda}_{V_1} = \min(\lambda_{\min}(P_x), \lambda_{\min}(P_\eta), 1); \tag{3.29}$$
$$\bar{\lambda}_{V_1} = \max(\lambda_{\max}(P_x), \lambda_{\max}(P_\eta), \Lambda\mu_1) \tag{3.30}$$

Remark 3.3.1. *Note that the solutions of the LMI (3.27) are the matrices $P = diag\{P_x, P_\eta\}$, $Q = diag\{Q_x, Q_\eta\}$ and the observer gain K_3. According to corollary 3.2.7, the eigenvalues of the matrix M will be designed based on P and Q, which depend implicitly on K_3. Then, according to Lemma 3.2.6, these eigenvalues can be achieved by computing the appropriate matrices K_1 and K_2. These matrices can be determined by applying the algorithm given by remark 2.2 in (Tréangle, Farza, and M'Saad, 2019a).*

Proof of Theorem 3.3.1 :
From (3.19), the dynamic of $\tilde{x}(t)$ and $\tilde{z}(t)$ are given as follows:

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + \tilde{\varphi}_i(t) + B\xi(t) - K1\Lambda\Delta\eta(t) \tag{3.31}$$
$$\begin{aligned}\dot{\tilde{z}}(t) &= CA\left[\tilde{x}(t) - \tilde{x}(t-\tau)\right] + C\left[\tilde{\varphi}(t) - \tilde{\varphi}(t-\tau)\right] + K_3C\tilde{x}(t) - K_3\tilde{z}(t) - \dot{\upsilon}(t) \\ &= CA\int_{t-\tau}^{t}\dot{\tilde{x}}(s)ds + C\left[\tilde{\varphi}(t) - \tilde{\varphi}(t-\tau)\right] - K_3\tilde{z}(t) + K_3C\tilde{x}(t) - \dot{\upsilon}(t)\end{aligned} \tag{3.32}$$

Using the facts that $\Delta^{-1}A\Delta = \Lambda A$, $C\Delta = C$, $CAB = 0$ and $CA\Delta = \Lambda CA$ one gets:

$$\dot{\varepsilon}_x(t) = \Lambda A\varepsilon_x(t) + \Delta^{-1}\tilde{\varphi}(t) + \Lambda^{1-m}B\xi(t) - K1\Lambda\eta(t) \tag{3.33}$$
$$\dot{\eta}(t) = \Lambda(-K_2 + A^T)\eta(t) - \Lambda^{l+1}C^T\varepsilon_z(t) + \Lambda C^TC\varepsilon_x(t) \tag{3.34}$$
$$\begin{aligned}\dot{\varepsilon}_z(t) &= \Lambda^{1-l}CA\int_{t-\tau}^{t}\dot{\varepsilon}_x(s)ds + \Lambda^{-l}C\left[\tilde{\varphi}(t) - \tilde{\varphi}(t-\tau)\right] - \Lambda K_3\varepsilon_z(t) + \Lambda^{1-l}K_3C\varepsilon_x(t) \\ &\quad - \Lambda^{-l}\dot{\upsilon}(t)\end{aligned} \tag{3.35}$$

One gets:

$$\|CA\dot{\varepsilon}_x(t)\| \le \Lambda\Omega_1\left\|\tilde{X}_1\right\|; \ \ \|C\dot{\varepsilon}_x(t)\| \le \Lambda\Omega_1\left\|\tilde{X}_1\right\| \tag{3.36}$$
$$\tilde{X}_1 = \begin{bmatrix}\varepsilon_x^T(t) & \eta^T(t)\end{bmatrix}^T; \ \Omega_1 = max\left(1 + \Lambda^{-1}L_\varphi; \lambda_{\max}(K_1)\right) \tag{3.37}$$

Consider now the following Lyapunov–Krasovskii function with time delay $\tau > 0$:

$$V_1 = \varepsilon_x^T(t)P_x\varepsilon_x(t) + \eta^T(t)P_\eta\eta(t) + \varepsilon_z^T(t)\varepsilon_z(t) + \tau\alpha_1\int_{-\tau}^{0}\int_{t+r}^{t}\left\|\tilde{X}_1(s)\right\|^2 dsdr \tag{3.38}$$

where α_1 is a positive scalar given as:

$$\alpha_1 = \Lambda^{2-l}\Omega_1^2 + \frac{\Lambda\mu}{\tau} \tag{3.39}$$

and the scalar μ_1 is chosen such that:

$$\tau^2\alpha_1 - \Lambda\mu_1 \le 0 \tag{3.40}$$

Let's define $W_1 = \begin{bmatrix}\varepsilon_x^T & \eta^T & \varepsilon_z^T\end{bmatrix}^T$, from(3.15) and (3.40) we have:

$$\underline{\lambda}_{V_1}W_1^TW_1 \le V1 \le \bar{\lambda}_{V_1}W_2^TW_2 \tag{3.41}$$

From (3.60), (3.61), (3.35), (3.62) and by taking into account assumption 3.2.4 one gets:

$$\begin{aligned}\dot{V}_1 &\le \Lambda\tilde{X}_1^T\left(PM + M^TP\right)\tilde{X}_1 + 2L_\varphi\varepsilon_x^T(t)P_x\varepsilon_x(t) - 2\eta^T(t)P_\eta\Lambda^{l+1}C^T\varepsilon_z(t) \\ &\quad + 4L_\varphi\Lambda^{-l}\left\|\varepsilon_z(t)\right\|\left\|C\varepsilon_x(t)\right\| + 2\Lambda^{2-l}\left(1 + \frac{L_\varphi}{\Lambda}\right)\Omega_1\left\|\varepsilon_z(t)\right\|\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|ds \\ &\quad - 2\Lambda\varepsilon_z^T(t)K_3\varepsilon_z(t) + 2\varepsilon_z(t)\Lambda^{1-l}K_3C\varepsilon_x(t) + \tau^2\alpha_1\left\|\tilde{X}_1\right\|^2 - \tau\alpha_1\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2 ds \\ &\quad + 2\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}\left\|\varepsilon_x(t)\right\| + 2\frac{\epsilon_3}{\Lambda^l}\left\|\varepsilon_z(t)\right\| + \Lambda\mu_1\left\|W_1\right\|^2 - \Lambda\mu_1\left\|W_1\right\|^2\end{aligned} \tag{3.42}$$

Using Young's inequality we obtain:

$$2\left(1+\frac{L_\varphi}{\Lambda}\right)\Omega_1\left\|\varepsilon_z(t)\right\|\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|ds \leq \left(1+\frac{L_\varphi}{\Lambda}\right)^2\left\|\varepsilon_z(t)\right\|^2 + \Omega_1^2\int_{t-\tau}^{t}\tilde{X}_1^T(s)ds\int_{t-\tau}^{t}\tilde{X}_1(s)ds \tag{3.43}$$

Using the Jensen's inequality one gets:

$$\Lambda^{2-l}\Omega_1^2\int_{t-\tau}^{t}\tilde{X}_1^T(s)ds\int_{t-\tau}^{t}\tilde{X}_1(s)ds \leq \tau\Lambda^{2-l}\Omega_1^2\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2ds \tag{3.44}$$

Thus, from (3.39), (3.43) and (3.44) we have:

$$\begin{aligned}&-\tau\alpha_1\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2ds+\Lambda^{2-l}\Omega_1^2\int_{t-\tau}^{t}\tilde{X}_1^T(s)ds\int_{t-\tau}^{t}\tilde{X}_1(s)ds \\ &\leq -\Lambda\mu\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2ds\end{aligned} \tag{3.45}$$

Hence, from (3.14), (3.39) (3.43), (3.44) and by taking into account corollary 3.2.7 one gets:

$$\begin{aligned}\dot{V}_1 &\leq \Lambda W_1^T\Phi_1W_1+\tau^2\alpha_1\left\|W_1\right\|^2-\Lambda\mu_1\left\|W_1\right\|^2+2\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}+\frac{\epsilon_3}{\Lambda^l}\right)\left\|W_1\right\| \\ &\quad-\Lambda\mu\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2ds\end{aligned} \tag{3.46}$$

Then, from (3.40) we obtain:

$$\begin{aligned}\dot{V}_1 &\leq -\Lambda\mu\left\|W_2\right\|^2+2\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}+\frac{\epsilon_3}{\Lambda^l}\right)\left\|W_1\right\| \\ &\leq -\frac{\Lambda\mu}{\bar{\lambda}_{V_1}}V_1+\frac{2}{\sqrt{\underline{\lambda}_{V_1}}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}+\frac{\epsilon_3}{\Lambda^l}\right)\sqrt{V_1}\end{aligned} \tag{3.47}$$

Or equivalently:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sqrt{V_1} \leq -\frac{0.5\Lambda\mu}{\bar{\lambda}_{V_1}}\sqrt{V_1}+\frac{1}{\sqrt{\underline{\lambda}_{V_1}}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}+\frac{\epsilon_3}{\Lambda^l}\right) \tag{3.48}$$

Using the comparison lemma we obtain:

$$\sqrt{V_1} \leq \sup_{c\in[t_0-\tau,\,t_0]}\left\|\sqrt{V_1(c)}\right\|e^{-\frac{0.5\Lambda\mu}{\bar{\lambda}_{V_1}}(t-t_0)}+\frac{2\bar{\lambda}_{V_1}}{\Lambda\mu\sqrt{\underline{\lambda}_{V_1}}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}+\frac{\epsilon_3}{\Lambda^l}\right) \tag{3.49}$$

Let's define $W_2=\left[\varepsilon_x^T \quad \eta^T \quad \varepsilon_z^T \quad \sqrt{\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2ds}\right]^T$ hence one gets:

$$\left\|W_2\right\| \leq \sqrt{\frac{\bar{\lambda}_{V_1}}{\underline{\lambda}_{V_1}}}\sup_{c\in[t_0-\tau,\,t_0]}\left\|W_2(c)\right\|e^{-\frac{0.5\Lambda\mu}{\bar{\lambda}_{V_1}}(t-t_0)}+\frac{2\bar{\lambda}_{V_1}}{\Lambda\mu\underline{\lambda}_{V_1}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}+\frac{\epsilon_3}{\Lambda^l}\right) \tag{3.50}$$

From the above equation and using $\varepsilon_z(t)=\Delta^{-1}\tilde{z}(t)$, the result given in (3.28) can be obtained.

This ends the proof of theorem 3.3.1

3.4 Sampled-delay observer

The observer designed above assumes that the outputs are continuously available. However, this is rarely in practice since these outputs are most of the time available at sampling instants $0 \leq t_0 < t_1 < \cdots < t_k < t_{k+1} < \cdots$ with sampling interval $h = t_{k+1} - t_k$. In this section, we propose an extended version of the above observer that deals simultaneously with sampled and delayed outputs problem. For this sake, we assume that the sampling interval h is bounded with a lower bound h_m and an upper bound h_M:

$$0 < h_m \leq h = t_{k+1} - t_k \leq h_M,\ \forall k \geq 0 \tag{3.51}$$

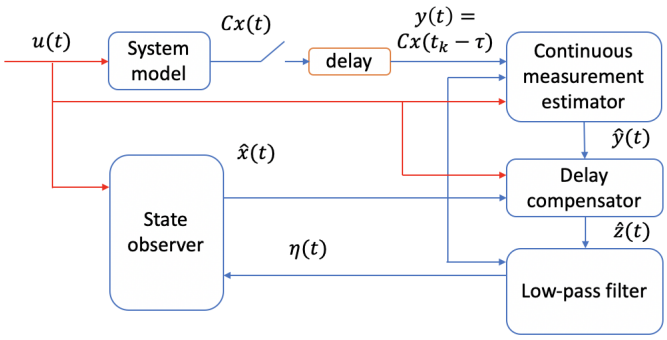


Figure 3.2: Synoptic diagram of the proposed sampled-delay observer

3.4.1 Observer design

In this part, a FHGO is re-designed. The proposed observer structure (depicted in Figure. 3.2) consists of four main cascaded sub-systems extending the observer (3.19). Hence the first three subsystems are similar to (3.19). The last subsystem is devoted to continuous-measurement estimation. Last but not least, this correction term allows us to increase the MAVTD, compared to previous works, while keeping the simplicity of the observer structure.

The proposed observer structure is as follows:

$$\begin{cases} \dot{\hat{x}}(t) & = A\hat{x}(t) + \varphi(u(t), \hat{x}(t)) + K_1 \Lambda \Delta \eta(t) \\ \dot{\eta}(t) & = \Lambda(A^T - K_2)\eta(t) + \Lambda C^T \left[\hat{z}(t) - C\hat{x}(t)\right] \\ \hat{z}(t) & = \hat{y}(t) + \int_{t-\tau}^{t} C\left[A\hat{x}(s) + \varphi(u(s), \hat{x}(s))\right] ds - \int_0^t \Lambda K_3 \left[\hat{z}(s) - C\hat{x}(t)\right] ds \\ \dot{\hat{y}}(t) & = C\left[A\hat{x}(t-\tau) + \varphi(u(t-\tau), \hat{x}(s-\tau))\right] - \Lambda K_4 \left[\hat{y}(t) - C\hat{x}(t-\tau)\right], \ t \in [t_k, t_{k+1}[\\ \hat{y}(t_k) & = y(t_k), \ t \in [t_k, t_{k+1}[\end{cases} \tag{3.52}$$

where $\hat{y}(t)$ represents the estimation of the output in the interval $[t_k;\ t_{k+1}[$. This estimation is updated at each instant t_k as shown by the last equation in (3.52). K_1,K_2, $K_3 = k_3 I_n$ and $K_4 = k_4 I_n$ are the proposed observer gains which are designed, under some conditions, in order to ensure the stability of the proposed observer (3.52), as given by theorem 3.4.1. Λ is the high-gain parameter and Δ is the block diagonal matrix given by:

$$\Delta = \text{diag}[I_n,\ \Lambda I_n,\ \cdots,\ \Lambda^{m-1} I_n],\ \Lambda > 0 \tag{3.53}$$

Theorem 3.4.1. *Consider system (3.2) under the assumptions 3.2.1, 3.2.3 and 3.2.4. For three positive scalars μ, μ_τ and μ_h, assume that there exists symmetric, positive and definite matrices $R = rI_p, Q_x, Q_\eta, P_x, P_\eta$ and a gain K_3 such that the LMI (3.54) holds.*

$$\Phi_2 = \begin{bmatrix} -Q_x + \frac{2L_\varphi P_x}{\Lambda} + \frac{R}{\Lambda} + \mu_\tau I_p & 0_p & \frac{A^T C^T + C^T K_3}{\Lambda^l} + \frac{L_\varphi}{\Lambda^{l+1}} C^T & 0_p \\ * & -Q_\eta + \mu_\tau I_p & -\Lambda^l P_\eta C^T & 0_p \\ * & * & -2K_3 + \frac{K_4}{\Lambda^{l-1}} & \frac{K_4 C}{\Lambda^l} \\ * & * & * & -\frac{R}{\Lambda} + \frac{\mu_h}{\Lambda} I_p \end{bmatrix} \leq -\mu I_{4p} \tag{3.54}$$

$$K_4 = k_4 I_n;\ \Lambda k_4 = 1. \tag{3.55}$$

then the observer (3.52) converges exponentially according to the following estimation errors bound:

$$\lim_{t\to\infty} \|\tilde{x}_i(t)\| \leq \frac{2\bar{\lambda}_V}{\Lambda^{1-i}\mu\underline{\lambda}_V} \left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{-l}\epsilon_2}{e^{(t-t_k)}} \right) \tag{3.56}$$

$$\lim_{t\to\infty} \|\tilde{z}(t)\| \leq \frac{2\bar{\lambda}_V}{\Lambda^{1-l}\mu\underline{\lambda}_V} \left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{-l}\epsilon_2}{e^{(t-t_k)}} \right) \tag{3.57}$$

where:

$$\underline{\lambda}_V = \min(\lambda_{\min}(P_x); 1) \tag{3.58}$$

$$\bar{\lambda}_V = \max(\lambda_{\max}(P_x); \lambda_{\max}(P_\eta); \Lambda\mu_\tau; \mu_h) \tag{3.59}$$

Proof of Theorem 3.4.1 :

Let us firstly define the estimation errors as follows:$\tilde{x}(t) = x(t) - \hat{x}(t)$, $\tilde{z}(t) = Cx(t) - \hat{z}(t)$, $\varepsilon_x(t) = \Delta^{-1}\tilde{x}(t)$, $\varepsilon_z(t) = \Lambda^{-l}\tilde{z}(t), l > 0$, $\tilde{\varphi}_i(t) = \varphi_i(u(t), x(t)) - \varphi_i(u(t), \hat{x}(t))$, $\tilde{\varphi}_i(t-\tau) = \varphi_i(u(t-\tau), x(t-\tau)) - \varphi_i(u(t-\tau), \hat{x}(t-\tau))$.

Using the facts that $\Delta^{-1}A\Delta = \Lambda A$, $C\Delta = C$, $CAB = 0$ and $CA\Delta = \Lambda CA$ one gets:

$$\dot{\varepsilon}_x(t) = \Lambda A\varepsilon_x(t) + \Delta^{-1}\tilde{\varphi}(t) + \Lambda^{1-m}B\xi(t) - K_1\Lambda\eta(t) \tag{3.60}$$

$$\dot{\eta}(t) = \Lambda(-K_2 + A^T)\eta(t) - \Lambda^{l+1}C^T\varepsilon_z(t) + \Lambda C^T C\varepsilon_x(t) \tag{3.61}$$

One gets:

$$\|CA\dot{\varepsilon}_x(t)\| \leq \Lambda\Omega_1 \left\|\tilde{X}_1\right\|; \ \ \|C\dot{\varepsilon}_x(t)\| \leq \Lambda\Omega_1 \left\|\tilde{X}_1\right\| \tag{3.62}$$

$\tilde{X}_1 = \begin{bmatrix}\varepsilon_x^T(t) & \eta^T(t)\end{bmatrix}^T$, $\Omega_1 = max\left(1 + \Lambda^{-1}L_\varphi; \lambda_{\max}(K_1)\right)$

Based on (3.52), the dynamics of $\tilde{y}(t) = Cx(t-\tau) - \hat{y}(t)$ and $\tilde{z}(t) = Cx(t) - \hat{z}(t)$ are given as follows:

$$\dot{\tilde{y}}(t) = (CA + \Lambda K_4)\,\tilde{x}(t-\tau) + C\tilde{\varphi}(t-\tau) - \Lambda K_4\tilde{y}(t) \tag{3.63}$$

$$\dot{\tilde{z}}(t) = \dot{\tilde{y}}(t) + CA\int_{t-\tau}^{t}\dot{\tilde{x}}(s)ds + C\left[\tilde{\varphi}(t) - \tilde{\varphi}(t-\tau)\right] + K_3C\tilde{x}(t) - K_3\tilde{z}(t) \tag{3.64}$$

By taking into account that $\tilde{y}(t_k) = \upsilon(t_k)$ and since the estimation of y, namely *haty* is computed in $t \in \left[t_k, \ \ t_{k+1}\right]$. Then the error estimation $\tilde{y}(t)$ can be obtained by solving the ODE (3.63) as follows:

$$\tilde{y}(t) = e^{-\Lambda K_4(t-t_k)}\int_{t_k}^{t}\left[(CA + \Lambda K_4)\,\tilde{x}(s-\tau) + C\tilde{\varphi}(s-\tau)\right]e^{-\Lambda K_4(s-t_k)}ds + \upsilon(t_K)e^{-\Lambda K_4(t-t_k)} \tag{3.65}$$

From (3.64) and (3.65) one gets:

$$\begin{aligned}\dot{\varepsilon}_z(t) = &-\Lambda^{2-l}K_4e^{-\Lambda K_4(t-t_k)}\int_{t_k}^{t}\left[(CA + K_4)\,\varepsilon_x(s-\tau) + \frac{C}{\Lambda}\tilde{\varphi}(s-\tau)\right]e^{-\Lambda K_4(s-t_k)}ds \\ &- \Lambda^{1-l}K_4\upsilon(t_K)e^{-\Lambda K_4(t-t_k)} + \Lambda^{1-l}\left(CA + K_3C\right)\varepsilon_x(t) + \Lambda^{-l}C\tilde{\varphi}(t) - \Lambda K_3\varepsilon_z(t) + \Lambda^{1-l}K_4C\varepsilon_x(t-\tau)\end{aligned} \tag{3.66}$$

Using the fact that in the interval $s \in [t_k;\ t]$, the functions $\|\varepsilon_x(s-\tau)\|$ and $e^{-\Lambda K_4(s-t_k)}$ are non-negative integrable functions and since:

$$\sup_{s\in[t_k,\ t]} e^{-\Lambda K_4(s-t_k)}ds = 1 \tag{3.67}$$

we can obtain the following inequality:

$$\int_{t_k}^{t}\left[(CA + K_4)\,\varepsilon_x(s-\tau) + \frac{C}{\Lambda}\tilde{\varphi}(s-\tau)\right]e^{-\Lambda K_4(s-t_k)}ds \leq \int_{t_k}^{t}\left(1 + k_4 + \frac{L_\varphi}{\Lambda}\right)\|\varepsilon_x(s-\tau)\|\,ds \tag{3.68}$$

Note that the dynamic equations of $\varepsilon_x(t)$ and $\eta(t)$ are similar to (3.60) and (3.61).

Consider now the following Lyapunov–Krasovskii function with time delay $\tau > 0$ and $d = t - t_k > 0$:

$$\begin{aligned}V = &\varepsilon_x^T(t)P_x\varepsilon_x(t) + \alpha\int_{-\tau}^{0}\int_{t+r}^{t}\left\|\tilde{X}_1(s)\right\|^2 dsdr + \eta^T(t)P_\eta\eta(t) + \beta(h-d)\int_{t_k}^{t}\|\varepsilon_x(s-\tau)\|^2 ds \\ &+ \varepsilon_z^T(t)\varepsilon_z(t)\end{aligned} \tag{3.69}$$

where α and β are positive scalars given as:

$$2r\Lambda\Omega_1 + \Lambda\mu = \ \alpha \tag{3.70}$$

$$h\Lambda^{2-l}k_4\left(1 + k_4 + \frac{L_\varphi}{\Lambda}\right)^2 + \Lambda\mu = \beta \tag{3.71}$$

the parameter μ_h and μ_τ are chosen as:

$$\beta h \leq \mu_h; \ \tau\alpha \leq \Lambda\mu_\tau \tag{3.72}$$

From (3.15) and (3.72) one gets

$$\underline{\lambda}_V W_5^T W_5 \leq V \leq \bar{\lambda}_V W_3^T W_3 \tag{3.73}$$

where:

$$W_3 = \begin{bmatrix} W_4^T & \sqrt{\int_{t_k}^t \|\varepsilon_x(s-\tau)\|^2 ds} & \sqrt{\int_{t-\tau}^t \left\|\tilde{X}_1(s)\right\|^2 ds} \end{bmatrix}^T \tag{3.74}$$

$$W_4 = \begin{bmatrix} \varepsilon_x^T(t) & \eta^T(t) & \varepsilon_z^T(t) & \varepsilon_x^T(t-\tau) \end{bmatrix}^T \tag{3.75}$$

$$W_5 = \begin{bmatrix} \varepsilon_x^T(t) & \varepsilon_z^T(t) \end{bmatrix}^T \tag{3.76}$$

From (3.60), (3.61), (3.62), (3.66), (3.68) and by taking into account the assumption (3.2.3) one gets:

$$\begin{aligned} \dot{V} \leq\ & \Lambda \tilde{X}_1^T \left(PM + M^T P\right) \tilde{X}_1 + 2L_\varphi \varepsilon_x^T(t) P_x \varepsilon_x(t) - 2\eta^T(t) P_\eta \Lambda^{l+1} C^T \varepsilon_z(t) + 2L_\varphi \Lambda^{-l} \|\varepsilon_z(t)\| \|C\varepsilon_x(t)\| \\ & - 2\Lambda \varepsilon_z^T(t) K_3 \varepsilon_z(t) + 2\Lambda^{1-l} \varepsilon_z^T(t) \left(CA + K_3 C\right) \varepsilon_x(t) \\ & + 2\Lambda^{2-l} k_4 e^{-\Lambda K_4 (t-t_k)} \|\varepsilon_z(t)\| \int_{t_k}^t \left(1 + k_4 + \frac{L_\varphi}{\Lambda}\right) \|\varepsilon_x(s-\tau)\| \, ds + \tau\alpha \left\|\tilde{X}_1\right\|^2 \\ & + 2\Lambda^{1-l} \varepsilon_z^T(t) K_4 C \varepsilon_x(t-\tau) - \alpha \int_{t-\tau}^t \left\|\tilde{X}_1(s)\right\|^2 ds - \beta \int_{t_k}^t \|\varepsilon_x(s-\tau)\|^2 ds + \beta(h-d) \|\varepsilon_x(t-\tau)\|^2 \\ & + 2\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} \|\varepsilon_x(t)\| + 2\frac{\Lambda^{1-l} K_4 \epsilon_2}{e^{\Lambda K_4 (t-t_k)}} \|\varepsilon_z(t)\| \end{aligned} \tag{3.77}$$

Using Young's inequality we obtain:

$$2\left\|\varepsilon_z^T(t)\right\| \int_{t_k}^t \left(1 + k_4 + \frac{L_\varphi}{\Lambda}\right) \|\varepsilon_x(s-\tau)\| \, ds \leq \|\varepsilon_z(t)\|^2 + \left[\left(1 + k_4 + \frac{L_\varphi}{\Lambda}\right) \int_{t_k}^t \|\varepsilon_x(s-\tau)\| \, ds\right]^2 \tag{3.78}$$

Using Jensen's inequality one gets:

$$\left(\int_{t_k}^t \|\varepsilon_x(s-\tau)\| \, ds\right)^2 \leq h \int_{t_k}^t \|\varepsilon_x(s-\tau)\|^2 ds \tag{3.79}$$

From (3.67), (3.71), (3.79), one gets:

$$\begin{aligned} & \Lambda^{2-l} k_4 e^{-\Lambda K_4 (t-t_k)} \left(\int_{t_k}^t \left(1 + k_4 + \frac{L_\varphi}{\Lambda}\right) \|\varepsilon_x(s-\tau)\| \, ds\right)^2 - \beta \int_{t_k}^t \|\varepsilon_x(s-\tau)\|^2 ds \\ & \leq -\Lambda\mu \int_{t_k}^t \|\varepsilon_x(s-\tau)\|^2 ds \end{aligned} \tag{3.80}$$

From (3.80) and by adding the term $0 = \mu_\tau \left\|\tilde{X}_1\right\|^2 - \mu_\tau \left\|\tilde{X}_1\right\|^2 + \mu_h \|\varepsilon_x(t-\tau)\|^2 - \mu_h \|\varepsilon_x(t-\tau)\|^2 + r\varepsilon_x(t)^T \varepsilon_x(t) - r\varepsilon_x(t-\tau)^T \varepsilon_x(t-\tau - r\varepsilon_x(t)^T \varepsilon_x(t) + r\varepsilon_x(t-\tau)^T \varepsilon_x(t-\tau)$ into (3.77) and from that fact that $\beta(h-d) \leq \beta h, \ \forall\, t \in \begin{bmatrix} t_k & t_{k+1} \end{bmatrix}$ we obtain:

$$\begin{aligned} \dot{V} \leq\ & \Lambda W_4^T \Phi W_4 + \tau\alpha \left\|\tilde{X}_1\right\|^2 - \Lambda\mu_\tau \left\|\tilde{X}_1\right\|^2 - \alpha \int_{t-\tau}^t \left\|\tilde{X}_1(s)\right\|^2 ds + \beta h \|\varepsilon_x(t-\tau)\|^2 - \mu_h \|\varepsilon_x(t-\tau)\|^2 \\ & - r\varepsilon_x(t)^T \varepsilon_x(t) + r\varepsilon_x(t-\tau)^T \varepsilon_x(t-\tau) - \Lambda\mu \int_{t_k}^t \|\varepsilon_x(s-\tau)\|^2 ds \\ & + \frac{2\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} \|\varepsilon_x(t)\| + \frac{2\Lambda^{1-l} k_4 \epsilon_2}{e^{\Lambda k_4 (t-t_k)}} \|\varepsilon_z(t)\| \end{aligned} \tag{3.81}$$

Using Leibniz's integration formula and (3.62) we obtain:

$$-r\varepsilon_x(t)^T \varepsilon_x(t) + r\varepsilon_x(t-\tau)^T \varepsilon_x(t-\tau) = -2r \int_{t-\tau}^t \varepsilon_x(s)^T \dot{\varepsilon}_x(s) ds \leq 2r\Lambda\Omega_1 \int_{t-\tau}^t \left\|\tilde{X}_1(s)\right\|^2 ds \tag{3.82}$$

Then from (3.70) one gets:

$$r\varepsilon_x(t-\tau)^T\varepsilon_x(t-\tau) - r\varepsilon_x(t)^T\varepsilon_x(t) - \alpha\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2 ds \le -\Lambda\mu\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2 ds \tag{3.83}$$

From (3.72) one gets: $\tau\alpha\left\|\tilde{X}_1\right\|^2 - \Lambda\mu_\tau\left\|\tilde{X}_1\right\|^2 \le 0$ and $\beta h\left\|\varepsilon_x(t-\tau)\right\|^2 - \mu_h\left\|\varepsilon_x(t-\tau)\right\|^2 \le 0$. Hence:

$$\begin{aligned}\dot{V} &\le \Lambda W_4^T\Phi_2 W_4 + \frac{2\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}}\|W_1\| + \frac{2\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda k_4(t-t_k)}}\|W_1\| - \Lambda\mu\int_{t_k}^{t}\|\varepsilon_x(s-\tau)\|^2 ds \\ &\quad - \Lambda\mu\int_{t-\tau}^{t}\left\|\tilde{X}_1(s)\right\|^2 ds\end{aligned} \tag{3.84}$$

From (3.54), we obtain:

$$\begin{aligned}\dot{V} &\le -\Lambda\mu\|W_3\|^2 + 2\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda K_4(t-t_k)}}\right)\|W_1\| \\ &\le -\frac{\Lambda\mu}{\bar{\lambda}_V}V + \frac{2}{\sqrt{\underline{\lambda}_V}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda K_4(t-t_k)}}\right)\sqrt{V}\end{aligned} \tag{3.85}$$

Or equivalently:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sqrt{V} \le -\frac{\Lambda\mu}{2\bar{\lambda}_V}\sqrt{V} + \frac{1}{\sqrt{\underline{\lambda}_V}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda k_4(t-t_k)}}\right) \tag{3.86}$$

Using the comparison lemma one gets:

$$\sqrt{V} \le \sup_{c\in[t_0-\tau,\,t_0]}\sqrt{V(c)}e^{-\frac{0.5\Lambda\mu}{\bar{\lambda}_V}(t-t_0)} + \frac{2\bar{\lambda}_V}{\Lambda\mu\sqrt{\underline{\lambda}_V}}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda K_4(t-t_k)}}\right) \tag{3.87}$$

From (3.73) one gets: $\underline{\lambda}_V W_5^T W_5 \le V \le \bar{\lambda}_V W_3^T W_3$ we can obtain: $\sqrt{\underline{\lambda}_V}\|W_5\| \le \sqrt{V}$; $\sup_{c\in[t_0-\tau,\,t_0]}\sqrt{V(c)} \le \sqrt{\bar{\lambda}_V}\sup_{c\in[t_0-\tau,\,t_0]}\|W_3(c)\|$. Hence:

$$\|W_5\| \le \sqrt{\frac{\bar{\lambda}_V}{\underline{\lambda}_V}}\sup_{c\in[t_0-\tau,\,t_0]}\|W_3(c)\|\,e^{\frac{-0.5\Lambda\mu}{\bar{\lambda}_V}(t-t_0)} + \frac{2\bar{\lambda}_V}{\Lambda\mu\underline{\lambda}_V}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda k_4(t-t_k)}}\right) \tag{3.88}$$

Or equivalently:

$$\lim_{t\to\infty}\|W_5\| \le \frac{2\bar{\lambda}_V}{\Lambda\mu\underline{\lambda}_V}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{1-l}k_4\epsilon_2}{e^{\Lambda k_4(t-t_k)}}\right) \tag{3.89}$$

Since the time t reach to sufficient large value, the convergence of (3.89) is guaranteed. Therefore, from (3.63), one can obtain:

$$\dot{\tilde{y}}(t) + \Lambda K_4\tilde{y}(t) = (CA + \Lambda K_4)\,\tilde{x}(t-\tau) + C\tilde{\varphi}(t-\tau) \le \alpha_y \tag{3.90}$$

where α_y is a small positive matrix with the same dimension with $\tilde{y}(t)$. Thanks to the existent of K_4 in (3.90), the convergence of $\tilde{y}(t)$ will be guaranteed as:

$$\lim_{t\to\infty}\|\tilde{y}(t)\| \le \frac{\|\alpha_y\|}{\Lambda k_4} \tag{3.91}$$

It is worth noting that the existent of K_4 guarantees the exponential convergence of $\tilde{y}(t)$. This estimation error is reversely proportional to k_4. However the estimation errors (3.56), (3.57) are proportional to this value due to measurement noise. Hence, in order to make sure the estimation errors converge to small values for $\|\tilde{y}(t)\|$ as well as $\|\tilde{z}(t)\|$, $\|\tilde{x}(t)\|$, one proposed to design the gain $K_4 = k_4 I_n$ such that:

$$\Lambda k_4 = 1 \tag{3.92}$$

Therefore, one gets:

$$\lim_{t\to\infty}\|W_5\| \le \frac{2\bar{\lambda}_V}{\Lambda\mu\underline{\lambda}_V}\left(\frac{\lambda_{\max}(P_x)\epsilon_1}{\Lambda^{m-1}} + \frac{\Lambda^{-l}\epsilon_2}{e^{(t-t_k)}}\right) \tag{3.93}$$

Recall $\varepsilon_x(t) = \Delta^{-1}\tilde{x}(t)$, $\varepsilon_z(t) = \Lambda^{-l}\tilde{z}(t), l > 0$ and $W_2 = \begin{bmatrix}\varepsilon_x^T(t) & \varepsilon_z^T(t)\end{bmatrix}^T$, one derives the results in (3.56), (3.57).

This ends the proof of theorem 3.4.1.

Remark 3.4.1. *According to theorem 3.4.1, the stability of the observer (3.52) can be achieved by computing the appropriate gains K_1, K_2, K_3 and K_4 such that the LMI (3.54) hold. These gains can be determined by applying the following procedure:*

- *Step 1: Choosing the parameters Λ, μ so that the estimation errors (3.56), (3.57) are small enough and the convergence time as desired. Note that, the value of Λ should be chosen high and $\Lambda k_4 = 1$.*
- *Step 2: Based on the Step 1, solve (3.71) to obtain β. Then choosing the positive constant μ_h so that (3.72) is satisfied.*
- *Step 3: From the obtained μ_h in Step 2, choose a maximum value for r such that $r < r_{\max} = 2(\Lambda\mu + \mu_h)$ (this value can guarantee the feasibility of the LMI (3.54). i.e $-r + \mu_h + \Lambda\mu < 0$). Then solve (3.70) to obtain α. Then choosing the positive constant μ_τ so that (3.72) is satisfied.*
- *Step 4: Solving the LMI (3.54) based on the above condition, one obtains K_3, P_x, P_η Q_x and Q_η.*
- *Step 5: Replacing $P = \text{diag}\{P_x, P_\eta\}$, $Q = \text{diag}\{Q_x, Q_\eta\}$ into (3.14) results us the appropriate the eigenvalues of matrix M.*
- *Step 6: Applying the algorithm given by Remark 2.2 in (Tréangle, Farza, and M'Saad, 2019a), the appropriate matrices K_1 and K_2 can be computed.*

Remark 3.4.2. *It is worth noting that the method proposed in this paper provides a larger MAVTD than those given in the literature. For example, when comparing it with the use of the High-Gain Observer in (Adil et al., 2020b), the authors imposed a condition: $\frac{1}{\tau} - 2\mu\Lambda^2 > 0$. In (Assche et al., 2011b), the authors presented a condition: $\tau \in \left[0; \frac{1}{2k_2\Lambda}\right]$; $k_2 \geq 2$. It is evident that the standard HGO cannot tolerate high MAVTD values when the high-gain parameter is chosen to be high. In other words, the higher the Λ, the lower the τ. Contrastingly, the method proposed in this work provides a tool to design the observer that satisfies any value of Λ and τ. Specifically, the LMI (3.54) is designed to ensure the satisfaction of (3.72). Consequently, the high-gain parameter, time delay, and sampling period are considered as the process parameters used to solve the LMI (3.54). Therefore, the conditions for not only the high-gain parameter but also the time delay and sampling period can be arbitrary based on the developer's preferences.*

3.5 Application to state estimation of a quadrotors UAV

3.5.1 Delay measurement problem

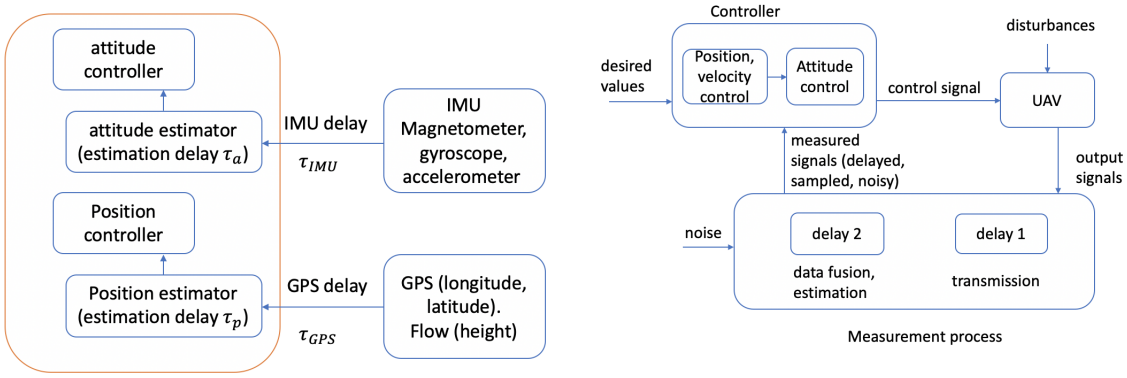


Figure 3.3: The origin of the measurement problem

Parameter	Simulation	Experiment	Unit
m	0.7427234	0.9885	kg
I_x	$1.5044e-02$	0.028	kgm^2
I_y	$1.5044e-02$	0.028	kgm^2
I_z	$2.8369e-02$	0.0397	kgm^2
J_r	$6.2342e-06$	$6.2342e-06$	kgm^2
τ_{IMU}	0.001	0.001	s
τ_{GPS}	0.17	0.17	s
τ_a	0.01	0.01	s
τ_p	0.01	0.01	s
h	0.005	0.005	s
g	9.81	m/s^2	
Max speed	3.5	3.5	m/s
Max acceleration	1	1	m/s^2
Max roll/pitch angle	45	45	deg

Table 3.1: Parameter of the UAV used in the application validation

In the UAV application, the delay effect is influenced by two main processes, as illustrated in Figure 3.3. The first process involves the intervention or induction in the measurement instrument, which leads to measurement delays (τ_{IMU} and τ_{GPS}). These delays represent the normal response time of the sensor, and in this study, we consider them as contributing factors to the overall measurement delay.

The second source of delay arises from data fusion processes, including filtering and attitude and position estimation. It should be noted that attitude, represented by Euler angles or quaternions, and the position in the x, y, z coordinates are not directly measured. Instead, they are estimated based on the available measurements.

For instance, attitude is typically estimated using values obtained from the magnetometer and gyroscope, resulting in a delay denoted as τ_a. Similarly, the position in Euclidean coordinates is estimated using longitude and latitude values from GPS, leading to a delay denoted as τ_p.

Accordingly, the natural delay affected on the measurements can be summarised as:

$$\text{Attitude measurement } :\tau_1 = \tau_a + \tau_{IMU} \tag{3.94}$$

$$\text{Position measurement } :\tau_2 = \tau_p + \tau_{GPS} \tag{3.95}$$

The physical parameters of the UAV used are given in Table 3.1.

3.5.2 Simulation validation

In this section, the application of the proposed observers to the quadrotor UAV is studied in a simulation environment. In the first section, the simulation environment using Simscape-multibody 3D model is presented, explaining how to run simulations of the system and how we embedded the observer. In a second part, the obtained simulation results will be shown and discussed.

Simscape-multibody 3D simulation

Simscape-multibody is a toolbox provided by Mathworks in Matlab-simulink environment. Simscape Multibody provides a remarkable simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear.

Simscape Multibody provides very useful tool for developing control systems, observer and test system level performance. We can parameterize our models using MATLAB variables and expressions, and design controller, observers for our multibody system in Simulink.

In this simulation, we use the simscape 3D system of a quadrotor UAV as shown in Figure . 3.4. The model parameters are given in Table 3.1.

Figure 3.4: Simscape-multibody 3D model of a quadrotor UAV

The simulation setup is shown in Figure. 3.5 which consists of seven parts:

- Guidance block: This guidance block is in charge of providing the desired position and the tearing angle for the quadrotor UAV.
- Controller block: In this block, a controller is employed to control the quadrotor UAV toward the desired position provided by the guidance part. The output of the controller block will be three control rotation torques and a thrust force.
- Thrust force generator block: From the obtained three control rotation torques and a thrust force, this block will be in charge of generating the appropriate thrust force needed for each rotor and its corresponding rotation speed. Hence, these value will be transmit to the quadrotor UAV block.
- Quadrotor system block: This is the mechanical model of quadrotor system used in this study. The thrust forces generated from "Thrust force generator block" will operate the quadrotor so that it can execute the needed task.
- Output block: This block simulates the sensor and provides the output measurement for the observer and controller studies.
- Virtual environment block: This is where we define the environment of our simulation environment such as gravity, external disturbance, ground ...

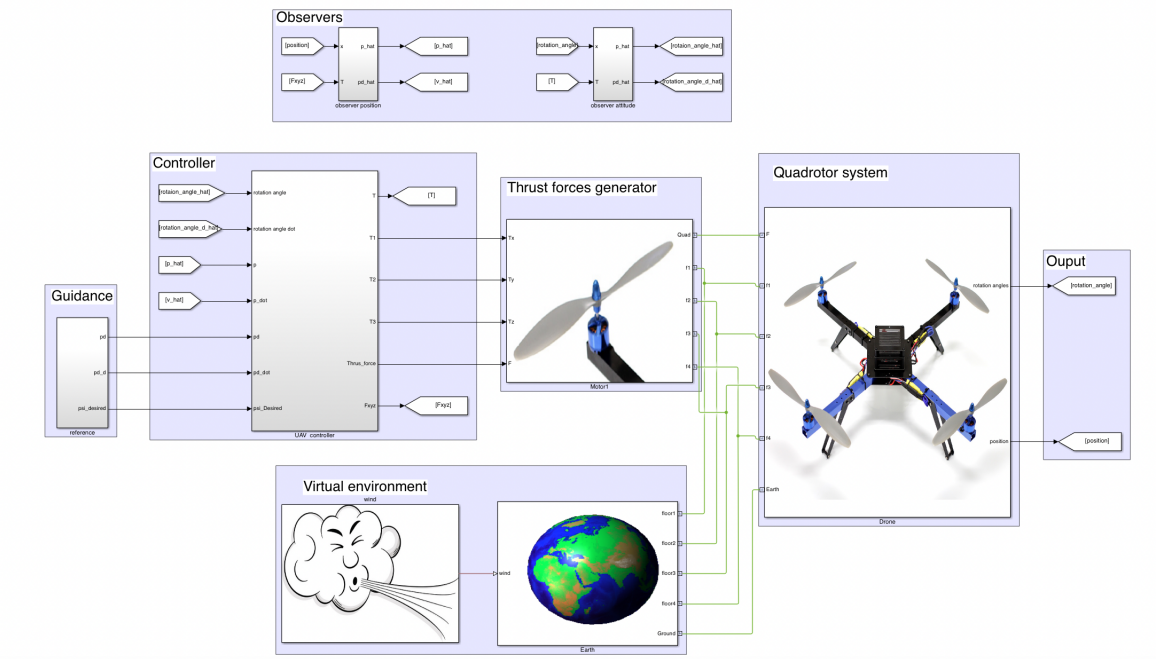


Figure 3.5: Simulation framework

- Observer block: this is where we can employ our observer to verify its performance.

Two simulations will be carried out. The first simulation is for validating the performance of the proposed observer in comparison with the other methods, namely (Kalman-Like Observer) KLO in (Cuny et al., 2020) and HGO in (Assche et al., 2011b). The second simulation is devoted to show the effects of adding observer to improve the control performance when the measurement is delayed and noisy.

Based on the given parameters , one set $L_\varphi = 3$, the high gain parameter and μ is chosen as $\Lambda = 20$, $\mu = 1$. Based on remark 3.4.1, the observer for position and attitude estimation are as follows:

Position estimation

Based on (3.70-3.72) one gets: $\beta = 20.144$, $\mu_h = 0.1$, $\alpha = 1628$ and $\mu_\tau = 13.84$ for the case of position observer. Solving the LMI (3.54) one gets $K_3 = 0.6773I_3$; $K_4 = 0.05I_3$, the eigenvalues (triple) of the matrix M can be obtained: $[-0.3511 \quad -0.3525 \quad -15.8254 \quad -0.5389]$. The gains K_1 and K_2 are given as $K_1 = \text{diag}[0.5071I_3,\ 1.0555I_3]$, $K_2 = \text{diag}[1.2422I_3,\ 15.8257I_3]$.

Attitude estimation

Similarly, one gets: $\beta = 20.144$, $\mu_h = 0.1$, $\alpha = 1628$ and $\mu_\tau = 0.9$ for the case of attitude observer. Solving the LMI (3.54) one gets $K_3 = 0.6934I_3$; $K_4 = 0.05I_3$, the eigenvalues (triple) of the matrix M can be obtained: $[-0.3381 \quad -0.3346 \quad -16.3168 \quad -0.4968]$. The gains K_1 and K_2 are given as $K_1 = \text{diag}[0.4508I_3,\ 0.9170I_3]$, $K_2 = \text{diag}[1.1693I_3,\ 16.3170I_3]$.

For the sake of validating the observer performance in term of compensating the delay effects and noise measurement. The measurement noise is modeled as in (3.4) with the following parameters: $a = [0.02 \quad 0.01 \quad 0.007]^T$, $\omega = [10\pi \quad 25\pi \quad 12.5\pi]^T$ and $f = [\pi/5 \quad \pi/7 \quad \pi/3]^T$. To the output measurement of attitude together with time delay $\tau_{IMU} = 0.02s$. $a = [0.06 \quad 0.03 \quad 0.031]^T$, $\omega = [10\pi \quad 25\pi \quad 12.5\pi]^T$ and $f = [\pi/5 \quad \pi/7 \quad \pi/3]^T$ to the output measurement of attitude together with time delay $\tau_{GPS} = 0.18s$.

Since $\Lambda = 20$, the MAVTD for KLO and HGO are $\tau \leq 0.0354s$ and $\tau \leq 0.0125s$. From these conditions, we expect that the results given by KLO and HGO will diverge in the application for position observer, due to the time delay in this case is determined as $\tau_2 = 0.18s$.

Simulation results

The state estimations are given in the Figure.3.6 and 3.7. The proposed FHGO provides a relatively smooth state estimations comparing with HGO and KLO are highly affected by measurement noise.

Moreover, as depicted in Figure. 3.7 when the time delay increases to $\tau = 0.18$, the estimations provided by KLO and HGO diverge while that given by FHGO converges. This results verify the effectiveness of our proposed structure.

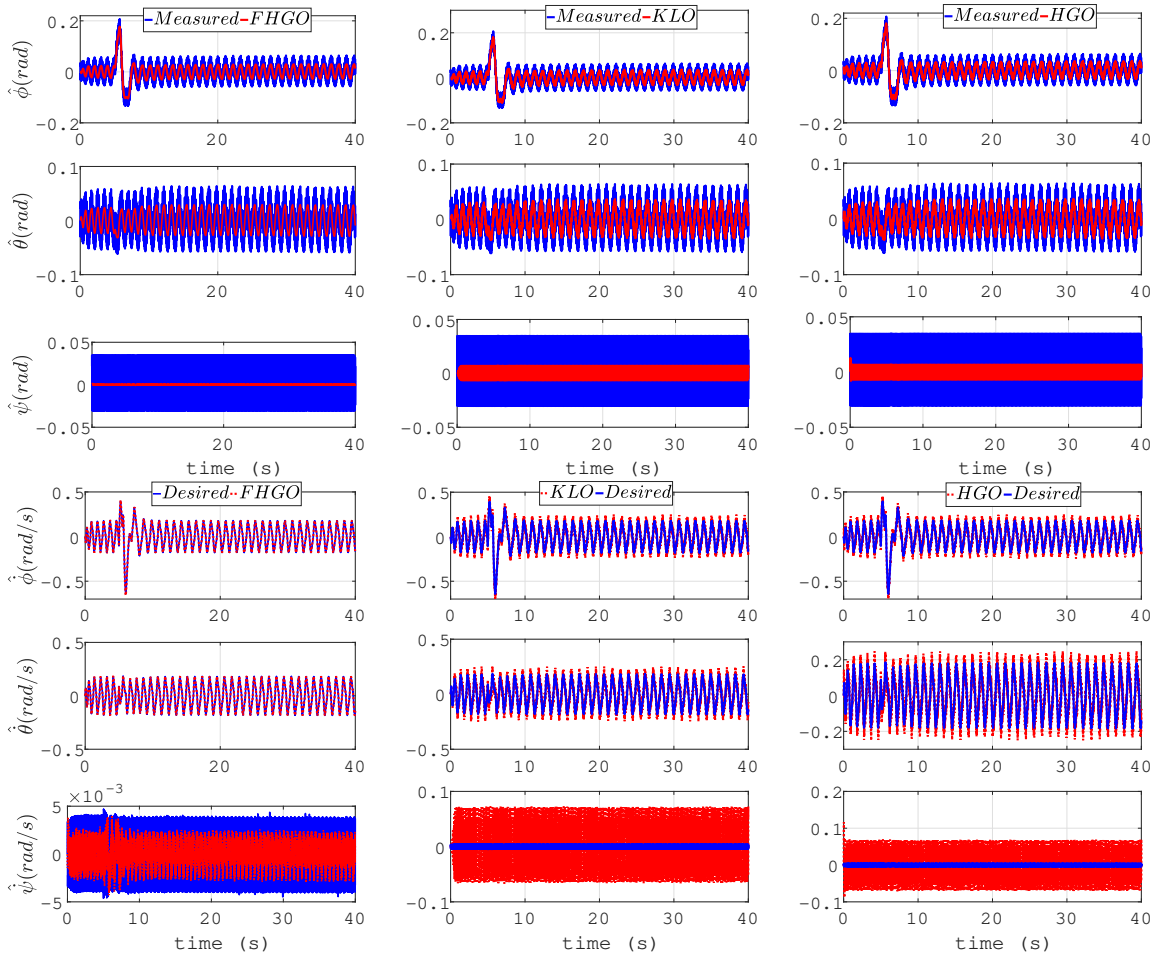


Figure 3.6: Simulation results of state estimation for attitude estimation using FHGO (left), KLO (Cuny et al., 2020) (middle), HGO (Assche et al., 2011b) (right).

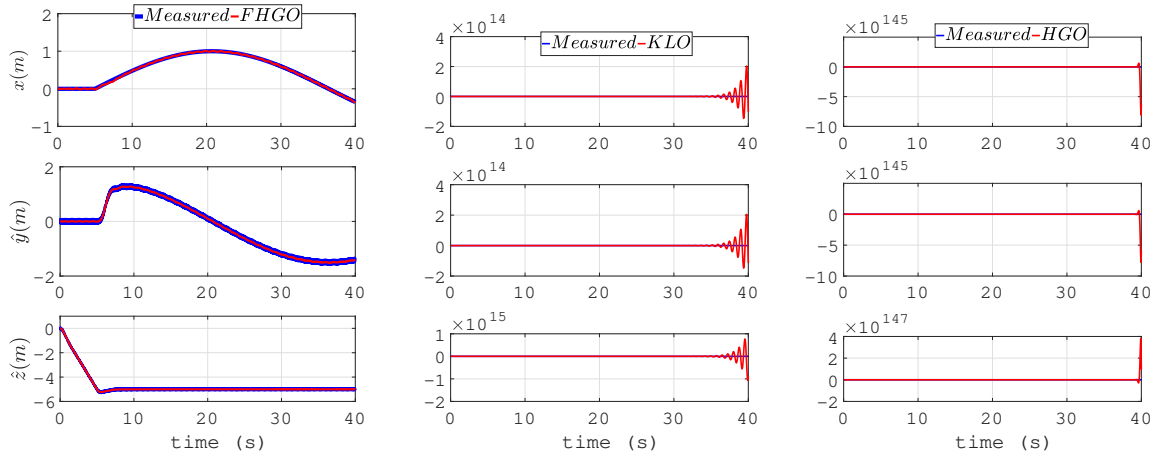


Figure 3.7: Simulation results of position estimation using FHGO (left), KLO (Cuny et al., 2020) (middle), HGO (Assche et al., 2011b) (right).

3.5.3 Experimental validation

In this section, the experiment validation is carried out to illustrate the effectiveness of the proposed FHGO compared to KLO in (Cuny et al., 2020) and HGO in (Assche et al., 2011b).

Experimental setup

In order to validate the proposed observer, a real outdoor flight experiment is performed. The experimental setup is shown in Figure. 3.8. The platform used in this experiment consists of three main components:

- Quadrotor UAV: The UAV uses s500 quadcopter frame, enquired with a Pixhawk 4 autopilot module. The Pixhawk 4 board includes two processors. A main one devoted to computation and another one dedicated to input/output processing. Four sensors are embedded, two gyroscopes, a magnetometer, an accelerometer and a barometer. A GPS module is also enquired to determine the position of UAV. A radio receiver is connected with Pixhawk for receiving the radio controller signal for manual control mode. A telemetry module used to communicate with Ground station PC.

- Radio controller: The remote controller (FrSky Taranis X9-Lite) will be used to manually control the UAV.

- Ground station PC. The ground station laptop allows us to install the observer in Matlab Simulink, The laptop communicate with the UAV through the telemetry radio (200Hz) communication using MAVlink protocol.

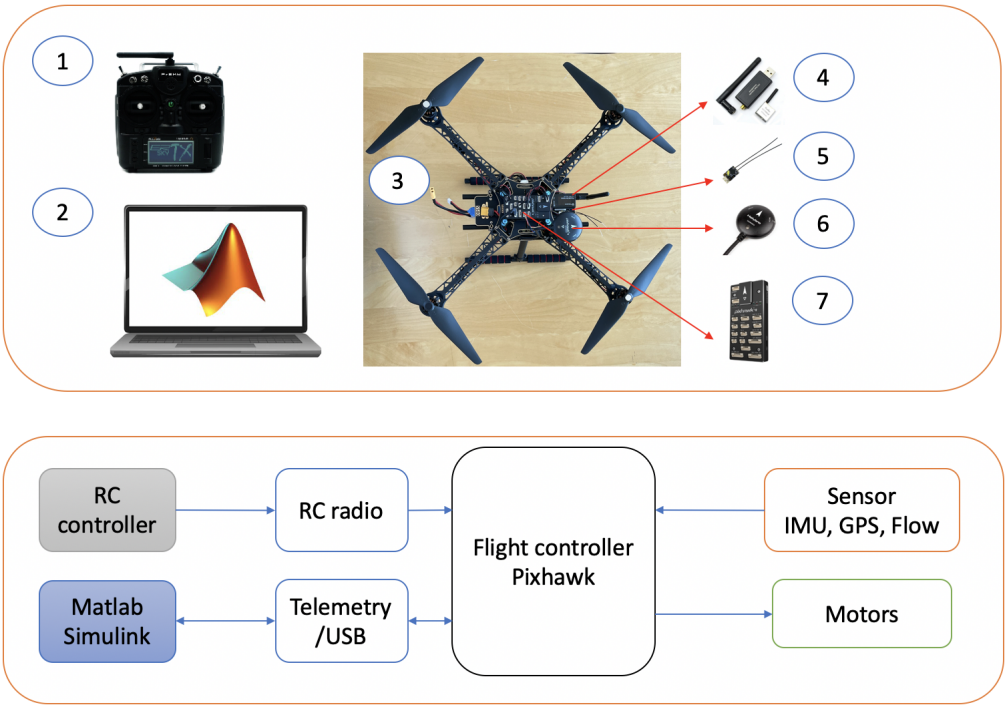


Figure 3.8: (Above) Experimental setup for the outdoor flight: 1. Radio controller; 2. Ground station laptop with Matlab Simulink; 3. Quadrotor UAV; 4. Radio telemetry module; 5. Radio receiver; 6. GPS module; 7. Pixhawk card. (Below) Experimental setup schema.

The Pihawk (Px4) provides a dynamic platform for academy research. The Firmware installed inside the board is suitable for many aerospace research application such as bicopter, quadcopter, hexacopter, helicopter and so on. We can easily configure the Px4 board to the suitable platform. However, this is also the main disadvantages of using Px4 for a specific application, when specialization is more emphasized. Due to heavy Firmware has been installed inside, the usable memory for developing and embedding the other algorithm into the Px4 becoming difficult. Many modules are hidden and untouchable, while some of them might not necessary for some specific research. For example, we need only to build the observer. Moreover, the source code using for Pixhawk is under developing, and being updated daily. As a results when we are using old Pixhawk board, this board might not be synchronized with the current open source Firmware. For this reason, the researcher is required to update the Pixhawk board to the higher model, and that might be expensive.

Due to the limitation in the memory resource, installing multiple observers into the card is impossible. Hence, we chose to install the observers in the station laptop, the input of the observer obtained from the measurement given by the UAV is the same as if install in the card, then the property and the effective of the observer will be similar. Moreover, the comparative analysis with other observers can be easily executed.

The UAV is controlled by the Radio controller outdoor. The measurements such as position and attitude of the UAV then will be obtained using Pixhawk board. These signal then will be transmitted to the Ground PC through telemetry radio using MAVlink protocol.

The observer parameter used in this section will be the same as in simulation.

Experimental results

Figure 3.9 depicts the state estimation for attitude estimation application. The proposed FHGO performs a relatively smooth estimation, thank to the adding low-pass filter $\eta(t)$. By contrast, the results given by KLO and HGO are highly affected by measurement noise and this effects proportionally increases to the order of the state (effects in rate estimation is higher than that in angle estimation). These results have well presented the disadvantages of using classical high gain technique with measurement noise.

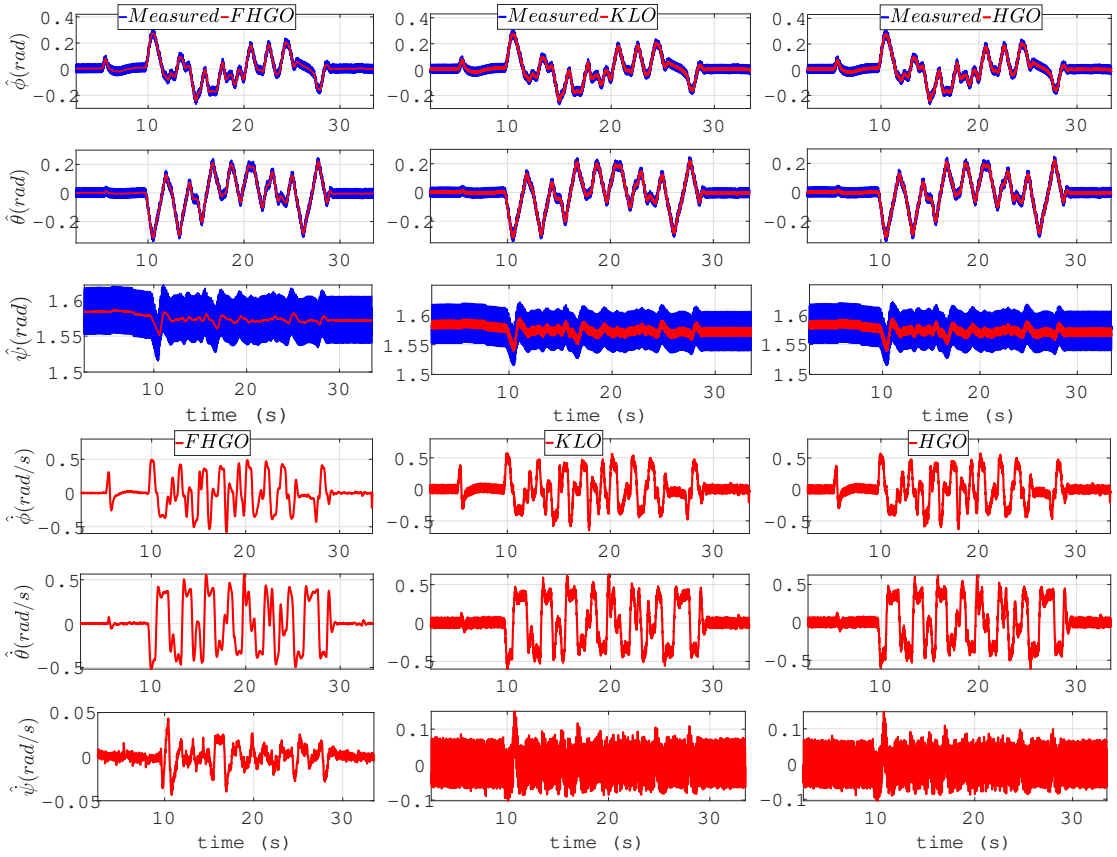


Figure 3.9: Experimental results of state estimation for attitude estimation using FHGO (left), KLO (Cuny et al., 2020) (middle), HGO (Assche et al., 2011b) (right).

Figure 3.10 shows the position estimation results using different method. As discussed earlier, since the delay in this case is larger, namely $\tau_2 = 0.18$, the stabilities of KLO and HGO are lost, which results the estimations given by KLO and HGO diverge. This theoretical conditions is experimentally confirmed in the application for position estimation. However, as depicted in Figure 3.10, the proposed method provides a relatively smooth estimation and the noise rejection ability is well performed.

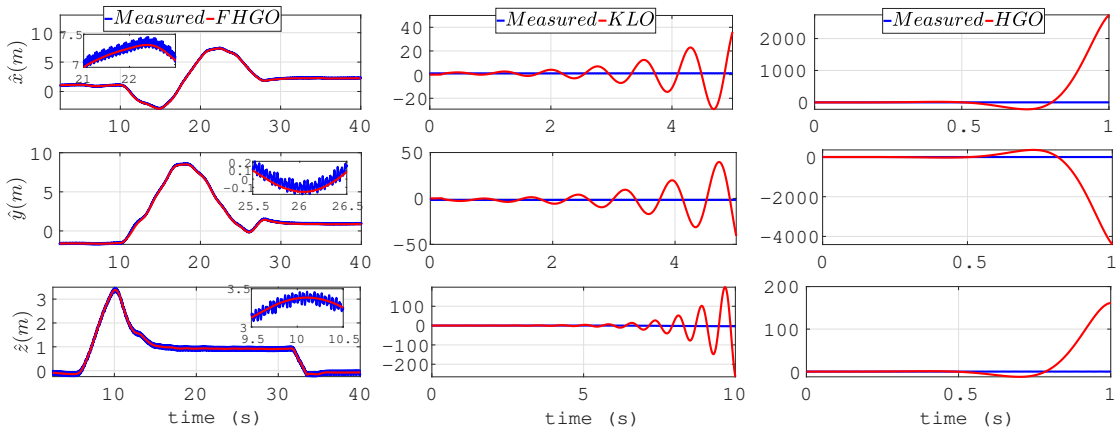


Figure 3.10: Experimental results of position estimation using FHGO (left), KLO (Cuny et al., 2020) (middle), HGO (Assche et al., 2011b) (right).

3.6 Discussion

The observers proposed in this chapter provide a very useful tool for the state estimation in the existence of delay measurement. However, In the context of delay system, most of the works concerning the delay problem including the previously proposed observer, the time delay is assumed known. This assumption, although it simplifies the problem, can be restricted in applications due to the difficulty of measuring the time-delay. Therefore, the identifiability and the estimation of the delay play an important role in the field of time-delay system.

Motivated by this problem, the delay identification for an unknown output delay was studied. In order to preserve the generality of the study, the estimator will be proposed for a drift observable non-linear system in general.

3.7 Delay identification for an unknown output delay

Regarding the delay identification problem, many methods have been proposed, such as the identification by using variable structure observers (Drakunov et al., 2006),(Orlov et al., 2003) and neural network algorithms (Yadaiah et al., 2007). The reader can refer to (Belkoura et al., 2004; Drakunov et al., 2006) and references therein for additional methods and details. In (Orlov et al., 2002) and (Deng et al., 2021), the problems of identifiability and estimation of time-delay are studied for linear systems, where a simple sufficient condition for delay identifiability was proposed. However, for a nonlinear time-delay systems, this problem remains an open challenging issue.

In (Nguyen, Tan, and Trinh, 2021), a sliding mode observer has been proposed for nonlinear systems to estimate the delay by minimizing a cost function. In (Cacace et al., 2016) and (Cacace et al., 2017), a high-gain observer has been used to achieve simultaneous estimation of the states and the time-delay. In (Anguelova and Wennberg, 2008), state elimination has been used to identify the delay. However, the existing methods in the literature ((Cacace et al., 2016), (Nguyen, Tan, and Trinh, 2021), (Cacace et al., 2017), (Anguelova and Wennberg, 2008)) are proposed under a restrictive condition. Indeed, the system trajectory has to be monotonic, which is hard to satisfy from a practical point of view.

On the other hand, in many works such as (Cacace et al., 2016), (Cacace et al., 2017), (Drakunov et al., 2006), and (Zheng, Polyakov, and Levant, 2018), where the state and the delay have been estimated simultaneously, a partial derivative with respect to the estimated delay has been used to design the time delay estimator. It is clear that using a partial derivative with respect to a time-varying function requires heavy computation and may not be easily obtained.

In this part, the problem of simultaneously estimating the states and the output constant delay for nonlinear systems is addressed. The proposed estimator uses a condition that is less restrictive than the ones previously mentioned. Moreover, the need for the partial derivative with respect to the estimated delay is eliminated, as the proposed delay estimator is based on the derivatives of known functions with respect to time.

3.7.1 Problem statement

In order to preserve the generality of the work, let us consider the following nonlinear system with output delay as follow:

$$\begin{cases} \dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\ y(t) &= h(x(t-\tau)) \\ x(t) &= \Psi(t) \ \text{ where } \ t \in [-\tau_{\max},\ 0] \end{cases} \tag{3.96}$$

where $x(t) \in \mathbb{X} \subset \mathbb{R}^n$ is the system state which assumed to be bounded, $u(t) \in \mathbb{U} \subset \mathbb{R}$ is the known input which assumed to be predefined, $y(t) \in \mathbb{R}$ is the output. f, g and h are differentiable vector functions, τ is the unknown constant delay which affects the output measurement and is assumed to be bounded $\tau \in [\tau_{\min}, \tau_{\max}]$. $\Psi(t)$ represents the unknown continuous initial functions.

Let us now consider the change of coordinate $w(t) = x(t-\tau)$. Since τ is constant, one gets : $\dot{w}(t) = \dot{x}(t-\tau)$. Under the assumptions that the functions f, g and h are differentiable, the system (3.96) can be represented in the new coordinates as follows:

$$\begin{cases} \dot{w}(t) &= f(w(t)) + g(w(t))u(t-\tau) \\ y(t) &= h(w(t)) \end{cases} \tag{3.97}$$

It is worth noting that from (3.96) , the initial condition of $w(t)$ is determined then $w(t)$ is smooth $\forall t >\geq 0$ if f, g, u are smooth functions.

Remark 3.7.1. *The time-delay τ is assumed to be unknown. As a result, the function $u(t-\tau)$ is considered also unknown. Therefore, the system (3.97) can be considered as a non-delayed system affected by unknown inputs.* □

If $f(w(t))$, $g(w(t))$ and $h(w(t))$ have uniform observation relative degree equals to n $\forall x \in \mathbb{X}$, i.e.

$$L_g L_f^i h(x(t)) = 0;\ i = 0, 1, \cdots, n-2 \tag{3.98}$$

where $L_f h(w(t))$ is the Lie derivative of the function h with respect to the vector f defined such that:

$$L_f h(w(t)) = \sum_{i=1}^{n} \frac{\partial h(w(t))}{\partial w_i(t)} f_i(w(t)) \tag{3.99}$$

and $L_f^k h(w(t))$ represents the k-times repeated iteration of $L_f h(w(t))$, then, the following mapping $z(t) = \Phi(w(t))$ can be proposed:

$$z(t) = \Phi(w(t)) = \begin{bmatrix} h(w(t)) \\ L_f h(w(t)) \\ \vdots \\ L_f^{n-1} h(w(t)) \end{bmatrix} \tag{3.100}$$

Based on (Cacace, Germani, and Manes, 2010b) and using the change of coordinates $z(t) = \Phi(w(t))$, the dynamic of system (3.97) in the new base $z(t)$ can be obtained as follows:

$$\begin{cases} \dot{z}_1(t) &= z_2(t) \\ \dot{z}_2(t) &= z_3(t) \\ &\vdots \\ \dot{z}_n(t) &= F(z(t), u(t-\tau)) \\ y(t) &= z_1(t) \end{cases} \tag{3.101}$$

where $F(z(t), u(t-\tau))$ is defined as:

$$\begin{aligned} F(z(t), u(t-\tau)) &= \bar{F}(w, u, t, \tau)|_{w(t)=\Phi^{-1}(z(t))} \\ &= f_F(z,t) + g_F(z,t)u(t-\tau) \end{aligned} \tag{3.102}$$

where

$$f_F(z,t) = L_f^n h(w(t))|_{w(t)=\Phi^{-1}(z(t))} \tag{3.103}$$

$$g_F(z,t) = L_g L_f^{n-1} h(w(t))|_{w(t)=\Phi^{-1}(z(t))} \tag{3.104}$$

For the sake of notation simplicity, the function $F(z(t), u(t-\tau))$ will be noted as:

$$F(z(t), u(t-\tau)) = F(z, u, t, \tau) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R} \tag{3.105}$$

Since the function $F(z, u, t, \tau)$ is nonlinear, it is worth noticing that the identifiability of time-delay in this nonlinear case is challenging. Therefore, the following part will be devoted for solving this problem.

The proposed delay estimator will be designed under the following assumptions:

Assumption 3.7.1. *With the initial function $\Psi(t)$, the system (3.96) has a unique smooth and bounded solution on the compact set $\mathbb{X} \subset \mathbb{R}^n$, i.e $x(t) \in \mathbb{X}$; $\forall t \geq -\tau_{\max}$*

According to the assumption 3.7.1, the image $z(t)$ of $x(t)$, using the mapping (3.100), is smooth and bounded i.e $z(t) \in \mathbb{Z} \subset \mathbb{R}^n$. Thus the i^{th} derivative of output exists such that $y^{(i)}(t)$ is smooth and bounded for $1 \leq i \leq n$. Therefore, the following assumption can be considered:

Assumption 3.7.2. *The $(n+1)th$ derivative of the output measurement $y(t)$ is bounded i.e:*

$$\exists\ L_y > 0 \text{ such that } \forall t > 0, \left| y^{(n+1)}(t) \right| \leq L_y \tag{3.106}$$

This assumption leads to the following inequality:

$$\left| \frac{d}{dt} F(z, u, t, \tau) \right| \leq L_y, \quad \forall t > 0 \tag{3.107}$$

Assumption 3.7.3. *$f(w(t))$, $g(w(t))$ and $h(w(t))$ of the system (3.97) have uniform observation relative degree which is, at least, equal to n.*

Assumption 3.7.4. *The control function $u(t)$ is a smooth and differentiable function.*

3.7.2 Delay identifiability

Let us firstly introduce the following distinguishability definitions regarding different delays.

Definition 3.7.5. *$\forall z \in \mathbb{Z}, u \in \mathbb{U}$, two different delays τ and $\bar{\tau}$ are said to be distinguishable if there exists an interval $[t-T; t]$; $T > 0$ such that:*

$$\int_{t-T}^{t} |F(z, u, s, \tau) - F(z, u, s, \bar{\tau})|\, ds > 0 \quad \forall\ t > 0 \tag{3.108}$$

They are globally distinguishable if (3.108) is satisfied for all τ, $\bar{\tau} \in \mathbb{R}^+$ and locally distinguishable if (3.108) is satisfied for τ, $\bar{\tau}$ in the interval $[\tau_{\min},\ \tau_{\max}]$.

Definition 3.7.6. *$\forall z \in \mathbb{Z}, u \in \mathbb{U}$, the time-delay τ is said to be identifiable if in the interval $[t-T; t]$; $T > 0$ the following condition is satisfied :*

$$\int_{t-T}^{t} |F(z, u, s, \tau) - F(z, u, s, \bar{\tau})|\, ds = 0;\ \forall t > T \tag{3.109}$$

which implies $\tau = \bar{\tau}$ for all $t > T_c$. If (3.109) is satisfied for all $\bar{\tau} \in \mathbb{R}^+$, then the identifiability is global. If (3.109) is satisfied for $\bar{\tau}$ in the interval $[\tau_{\min},\ \tau_{\max}]$, then the identifiability is local. It is worth noting that if condition (3.109) is verified, then one gets the following consequence:

$$|F(z, u, t, \tau) - F(z, u, t, \bar{\tau})| = 0 \quad \forall\ t > 0 \tag{3.110}$$

In the following, a finite-time observer, using implicit Lyapunov function approach, will be applied to estimate the state in the new base $z(t)$ and the unknown constant delay.

3.7.3 State and delay estimation design

For the sake of delay estimation, the following assumptiion will be considered:

Assumption 3.7.7. *The delay τ is at least locally identifiable.*

Based on assumptions 3.7.1 and 3.7.3 the system (3.96) can be transformed into the canonical form (3.101) contains $\dot{z}_i = y^{(i)}$ for $1 \leq i \leq n$.

In the following, based on assumption 3.7.2, 3.7.4, 3.7.7 the state and delay estimator will be proposed. Notice that the estimation procedure is done into two steps: Firstly, a high order sliding mode observer (HOSMO) based on (Levant, 2003) is proposed for the system (3.101) to estimate the state $z(t)$ and the function $F(z,u,t,\tau)$. It is worth noting that, the estimation of $F(z,u,t,\tau)$ is crucial for delay estimation. Hence, in this case, the extended state technique is applied and $F(z,u,t,\tau)$ is considered as an augmented state. This estimation is also known as input estimation and has been studied in (Dam et al., 2021). The state estimation is proposed as follows:

$$\dot{\hat{z}}_1(t) = \hat{z}_2(t) + k_1 \left|\hat{z}_1(t) - y(t)\right|^{\frac{-1}{n+1}} \left(\hat{z}_1(t) - y(t)\right) \tag{3.111}$$

$$\dot{\hat{z}}_2(t) = \hat{z}_3(t) + k_2 \left|\hat{z}_1(t) - y(t)\right|^{\frac{-2}{n+1}} \left(\hat{z}_1(t) - y(t)\right) \tag{3.112}$$

$$\vdots$$

$$\dot{\hat{z}}_n(t) = k_n \left|\hat{z}_1(t) - y(t)\right|^{\frac{-n}{n+1}} \left(\hat{z}_1(t) - y(t)\right) + \hat{F}(z,u,t,\tau) \tag{3.113}$$

$$\dot{\hat{F}}(z,u,t,\tau) = k_{n+1} \left|\hat{z}_1(t) - y(t)\right|^{-1} \left(\hat{z}_1(t) - y(t)\right) \tag{3.114}$$

where $\Delta_{(t,\hat{\tau})}F = \hat{F}(z,u,t,\tau) - F(\hat{z},u,t,\hat{\tau})$, $\hat{\tau}(0)$ is initial condition of the delay estimator and Γ presents the estimator gain which should be just positive.

Remark 3.7.2. *The observer (3.111-3.114) is designed based on HOSMO in (Levant, 2003). As mentioned in (Levant, 2003) with the appropriate design of the gains k_1, k_2, $\cdots$, k_{n+1}, the proposed observer will converge in a finite time T_f. Thus, since $t \geq T_f$ one gets $\hat{z}_i(t) = z_i(t)$ for $1 \leq i \leq n$ and $\hat{F}(z,u,t,\tau) = F(z,u,t,\tau)$. The design of the gains k_1, k_2, $\cdots$, k_{n+1} is detailed using implicit Lyapunov functionnal given in (Polyakov, Efimov, and Perruquetti, 2015) will be designed in the following theorem.*

Theorem 3.7.8. *Consider the system (3.96) under the assumptions 3.7.1, 3.7.2. If there exists the SPD matrices P and Z, and a matrix $Y = PK$ (where K is the observer gain) given in , and $\alpha \geq L_y$ such that the following LMIs hold:*

$$\begin{bmatrix} P\bar{A} + \bar{A}^T P - \bar{C}^T Y^T - YC + (1+\alpha)P & P & PB \\ P & -Z & 0 \\ B^T P & 0 & -L^{-1} \end{bmatrix} \leq 0 \tag{3.115}$$

$$\begin{bmatrix} 1 & Y^T \\ Y & P \end{bmatrix} \geq 0; \; Z \leq P. \tag{3.116}$$

then the observer (3.111-3.114) is uniformly finite-time stable and the observer gain K can be computed as $K = P^{-1}Y$

Secondly, based on the state estimation $\hat{z}(t)$ and the estimated function $\hat{F}(z,u,t,\tau)$ resulting from the finite time observer, the delay estimation is established (3.117)-(3.118) in the following theorem:

Theorem 3.7.9. *Consider the system (3.96) under the assumptions 3.7.1, 3.7.2, 3.7.3 and 3.7.4. If $|g_F(\hat{z},t)\dot{u}(t-\hat{\tau})| > 0 \; \forall t > \max(T_f; \tau_{\max})$, and the estimation of time delay is designed as follows:*

$$\dot{\hat{\tau}}(t) = \frac{-\dot{\hat{F}}(z,u,t,\tau) + \dot{f}_F(\hat{z},t) + \dot{g}_F(\hat{z},t)u(t-\hat{\tau})}{g_F(\hat{z},t)\dot{u}(t-\hat{\tau})} + \frac{g_F(\hat{z},t)\dot{u}(t-\hat{\tau}) - \Gamma\Delta_{(t,\hat{\tau})}F}{g_F(\hat{z},t)\dot{u}(t-\hat{\tau})}; \; \forall t > \max(T_f; \tau_{\max}) \tag{3.117}$$

$$\hat{\tau}(t) = \hat{\tau}(0); \; \forall t \leq \max(T_f; \tau_{\max}) \tag{3.118}$$

then $\hat{\tau}(t)$ converges locally and exponentially to τ $\forall t \geq T_f$. □

Proof of Theorem 3.7.8: The problem of design finite-time observer (3.111-3.114) will be studied in this part using implicit Lyapunov functionnal method proposed in (Polyakov, Efimov, and Perruquetti, 2015).

Let us define the estimation error of the finite-time observer as follow:

$$e = \begin{bmatrix} z_1(t) - \hat{z}_1(t) & \cdots & z_n(t) - \hat{z}_n(t) & -\Delta_{(t,\hat{\tau})}F \end{bmatrix}^T \tag{3.119}$$

Let us recall the framework of implicit Lyapunov functionnal defined as follows (Polyakov, Efimov, and Perruquetti, 2015) :

$$Q(V,e) = e^T D_\theta(V^{-1}) P D_\theta(V^{-1}) e - 1 \tag{3.120}$$

where $V \in \mathbb{R}^+$ is a Lyapunov function, $e \in \mathbb{R}^{n+1}$ represents the estimation error, $P \in \mathbb{R}^{(n+1)\times(n+1)}$ is a SPD matrix and $D_\theta(.)$ is the diagonal matrix defined as:

$$D_\theta(\epsilon) = diag(\epsilon^{\theta_1}, \epsilon^{\theta_2}, \cdots, \epsilon^{\theta_{n+1}}) \tag{3.121}$$

with $\theta = [\theta_1,\ \theta_2,\ ,\cdots,\ \theta_{n+1}]^T \in \mathbb{R}^{n+1}$ and θ_i $(\forall i = 1,\ldots,n+1)$ are a positive constants. In this study we consider θ in (3.120) as $\theta = \left[1,\ \frac{n}{n+1},\ \cdots,\ \frac{1}{n+1}\right]^T$. It is worth noting that the definition of matrix $D_\theta(.)$ will be used through the proof with different parameter, then general definition (3.121) is necessary.

Based on (3.119) and the observer (3.111-3.114, the dynamic of the estimation error is given as:

$$\dot{e} = \left(\bar{A} - D_{\bar{\theta}}\left(\left|\bar{C}e\right|^{-1}\right) KC\right) e + \bar{B}\delta \tag{3.122}$$

where $\bar{\theta} = \left[\frac{1}{n+1},\ \frac{2}{n+1},\ \cdots,\ \frac{n}{n+1}, 1\right]^T$, $K = [k_1, \cdots,\ k_n,\ k_{n+1}]^T$ where $k_i > 0$,
$\bar{A} : \left\{\left(\bar{A}\right)_{i,j} = 1 \text{ if } j = i+1; \text{else } \left(\bar{A}\right)_{i,j} = 0\right\}; i,j : 1 \to n+1$, $\bar{B} = \begin{bmatrix}0 & 0 & \cdots & 1\end{bmatrix}^T \in \mathbb{R}^{(n+1)\times 1}$, $\bar{C} = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix} \in \mathbb{R}^{1\times(n+1)}$, $\delta(t) = \frac{d}{dt}F(z,u,t,\tau)$.

Taking the partial derivative of $Q(V,e)$ with respect to e and based on (3.122), one gets:

$$\frac{\partial Q(V,e)}{\partial e}\dot{e} = 2e^T D_\theta(V^{-1}) P D_\theta(V^{-1}) \left[\left(\bar{A} - D_{\bar{\theta}}\left(\left|\bar{C}e\right|^{-1}\right) KC\right) e + \bar{B}\delta\right] \tag{3.123}$$

Let $\varepsilon_1 = D_\theta(V^{-1})e$, $\varepsilon_2 = -\Theta(\lambda)K\frac{\bar{C}e}{|\bar{C}e|}$, where $\lambda = \frac{|\bar{C}e|}{V} \in [0,1]$, $\Theta(\lambda) = \lambda\left[D_{\bar{\theta}}\left(\lambda^{-1}\right) - I_{n+1}\right]$. and $\varepsilon_3 = V^{\frac{-1}{n}}\delta$. Taking into account the fact that:
$D_{\bar{\theta}}\left(\left|\bar{C}e\right|^{-1}\right) = D_{\bar{\theta}}\left(\lambda^{-1}\right) D_{\bar{\theta}}\left(V^{-1}\right)$, $D_\theta(V^{-1})\bar{A}D_\theta^{-1}(V^{-1}) = V^{\frac{-1}{n}}\bar{A}$
and $D_{\bar{\theta}}(V^{-1})K\bar{C} = V^{\frac{-1}{n}}D_\theta^{-1}(V^{-1})K\bar{C}D_\theta(V^{-1})$, then the equation (3.123) can be rewritten as:

$$\frac{\partial Q(V,e)}{\partial e}\dot{e} = V^{\frac{-1}{n}}\begin{bmatrix}\varepsilon_1\\ \varepsilon_2\end{bmatrix}^T \Phi_3 \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\end{bmatrix} + 2\varepsilon_1^T P B\varepsilon_3 - \frac{V^{\frac{1}{n}}}{L}\varepsilon_3^T\varepsilon_3 + V^{\frac{-1}{n}}\left(\varepsilon_2^T Z\varepsilon_2 - \varepsilon_1^T P\varepsilon_1\right) - V^{\frac{-1}{n}}\alpha\varepsilon_1^T P\varepsilon_1 + \frac{V^{\frac{1}{n}}}{L}\varepsilon_3^T\varepsilon_3 \tag{3.124}$$

where

$$\Phi_3 = \begin{bmatrix}P\bar{A} + \bar{A}^T P - \bar{C}^T Y^T - YC + (1+\alpha)P & P\\ P & -Z\end{bmatrix} \tag{3.125}$$

where Z is a SPD matrix, a matrix $Y = PK$ (K is the observer gain). If the matrix Y is design such as

$$\begin{bmatrix}1 & Y^T\\ Y & P\end{bmatrix} \geq 0;\ \Theta(\lambda)Z\Theta(\lambda) \leq P,\ \forall\ \lambda \in [0,1] \tag{3.126}$$

One gets $\varepsilon_2^T Z\varepsilon_2 \leq 1$, due to $Q(V,e) = 0$ one gets $\varepsilon_1^T P\varepsilon_1 = 1$, hence:

$$V^{\frac{-1}{n}}\left(\varepsilon_2^T Z\varepsilon_2 - \varepsilon_1^T P\varepsilon_1\right) \leq 0 \tag{3.127}$$

Based on Assumption 3.7.2, we have $L_y^2 V^{\frac{-2}{n}} \geq \sup_{t\geq 0}\left(\varepsilon_3^T\varepsilon_3\right)$. Then, for a given $\alpha > L_y$, one gets:

$$-V^{\frac{-1}{n}}\alpha\varepsilon_1^T P\varepsilon_1 + \frac{V^{\frac{1}{n}}}{L}\varepsilon_3^T\varepsilon_3 \leq 0 \tag{3.128}$$

Hence:

$$\frac{\partial Q(V,e)}{\partial e}\dot{e} \leq \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\end{bmatrix}^T \underbrace{\begin{bmatrix} V^{\frac{-1}{n}}\Phi_3 & \begin{bmatrix}PB\\ 0\end{bmatrix}\\ \begin{bmatrix}B^T P & 0\end{bmatrix} & -\frac{V^{\frac{1}{n}}}{L_y}\end{bmatrix}}_{\Phi_4} \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\end{bmatrix} \tag{3.129}$$

If the matrix Y is designed as follow

$$\begin{bmatrix} P\bar{A}+\bar{A}^TP-\bar{C}^TY^T-YC+(1+\alpha)P & P & PB \\ P & -Z & 0 \\ B^TP & 0 & -L^{-1} \end{bmatrix} \leq 0 \tag{3.130}$$

then from the Schur complement, the matrix Φ_4 is negative semidefinite.

Now, let us give the expression of the partial derivative of $Q(V,e)$ with respect to V:

$$\frac{\partial Q(V,e)}{\partial V} = -V^{-1}e^T D_\theta(V^{-1})\left(H_\theta P + PH_\theta\right) D_\theta(V^{-1})e < 0 \tag{3.131}$$

where $H_\theta = diag(\theta_1, \theta_2, \cdots, \theta_{n+1})$ is a positive matrix.

Therefore, from (3.131) and (3.129) the derivative of Lyapunov function V can be derived as:

$$\dot{V} = -\left(\frac{\partial Q(V,e)}{\partial V}\right)^{-1}\frac{\partial Q(V,e)}{\partial e}\dot{e} \leq 0 \tag{3.132}$$

This completes the proof □

It is worth noting that the conditions (3.126) is nonlinear then for the sake of observer design, one shall simplify this condition into the linear form. Due to $\lambda \in [0,1]$, hence, $\Theta(\lambda) \leq I_{n+1}$, then $\Theta(\lambda)Z\Theta(\lambda) \leq Z$, then the conditions (3.126) can be rewrite as:

$$\begin{bmatrix} 1 & Y^T \\ Y & P \end{bmatrix} \geq 0;\ Z \leq P. \tag{3.133}$$

Remark 3.7.3. *According the above proof, the finite-time convergence behaviour of the observer (3.111-3.114) can be achieved by computing the appropriate gain K. This gain can be determined by Solving the LMI (3.130) under the conditions (3.133) to obtain P and Y, then one gets $K = P^{-1}Y$.*

Proof of Theorem 3.7.9: According to Remark 3.7.2, in a finite time T_f we have $\hat{z}(t) = z(t)$ and $\hat{F}(z,u,t,\tau) = F(z,u,t,\tau)$. Therefore, when $t > T_f$, one gets:

$$\begin{aligned} \Delta_{(t,\hat{\tau})}F &= \hat{F}(z,u,t,\tau) - F(\hat{z},u,t,\hat{\tau}) \\ &= F(z,u,t,\tau) - F(z,u,t,\hat{\tau}) \end{aligned} \tag{3.134}$$

Let us now consider the first derivative of $\left|\Delta_{(s,\hat{\tau})}F\right|$ which is given as follow:

$$\frac{d\left|\Delta_{(t,\hat{\tau})}F\right|}{dt} = sign(\Delta_{(t,\hat{\tau})}F)\frac{d\Delta_{(t,\hat{\tau})}F}{dt} \tag{3.135}$$

From (3.102) and (3.134) one gets:

$$\begin{aligned} \frac{d\Delta_{(t,\hat{\tau})}F}{dt} &= \dot{\hat{F}}(z,u,t,\tau) - \dot{f}_F(\hat{z},t) - \dot{g}_F(\hat{z},t)u(t-\hat{\tau}) \\ &\quad - g_F(\hat{z},t)\dot{u}(t-\hat{\tau})(1-\dot{\hat{\tau}}) \end{aligned} \tag{3.136}$$

From (3.117) and (3.136) together with the condition $|g_F(\hat{z},t)\dot{u}(t-\hat{\tau})| > 0$ one gets:

$$\frac{d\Delta_{(t,\hat{\tau})}F}{dt} = -\Gamma\Delta_{(t,\hat{\tau})}F \tag{3.137}$$

Consider now the following Lyapunov function:

$$V_\tau = 0.5\left|\Delta_{(t,\hat{\tau})}F\right|^2 \tag{3.138}$$

From (3.117) and (3.135) one gets:

$$\dot{V}_\tau = \left|\Delta_{(t,\hat{\tau})}F\right| sign(\Delta_{(t,\hat{\tau})}F)\frac{d\Delta_{(t,\hat{\tau})}F}{dt} \tag{3.139}$$

By replacing (3.137) into (4.63) one obtains the following results:

$$\dot{V}_\tau = -\Gamma\left|\Delta_{(t,\hat{\tau})}F\right| sign(\Delta_{(t,\hat{\tau})}F)\Delta_{(t,\hat{\tau})}F \tag{3.140}$$

Then, for $\Gamma > 0$ one gets $\dot{V}_\tau \leq 0$, which implies that $|F(z,u,t,\tau) - F(z,u,t,\hat{\tau})|$ converges to zero in a finite time, consequently, $\int_{t-T}^{t}|F(z,u,s,\tau) - F(z,u,s,\bar{\tau})|\,ds$ converges to zero.

According to Definition 3.7.6 $\hat{\tau}(t)$ exponentially converges to τ. This completes the proof of Theorem 3.7.9 □

Remark 3.7.4. *Theorem 1 is based on the condition*

$$|g_F(\hat{z},t)\dot{u}(t-\hat{\tau})|>0;\ \forall t>\max\left(T_f;\tau_{\max}\right) \tag{3.141}$$

As the proposed method in this work is devoted to delay estimation and since the control input $u(t)$ can be given by the developer, then the value of $u(t)$ can be arbitrary designed such that $\dot{u}(t-\hat{\tau})\neq 0$ $\forall t>\max\left(T_f;\tau_{\max}\right)$. Therefore, the needed condition for delay estimator design can be rewritten as:

$$|g_F(\hat{z},t)|>0;\ \forall t>\max\left(T_f;\tau_{\max}\right) \tag{3.142}$$

Remark 3.7.5. *It is worth noting that the condition (3.142), considered in this paper, is less restrictive than those given in the literature (Drakunov et al., 2006), (Cacace et al., 2016), (Cacace et al., 2017), (Zheng, Polyakov, and Levant, 2018). Indeed, in (Drakunov et al., 2006) the authors imposed the strictly monotonic condition: $\left|\frac{dF(z,u,t,\tau)}{dt}\right|>0$, in (Cacace et al., 2016), (Cacace et al., 2017), the authors proposed: $x(t-\tau)-x(t-\hat{\tau})=0$ only if $\tau=\hat{\tau}$. These methods cannot be applied to non-monotonic function. In (Zheng, Polyakov, and Levant, 2018) the problem of non-monotonic function is considered under the very restrictive condition given as:*

$$\int_{t-T}^{t}(F(z,u,s,\tau)-F(z,u,s,\bar{\tau}))\frac{\partial F(z,u,s,\bar{\tau})}{\partial\bar{\tau}(t)}ds\neq 0 \tag{3.143}$$

It is clearly that this condition will not be satisfied if (3.110) is achieved. □

Remark 3.7.6. *Notice that the delay estimator (3.117) proposed in this work is based on the derivative of known functions w.r.t time, which is an advantage comparing to the other methods in the literature that exploit the partial derivative w.r.t the estimation of delay, which is not always easy to obtain. Indeed, in (Cacace et al., 2016), (Cacace et al., 2017) the author used $\frac{\partial\hat{x}_{\hat{\tau}}(t)}{\partial\hat{\tau}}$ and in (Zheng, Polyakov, and Levant, 2018) the author used $\frac{\partial F(z,u,t,\hat{\tau})}{\partial\hat{\tau}(t)}$.*

3.8 Conclusion

This chapter presents a novel structure of FHGO for a class of nonlinear systems subject to sampled-delayed measurements, bounded disturbances, and noises. The proposed method provides a new tool to design the observer based on LMI, which can adapt to a wide range of values for high gain parameters, time delay, and sampling period conditions. The relevance of our proposed method in compensating for the effects resulting from sampled data measurements, output time delay, and filtering output noise has been demonstrated through simulation and experimental results performed on a Quadrotor UAV. To assess the performance of our proposed observer, we conducted a comparison with both high-gain and Kalman-like observers.

Additionally, we have developed a novel method to estimate the unknown delay affecting output measurements. Our method is based on a mathematical model that considers the characteristics of the input signal and the measurement system. By applying our proposed method, we have proved its stability and proposed a method to calculate the observer gain based on LMI. Our method has several advantages over existing approaches, including its simplicity, computational efficiency, and ability to handle nonlinear systems. In addition, our method does not require non-monotonic assumptions.

The proposed method in this chapter has the potential to make a significant contribution to various fields that rely on accurate measurements, such as control systems, signal processing, and communication systems. Future work could involve further validation of the method with additional experimental data, as well as exploring ways to extend our approach to more complex systems such as multi-agent systems, where delay communication is an interesting challenge.

Building on the observer developed in this chapter, the next step in this thesis will be to design and implement a controller for the quadrotor UAV system. By using the insights gained from the observer, the controller will be able to effectively regulate the quadrotor's motion and improve its stability and performance. Together, the observer and controller will form a robust control

CHAPTER 4

Flight control design for a quadrotor UAVs

4.1 Introduction

Quadrotors UAV, with four propellers symmetrically aligned and facing upwards, has become increasingly popular for various applications in the field of UAVs. This is largely due to the quadrotor's mathematical simplicity, flight capabilities, and relatively low-cost implementation. Over the past two decades, the quadrotor UAV has attracted a lot of interest in the automation community, with researchers focusing on various aspects such as controller design, observation device design, fault detection, and more. Among these areas of research, designing a controller for a quadrotor UAV has been the most popular topic. The controller plays a crucial role in ensuring the stability and maneuverability of the quadrotor, which is essential for its effective operation. Numerous control methods have been proposed for quadrotor UAVs, ranging from the simple yet effective method of a PID controller to more complex structures such as adaptive and intelligent controllers.

The linear control flight technique represents the basic structure for UAV, several control designs have been proposed such as PID controllers, which have been studied extensively in (Pounds, 2012) and (Moreno-Valenzuela et al., 2018), as well as LQR/LQG controllers, which have been analyzed in (Cowling et al., 2007) and (Minh and Ha, 2010). State feedback control, another popular control design, has been explored in (Ajmera and Sankaranarayanan, 2016) and other studies.

The PID controller is widely used in both flight control and various industrial applications due to its straightforward controller design and easy-to-tune parameters. However, it is important to note that the PID controller has a significant limitation concerning control parameters. These parameters are typically tuned to work in specific conditions, which may not always reflect the practical working conditions that can vary and be challenging to determine accurately due to the uncertainty surrounding the system's model.

The other popular control technique which belong to linear controller, namely Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG). These techniques are classified as optimal control technique, in the sense that they are based on minimizing a quadratic cost function to optimise the control parameters. Compared to PID control technique, LQR/LQG provide a more optimal control performance. However, these techniques require to linearized the system at the working point. It is a matter of fact that these methods are more suitable for some simple task such as stabilise the UAV, while for more complicated task where the working point is variate, the pre-determined control parameter can not give the optimal control performances.

On the other hand, state feedback control is also another popular control technique for linear control design. The method is based on approximating the nonlinear model of the UAV to a linear one. Compared to the PID controller, the state feedback technique suggests that we can arbitrarily place the poles of a system to design the control gain. Regardless of the controller's design ability compared to PID, the state feedback technique cannot overcome the existing problems of using a linear controller as aforementioned.

The main advantage of using linear control is its simplicity in control structure. However, the system linearization might degrade the control performance due to the highly coupled nonlinear model of the UAV. Moreover, another main problem when designing a flight control for a UAV is the presence of external disturbances, which may not be well rejected by using a linear controller. To overcome such

drawbacks, alternative control methods based on nonlinear control techniques were investigated in the literature.

To solve the problems and overcome the limitations of using linear control approaches, many nonlinear controllers have been studied from simple structure which is based on mathematical model of the system to more complicated method using intelligent controller based on some self learning methods. To summarise the state of the art of nonlinear control, a brief overview of the different non-linear controller will be presented in this chapter,

The model-based control approach represents the foundational and most basic methodology for designing nonlinear controllers. This approach involves using a mathematical model of the system being controlled to design a controller that will steer the system towards a desired performance. Several widely used techniques have been proposed to enhance the performance of model-based controllers, including SMC and backstepping control.

SMC is a control technique that enables robust control of nonlinear systems in the presence of uncertainties and disturbances by designing a sliding surface to govern the system's behavior. The basic concept of SMC design can be found in (Utkin, 1992). SMC control approach is one of the efficient tools to design robust controllers for complex and high-order nonlinear systems under uncertainties. Research in this area was initiated in the former Soviet Union about 60 years ago, and since then, the SMC technique has received much attention from the international control community in various domains, including UAV applications (Cheng and Liu, 2019; Falcón, Ríos, and Dzul, 2019; L'Afflitto, Anderson, and Mohammadi, 2018; Mofid and Mobayen, 2018; Safeer Ullah and Iqbal, 2020). The backstepping technique was first proposed by (Kokotovic, 1992). This method involves recursively designing controllers to stabilize the system's states towards a desired equilibrium. These techniques have proven to be effective in enhancing the performance of model-based controllers and have been widely implemented in various control systems.

These mentioned methods are simple and easy to implement. However, their accuracy depends on the model parameters, and the control parameters are tuned to work in specific conditions. In practical applications, the model parameters such as mass and inertia are usually unknown, and the working conditions can vary. Therefore, more intelligent techniques that can guarantee acceptable control performance without knowing the mathematical model are presented. Model Predictive Control (MPC) is proposed and its application to a quadrotor UAV is shown in (Younes Al Younes and Hajjaji, 2016; Eskandarpour and Sharf, 2020; Jalili, Rezaie, and Rahmani, 2018). A Fuzzy Logic Controller (FLC) is used for a trajectory tracking problem of a quadrotor in (Nasr et al., 2018; Harik et al., 2017). Fuzzy logic rules are used for online gain tuning of a PID controller in (Andrade et al., 2022; Dong J, 2018; Lei, Li, and Chen, 2019). The adaptive fuzzy gain scheduling SMC technique is introduced in (Eltayeb et al., 2019). These mentioned methods significantly improve the adaptive characteristics and the control performance in the absence of a mathematical model.

However, it should be mentioned that the main limitation of the quadrotor UAV is their lightweight which makes them sensitive to disturbances such as wind, rotor vibrations, with or without external payload. Therefore, a robust and adaptive controller is required for such system. In an effort to address the limitations of traditional control approaches in handling disturbances and uncertainties, additional adaptive techniques have been investigated. These techniques aim to adapt the controller's parameters to accommodate changing conditions and improve performance. For instance, an intelligent fuzzy controller has been proposed in (Kayacan and Maslim, 2017) to enhance the robustness of the controller against uncertainties. Moreover, an adaptive observer-based fault estimation approach has been deployed in (Kayacan and Maslim, 2017) to improve the fault tolerance capability of the control system.

Other studies have explored the use of neural network techniques to estimate and compensate for disturbances and uncertainties. For example, an adaptive sliding mode controller has been introduced in (Alqaisi et al., 2019), where a Radial Basis Function Neural Network (RBFNN) is used to estimate disturbances and uncertainties in conjunction with a classical sliding mode controller. Similarly, in (Doukhi and Lee, 2019), the authors combine an RBFNN and a PID controller to improve the robustness of the control system. In (Qingzheng Xu and Zhen, 2019), the combination of RBFNN and backstepping controller is studied. Among these methods, the use of neural networks to approximate and compensate for disturbances and uncertainties has gained significant attention. The main advantage of using neural networks is their self-learning capacity, which allows them to adapt to uncertain and unknown parts of the system model without prior knowledge. As a result, the adaptive behavior of the system is significantly improved, particularly in situations where the system model is uncertain or unknown.

Despite the mentioned advantages, using neural networks also exhibits some drawbacks, such as: 1) high computational cost when there are a high number of neural nodes, and 2) the estimation provided by the neural network takes time to converge, which can affect the control performance. The first problem can be easily solved by using a small number of nodes. However, the second problem is more complicated

when the control parameters are pre-designed. For example, in (Alqaisi et al., 2019), the proposed controllers demonstrate their ability to compensate for uncertainties and bounded disturbances in the simulation environment. However, in the experimental tests, there was inaccurate tracking of the desired trajectory due to the requirement for faster convergence of disturbance estimation in the experiment.

After numerous years of research and development, quadrotor UAV have become highly sought-after and there are a plethora of options available in the market. These options range from quadrotors designed for entertainment purposes to quadrotors intended for research with highly stable controllers that have been created and incorporated. As shown in the literature, the control structures of quadrotor UAV consists of two control loop, namely the inner controller and the outer controller, both of them are in charged of achieving trajectory tracking tasks. In the field of quadrotor UAVs, both for entertainment and research purposes, the inner loop controller has been extensively developed and is generally considered to be functioning optimally. Therefore, in the design of the overall control system, the inner control loop is often treated as a closed loop, and attention is directed towards the outer loop controller design. This approach allows for a more targeted and efficient design process, as it eliminates the need to modify a well-functioning inner loop controller, and instead allows for a focus on improving the overall system performance by optimizing the outer loop controller.

Motivated by this problem and conditions, we aim to develop an adaptive controller for trajectory tracking of a quadrotor, under the assumption that the model parameter such as mass, moment of inertia are unknown, the UAV is working under disturbance and aerospace-dynamic effects. The inner loop controller in this case is supposed to work perfectly. This work focus on designing a controller for the outer loop, which delivers desired attitude set points for the inner loop controller. To achieve this objective, the sliding mode technique is firstly applied to build a robust non-linear controller, the RBFNN then is deployed to estimate the value of model uncertainties and unknown disturbances. Using the tracking error knowledge, a Fuzzy logic control law is used to tune the control parameters. This method present a remarkable adaptive behaviour.

It is worth noting that while using neural networks, specifically RBFNNs, to approximate and compensate for disturbances and uncertainties has gained significant attention, there are still limitations to their self-learning capabilities. This is because the parameters of the neural network such as node center value and width value of Gaussian function are typically chosen by the developer, which can restrict the network's ability to learn and adapt in real-time. To overcome such drawback, a second approach has been proposed to address this limitation, which involves utilizing backpropagation to estimate the value of model uncertainties and unknown disturbances based on the tracking error knowledge. By incorporating this method, the controller can more accurately adapt to changes in the system, leading to improved overall performance.

By combining the advantage of adaptive neural network and the robustness of sliding mode technique. The proposed structure remarkably enhances adaptive behaviour of the controller, and the uncertainties are approximated more effectively in comparison to the previous work in the literature.

The chapter is organized as follows: The control problem statement is discussed in section 4.2. The main part of this chapter, namely controller design will be presented in section 4.3. The simulation and experimental validation will be given in section 4.4 and 4.5 respectively. Some conclusions will be made in section 4.7 to summarise the different results presented in this chapter.

4.2 Problem statement

Industrial autonomous UAVs typically consist of two main control blocks: guidance and control, as shown in Figure 4.1. The control station sends the desired trajectory to the drone, which first passes through the guidance block. The guidance block performs three procedures: desired altitude (which sets the desired altitude for the UAV based on the desired trajectory), heading (which steers the UAV to the desired orientation), and move to desired position (which sets the desired x and y coordinates for the UAV based on the desired trajectory). After these procedures, the guidance block provides the desired position ($x_d,\ y_d,\ z_d$) and the desired steering angle ψ_d to the control block. The control block then tracks the desired signals to control the UAV. This part plays a crucial role in the control of a UAV system, and hence, this work focuses on studying the control block.

The control block consists of two main different loops, namely the outer loop controller (position controller and/or speed controller that provides the desired attitude for the inner loop), and the inner loop controller (attitude controller).

For the sake of control design, the dynamic model of quadrotor in term of translation and rotation will be presented in the next section.

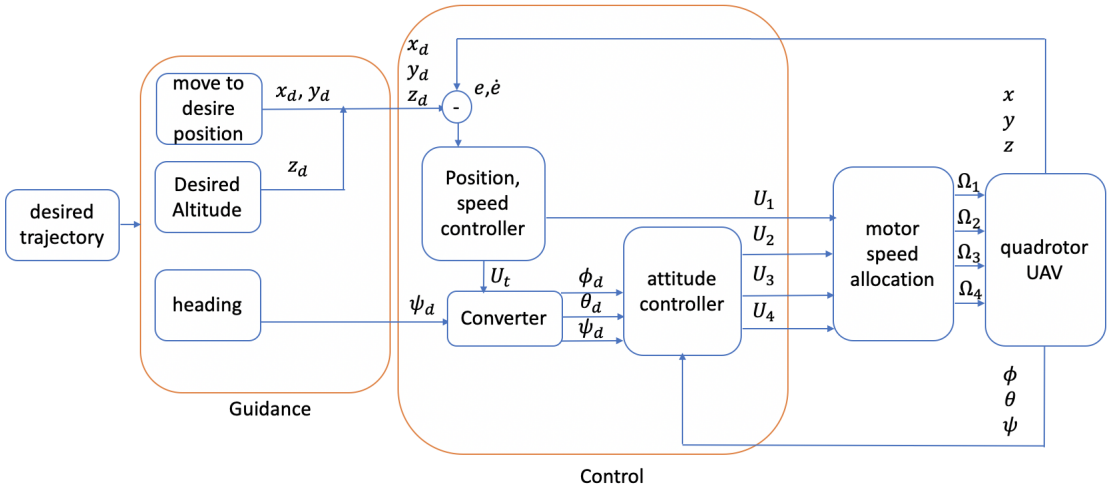


Figure 4.1: Overall structure for the control design of a quadrotor UAV

4.2.1 Position controller architecture

The outer loop controller must be designed so that the position of the quadrotor UAV can track the desirable trajectory and then calculate the desired attitude for the inner controller. Following (2.41) and (2.43)), the dynamic model of the quadrotor UAV for position control can be obtained as follows:

$$\begin{cases} \dot{X}_{t1} &= X_{t2} \\ \dot{X}_{t2} &= F_t(X_t) + G_t U_t + \delta_t \end{cases} \tag{4.1}$$

where $X_t = \begin{bmatrix} X_{t1}^T & X_{t2}^T \end{bmatrix}^T$ is the state vector of quadrotor in term of translation defined as follows:

$$X_{t1} = \begin{bmatrix} x & y & z \end{bmatrix}^T ; X_{t2} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{4.2}$$

$$F_t(X_t) = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} ; G_t = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \tag{4.3}$$

$$U_t = \begin{bmatrix} U_{tx} \\ U_{ty} \\ U_{tz} \end{bmatrix} = \begin{bmatrix} cos(\phi)sin(\theta)cos(\psi) + sin(\phi)sin(\psi) \\ cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi) \\ cos(\phi)cos(\theta) \end{bmatrix} F_T \tag{4.4}$$

$$\delta_t = \frac{1}{m} \begin{bmatrix} \delta_x & \delta_y & \delta_z \end{bmatrix}^T \tag{4.5}$$

In (4.4) three new control parameters U_{tx}, U_{ty} and U_{tz} are called virtual control signals. These are the projection of U_1 related to the three axes X, Y and Z, hence one gets the following property:

$$F_T = \sqrt{U_{tx}^2 + U_{ty}^2 + U_{tz}^2} \tag{4.6}$$

The virtual control signal together with desired heading (desired yaw angle) is used to generate the desired roll and pitch angles as follows:

$$\theta_d = arctan\left(\frac{U_{ty}sin(\psi_d) - U_{tx}cos(\psi_d)}{U_{tz}}\right) \tag{4.7}$$

$$\phi_d = arctan\left(\frac{U_{ty}cos(\psi_d) + U_{tx}sin(\psi_d)}{U_{tz}}cos(\theta_d)\right) \tag{4.8}$$

In the following part, one shall address the design of an adaptive controller for the system quadrotor UAV. To reduce the cumbersome, the controller will be design for the following class of nonlinear system (4.9), and applied to the case of quadrotor UAV system.

$$\begin{cases} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= F(X) + GU + \delta \end{cases} \tag{4.9}$$

where $X = \begin{bmatrix} X_1^T & X_2^T \end{bmatrix}^T$ is the state vector, $F(x)$ is nonlinear function, U is the control input, δ is the external disturbance. G is an invertible and positive-constant matrix.

Remark 4.2.1. *It is worth noting that the nonlinear function $F(X)$ can be easily determined in the simulation. Then the nonlinear control technique such as backstepping controller (Araar and Aouf, 2014), (Yu et al., 2019), sliding mode control in (Cheng and Liu, 2019) (Falcón, Ríos, and Dzul, 2019), (L'Afflitto, Anderson, and Mohammadi, 2018), (Mofid and Mobayen, 2018), (Safeer Ullah and Iqbal, 2020) can be used. However, in the experiment, the translation drag coefficient and the aerodynamic friction coefficients might not be determined and can not be measured. Then the function $F(X)$ is normally unknown, furthermore, the external disturbances δ is never identified accurately by any measurement method.*

The proposed controller will be designed under the following assumptions:

Assumption 4.2.1. *The disturbances δ_t and δ_r are assumed unknown but bounded with known bounds:*

$$\exists\ \epsilon_t,\ \epsilon_r \in \mathbb{R}^+,\ \sup_{t\geq 0} \|\delta_t\| \leq \epsilon_t,\quad \sup_{t\geq 0} \|\delta_r\| \leq \epsilon_r \tag{4.10}$$

Assumption 4.2.2. *The nonlinear functions $F_t(X_t)$ and $F_r(X_R)$ are assumed unknown but bounded with known bounds:*

$$\exists\ \epsilon_{ft},\ \epsilon_{fr} \in \mathbb{R}^+,\ \sup_{t\geq 0} \|F_t(X_t)\| \leq \epsilon_{ft},\quad \sup_{t\geq 0} \|F_r(X_r)\| \leq \epsilon_{fr} \tag{4.11}$$

The main objective of the controller design is to build an adaptive controller for system (4.9), the position controller then will be applied. The desired attitude for the inner loop controller will be derived based on (4.7-4.8). Based on sliding mode technique, where the unknown nonlinear function $F(X)$ and the external disturbance δ can be estimated by using an adaptive neural network. In order to improve the adaptive and robustness behaviour of the proposed controller, a fuzzy logic rule will be used for the propose of online tuning of control parameter.

4.3 Controller design

4.3.1 Sliding mode controller framework

In this part, one shall design a controller so that the location of the quadrotor UAV can track the desired trajectory. For this sake, let us define the desired trajectory as X_{1d} and its derivative as X_{2d}. One gets the following tracking errors:

$$e = X_{1d} - X_1;\ \dot{e} = X_{2d} - X_2;\ E = \begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}^T. \tag{4.12}$$

The objective can be realized by designing the controller such that the tracking error converges to zero i.e $\lim_{t\to\infty} e = 0,\ \lim_{t\to\infty} \dot{e} = 0$

For this sake, based on the sliding mode technique, let us define the sliding surface as following:

$$S = \dot{e} + ce \tag{4.13}$$

where c is a sliding surface coefficient, which is positive definite. From (4.9), (4.12) and (4.13) one gets:

$$G^{-1}\dot{S} = G^{-1}\left(\dot{X}_{2d} - F(X) - GU - \delta + c\dot{e}\right) = -U + G^{-1}\left(\dot{X}_{2d} - F(X) - \delta + c\dot{e}\right) \tag{4.14}$$

Now define $\Gamma = G^{-1}\left(\dot{X}_{2d} - F(X) - \delta + c\dot{e}\right)$, then (4.14) can be rewritten as follows:

$$G^{-1}\dot{S} = -U + \Gamma \tag{4.15}$$

Hence the sliding mode controller is given as:

$$U = KS + K_1 sign(S) + \Gamma \tag{4.16}$$

where K and K_1 are the controller gains. That shall be designed in theorem 4.3.4.

4.3.2 Back-propagation neural network

It is worth to mention that the function Γ is unknown due to lacking of knowledge about $F(X)$ and δ. Hence any controller requires the knowledge of Γ such as (4.16) can not be implemented. Motivated by this problem, in the following part, an artificial neural network using back-propagation technique will be exploited to estimate the unknown function Γ. Indeed, many studies have shown that for X restricted to a compact set, and the neural network have a sufficiently number of hidden layer, then one gets:

$$\Gamma = W\mu\left(\sigma E\right) + \zeta \tag{4.17}$$

where W is the ideal weight matrix of output layer, σ is the ideal weight matrix of hidden layer, ζ represents the neural network approximation error. $\mu\left(\sigma E\right) = \begin{bmatrix}\mu_1\left(\sigma E\right) & \cdots & \mu_n\left(\sigma E\right)\end{bmatrix}^T$ is the transfer function of the hidden neurons, which is given by the following expression:

$$\mu_i\left(\sigma E\right) = \frac{2}{1 + exp(-2\sigma_i E)} - 1;\ i = 1 \cdots n. \tag{4.18}$$

where n is the number of node of hidden layer, σ_i is the i^{th} row of σ.

Remark 4.3.1. *From (4.18), one gets the following property*

$$\left\|\mu_i\left(\sigma E\right)\right\| \leq 1 \tag{4.19}$$

It is worth to notice that the ideal weight matrices W and σ are not available in this context. Hence, the estimation of Γ will be provided based on the estimation of the idea weight matrices, namely $\hat{W}$ and $\hat{\sigma}$ such that:

$$\hat{\Gamma} = \hat{W}\mu\left(\hat{\sigma}E\right) \tag{4.20}$$

According to (4.16) and (4.20), the adaptive neural network-bsed sliding mode control structure is given as follows:

$$U = KS + K_1 sign(S) + \hat{W}\mu\left(\hat{\sigma}E\right) \tag{4.21}$$

The objective is to train the neural network together with the sliding mode controller to give us the online optimal estimate of Γ by guarantee $\hat{W}$ and $\hat{\sigma}$ converge to W and σ respectively. To achieve this objective, in the next section, one shall give the detail about designing the adaptation law of $\hat{W}$ and $\hat{\sigma}$ respectively.

• **Neural network design based on back-propagation technique**

Once the structure of neural network is defined, a learning rule will be proposed in this section to train the network. Using the back-propagation technique, let us consider the following adaptation law for $\hat{W}$ and $\hat{\sigma}$ as follows:

$$\dot{\hat{W}} = -\eta_w \frac{\partial \mathbf{C}}{\partial \hat{W}} - \rho_w \left\|S\right\| \hat{W}; \qquad \dot{\hat{\sigma}} = -\eta_\sigma \frac{\partial \mathbf{C}}{\partial \hat{\sigma}} - \rho_\sigma \left\|S\right\| \hat{\sigma} \tag{4.22}$$

where $\mathbf{C} = 0.5 S^T S$ is the objective cost function that should be minimized, η_w, ρ_w, η_σ, ρ_σ are the training gains and will be design in the Theorem 4.3.4.

The estimation of weight matrices will be deduced from (4.22) by solving the partial derivative of cost function. For this sake, let us define the following new variables:

$$\Omega_{\hat{W}} = \hat{W}\mu\left(\hat{\sigma}E\right); \qquad \Omega_{\hat{\sigma}} = \hat{\sigma}E \tag{4.23}$$

Accordingly, the partial derivative $\frac{\partial \mathbf{C}}{\partial \hat{W}}$ and $\frac{\partial \mathbf{C}}{\partial \hat{\sigma}}$ can be decoupled as following:

$$\frac{\partial \mathbf{C}}{\partial \hat{W}} = \frac{\partial \mathbf{C}}{\partial S} \times \frac{\partial S}{\partial \Omega_{\hat{W}}} \times \frac{\partial \Omega_{\hat{W}}}{\partial \hat{W}} \tag{4.24}$$

$$\frac{\partial \mathbf{C}}{\partial \hat{\sigma}} = \frac{\partial \mathbf{C}}{\partial S} \times \frac{\partial S}{\partial \Omega_{\hat{\sigma}}} \times \frac{\partial \Omega_{\hat{\sigma}}}{\partial \hat{\sigma}} \tag{4.25}$$

Using the cost function with (4.23) one gets:

$$\frac{\partial \mathbf{C}}{\partial S} = S^T;\ \frac{\partial \Omega_{\hat{W}}}{\partial \hat{W}} = \mu\left(\hat{\sigma}E\right);\ \frac{\partial \Omega_{\hat{\sigma}}}{\partial \hat{\sigma}} = E \tag{4.26}$$

According to (4.15), (4.20) and (4.21), one gets:

$$\frac{\partial G^{-1}\dot{S}}{\partial \Omega_{\hat{W}}} = -K\frac{\partial S}{\partial \Omega_{\hat{W}}} - I \tag{4.27}$$

$$\frac{\partial G^{-1}\dot{S}}{\partial \Omega_{\hat{\sigma}}} = -K\frac{\partial S}{\partial \Omega_{\hat{W}}} - \hat{W}\frac{\partial \mu\left(\hat{\sigma}E\right)}{\partial \Omega_{\hat{W}}} \tag{4.28}$$

From (4.18) and (4.23), the partial derivative $\frac{\partial \mu(\hat{\sigma}E)}{\partial \Omega_{\hat{W}}}$ can be deduced as:

$$\frac{\partial \mu\left(\hat{\sigma}E\right)}{\partial \Omega_{\hat{W}}} = \begin{bmatrix} 1-\mu_1^2\left(\hat{\sigma}E\right) & & 0 \\ & \ddots & \\ 0 & & 1-\mu_n^2\left(\hat{\sigma}E\right) \end{bmatrix} = D\left(\hat{\sigma}E\right) \tag{4.29}$$

Equations (4.27) and (4.28) present a set of so-called back-propagation nonlinear dynamical systems, which will be solved to obtain the solutions of $\frac{\partial S}{\partial \Omega_{\hat{W}}}$ and $\frac{\partial S}{\partial \Omega_{\hat{\sigma}}}$. According to (Rahimilarki et al., 2019) and due to the fact that the network converges relatively fast if the training gains are designed in a proper manner, one can assume the static approximation of the gradient i.e $\frac{\partial G^{-1}\dot{S}}{\partial \Omega_{\hat{W}}} = 0$ and $\frac{\partial G^{-1}\dot{S}}{\partial \Omega_{\hat{\sigma}}} = 0$. Hence, one gets:

$$\frac{\partial S}{\partial \Omega_{\hat{W}}} = \frac{-1}{K}I;\ \frac{\partial S}{\partial \Omega_{\hat{\sigma}}} = \frac{-1}{K}\hat{W}D\left(\hat{\sigma}E\right) \tag{4.30}$$

Consequently, by using above equation and (4.22), (4.24), (4.25), (4.26) the update rules of weight matrices can be obtained as:

$$\dot{\hat{W}} = \frac{\eta_w}{K}S\mu^T\left(\hat{\sigma}E\right) - \rho_w \left\|S\right\|\hat{W} \tag{4.31}$$

$$\dot{\hat{\sigma}} = \frac{\eta_\sigma}{K}\left(S^T\hat{W}D\left(\hat{\sigma}E\right)\right)^T E^T - \rho_\sigma \left\|S\right\|\hat{\sigma} \tag{4.32}$$

- **Adaptive neural network based sliding mode controller design (SMC-BPNN)**

As the structure of the controller and the training rules are known, it is possible to design the controller gains and the training gains in a way that guarantees global stability of the system. This is a critical requirement for any controller, as an unstable system can result in catastrophic consequences. In this section, we will propose a design for the necessary gains based on the Lyapunov's direct method. This method is a widely used technique for designing control laws that guarantee stability, and it relies on constructing a Lyapunov function that decreases along the trajectories of the system. By selecting appropriate gains, we can ensure that the Lyapunov function decreases asymptotically and ultimately converges to zero, which corresponds to the desired operating point of the system.

The proposed controller will be designed under the following assumptions:

Assumption 4.3.1. *The quadrotor system is a bounded input, bounded output system, i.e there exists the compact sets* $\mathbb{X} \subset \mathbb{R}^6$, $\mathbb{U} \subset \mathbb{R}^3$ *such that* $X \in \mathbb{X}$, $U \in \mathbb{U}$. *Moreover, for a bounded input, bounded output system one can easily obtain:*

$$\exists\ E_M \in \mathbb{R}^+,\ \sup_{t\geq 0}\left\|E\right\| \leq E_M \tag{4.33}$$

Assumption 4.3.2. *The desired weight matrices* W *and* σ *are bounded i.e:*

$$\exists\ W_M;\ \sigma_M \in \mathbb{R}^+,\ sup\left\|W\right\| \leq W_M,\ sup\left\|\sigma\right\| \leq \sigma_M \tag{4.34}$$

Assumption 4.3.3. *The neural network estimation error* ζ *is assumed small, unknown but bounded with known bounds:*

$$\exists\ \epsilon_\zeta <<, \in \mathbb{R}^+,\ \sup_{t\geq 0}\left\|\zeta\right\| \leq \epsilon_\zeta \tag{4.35}$$

The following theorem summarises the main results of this section.

Theorem 4.3.4. *Consider the system (4.9) under the assumption 4.2.1, 4.2.2, 4.3.3, 4.3.1, 4.3.2. If the controller gains and the training gains respectively in (4.21), (4.31) and (4.32) are chosen such that*

$$\frac{K\sqrt{\rho_w \rho_\sigma}}{\eta_\sigma} \geq E_M \tag{4.36}$$

$$K_1 \geq max\left(\rho_w W_M^2; \frac{1}{\rho_\sigma}\left(\rho_\sigma \sigma_M + \frac{E_M W_M \eta_\sigma}{K}\right)^2\right) + \epsilon_\zeta + 2W_M \tag{4.37}$$

$$\eta_w = K \tag{4.38}$$

Then the stability of the controller (4.21) will be asymptotically and globally ensured.

Proof:

Let us define $\tilde{W} = W - \hat{W}$ and $\tilde{\sigma} = \sigma - \hat{\sigma}$ are the estimation errors of weight matrices. From (4.31) and (4.32), the first derivative of these estimation errors w.r.t times are given as:

$$\dot{\tilde{W}} = -\frac{\eta_w}{K} S\mu^T(\hat{\sigma}E) + \rho_w \|S\|\left(W - \tilde{W}\right) \tag{4.39}$$

$$\dot{\tilde{\sigma}} = -\frac{\eta_\sigma}{K}\left(S^T\left(W - \tilde{W}\right)D(\hat{\sigma}E)\right)^T E^T + \rho_\sigma \|S\|(\sigma - \tilde{\sigma}) \tag{4.40}$$

From (4.17), (4.20) and (4.21). (4.15) can be rewritten as follow:

$$G^{-1}\dot{S} = -KS - K_1 sign(S) + \tilde{W}\mu(\hat{\sigma}E) + \omega \tag{4.41}$$

Under the assumption 4.2.1, 4.2.2, 4.3.3, where $\omega = \zeta + W\left[\mu(\sigma E) - \mu(\hat{\sigma}E)\right]$. One can easily obtain:

$$\|\omega\| \leq \epsilon_\zeta + 2W_M \tag{4.42}$$

Let us consider the following Lyapunov function:

$$V = \frac{1}{2}S^T G^{-1} S + \frac{1}{2}tr(\tilde{W}^T\tilde{W}) + \frac{1}{2}tr(\tilde{\sigma}^T\tilde{\sigma}) \tag{4.43}$$

Derivating (4.43) w.r.t the time and together with (4.39), (4.40) and (4.41) yields :

$$\begin{aligned}\dot{V} = {} & S^T\left[-KS - K_1 sign(S) + \tilde{W}\mu(\hat{\sigma}E) + \omega\right] + tr\left(-\tilde{W}^T\frac{\eta_w}{K}S\mu^T(\hat{\sigma}E)\right) \\ & - tr\left(\tilde{\sigma}^T\frac{\eta_\sigma}{K}\left(S^T\left(W - \tilde{W}\right)D(\hat{\sigma}E)\right)^T E^T\right) + tr\left(\tilde{\sigma}^T\rho_\sigma\|S\|(\sigma - \tilde{\sigma})\right) \\ & + tr\left(\tilde{W}^T\rho_w\|S\|\left(W - \tilde{W}\right)\right)\end{aligned} \tag{4.44}$$

Taking into account the expression of η_w in (4.38). One gets:

$$S^T\tilde{W}\mu(\hat{\sigma}E) - tr\left(\tilde{W}^T\frac{\eta_w}{K}S\mu^T(\hat{\sigma}E)\right) = 0 \tag{4.45}$$

Taking into account the fact that $\|D(\hat{\sigma}E)\| \leq 1$, the following inequalities can be derived:

$$tr\left(\tilde{W}^T\rho_w\|S\|\left(W - \tilde{W}\right)\right) \leq -\rho_w\|S\|\left\|\tilde{W}\right\|^2 + \rho_w W_M\|S\|\left\|\tilde{W}\right\| \tag{4.46}$$

$$-tr\left(\tilde{\sigma}^T\frac{\eta_\sigma}{K}\left(S^T\left(W - \tilde{W}\right)D(\hat{\sigma}E)\right)^T E^T\right) \leq \frac{\eta_\sigma E_M W_M\|S\|\|\tilde{\sigma}\|}{K} + \frac{\eta_\sigma E_M\|S\|\|\tilde{\sigma}\|\left\|\tilde{W}\right\|}{K} \tag{4.47}$$

$$tr\left(\tilde{\sigma}^T\rho_\sigma\|S\|(\sigma - \tilde{\sigma})\right) \leq \rho_\sigma\sigma_M\|S\|\|\tilde{\sigma}\| - \rho_\sigma\|S\|\|\tilde{\sigma}\|^2 \tag{4.48}$$

Substituting (4.45), (4.46), (4.47) and (4.48) into (4.44) one can obtain:

$$\begin{aligned}\dot{V} \leq {} & \rho_w W_M\|S\|\left\|\tilde{W}\right\| + \left(\rho_\sigma\sigma_M + \frac{\eta_\sigma E_M W_M}{K}\right)\|S\|\|\tilde{\sigma}\| - (K_1 - \omega)\|S\| \\ & - \frac{1}{2}\|S\|\left(\rho_w\left\|\tilde{W}\right\|^2 + \rho_\sigma\|\tilde{\sigma}\|^2 - \frac{2\eta_\sigma E_M}{K}\|\tilde{\sigma}\|\left\|\tilde{W}\right\|\right) - K\|S\|^2 \\ & - \frac{1}{2}\|S\|\left(\rho_w\left\|\tilde{W}\right\|^2 + \rho_\sigma\|\tilde{\sigma}\|^2\right)\end{aligned} \tag{4.49}$$

Using the Young inequality one derives:

$$\frac{(K_1-\omega)\|S\|}{2}+\frac{\rho_w\|S\|}{2}\left\|\tilde{W}\right\|^2\geq\sqrt{(K_1-\omega)\rho_w}\|S\|\left\|\tilde{W}\right\| \tag{4.50}$$

$$\frac{(K_1-\omega)\|S\|}{2}+\frac{\rho_\sigma\|S\|}{2}\|\tilde{\sigma}\|^2\geq\sqrt{(K_1-\omega)\rho_\sigma}\|S\|\|\tilde{\sigma}\| \tag{4.51}$$

$$\rho_w\left\|\tilde{W}\right\|^2+\rho_\sigma\|\tilde{\sigma}\|^2\geq 2\sqrt{\rho_w\rho_\sigma}\left\|\tilde{W}\right\|\|\tilde{\sigma}\| \tag{4.52}$$

From (4.37), (4.42), (4.50) and (4.51) one can obtain:

$$\rho_w W_M\|S\|\left\|\tilde{W}\right\|+\left(\rho_\sigma\sigma_M+\frac{\eta_\sigma E_M W_M}{K}\right)\|S\|\|\tilde{\sigma}\|-(K_1-\omega)\|S\|-\frac{1}{2}\|S\|\left(\rho_w\left\|\tilde{W}\right\|^2+\rho_\sigma\|\tilde{\sigma}\|^2\right) \tag{4.53}$$

$$\leq 0$$

Similarly, from (4.36) and (4.52) one gets:

$$-\frac{1}{2}\|S\|\left(\rho_w\left\|\tilde{W}\right\|^2+\rho_\sigma\|\tilde{\sigma}\|^2-\frac{2\eta_\sigma E_M}{K}\|\tilde{\sigma}\|\left\|\tilde{W}\right\|\right)\leq 0 \tag{4.54}$$

By replacing (4.53) and (4.54) into (4.49) we can obtain:

$$\dot{V}\leq -K\|S\|^2 \tag{4.55}$$

This ends the proof of Theorem 4.3.4.

4.3.3 RBF Neural network and Fuzzy logic

The above back-propagation neural network provides a highly adaptive tool for designing the controller. However, this technique requite a heavy computation. To tackle this problem, in this section, an artificial neural network using RBF neural network will be used to estimate the unknown function Γ.

• **RBFNN based sliding mode controller design (SMC-RBFNN)**

The RBFNN consists three layers, namely input layer, output layer and a hidden layer. The hidden layer consists n nodes, each node uses a Gaussian radial basis function as follow:

$$\mu_i(E)=exp\left(-\frac{\|E-r_i\|^2}{\eta_i}\right);\ i=1\cdots n. \tag{4.56}$$

where r_i is the node center of basic function, η_i represents the width value of Gaussian function for neural net i

Since we have a sufficiently number of hidden layer, the ideal value of Γ is computed by RBFNN is given by the following expression:

$$\Gamma=W^T\mu(E)+\zeta \tag{4.57}$$

where W is the idea weight matrix, $\mu(E)=\begin{bmatrix}\mu_1(E) & \cdots & \mu_n(E)\end{bmatrix}^T$, ζ represents the neural network approximation error. In the context of the problem at hand, the ideal weight matrix W is not available, and therefore the desired value of Γ cannot be obtained directly from (4.57). Consequently, an alternative approach is required to estimate the value of Γ. To this end, the estimation of the ideal weight matrix W, denoted as $\hat{W}$, will be used to obtain the estimation of Γ. This approach is based on the assumption that the ideal weight matrix W and the unknown function Γ are related. By estimating $\hat{W}$, an estimation of Γ can be obtained, thus allowing for the design of the controller. Hence, the estimation of Γ will be given as follow:

$$\hat{\Gamma}=\hat{W}^T\mu(E) \tag{4.58}$$

From (4.16) and (4.58), the adaptive neural network-based sliding mode controller will be designed as follows:

$$U=KS+K_1 sign(S)+\hat{W}^T\mu\left(E\right) \tag{4.59}$$

The objective is to train the neural network to give us the online optimal estimate of Γ by guarantee $\hat{W}$ converge to W.

Theorem 4.3.5. *Let us consider the system (4.9) under the assumption 4.2.1, 4.2.2, 4.3.3 . If the controller gains and the updated law for weight matrix are designed as follow:*

$$K > 0;\ K_1 \geq \epsilon_\zeta \tag{4.60}$$

$$\dot{\hat{W}} = \mu(E)S^T \tag{4.61}$$

Then the stability of the controller (4.59) will be asymptotically and globally ensured.

Proof: Let $\tilde{W} = W - \hat{W}$ and $\tilde{\Gamma} = \Gamma - \hat{\Gamma}$ are the estimation errors of the ideal weight matrix and the unknown function Γ respectively.

Consider the following Lyapunov function:

$$V = \frac{1}{2}S^T G^{-1} S + \frac{1}{2} tr\left(\tilde{W}^T \tilde{W}\right) \geq 0 \tag{4.62}$$

From (4.15), (4.60) one gets:

$$\dot{V} = S^T\left(-KS - K_1 sign(S) + \Gamma - \hat{\Gamma}\right) - tr\left(\tilde{W}\dot{\tilde{W}}\right) \tag{4.63}$$

Replacing Γ and $\hat{\Gamma}$ by (4.57) and (4.58) respectively results:

$$\dot{V} = S^T\left(\zeta - K_1 sign(S)\right) - KS^T S + S^T\tilde{W}^T\mu - tr\left(\tilde{W}\dot{\tilde{W}}\right) \tag{4.64}$$

Since $\dot{\tilde{W}} = \mu S^T$, which implies that $S^T\tilde{W}^T\mu - tr\left(\tilde{W}\dot{\tilde{W}}\right) = 0$ and following the assumption 4.3.3 which implies that:
$S^T\left(\zeta - K_1 sign(S)\right) \leq 0$, one gets:

$$\dot{V} \leq -KS^T S \leq 0 \tag{4.65}$$

This ends the proof of Theorem 4.3.5.

SMC is known for its robustness, which is an outstanding advantage. However, this feature is only applicable when the error dynamic reaches the sliding manifold. In order to ensure stability, it is required that the control parameter k and sliding surface coefficient c satisfy certain positive sufficient conditions. Different control gains are suitable for each stage of the sliding surface approach toward zero and the error dynamic approach to the sliding surface. These conditions are detailed in several studies, including (Cheng and Liu, 2019), (Falcón, Ríos, and Dzul, 2019), (L'Afflitto, Anderson, and Mohammadi, 2018), (Mofid and Mobayen, 2018), and (Safeer Ullah and Iqbal, 2020).

- **RBFNN and fuzzy logic based sliding mode controller design (SMC-RBFNN-F)**

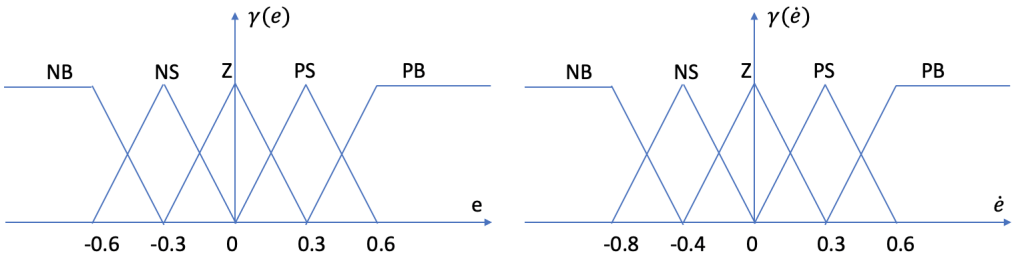


Figure 4.2: FL inputs membership functions

Hence choosing suitable values of c and k to optimal the tracking process is the next mission of this work. To achieve this task, the Fuzzy Logic using Sugeno fuzzy inference will be used, with inputs membership functions as shown in Figure 4.2. The rule i^{th} designed according to the tracking error and its derivative is designed as follows:

$$\text{If } e \text{ is } x_i \text{ and } \dot{e} \text{ is } y_i, \text{ then the output level } u_i = con_i.$$

(con_i is a constant). The final output is given by the weighted average over all rule outputs:

$$output = \frac{\sum_{i=1}^{n} w_i u_i}{\sum_{i=1}^{n} w_i} \tag{4.66}$$

where n is the number of rules, w_i is the rule firing strength derived from the rule antecedent:

$$w_i = \text{And-Method}(\gamma(e), \gamma(\dot{e})) \tag{4.67}$$

The fuzzy rules and the outputs are illustrated in table 4.1.

Fuzzy rule for c					
e	$\dot{e}$				
	NB	NS	Z	PS	PB
NB	B	VB	VS	VB	B
NS	S	M	B	M	S
Z	VS	S	M	S	VS
PS	S	M	B	M	S
PB	B	VB	VS	VB	B

Fuzzy rule for k					
e	$\dot{e}$				
	NB	NS	Z	PS	PB
NB	VB	B	M	B	VB
NS	B	M	S	M	B
Z	M	S	VS	S	M
PS	B	M	S	M	B
PB	VB	B	M	B	VB

Table 4.1: Fuzzy rule for c and k for the Sliding surface and controller design

SMC provides significant advantages, particularly in terms of robustness. This is due to the use of the *sign* function, which enables the error dynamics to quickly reach the sliding manifold. However, the use of the *sign* function is also known to cause chattering, which can be reduced by replacing it with an alternative function, namely *sat* as following:

$$\begin{cases} sat(S) & = 1 \text{ if } S \geq r \\ sat(S) & = -1 \text{ if } S \leq -r \\ sat(S) & = S \text{ otherwise} \end{cases} \tag{4.68}$$

The use of the *sat* function has its limitations in ensuring the negative definiteness of $\dot{V}$, as it can only guarantee this condition when S is outside the ball defined as $|S| = r$. When S moves towards the inside of the ball, equations (4.65) and (4.55) may not hold, causing an increase in both $\dot{V}$ and V. However, this increase in V leads to an increase in S, which pushes S outside the ball where $\dot{V}$ is negative, ultimately reducing both V and S. This analysis shows that S is ultimately bounded, which also leads to the ultimate boundedness of the tracking error E.

4.4 Simulation validation

This section deals with the simulation validation of our designed controllers. First, the simulation setup is depicted, explaining how to run simulations and how we embedded the controller. Second, the obtained simulation results will be shown and discussed.

4.4.1 Hardware In The Loop (HITL) simulation

The PX4 firmware, allows to run some simulations directly in the Px4 card, this is so called Hardware in the loop simulation. The controller is firstly programmed using **C++** programming language and then upload to Px4 board. The Px4 then will be connected to the laptop where a simulator such as Gazebo is used to visualize the operation of the quadrotor UAV. Then the common UDP ports (14550) are used to connect to the Qgroundcontrol with the simulator. The desired position for the simulation will be then designed in Qgroundcontrol. Figure 4.3 shows diagram from the PX4 documentation. It illustrates a typical **HITL environment** and the HITL setup for the validation.

In order to demonstrate the effectiveness of the proposed controller, a comparative analyse with other method, namely PID-RBFNN (Doukhi and Lee, 2019) controller and SMC-RBFNN (Alqaisi et al., 2019) will be carried in this part. The simulation will be conducted as follow: the quadrotor UAV firstly takes of to the desired altitude of 4 m, then the UAV will be tracking the pre-defined desired trajectory provided by Qgroundcontrol as shown in Figure .4.4 (this trajectory can chosen based on the user), in this task the controller have to guarantee the uav remains at the same altitude level 4 m. Finally, the UAV will land to the ground since it reaches the end point.

The parameters for the controller (4.21) and the BPNN (4.31), (4.32) were chosen according to theorem 4.3.4. as: $c = 0.8$, $K = 2.7$, $K_1 = 2.5$, $\eta_w = 2.7$, number of node $n = 10$, $\eta_\sigma = 0.14$, $\rho_w = \rho_\sigma = 0.1$. The SMC in (Alqaisi et al., 2019) will be designed with the same parameter , the PID controller in (Doukhi and Lee, 2019) is designed with $K_p = 2.16$, $K_d = 2.7$, $K_i = 0$. The RBFNN is designed as follow:

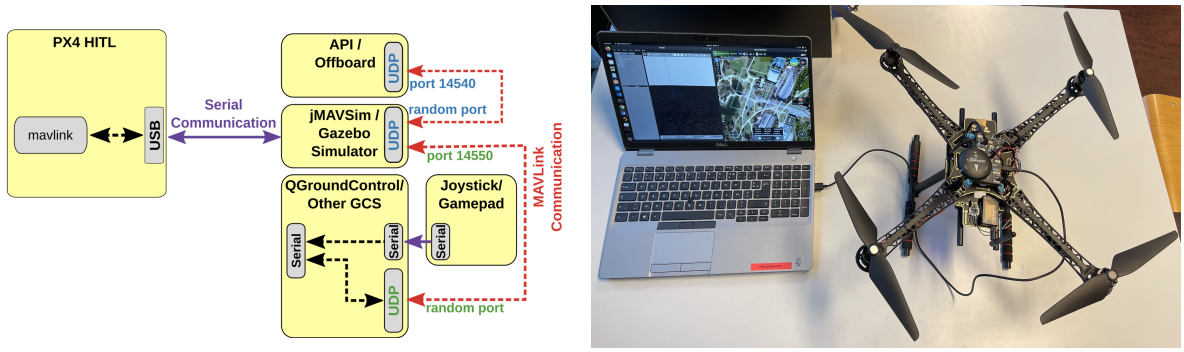


Figure 4.3: HITL environment (left), and HITL setup for the validation (right)

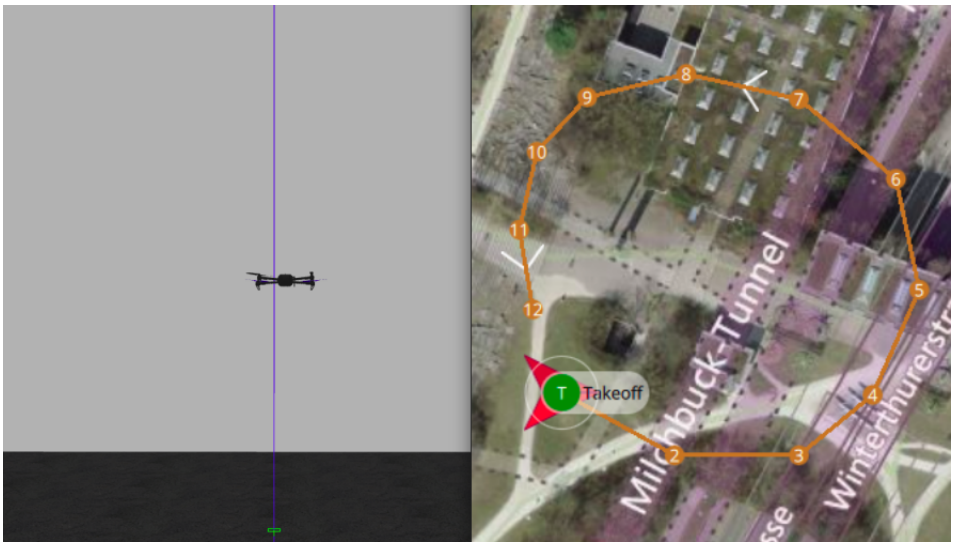


Figure 4.4: Quadrotor UAV is flying in the Gazebo simulator (left), Qgroundcontrol mission control platform (right) and the desired trajectory (brown line).

Output	VS	S	M	B	VB
c	0.6	0.7	0.8	0.9	1.0
k	2.4	2.6	2.8	3.0	3.2

Table 4.2: Fuzzy output value for c and k

$r_x = \text{linspace}(-0.5, 0.5, 10)$, $r_y = \text{linspace}(-0.5, 0.5, 10)$, $r_z = \text{linspace}(-0.7, 0.7, 10)$, $\eta_i = 2$. The gravity is set as $g = 9.81 m/s^2$. The parameter for the controller (4.59) will be trained using Fuzzy logic based on the the rules are given in table 4.1 and the output value given in table 4.2.

4.4.2 Simulation results

The tracking control performances obtained from the PID-RBFNN, SMC-RBFNN, SMC-BPNN, and SMC-RNFNN-F controllers proposed in this work are shown in Figure 4.5. It can be observed that all the proposed controllers successfully guarantee the trajectory tracking task. However, the results obtained from the PID-RBFNN controller are less effective compared to the others. Figure 4.6 illustrates the tracking errors of these controllers, clearly demonstrating that our controller maintains the quadrotor's location closest to the desired position with the smallest tracking error. Additionally, the control performance of SMC-RBFNN-F is relatively good compared to SMC-BPNN, while requiring less computational resources.

Figure 4.7 presents the desired attitude given by the studied controllers (calculated based on (4.7) and (4.8)) compared to the measured values. These results demonstrate that our controllers achieve smooth and stable attitudes for the tracking task, resulting in a stable flight state for the drone. However, the attitudes obtained using PID-RBFNN and SMC-RBFNN exhibit high oscillations, indicating that the quadrotor is flying in a less stable state.

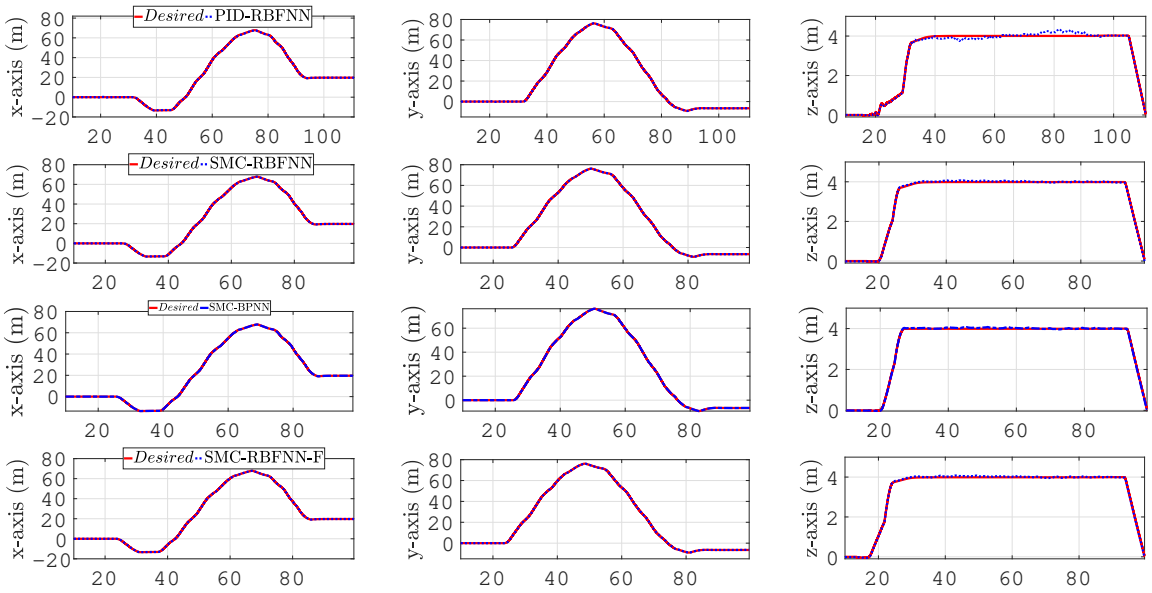


Figure 4.5: Position tracking results using PID-RBFNN , SMC-RBFNN, SMC-BPNN and SMC-RBFNN-F. Horizontal axis is time in seconds

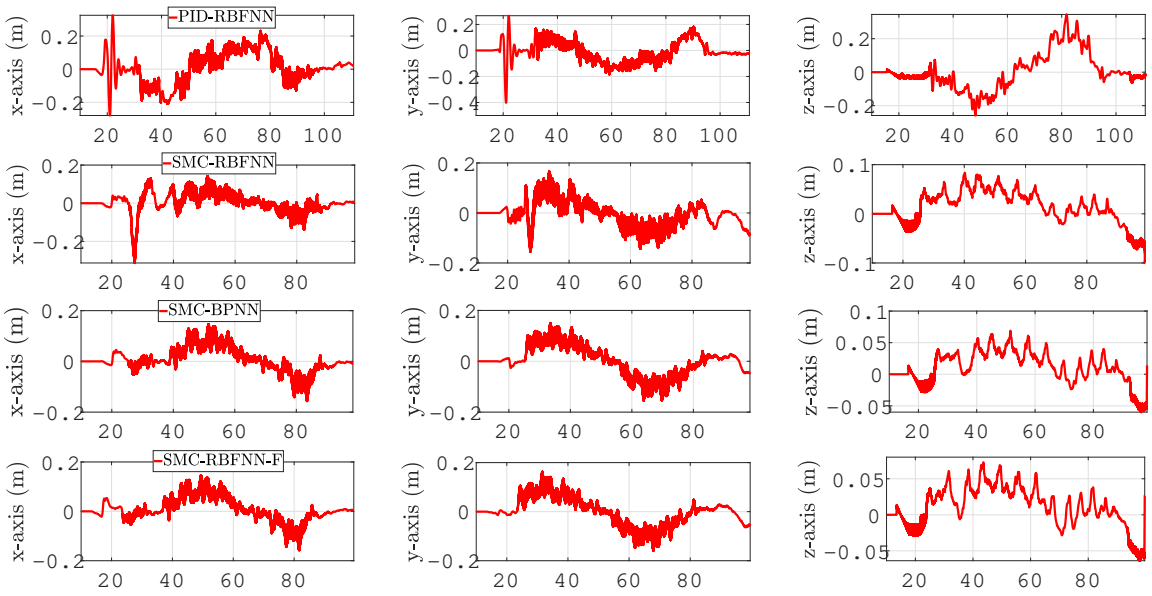


Figure 4.6: Position tracking errors using PID-RBFNN, SMC-RBFNN, SMC-BPNN-F and SMC-RBFNN-F. Horizontal axis is time in seconds

The output of the Fuzzy law and the sliding surface are shown in Figure. 4.8.

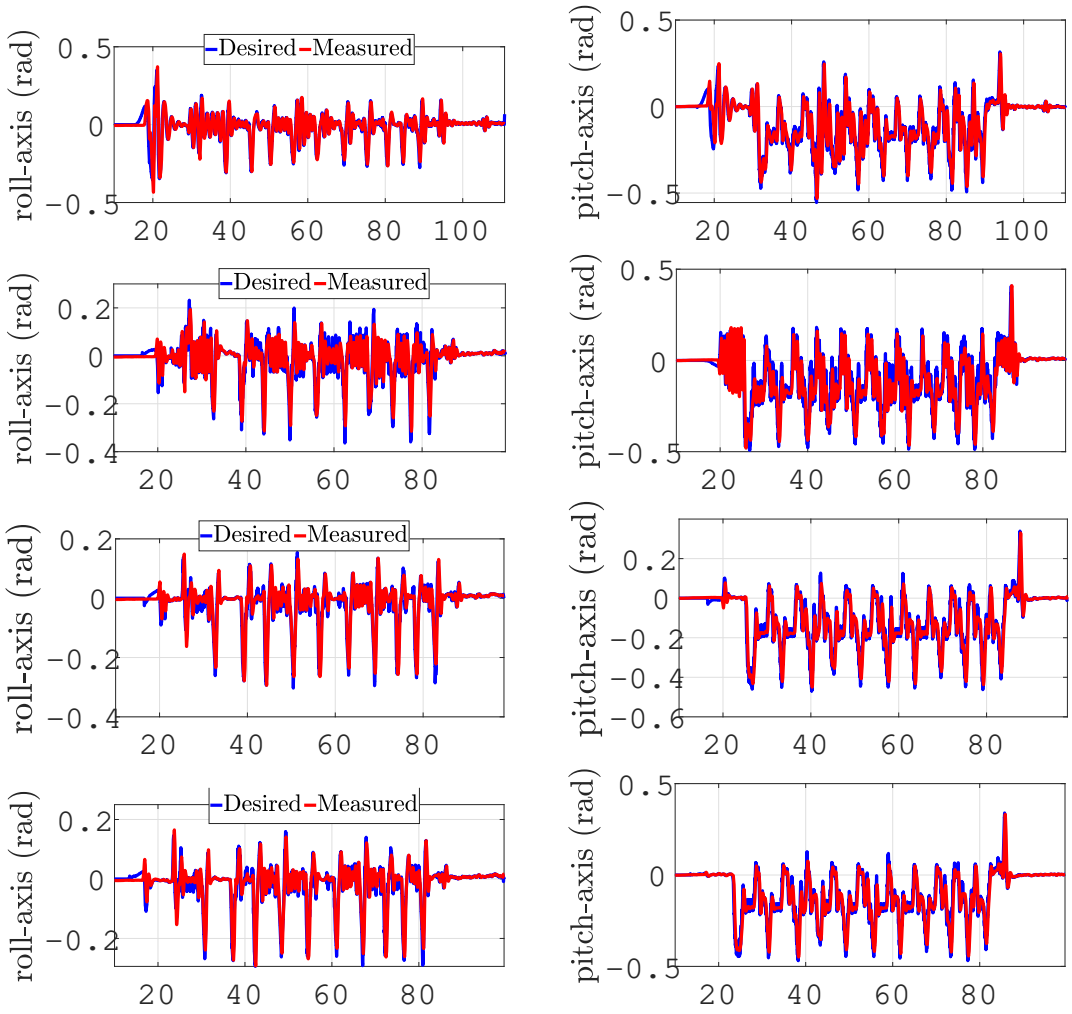


Figure 4.7: Desired attitude given by PID-RBFNN (first- up to down), SMC-RBFNN (second- up to down), SMC-BPNN (third- up to down), SMC-RBFNN-F (last- up to down), compared to its measured value, horizontal axis is time in seconds

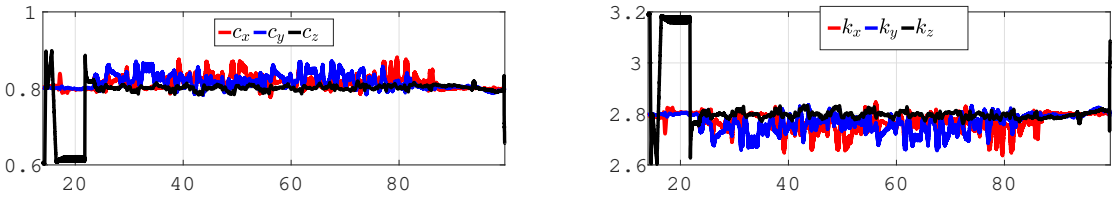


Figure 4.8: Tuned control gains c and k using SMC-RBFNN-F for the simulation, horizontal axis is time in seconds

4.5 Experimental validation

This section focuses on the experimental validation of the proposed controller. The first part will present the hardware setup, including the components of the quadcopter, their assembly, and configuration. In the second part, the experimental results will be presented and discussed.

4.5.1 Experimental setup

In order to validate the proposed controllers, a real outdoor flight experiment was performed. The experimental setup is shown in Fig. 4.9. The platform used in this experiment consists of three main components:

- Quadrotor UAV: The UAV uses s500 quadcopter frame, enquired with a Pixhawk 4 autopilot mod-

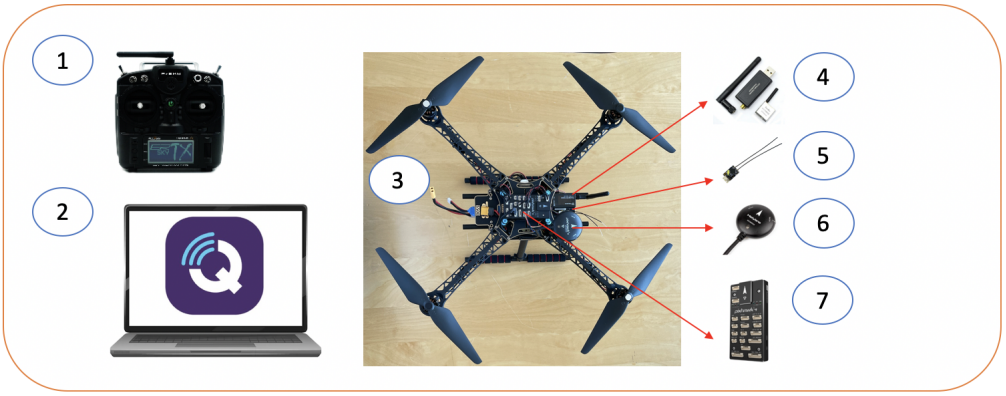


Figure 4.9: Experimental setup for the outdoor flight: 1. Radio controller; 2. Ground station laptop; 3. Quadrotor UAV; 4. Radio telemetry module; 5. Radio receiver; 6. GPS module; 7. Pixhawk card

ule. The controller is embedded directly into the Pixhawk 4 using **C++** programming language. The UAV is connected with Qgroundcontrol in station PC through telemetry radio (200Hz) communication using MAVlink protocol.

- Radio controller: The remote controller will be used to manually control the UAV (manual flight mode) in the case of accident (lost connection with Qgroundcontrol). In the mission flight mode, manual flight mode is deactivated as depicted in the Figure .4.10.

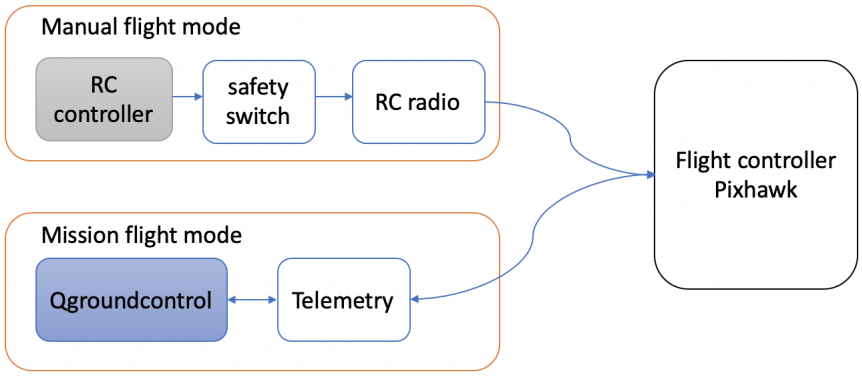


Figure 4.10: Experimental setup schema for safety mode

- Ground station PC. The ground station laptop allows us to provide the desired trajectory for the UAV using Qgroundcontrol. Since the desired trajectory is designed, it will then be uploaded to quadrotor using MAVlink protocol.

Figure 4.11: Desired trajectory from left to right for PID-RBFNN (first), SMC-RBFNN (second), SMC-RBFNN-F (third), SMC-BPNN (last)

It is worth noting that for validating each controller, we have to update the corresponding Firmware into the Pixhawk 4. After updating the Firmware, the configuration has to be redone for re-synchronizing

the sensor with the new Firmware. Consequently, for each controller, we have to re-design the desired trajectory. In order to make sure that the comparison with other method is fair, the desired trajectory for each experiment will be chosen approximately similar as shown in Figure. 4.11.

The successful validation of the proposed controller is illustrated in Figure. 4.12, which depicts the quadrotor UAV completing a mission flight assigned by qgroundcontrol.

Figure 4.12: Quadrotor UAV flies for the experimental validation

4.5.2 Experimental results

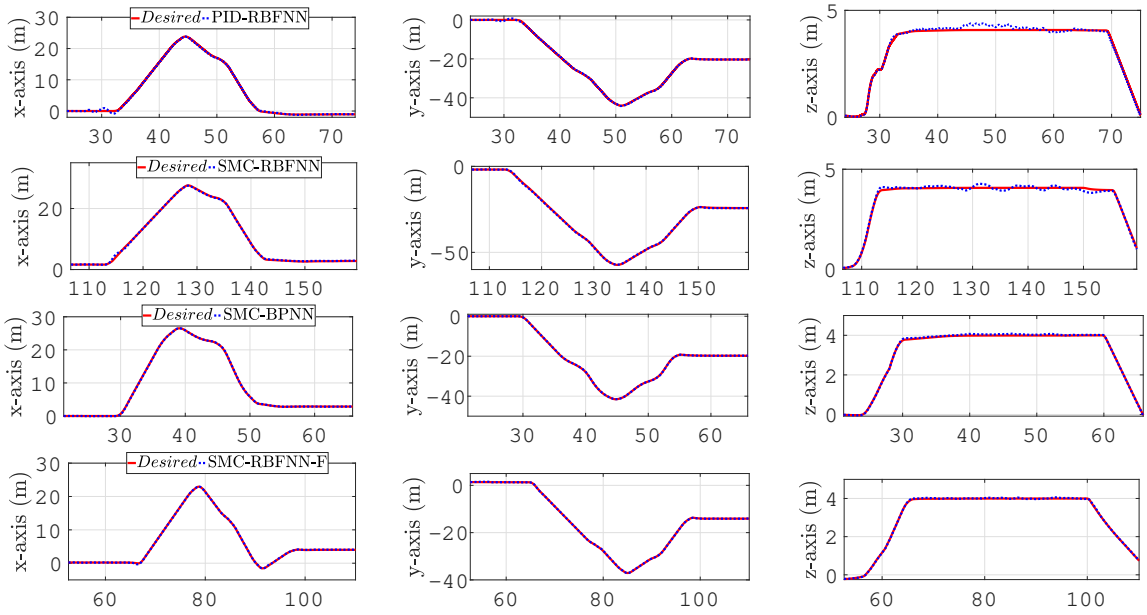


Figure 4.13: Position tracking results using PID-RBFNN , SMC-RBFNN, SMC-BPNN and SMC-RBFNN-F. Horizontal axis is time in seconds

As illustrated in Figure 4.13 and 4.14, the proposed controller has the capacity to provide a notable trajectory tracking performance in a very uncertain environment, compared to PID-RBFNN and SMC-RBFNN. These results showcase the power of Fuzzy logic in terms of tuning appropriate values for the controller, as well as the adaptive property of the backpropagation technique, which guarantees better tracking performance with smaller tracking errors. The output of the Fuzzy law and the sliding surface are shown in Figure 4.16.

The adaptive characteristics of the proposed controller demonstrate higher stability quality in comparison to PID-RBFNN and SMC-RBFNN, as shown in Figure 4.15. The SMC-RNFNN-F and SMC-BPNN provide relatively smooth attitudes, while those given by PID-RBFNN and SMC-RBFNN exhibit high oscillations.

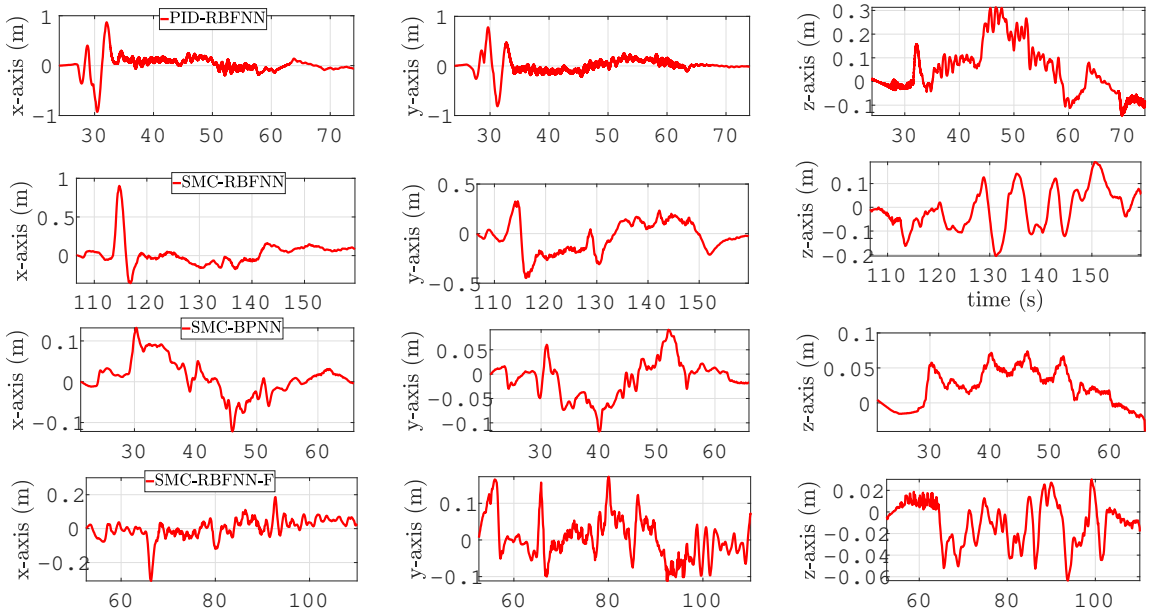


Figure 4.14: Position tracking errors using PID-RBFNN , SMC-RBFNN, SMC-BPNN and SMC-RBFNN-F. Horizontal axis is time in seconds

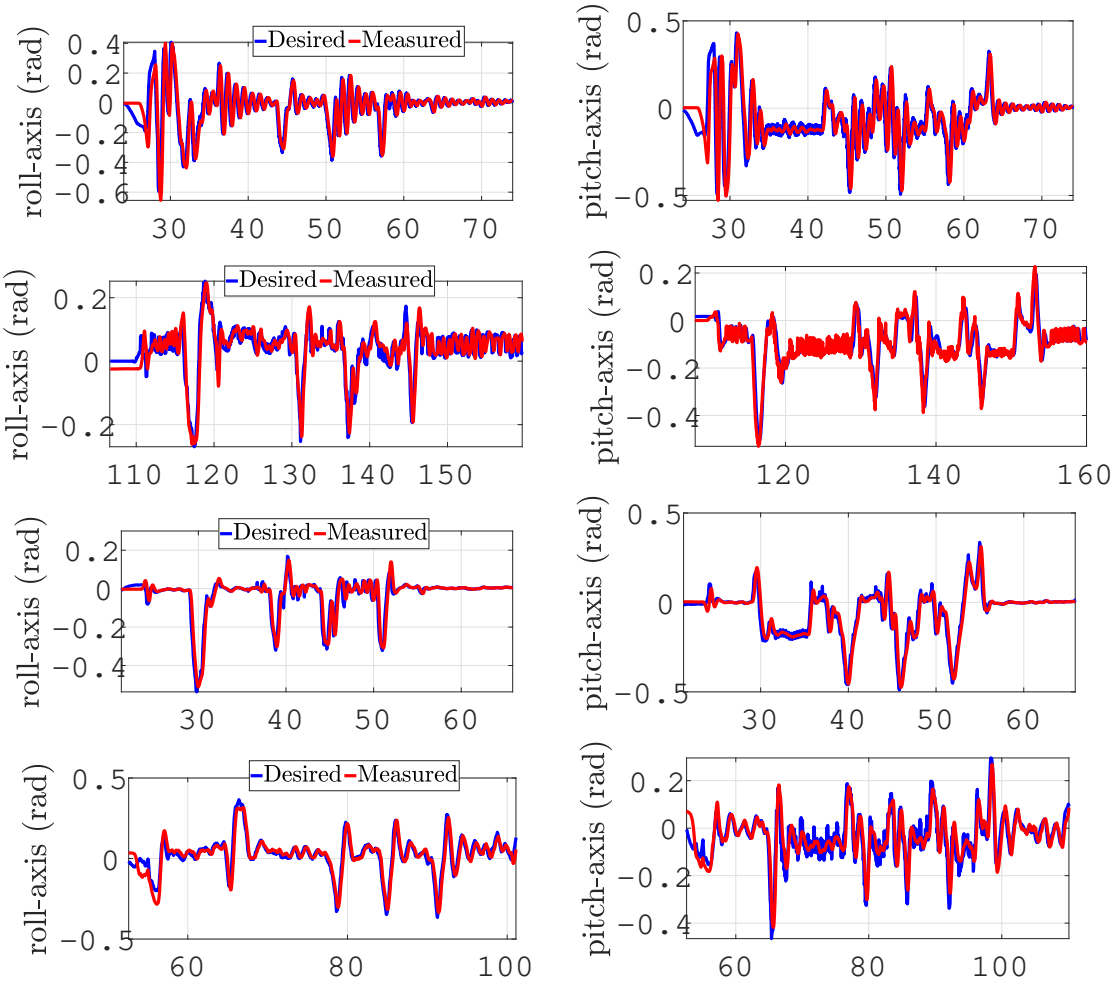


Figure 4.15: Desired attitude given by PID-RBFNN (first- up to down), SMC-RBFNN (second- up to down), SMC-BPNN (third- up to down), SMC-RBFNN-F (last- up to down), compared to its measured value. Horizontal axis is time in seconds

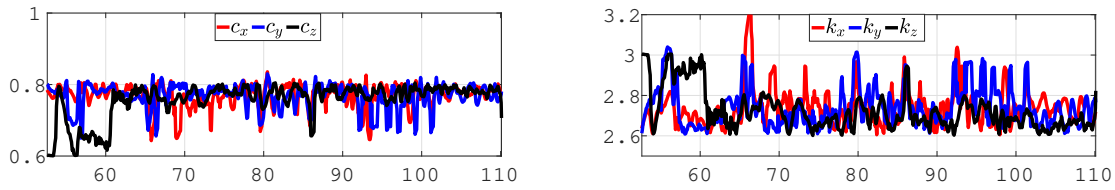


Figure 4.16: Tuned control gains c and k using SMC-RBFNN-F for the experiment, horizontal axis is time in seconds

4.6 Controller performance under the delay affected

The previous simulations and experiments have illustrated the performance of the proposed controller in comparison with other control methods. In this section the control performance of proposed controller under delay effects in the measurement wil be evaluated. As demonstrated from the previous chapter, when the delay exists, it might cause some instability in the control performance. Hence, the UAV might cashed during the flight mode due to the delay affected in the measurement. Therefore, for the safety reason, in this part, one shall present only the simulation.

The simulation setup for validating the controller is shown in Figure. 3.5. The controller block in this simulation is SMC-RBFNN-F.

To evaluate the effects of delay into the control performance. Two simulations will be carried out. The first simulation will use directly the delayed-noisy measurement. While the second simulation will deploy a FHGO developed in the chapter 3 to compensate the effect of delay and noisy measurement

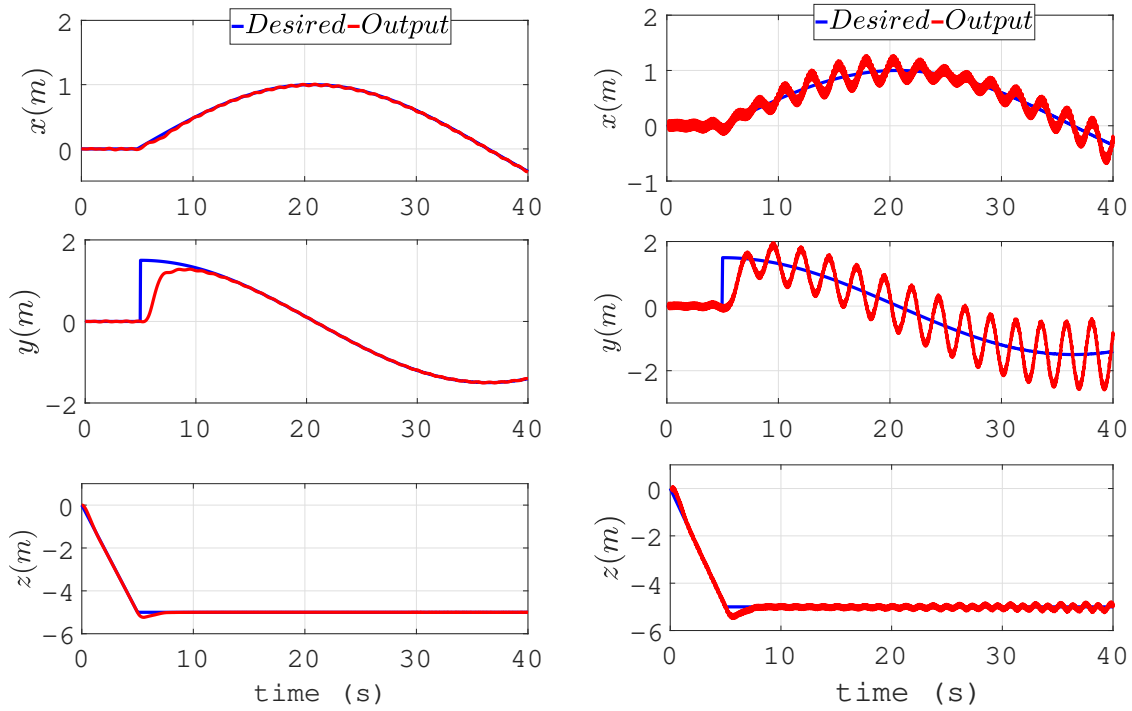


Figure 4.17: Control performance using the observer (left), using directly the delayed ($\tau = 0.18s$), noisy measurement (right)

The Figure 4.17 shows the control performance using the same controller. One used directly the delayed, noisy measurement and one used the proposed observer to compensate the delay and noise. As can be seen from this figure, the adding observer significantly improves the control results. The tracking error is relatively small in comparison with using poor measurement. This results have demonstrated the importance of the observer.

4.7 Conclusion

This chapter outlines the process of designing a control system for a quadrotor UAV. Multiple controllers are proposed and their respective advantages and disadvantages are thoroughly discussed. Notably, an adaptive law for the neural network is derived, which allows the controller to compensate for the effects of disturbance, model uncertainty, and unknown parameters without requiring knowledge of the system's model parameters.

The effectiveness of our proposed control method is demonstrated through simulation and experimental results obtained from the quadrotor UAV system. In particular, our approach is shown to successfully compensate for the aforementioned effects, highlighting its practical relevance and potential applications.

To evaluate the performance of our proposed controller, we conduct a comparison with recent work in the literature. This comparison serves as a comprehensive assessment of the efficacy of our control method and its potential to outperform existing approaches.

Furthermore, the outstanding property in term on improving the control quality under delayed-noisy measurement of the FHGO has been well emphasized in the last simulation.

Building on the control system designed in this chapter, the next step in this thesis will be to extend this method to control a swarm of UAVs. The observer developed in the previous chapter and the controller proposed in this chapter will serve as the foundation for this extension, allowing for the precise and coordinated control of multiple UAVs.

By leveraging the insights gained from the development of the observer and controller, the swarm control system will be capable of compensating for the effects of delay communication between multiple UAVs.

The subsequent chapter will delve into the study of formation control for a swarm of UAVs, where the SMC-RBFNN-F position controller proposed in this work will be employed to govern the position control of each individual UAV in the swarm.

CHAPTER 5

Formation flight of quadrotor UAVs

5.1 Introduction

Using a single quadrotor UAV can be limiting for certain complex tasks due to factors such as its size, payload capacity, and ability to cover large areas. Applications such as agriculture, search and rescue, environmental monitoring, infrastructure inspection, disaster response, military surveillance, film and TV production, wildlife conservation, sports broadcasting, and construction site monitoring require more extensive coverage and capabilities than a single UAV can provide.

To overcome these limitations, swarms of cooperating UAVs can be employed. Research in cooperative control of multiple UAVs, or multi-agent control, is gaining interest due to its high level of autonomy and ability to significantly reduce the human resources required to operate the system. Swarms can communicate and share information with each other, allowing agents to work independently with a central control station to achieve desired tasks. Compared to central-based control, cooperative control is far more autonomous.

The study of multi-agent control has wide-ranging applications, from mobile robots to UAV systems. Some examples of research in this field include studies on formation control, task allocation, communication protocols, and obstacle avoidance. References to these studies can be found in works such as (Maneekittichote and Chanthasopeephan, 2020; Yang et al., 2015; Senanayake et al., 2016; Grandi, Falconi, and Melchiorri, 2012; McLain et al., 2001; Acevedo et al., 2013; Hu and Jin, 2022; Borrelli, Keviczky, and Balas, 2004; Ali et al., 2021; Gozzini et al., 2020). The potential for cooperative control to overcome the limitations of single UAVs and enable more complex and extensive applications makes it a valuable area of research in the control community. Multi-agent control or swarm control strategies often consists of two component: mission control and formation control.

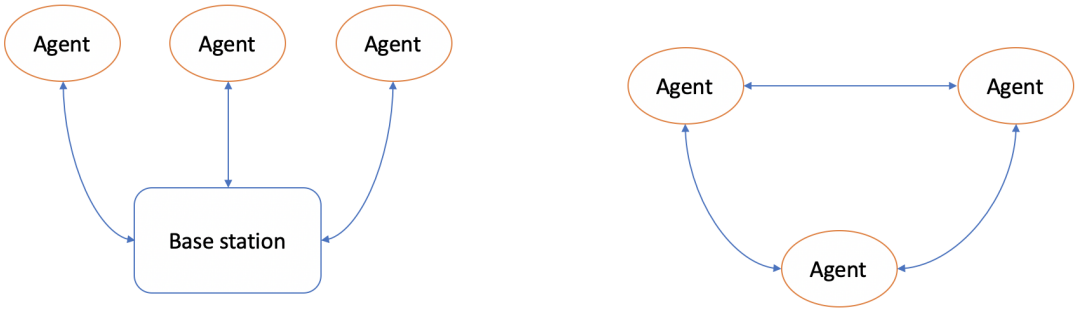
5.1.1 Mission control

Mission control in swarm control refers to the planning and coordination of tasks between the individual agents in a swarm. This involves determining the objectives of the swarm as a whole, as well as assigning specific tasks to individual agents to achieve those objectives. Mission control also involves routing and coordination of the agents' movements to ensure that they work together efficiently and effectively. Depending on the communication strategy adopted, mission planning for swarm control can fall under two main categories: centralized and decentralized architectures..

The centralized approach use a base station as depicted in Figure. 5.1a. All the agents in swarm will communicate with this station and the station give the control command directly to each agent. All computations and critical decisions are made at central base.

By contrast, the decentralized approach allows each agent in swarm have more autonomy property as depicted in Figure. 5.1b. Each agent is capable of making its own decisions and communicate with its neighbour agent. This approach eliminates the communication through the ground station. Hence, the operation rang is significantly improved.

The critical advantages and disadvantages of centralized and decentralized architectures is compared in Table. 5.1.



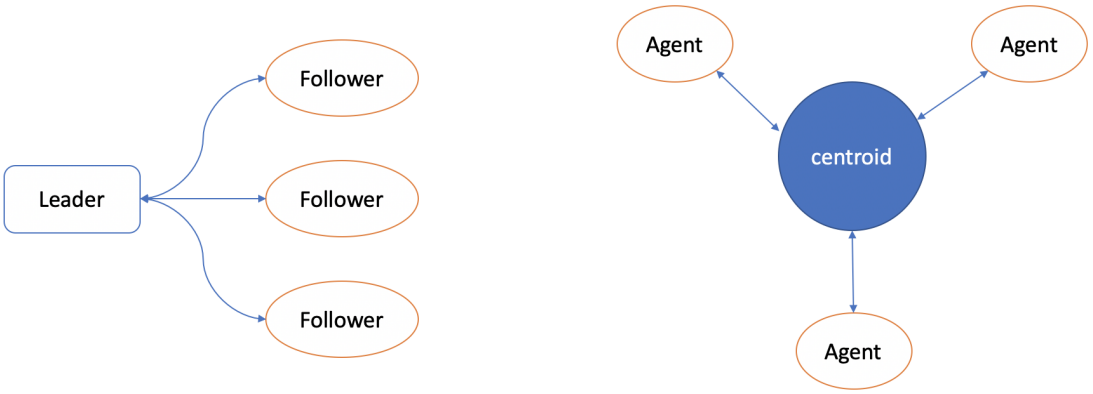
(a) Centralized (b) Decentralized

Figure 5.1: Visualization of communication strategies for mission control

Method	Advantages	Disadvantages
Centralized	Central authority responsible for critical decisions. No need for agent-to-agent communications. Agent loss has minimal impact on mission objectives	Communications limited to base station range. Complete reliance on base station availability. Heavy computational requirements.
Decentralized	Individual agent autonomy. Not limited to central base station range. System scales well.	Hierarchy or a coordination algorithm is needed.

Table 5.1: Advantages and disadvantages of centralized and decentralized architectures

5.1.2 Formation control



(a) Leader-Follower (b) Virtual Structure

Figure 5.2: Visualization of communication strategies for formation control

Formation control is the most actively studied topics for multi-agent systems, it refers to the ability of multiple unmanned aerial vehicles (UAVs) to maintain a specific geometric arrangement or pattern while performing a task. This can be achieved through communication and coordination between the UAVs, which allows them to work together to achieve a common goal. The formation can be fixed or dynamic, depending on the task and the environment, and the UAVs can adjust their position and trajectory relative to each other to maintain the desired formation. The overview of formation control can be found in following excellent surveys (Oh, Park, and Ahn, 2015; Yang, Xiao, and Li, 2021; Shi and Yan, 2021).

Depending on the communication strategy adopted, the formation control strategy can be categorized into two main approaches: leader-follower and virtual structure.

In the leader-follower approach, one agent is designated as the leader and the other agents follow its movements to maintain a desired formation as depicted in Figure. 5.2a. Depends on the used algorithm, the role of leader and followers can be switched between agents or not. In this configuration, the leader plays a role of leading the swarm tracking the mission. Each follower will have different predetermined offset that they follow depending on the mission requirement. This approach can be useful when there is a need for a clear hierarchy of control and a designated leader who can make decisions for the group. However, it can also present limitations in terms of scalability and robustness, as the entire swarm may be affected if the leader agent fails or becomes unavailable.

On the other hand, the virtual structure approach is considered to be a rigid body and all the agent moves as one whole group. The agent position will be maintained relative to the centroid of a virtual structure as depicted in Figure. 5.2b. Hence, all the agents in this approach is independent to the other agent. However, this lacking of communication between agents might caught the collision. Therefore, the trajectories for each agent in swarm are constantly calculated and each agent will be transmitting and receiving position information frequently. Consequently, the communication in this case is required high speed and low latency.

The critical advantages and disadvantages of virtual structure and leader-follower architectures are summarized in the Table. 5.2.

Method	Advantages	Disadvantages
Leader-Follower	Simple communications Scalable to large group of followers Simple to implement	Single-point of failure (Leader) might cause a failure for the whole swarm
Virtual structure	Each drone is given precise trajectories More robust	Communication is complex Each drone needs a unique ID system Dependent on controller

Table 5.2: Advantages and disadvantages of Leader-Follower and Virtual structure architectures

These mentioned missions and formation control methods, provide us four combinations of control method to control a multi-agent system, namely: Centralized Leader-Follower, Centralized Virtual Structure, Decentralized Virtual Structure, Decentralized Leader-Follower. In what follows, we will shortly review these method.

• **Centralized Leader-Follower**

This method deploys Leader-Follower formation control and uses a central station base for the mission control. In Centralized Leader-Follower strategy, the leader will sent its position and velocity to the ground station computer. Then all the calculation will be done in this computer and send to the followers. In (Yun, Chen, and Lum, 2010), the author implemented this method for control a warm of unmanned helicopters and validate this method in simulation. In (Santana, Brandão, and Sarcinelli-Filho, 2021) this method is analysed experimentally for a two quadrotors UAV.

• **Centralized Virtual Structure**

Unlike Leader-Follower, where the agents are separated into to role, namely leader and follower. The Virtual Structure consider the swarm as a rigid body, thus makes each agent is independent to the swarm since all the calculations are done in the station base computer. This method theoretically is more robust than Centralized Leader-Follower. However, the required calculations to maintain this structure stable is heavier. The application of this method can be found in (Broek et al., 2011).

• **Decentralized Virtual Structure**

This strategy for swarm control involves treating the group of agents as a single, rigid body. However, with the implementation of decentralized mission control, each individual agent can operate with a greater degree of autonomy. This is made possible by equipping each agent with an autopilot system that enables it to track its own reference trajectory, velocity, altitude, and heading. To determine its reference trajectory, an agent communicates with its neighbors within the swarm. Researchers have explored this method in developing controllers for swarms of UAVs, as seen in (Li and Liu, 2009), where each agent is considered an individual access point within a self-configuring network. Another example is (CAI et al.,

2020), which uses the same control strategy to design a swarm intelligent optimization algorithm for UAV swarms.

• **Decentralized Leader-Follower**

Decentralized Leader-Follower is the most used control strategy to build the controller for multi-agent systems and swarm of UAVs. Based on this framework, many control techniques have been applied to build the controller for swarm of UAVs. In (Dong et al., 2017; Dong et al., 2015) the time-varying formation tracking is analysed and designed for a formation control of a swarm of quadrotor UAVs. In (Wang, Cheng, and Xiao, 2020), the author proposed a controller using Hopf bifurcation. Many other excellent works using this framework can be found in (Mechali et al., 2022; Li, Yang, and Zhang, 2019; Xia, Liu, and Liu, 2018; Zhi et al., 2021; Dong et al., 2014; Dong et al., 2019; Zhou, Dong, and Zhong, 2016; Zhou et al., 2014).

In this area, consensus control is the most popular topic, which aims to achieve the formation objective by coordinating multiple agents. The communication between agents play a crucial part to maintain the autonomous property of the system. Similar to many engineering system, the multi-agents system also has to face with one of the most well known problem in communication, namely, time-delay. The existence of delay makes the formation control design much more complicated, and is the resource of instability as analysed in Chapter 3.

To overcome such interesting problem, many methods have been proposed to design a formation control for multi-agents system subjected to delays. In (Wang et al., 2020), by assuming that the delay affected to the control input, the authors combined a cascade structure predictive observer design for consensus control. Using the same assumption, the consensus control for multi-agent have been studied in (Zhang et al., 2019; Hua, Li, and Guan, 2019). In (Wang et al., 2017), the formation control problem is studied for a system with delay measurements. In (Wang et al., 2019), the delay effect in both input and output is considered.

Motivated by this problem and its drawback, our work aims to develop a controller for formation tracking of a multi-agent system and application to a swarm of three quadrotor UAVs, under the assumption that the delay affects both output measurements and control inputs. Our work inherits the success of the proposed observer and controller given in Chapter 3 and 4. By assuming that the trajectory tracking of each agent works perfectly, a formation controller is designed using the predictor feedback method given in (Krstic, 2010).

5.2 Basic concept on formation topology

.

5.2.1 Communication topology

Communication topology is an important part to design the formation control. Indeed, this study is devoted to presenting the complex communication between one agent and another in the swarm into a simple algebraic representation. In the following, one shall give a brief introduction about concept on formation communication topology.

Let consider $G = \{V(G); E(G); W(G); L(G)\}$ is a weighted directed graph, where $V(G) = \{v_1, v_2, \cdots, v_n\}$ is the node set with n vertices where v_1 defines the leader and the others define the followers, $E(G) \subseteq \{(v_i, v_j) : v_i, v_j \in V(G), i \neq j\}$ is the edge set, $W(G)$ denotes the weighted adjacency matrix and be defined as: If there is information transfer between the node i and the node j, then $w_{ij} = 1$ otherwise $w_{ij} = 0$, the Laplacian matrix $L(G)$ can be defined as follows:

$$L(G) = \begin{cases} l_{ij} = -w_{ij} & \text{if } i \neq j \\ l_{ii} = \sum_{j=1, i\neq j}^{n} w_{ij} & i = 1 \cdots n \end{cases} \tag{5.1}$$

The graph G is said to have a spanning tree if there exists at least one vertex has a connection to all other vertices.

5.2.2 Relative position of agents

The main objective of formation control is to maintain the distance between agents in a desired value. Therefore, determine the relative position of agents is crucial, especially for leader-follower strategy, where the followers have to trace the leader with a desired formation.

In this study, we consider the formation of swarm will have the same altitude. Therefore, the main objective now is to determine the relative position of agents in XY plan. Figure. 5.3 shows an example

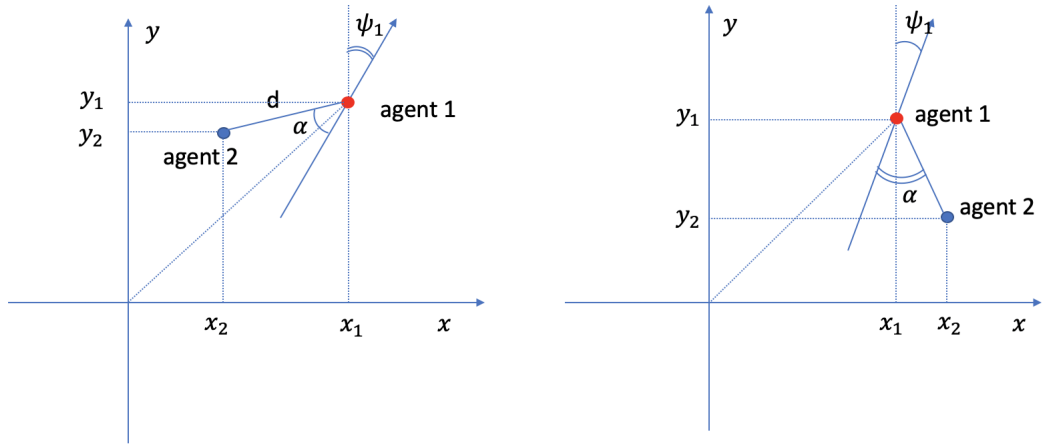


Figure 5.3: Example of relative position between two agents in a swarm

of two agents in XY plan. One assumes that the position $[x_1(t),\ y_1(t)]$ and the steering angle ψ_1 of agent 1 are known. Hence, to obtained the desired formation, one shall determine the desired position of agent 2 based on the desired relative distance to agent 1 d and the desired angle α. There are two possible positions for agent 2 satisfying the condition in which having a distance d away from agent 1 and having an angle α, namely on the left side and on the right side of agent 1. For example, in Figure. 5.3 (left) shows the agent 2 is in the left side of agent 1, 5.3 (right) shows the agent 2 is in the right side of agent 1.

Based on these properties, one shall give the expression of desired position for agent 2 regards to agent 1 as follow:

$$\text{On the left } \begin{cases} x_{2d} = x_1 + dcos(\frac{-\pi}{2} - \psi_1 - \alpha) \\ y_{2d} = y_1 + dsin(\frac{-\pi}{2} - \psi_1 - \alpha) \end{cases} ;\ \text{On the right } \begin{cases} x_{2d} = x_1 + dcos(\frac{-\pi}{2} - \psi_1 + \alpha) \\ y_{2d} = y_1 + dsin(\frac{-\pi}{2} - \psi_1 + \alpha) \end{cases} \tag{5.2}$$

5.3 Problem statement

Let us consider a multi-agent system with n agents. The communication between these agents is described by the graph G with each agent being a node in G . The swarm consists of one leader and $n-1$ followers. The main objective of formation control is to make the followers tracking a predefined formation regarding to the leader. Since the controller for each agent is considered well developed, the path tracking is considered working perfectly. Hence the formation control now turns into the design of the appropriate trajectory for the followers based on the desired formation and leader information. Let's now define the state vector of the agent i^{th} as follows:

$$x_i(t) = \begin{bmatrix} p_i^T(t) & v_i^T(t) & \psi_i(t) \end{bmatrix}^T ;\ i = 1 \cdots n. \tag{5.3}$$

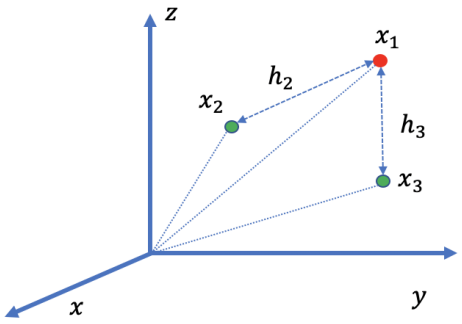


Figure 5.4: Example of formation with three agents

where $p_i(t) \in \mathbb{R}^{3\times1}$, $v_i(t)\mathbb{R}^{3\times1}$ and ψ_i are the position, the velocity and the steering angle of i^{th} agent respectively.

Let's define the desired formation vector for the followers as follows:

$$h_F = \begin{bmatrix} h_2^T(t) & \cdots & h_n^T(t) \end{bmatrix}^T \tag{5.4}$$

$$h_i(t) = \begin{bmatrix} h_{pi}^T(t) & h_{vi}^T(t) \end{bmatrix}^T ;\ i = 2 \cdots n. \tag{5.5}$$

where $h_{pi}(t)$ and $h_{vi}(t)$ corresponding to the position and velocity respectively.

Figure. 5.4 shows an example of a swarm enclosing with three agents. The distance between the followers 2 and leader 1 is h_2. The distance between the followers 3 and leader 1 is h_3.

Having established these definitions, we can now simplify the objective of formation control as follows: controlling the followers in such a way that the following condition is met:

$$x_i(t) - h_i(t) - x_1(t) = 0_{6\times 1} \tag{5.6}$$

In other words, we aim to control the follower in such a way that it can track a predetermined set point given as follows:

$$x_{id}(t) = h_i(t) + x_1(t) \tag{5.7}$$

Given that the path tracking controller for each agent is assumed to be perfect, the condition (5.7) can be easily achieved, which is $\lim_{t\to\infty} x_i(t) - x_1(t) = h_i(t)$. However, the primary objective of swarm control is generally to follow a desired formation, which allows for a less restrictive condition, namely $\lim_{t\to\infty} p_i(t) - p_1(t) = h_{pi}(t)$. By meeting this condition, the velocity of each agent can be considered as a control signal for further research.

Communication delay

The condition proposed above assumes that there is perfect communication between the agents and that data transmission occurs instantaneously. However, in the real application, there always exists the delay in the communication between these agents. Hence the desired state (5.7), which is normally calculated in the leader, will be affected by the delay when it arrived to the follower. Therefore, since the delay exists, instead of guarantee $\lim_{t\to\infty} p_i(t) - p_1(t) = h_{pi}(t)$. The formation can only obtain $\lim_{t\to\infty} p_i(t) - p_1(t) = h_{pi}(t-\tau)$, τ is the time delay.

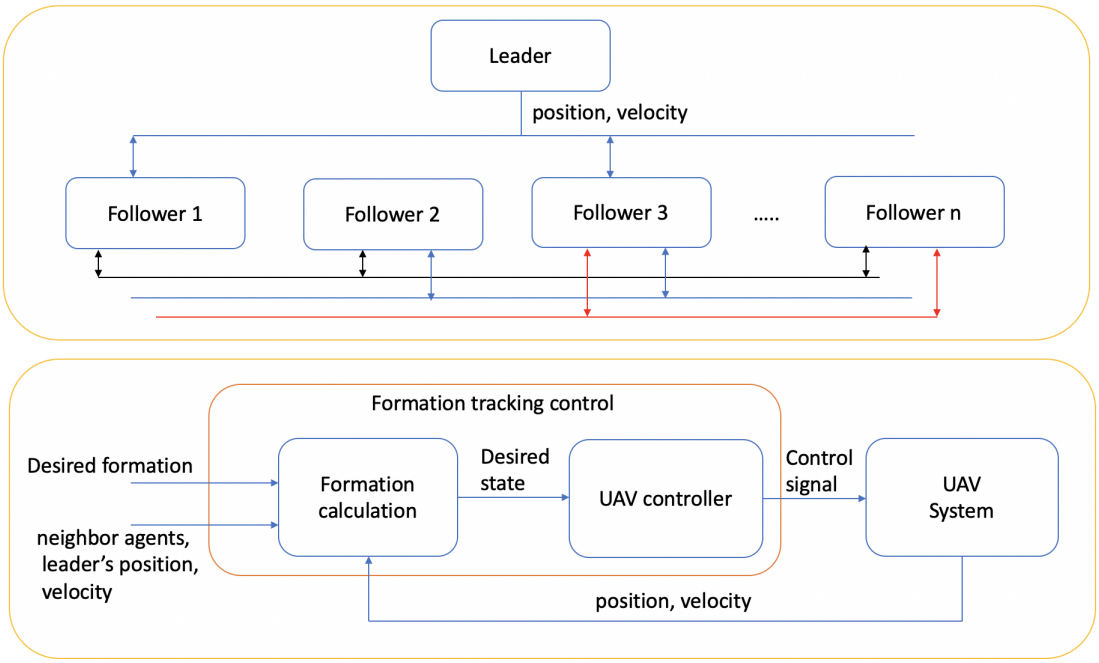


Figure 5.5: Block diagram of the formation swarm of UAVs. Example of communication in swarm (above). Formation control scheme for one agent (below)

Figure. 5.5 shows an example of communication between the agents in swarm and the control structure of each agent to obtain the formation tracking task. From this figure, we can see that, in decentralized leader-follower strategy, the leader will send the desired formation and its state to the directly connected agent (primary followers), for the agent does not have a connection with the leader (secondary follower). It can still receive these information through primary followers. In the following, one shall consider the dynamic model of a i^{th} primary followers with its measurement obtained at the leader. For the secondary follower, the same model can be delivered.

The upcoming section presents a proposed method that uses a delay observer, as discussed in Chapter 3, to mitigate the impact of delays on the system and enhance the quality of formation control. In order to achieve this, a dynamic model of the swarm will be introduced.

5.3.1 Dynamic model of multi-agent system with delay

The dynamic model of a multi-agent system in a leader- follower structure with delay is given as follows:

$$\begin{cases} \dot{p}_i(t) & = v_i(t) \\ y(t) & = x_i(t-\tau) + \xi(t) \end{cases} \tag{5.8}$$

In the above equation, $y(t)$ represents the measurement taken by the leader, which includes the position of the i-th agent, denoted as $x_i(t)$. The measurement is subject to a delay τ and measurement noise $\xi(t)$, which satisfies the assumption 3.2.2, 3.2.3.

The Figure. 5.6 demonstrates the delay effects in the communication between the agents in swarm.

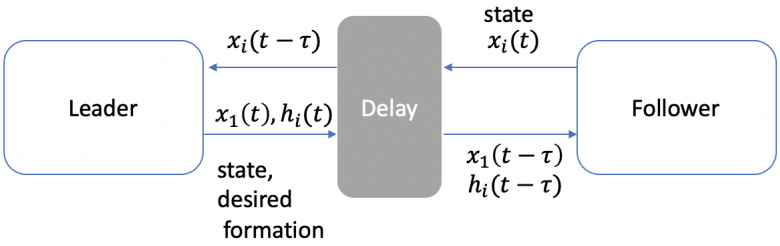


Figure 5.6: Block diagram of delay effects into the swarm communication

Since the controller is assumed to be perfect, we can approximate velocity of this agent by its given desired value, namely $v_1(t-\tau) + h_{vi}(t-\tau)$. then since the control task is achieved the dynamic model (5.8) can be rewritten by:

$$\begin{cases} \dot{p}_i(t) & = v_1(t-\tau) + h_{vi}(t-\tau) \\ y(t) & = p_i(t-\tau) + \xi(t) \\ p_i(t) & = \Psi(t) \quad \text{where} \quad t \in [-\tau,\ 0] \end{cases} \tag{5.9}$$

where $h_{vi}(t)$ will be considered as the control signal. Then the objective is now to design the formation control signal and the observer so that $\lim_{t\to\infty} p_i(t) - p_1(t) - h_{pi}(t) = 0_{3\times 1}$. For this sake the system (5.10) will be rewritten as follows:

$$\begin{cases} \dot{p}_i(t) & = v_1(t-\tau) + h_{vi}(t-\tau) \\ y(t) & = p_i(t-\tau) + \xi(t) \\ p_i(t) & = \Psi(t) \quad \text{where} \quad t \in [-\tau,\ 0] \end{cases} \tag{5.10}$$

Remark 5.3.1. *From Figure. 5.6 and the dynamic model (5.10), we can notice that the latency affects both side of communication. Therefore the desired formation, which is usually used as control signal for the agent arrives with the delay τ. Consequently, for an arbitrary desired formation, even if the trajectory tracking controller inside agent is working perfectly, hence, there always exists a control error. For example, the leader send the desired formation $h_{pi}(t)$ and its current position $p_1(t)$ to the i^{th} agent. This agent will receive $h_{pi}(t-\tau)$ and $p_1(t-\tau)$, since the tracking task in agent i^{th} is achieved, namely tracking $h_{pi}(t-\tau)$ and $p_1(t-\tau)$. there exists a formation control error: $e_F = h_{pi}(t) + p_1(t) - h_{pi}(t-\tau) - p_1(t-\tau)$. Therefore the formation control task can not be achieved*

The formation control problem will be solved under the following assumptions:

Assumption 5.3.1. *With the initial function $\Psi(t)$, the system (5.10) has a unique smooth, differentiable solution on the compact set $\mathbb{P} \subset \mathbb{R}^{3\times 1}$, i.e $p_i(t) \in \mathbb{P};\ \forall t \geq -\tau$*

Assumption 5.3.2. *All the follower in swarm have smooth, differentiable and bounded velocities on the compact set $\mathbb{V} \subset \mathbb{R}^{3\times 1}$, i.e $v_i(t) \in \mathbb{V};\ \forall t \geq 0$*

$$\exists\ \epsilon_v \in \mathbb{R}^+,\ \sup_{t\geq 0} \|v_i(t)\| \leq \epsilon_v;\ i = 2, \cdots, n \tag{5.11}$$

Assumption 5.3.3. *The leader in the swarm has smooth, differentiable and bounded velocity on the compact set $\mathbb{V}$ i.e $v_1(t) \in \mathbb{V};\ \forall t \geq -\tau$:*

$$\exists\ \sup_{t\geq 0} \|v_1(t)\| \leq \epsilon_v \tag{5.12}$$

Assumption 5.3.4. *The time-varying desired formation is smooth, differentiable and bounded i.e :*

$$\exists\ \epsilon_{h1} \in \mathbb{R}^+,\ \sup_{t\geq 0} \|h_{pi}(t)\| \leq \epsilon_{h1} \tag{5.13}$$

$$\exists\ \epsilon_{h2} \in \mathbb{R}^+,\ \sup_{t\geq 0} \left\|\dot{h}_{pi}(t)\right\| \leq \epsilon_{h2} \tag{5.14}$$

5.4 Formation tracking analyse

Let us rewrite the system (5.10) under the form of class (3.2) as follows (see in chapter 3):

$$\begin{cases} \dot{p}_i(t) & = u(t-\tau) \\ y(t) & = p_i(t-\tau) + \xi(t) \end{cases} \tag{5.15}$$

where $u(t-\tau) = v_1(t-\tau) + h_{vi}(t-\tau)$ is the control signal of the i-th agent.

Following the work given in chapter 3, we will propose an observer for system (5.15) as follows:

$$\begin{cases} \dot{\hat{p}}_i(t) & = u(t) + K_1 \Lambda \Delta \eta(t) \\ \dot{\eta}(t) & = -\Lambda K_2 \eta(t) + \Lambda \left[\hat{z}(t) - \hat{p}_i(t)\right] \\ \hat{z}(t) & = y(t) + \int_{t-\tau}^{t} u(s)ds - \int_0^t \Lambda K_3 \left[\hat{z}(s) - \hat{p}_i(s)\right] ds \end{cases} \tag{5.16}$$

For the appropriate designed values of K_1, K_2 and K_3 based on theorem 3.3.1, one gets in the finite time the estimation $\hat{p}_i(t)$. It will converge to $p_i(t)$ with a very small error. Hence one can assume the following assumption:

Assumption 5.4.1. *Since the convergence of the observer (5.16) is achieved. One can assume that*

$$\hat{p}_i(t) = p_i(t) \tag{5.17}$$

Let us now define the formation control error as: $e_{Fi} = p_i(t) - p_1(t) - h_{pi}(t)$.

Theorem 5.4.2. *Consider the swarm of UAVs whose dynamic is (5.8), under the assumption 5.3.1,5.3.2,5.3.3,5.3.4,5.4.1. If the controller for the agent i^{th} is chosen such that:*

$$u_i(t) = K_F \left(\hat{p}_i(t) - p_1(t) - h_{pi}(t) + \int_{t-\tau}^{t} u_i(s)ds \right) + \dot{h}_{pi}(t) \tag{5.18}$$

Then the stability of the proposed controller will be globally ensured, with the controller error converging exponentially to the following bound:

$$\|e_{Fi}(t)\| \leq \sup_{c\in[t_0-\tau,\ t_0]} \|e_{Fi}(c)\|\, e^{-|K_F|(t-t_0)} - \frac{\epsilon_F}{K_F} \tag{5.19}$$

where ϵ_F is defined in (5.25), K_F is a negative controller gain.

Proof of theorem 5.4.2 :

From (5.8) one gets the first derivative of the formation control error:

$$\dot{e}_{Fi} = v_1(t-\tau) + u_i(t-\tau) - v_1(t) - \dot{h}_{pi}(t) \tag{5.20}$$

where $u_i(t) = h_{vi}(t)$ is the formation controller for the i^{th} agent. Using the Leibniz integration formula for (5.20), one gets:

$$\begin{aligned} e_{Fi}(t+\tau) &= \int_0^{t+\tau} \dot{e}_{Fi}(s)ds = e_{Fi}(t) + \int_{t-\tau}^{t} \dot{e}_{Fi}(s+\tau)ds \\ &= e_{Fi}(t) + \int_{t-\tau}^{t} u_i(s)ds + \int_t^{t+\tau} \left(v_1(s-\tau) - v_1(s) - \dot{h}_{pi}(s)\right) ds \end{aligned} \tag{5.21}$$

Now from the assumption 5.4.1, the control signal of the i^{th} agent (5.18) can be rewritten as follows:

$$\begin{aligned} u_i(t) &= K_F \left(e_{Fi}(t) + \int_{t-\tau}^{t} u_i(s)ds \right) + \dot{h}_{pi}(t) \\ &= K_F e_{Fi}(t+\tau) - K_F \int_t^{t+\tau} \left(v_1(s-\tau) - v_1(s) - \dot{h}_{pi}(s)\right) ds + \dot{h}_{pi}(t) \end{aligned} \tag{5.22}$$

Combining (5.20), (5.21) and (5.22) one gets:

$$\dot{e}_{Fi} = K_F e_{Fi} \underbrace{-K_F \int_{t-\tau}^{t} \left(v_1(s-\tau) - v_1(s) - \dot{h}_{pi}(s)\right) ds - \int_{t-\tau}^{t} \dot{v}_1(s) ds - \int_{t-\tau}^{t} \ddot{h}_{pi}(d) ds}_{f_{Fi}(t)} \tag{5.23}$$

From (5.23) we derive finally:

$$\|e_{Fi}(t)\| \leq \sup_{c \in [t_0-\tau,\, t_0]} \|e_{Fi}(c)\| \, e^{-|K_F|(t-t_0)} - \frac{\|f_{Fi}(t)\|}{K_F} \tag{5.24}$$

Based on assumptions 5.3.2,5.3.3,5.3.4, one can assume that:

$$\exists\ \epsilon_F \in \mathbb{R}^+,\ \sup_{t \geq 0} \|f_{Fi}(t)\| \leq \epsilon_F \tag{5.25}$$

From (5.24) and (5.25) we can obtain:

$$\|e_{Fi}(t)\| \leq \sup_{c \in [t_0-\tau,\, t_0]} \|e_{Fi}(c)\| \, e^{-|K_F|(t-t_0)} - \frac{\epsilon_F}{K_F} \tag{5.26}$$

This ends the proof of theorem 5.4.2.

Remark 5.4.1. *The controller proposed in theorem 5.4.2 can guarantee the formation control error converges to a value $f_{Fi}(t)$, whose bound is given in (5.23). It is worth noting that this value presents a significant improvement compared to that given in remark 5.3.1. Indeed, the controller proposed in this section can guarantee the convergence of control error e_{Fi} to zeros in some specific condition. For example, the formation requires only $\dot{h}_{pi}$ is constant and the leader is moving with a constant speed, namely, $v_1(t-\tau) - v_1(t) - \dot{h}_{pi}(t) = 0$, $\dot{v} = 0$ and $\dot{h}_{pi}(t) = 0$. Since the decentralised leader-follower strategy requires only the follower traces the leader, not vice versa, then maintain the constant speed for leader is acceptable. Then in this case, with the desired constant formation, the proposed controller can achieve the formation tracking task with control error converges to zeros.*

5.5 Simulation validation

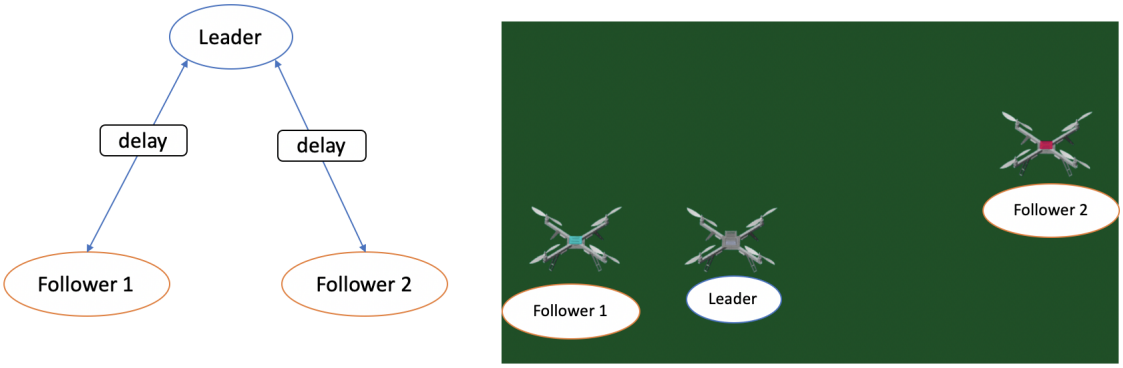


Figure 5.7: Simulation scenario for formation tracking validation. Communication strategy (left), Simscape-multibody 3D simulation platform (right).

In this section, the application of the proposed formation controller will be used to control a swarm of three quadrotors UAVs in the Simscape-multibody 3D simulation environment. The swarm of UAV is comprised of three Quadrotor UAV and will be controlled by decentralized leader-follower strategy. The communication strategy and UAV 3D model has been depicted in Figure. 5.7.

In this scenario, we assume that the leader can communicate with the followers, but the follower can only contact with the leader only. There exists a communication delay between UAVs equal to $\tau = 0.5s$.

One can refer to Figure. 5.8 to visualize the formation control structure. It is worth noting that in this chapter, the focus is on formation tracking, where the position and velocity controllers of each UAV are assumed to be ideal and integrated within the UAV itself.

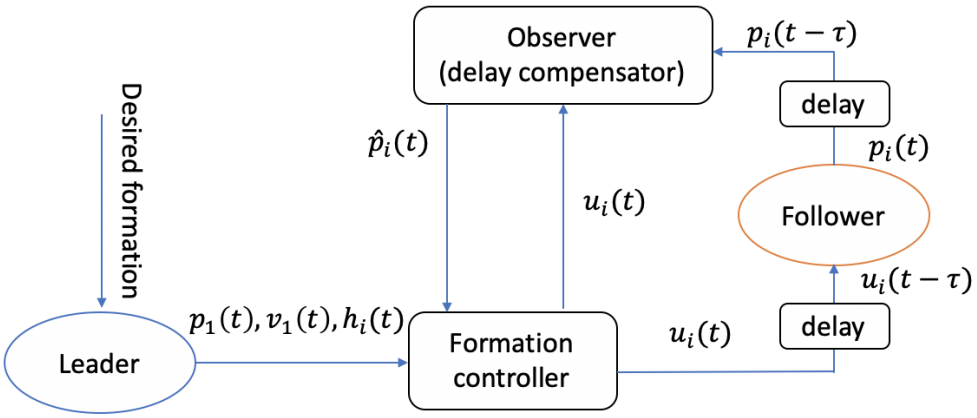


Figure 5.8: Control structure for formation tracking control

The control structure can be simply expressed as follows: The desired formation is firstly sent to the leader. The leader then will calculate the relative position for i^{th} follower , namely $h_i(t)$ based on its position $p_1(t)$ and velocity $v_1(t)$, using (5.2). These data then will be sent to the formation control block where the controller (5.18) will determine the appropriate control signal for each agent. Due to the delay communication effects, an observer for delay compensation is deployed for estimating the current position of follower, namely $\hat{p}_i(t)$.

In order to validate the formation tracking control performances of the proposed controller, the simulation will be carried out as follow:

- Take off: This is the first phase of the formation fly. All the UAVs in swarm will take off to $3m$

height, with constant speed $0.5m/s$. During the flight mode, this height will be maintained for all the agents in swarm. Then the formation can perform the tracking in the horizontal XY plane.

- Move to desired formation: In this step, two followers will move to the desired formation with a constant speed of $\dot{h}_{pi}(t) = 0.5m/s$. The leader will not move in this step. The formation is designed as follows: Three UAVs will be formed into an equilateral triangle shape with the sides of triangle equal to $4m$.

- Move the swarm to an other position: In this step, the leader will move to an other position with a constant speeds, namely $v_{1x} = 0.5m/s$ and $v_{1x} = 0.5m/s$. The followers will be controlled so that they follow the leader and maintain the formation form in equilateral triangle shape.

With these conditions, we expect that the proposed controller can control the swarm so that the formation tracking task is achieved with a tracking error which converges to zero as mentioned in remark 5.4.1.

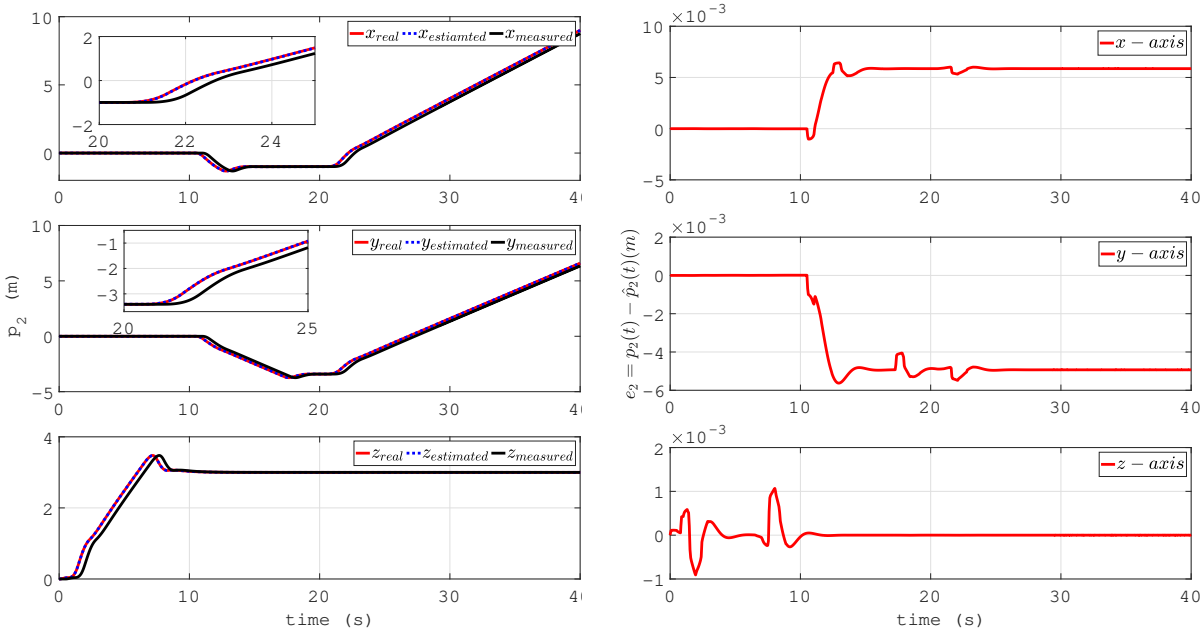
The initial position of each UAV in swarm is chosen as follows: (leader) $p_1(0) = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T (m)$, $p_2(0) = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T (m)$, $p_3(0) = \begin{bmatrix}3 & -1 & 0\end{bmatrix}^T (m)$.

In order to validate the advantages of using the proposed controller in this work. One shall present two simulations:

- 1: In the first simulation, the proposed controller will be used with the given control structure in Figure. 5.8.

- 2: In this simulation, the observer and the proposed controller will be removed. The feed back from the follower $(p_i(t))$ will be directly sent to leader, and the formation control signal $(p_1(t), v_1(t), h_i(t))$ will be sent directly to the follower.

Simulation 1 :



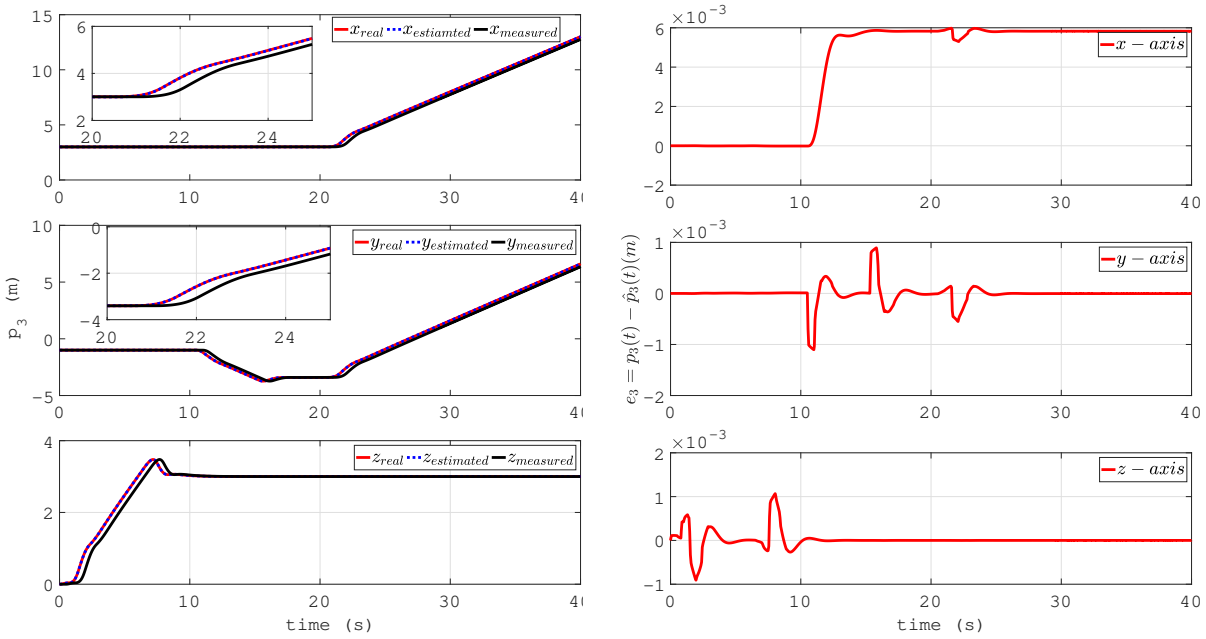
(a) Estimated state and its real measured values

(b) State estimation error

Figure 5.9: Simulation results for delay compensation for the follower 1 measurement

Figure. 5.9 and 5.10 illustrate the delay compensation performances by using the proposed state observer in this thesis for compensating the delay effects into the measurement of follower 1 and 2. As can be seen from these figures, the estimated values converge to their real values with very high speed and relatively small error. These highly precise estimation will be used to designed the formation control.

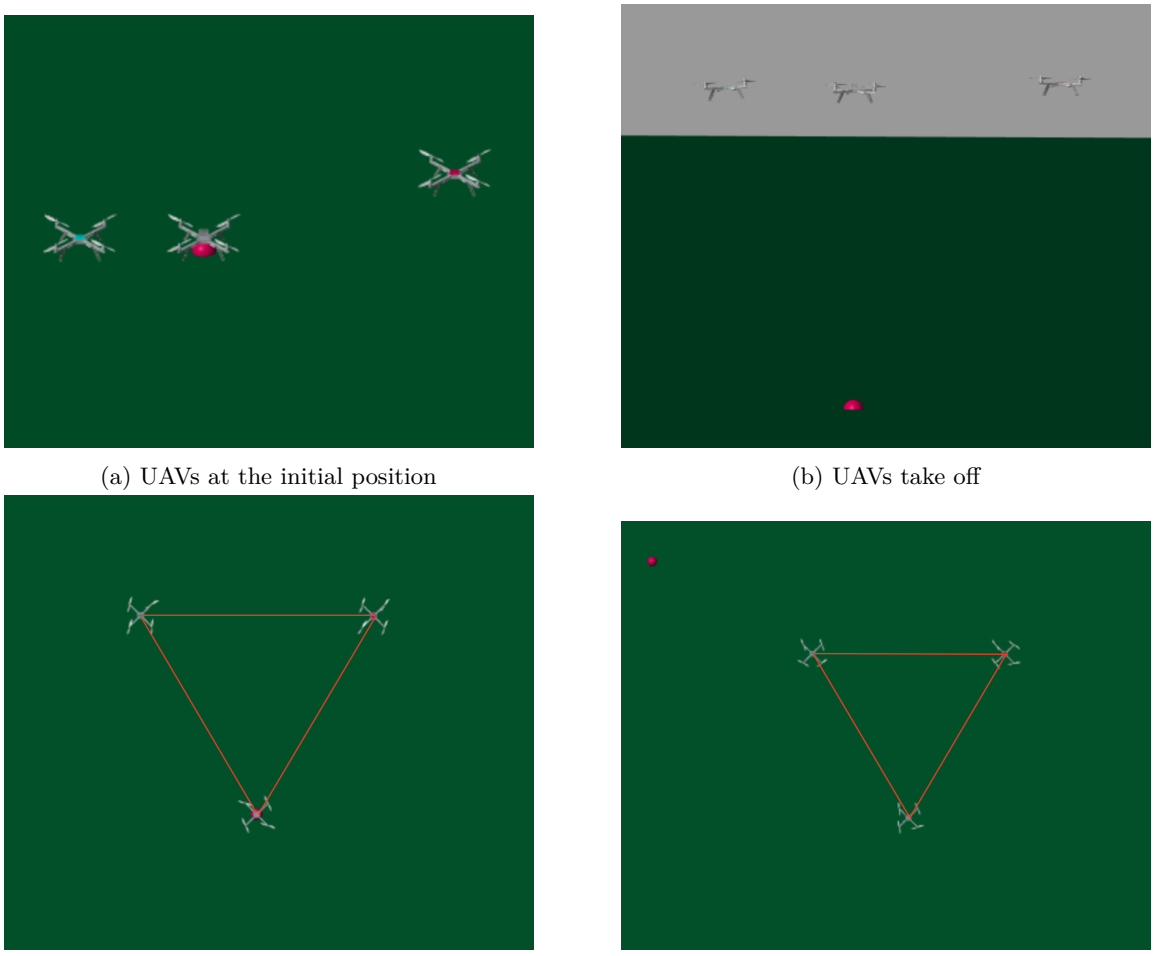
Figure. 5.11 visualises the formation tracking performances of the first simulation in 3D environment. We can see from this figure, the take off, the shape of the swarm (desired formation) and the movement of the formation are presented. The three UAVs have relatively formed into an equilateral triangle shape as can be seen form the figure. 5.11c. This desired shape is well preserved when the swarm is moving to an other position as can be seen from the figure. 5.11d.



(a) Estimated state and its real, measured values

(b) State estimation error

Figure 5.10: Simulation results for delay compensation for the measurement of follower 1



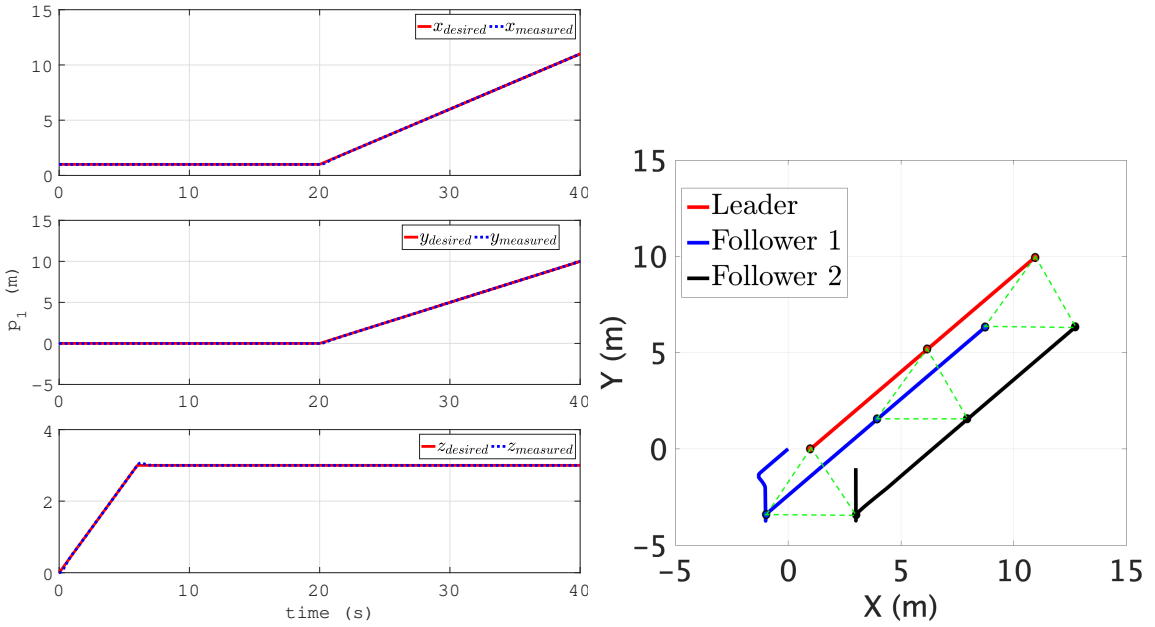
(a) UAVs at the initial position

(b) UAVs take off

(c) UAVs form an equilateral triangle shape

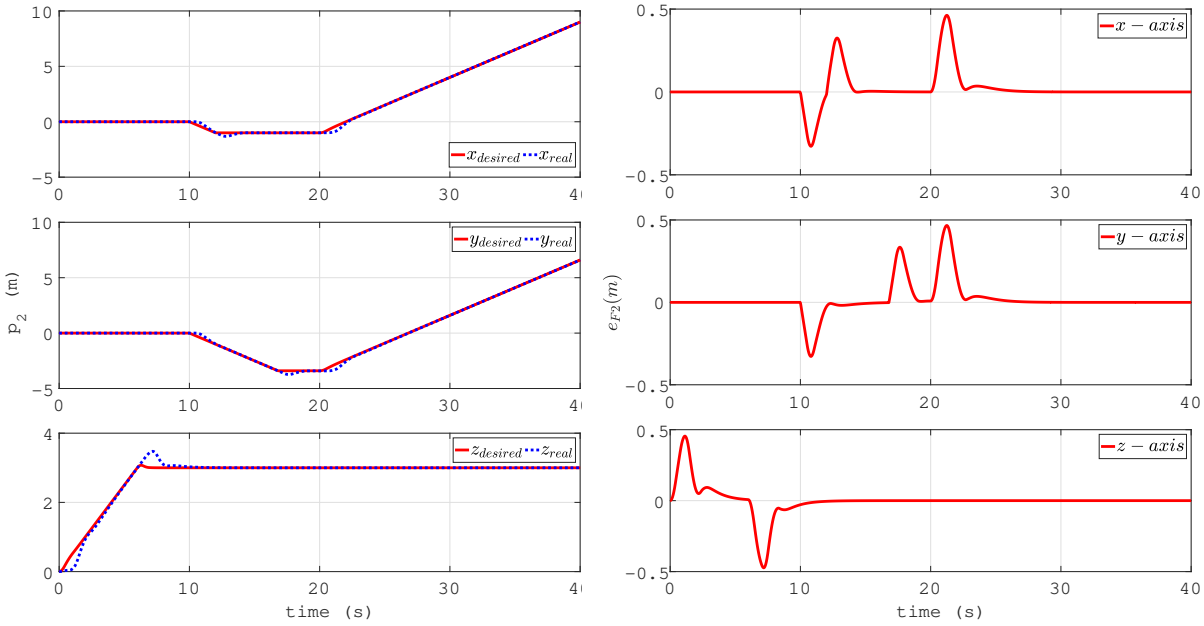
(d) Swarm is moving

Figure 5.11: Simulation results for formation tracking in 3D visualization (simulation 1)



(a) Position tracking of leader (b) Formation tracking of three UAVs

Figure 5.12: Simulation results for formation tracking of a swarm of three quadrotor UAVs (simulation 1)

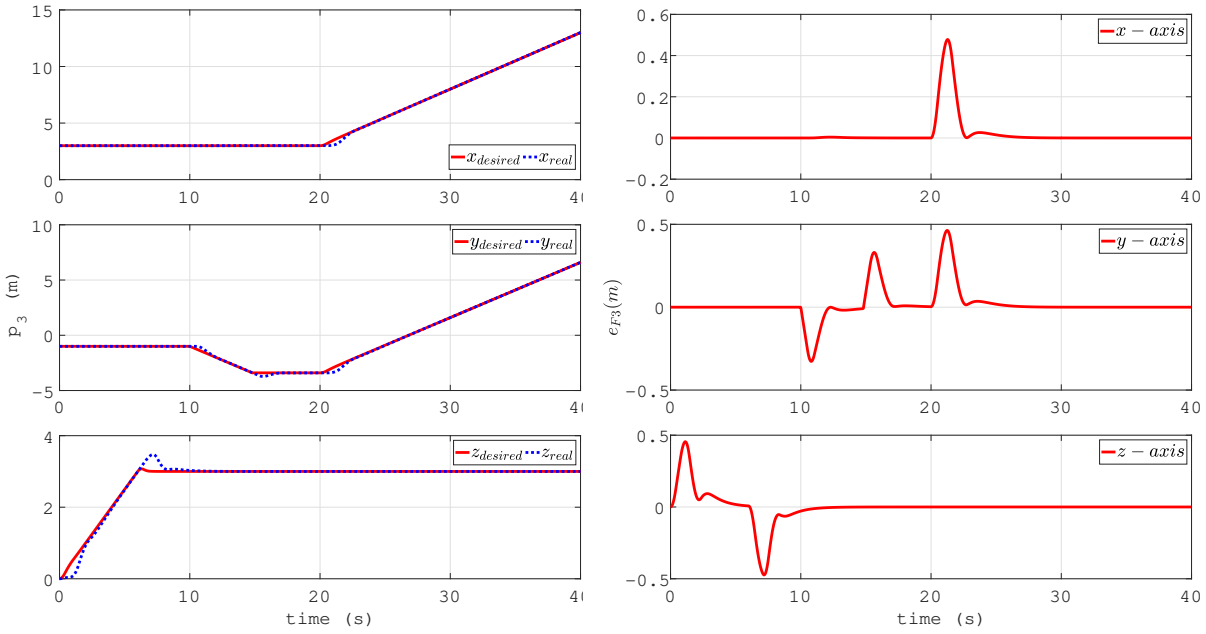


(a) Simulation results for formation tracking (b) Simulation results for formation tracking error

Figure 5.13: Simulation results for formation tracking of follower 1 (simulation 1)

Figure 5.12 demonstrates the position tracking performances of the leader and the formation tracking performances (Figure 5.12b) in 2D plan. It can be clearly seen that the leader quadrotors UAV has well achieved the position tracking for leading the follower. The swarm maintains an equilateral triangle formation.

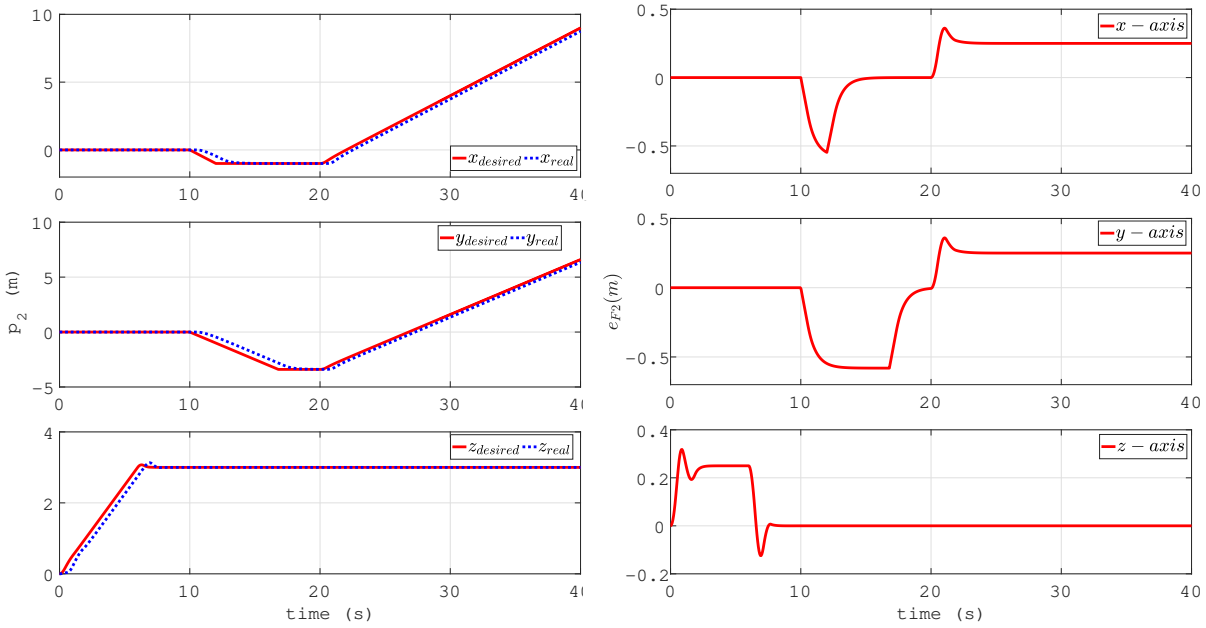
Figure 5.13 and 5.14 show the formation tracking performance of follower 1 and 2. It is worth noting that, the formation is achieved since the position of followers converge to their desired positions, namely $p_1(t) + h_{pi}(t)$. As can be seen from these figures, the proposed controller have well controlled the follower so that the formation tracking is well achieved. The formation tracking errors converge relatively to zero regardless the existent of delay communication between the UAVs. These results confirm the theoretical analyses and the statement given in remark 5.4.1.



(a) Simulation results for formation tracking (b) Simulation results for formation tracking error

Figure 5.14: Simulation results for formation tracking of follower 2 (simulation 1)

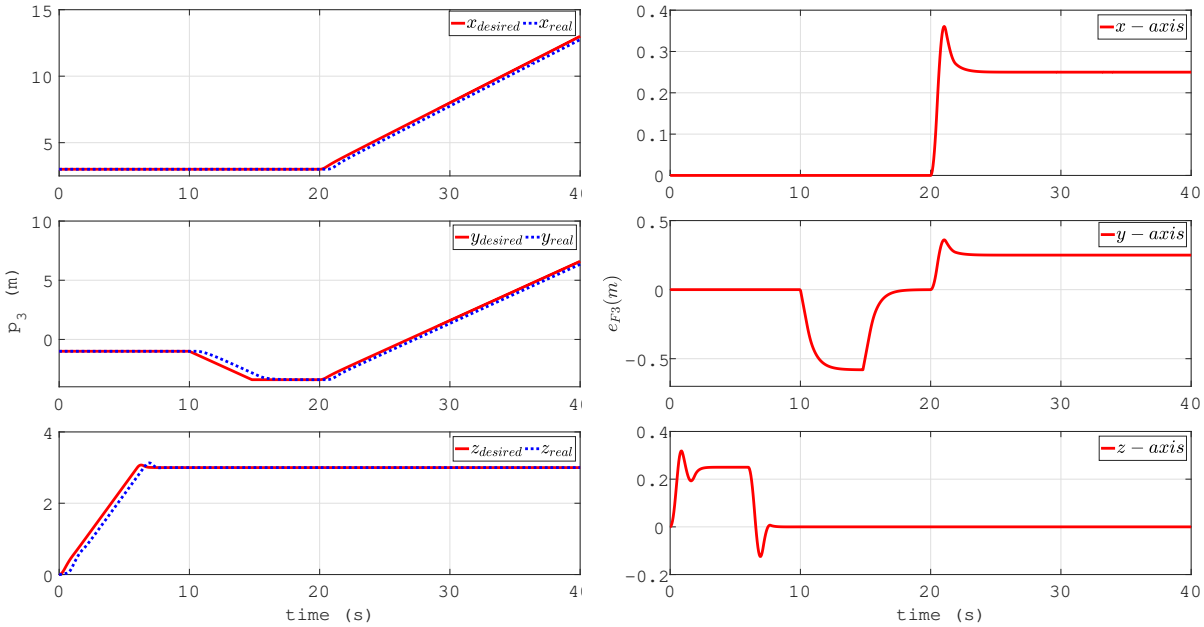
Simulation 2 :



(a) Simulation results for formation tracking (b) Simulation results for formation tracking error

Figure 5.15: Simulation results for formation tracking of follower 1 (simulation 2)

Figure 5.15 and 5.16 show the formation tracking performance of follower 1 and 2 without using the proposed controller and delay compensation. It is worth noting that, the communication delay is similar to the given in the first simulation. As can be seen from these figure, the delay communication effects are clearly shown in the tracking result. The follower can only track the delayed-desired position for the formation. Therefore the formation tracking can not be achieved. Figure 5.15b and 5.16b illustrate that, since the swarm is moving. The tracking errors can not converge to zero. These results confirm the statement given in remark 5.3.1.



(a) Simulation results for formation tracking

(b) Simulation results for formation tracking error

Figure 5.16: Simulation results for formation tracking of follower 2 for the simulation 2

5.6 Conclusion

In this chapter, we have presented a control system for multi-agent systems with communication delays. The main objective of the chapter was to develop a delay observer-based formation controller that can successfully handle formation tracking under specific conditions. Our proposed approach involved first implementing a delay observer in the leader agent to estimate the state of the followers in a gradual manner. Based on these predictions, we designed an adaptive formation controller that could effectively handle the formation tracking task.

The proposed approach was evaluated through a 3D simulation, and the results showed that it could successfully control the behavior of multiple agents under conditions of delay. Specifically, the simulation demonstrated the effectiveness of the proposed controller in ensuring successful formation tracking. We also showed that the stability of the proposed controller could be guaranteed, with the controller error converging exponentially to a given bound.

Overall, our study provides a novel approach to address the consensus problem in multi-agent systems with communication delays. By leveraging the concept of delay observer and an adaptive formation controller, we have developed a robust and effective control system that can ensure successful formation tracking even in the presence of delays. Future work could extend this approach to other multi-agent systems and explore the possibility of using different types of controllers to address different tasks.

Conclusion

In this thesis, we have presented a comprehensive study on the development of an observer and adaptive controllers for nonlinear systems with delayed measurement, specifically applied to the control of a UAV. The proposed observer and controller design methods incorporate RBF neural network, fuzzy logic, and back propagation neural network to improve the performance of the system.

The first part of the thesis is focused on developing an observer for nonlinear systems with delayed measurement. The objective was to design an observer that could estimate the states of the system accurately, even when the measurements were delayed. To accomplish this, the observer was designed to incorporate the delayed measurements and to use an adaptive sliding mode control. This observer has the advantage of being able to handle various types of nonlinear systems, and it can also cope with different types of delays. The proposed method was validated through simulations and experiments on a quadrotor UAV. The results demonstrated that the observer was effective in estimating the states of the system even in the presence of delay. Specifically, the observer was able to estimate the position and velocity of the UAV accurately, which is a critical component for control and navigation tasks. Moreover, in this chapter, we also addressed the problem of delay estimation under the condition that the value of delay is unknown.

In the second part of the thesis, we developed adaptive sliding mode controllers using RBF neural network and fuzzy logic for the single UAV system. The controllers were able to track the desired trajectory of the UAV while mitigating the effect of disturbances. In particular, the RBF neural network-based controller was able to deal with complex and uncertain nonlinear systems, while the fuzzy logic-based controller was able to handle imprecise or vague information. Then we designed a back propagation neural network for the control of a single UAV system. The proposed method was able to overcome the limitations of the traditional control methods by adapting the controller parameters online, which led to improved tracking performance and robustness against disturbances. The performance of the controllers was evaluated through simulations and experiment and compared to other existing control methods. The results showed that the proposed controllers were effective in controlling the UAV system and outperformed other existing control methods.

Finally, in the last chapter, we developed an adaptive controller for a swarm of UAVs affected by delay communication between the agents based on the observer and controllers developed in the previous chapters. The proposed method was able to coordinate the motion of the swarm while maintaining the desired formation, and it was shown to outperform other existing swarm control methods. The proposed method used the observer to estimate the state of the swarm, and the adaptive controllers to control the motion of the swarm. The adaptive controller was designed to deal with the uncertainties and disturbances in the system and to ensure the convergence of the formation error to zero. The simulation results showed that the proposed method was effective in controlling the swarm under delay communication and maintaining the desired formation.

Overall, the results of this thesis demonstrate the effectiveness of the proposed observer and control design methods for nonlinear systems with delayed measurement, as well as their potential for applications in UAV control and swarm coordination. However, there are still some areas that need further research.

One such area is the application of the proposed observer to different objects, beyond those considered in this thesis. Additionally, it is important to explore the effects of various types of delays, such as time-varying delays and uncertainties, on the performance of both the observer and controller.

Moreover, the application of unknown delay estimation techniques to quadrotor UAV systems and swarm systems of UAVs presents an intriguing avenue for future research. Furthermore, there is a need

for the development of more advanced and efficient control methods that can handle the effects of delays on quadrotor UAVs and multi-agent systems. These methods should aim to operate under less restrictive conditions than the existing approaches.

In conclusion, further investigations are needed to address these research gaps and enhance the understanding and applicability of observer and control design methods in systems with delayed measurement, specifically in the context of UAV control and swarm coordination.

CHAPTER 6

Source code for LMI solver

6.1 Source code for LMI (3.27)

Let us first recall the LMI (3.27) as follows:

$$\Phi_1 = \begin{bmatrix} -Q_x + \frac{L_\varphi P_x}{\Lambda} + \mu_1 I_p & 0_p & \frac{C^T K_3}{\Lambda^l} + 2\frac{L_\varphi}{\Lambda^l} C^T \\ * & -Q_\eta + \mu_1 I_p & -\Lambda^l P_\eta C^T \\ * & * & -2K_3 + \left(1 + \frac{L_\varphi}{\Lambda}\right)^2 \frac{I_n}{\Lambda^{l-1}} \end{bmatrix} \leq -\mu I_{2p+n}$$

The source for solving this LMI is given as following:
Step 1: Declare the matrices A,B,C :

```
A= [zeros(3,3) eye(3);
    zeros(3,3) zeros(3,3)]
B= [zeros(3,3);
    eye(3)]
C= [eye(3) zeros(3)];
```

Step 2: Determine the conditions for μ, μ_τ :

```
lambda=20;
mu=1;
Lph=3;
l=0.5;
tau=0.011;
factor=1;


alpha=lambda^(2-l)*factor^2+lambda*mu/tau;

mu1= tau^2*alpha/lambda;
```

From these calculated value, one shall chosen the value for solving the LMI as follow:

```
mu1=0.05;
mu_h=0.1;
mu=1;
```

where $Lph = L_\varphi$, $lambda = \Lambda$, $tau = \tau$, $mu1 = \mu_1$, $mu = \mu$.
Step 3: Solve the LMI using LMI tool in matlab :

```
setlmis([])
Qx= lmivar(1,[size(A,1) 1]);
Qeta= lmivar(1,[size(A,1) 1]);
```

```
Px = lmivar(1,[size(A,1) 1]);
Peta= lmivar(1,[size(A,1) 1]);
K3= lmivar(1,[size(C,1) 1]);

lmiterm([-1 1 1 Qx],1,1); %Qx>0
lmiterm([-2 1 1 Qeta],1,1); %Qeta>0
lmiterm([-3 1 1 Px],1,1); %Px>0
lmiterm([-4 1 1 Peta],1,1); %Peta>0

lmiterm([-6 1 1 K3],1,1); %K3>0


lmiterm([11 1 1 Qx],-1,1, 's');
lmiterm([11 1 1 Px],Lph/lambda,1);
lmiterm([11 1 1 0],mu1*eye(size(A,1)));
lmiterm([11 1 1 0],mu*eye(size(A,1)));

lmiterm([11 1 2 0],0*eye(size(A,1)));

lmiterm([11 1 3 0],2*C'*lambda^(-1)*Lph);
lmiterm([11 1 3 K3],C'*lambda^(-1),1);
lmiterm([11 1 3 0],(Lph)*C'*lambda^(-1-1));


lmiterm([11 2 2 Qeta],-1,1);
lmiterm([11 2 2 0],mu1*eye(size(A,1)));
lmiterm([11 2 2 0],mu*eye(size(A,1)));

lmiterm([11 2 3 Peta],-lambda^(1),C');

lmiterm([11 3 3 K3],-2,1);
lmiterm([11 3 3 0],(1+Lph/lambda)^2/(lambda(1-1)));
lmiterm([11 3 3 0],mu*eye(size(C,1)));



LMIs_theo1 = getlmis;
[TMIN_theo1, XFEAS_theo1]= feasp(LMIs_theo1 );

Qx= dec2mat(LMIs_theo1,XFEAS_theo1,Qx)
Px= dec2mat(LMIs_theo1,XFEAS_theo1,Px)

Qeta= dec2mat(LMIs_theo1,XFEAS_theo1,Qeta)
Peta= dec2mat(LMIs_theo1,XFEAS_theo1,Peta)

K3= dec2mat(LMIs_theo1,XFEAS_theo1,K3)
```

6.2 Source code for LMI (3.54)

Let us first recall the LMI (3.54) as follows:

$$\Phi_2 = \begin{bmatrix} -Q_x + \frac{2L_\varphi P_x}{\Lambda} + \frac{R}{\Lambda} + \mu_\tau I_p & 0_p & \frac{A^T C^T + C^T K_3}{\Lambda^l} + \frac{L_\varphi}{\Lambda^{l+1}} C^T & 0_p \\ * & -Q_\eta + \mu_\tau I_p & -\Lambda^l P_\eta C^T & 0_p \\ * & * & -2K_3 + \frac{K_4}{\Lambda^{l-1}} & \frac{K_4 C}{\Lambda^l} \\ * & * & * & -\frac{R}{\Lambda} + \frac{\mu_h}{\Lambda} I_p \end{bmatrix} \leq -\mu I_{4p}$$

$K_4 = k_4 I_n;\ \Lambda k_4 = 1.$

The source for solving this LMI is given as following:

Step 1: Declare the matrices A,B,C :

```
A= [zeros(3,3) eye(3);
    zeros(3,3) zeros(3,3)]
B= [zeros(3,3);
    eye(3)]
C= [eye(3) zeros(3)];
```

Step 2: Determine the conditions for μ, μ_τ and μ_h :

```
lambda=20;
mu=1;
Lph=3;
tau=0.011
factor=1;
k4 = 1/lambda;
h=0.005;
beta= h*lambda^2*k4*(1+k4+Lph/lambda)^2+lambda*mu
muh= beta*h

rmax=2*(muh+lambda*mu);
alpha=2*rmax*lambda*factor+lambda*mu
mutau= alpha*tau/lambda
```

From these calculated value, one shall chosen the value for solving the LMI as follow:

```
mu_tau=0.9;
mu_h=0.1;
mu=1;
l=0.5;
```

where $Lph = L_\varphi$, $lambda = \Lambda$, $tau = \tau$, $mu\_tau = \mu_\tau$, $mu = \mu$, $mu\_h = \mu_h$

Step 3: Solve the LMI using LMI tool in matlab :

```
setlmis([])
Qx= lmivar(1,[size(A,1) 1]);
Qeta= lmivar(1,[size(A,1) 1]);
Px = lmivar(1,[size(A,1) 1]);
Peta= lmivar(1,[size(A,1) 1]);
R= lmivar(1,[size(A,1) 1]);
K3= lmivar(1,[size(C,1) 1]);

lmiterm([-1 1 1 Qx],1,1); %Qx>0
lmiterm([-2 1 1 Qeta],1,1); %Qeta>0
lmiterm([-3 1 1 Px],1,1); %Px>0
lmiterm([-4 1 1 Peta],1,1); %Peta>0
lmiterm([-5 1 1 R],1,1); %R>0

lmiterm([-6 1 1 K3],1,1); %K3>0


lmiterm([11 1 1 Qx],-1,1, 's');
lmiterm([11 1 1 Px],Lph/lambda,2);
lmiterm([11 1 1 R],1,1/lambda);
lmiterm([11 1 1 0],mu_tau*eye(size(A,1)));
lmiterm([11 1 1 0],mu*eye(size(A,1)));

lmiterm([11 1 2 0],0*eye(size(A,1)));

lmiterm([11 1 3 0],A'*C'*lambda^(-l));
lmiterm([11 1 3 K3],C'*lambda^(-l),1);
lmiterm([11 1 3 0],(Lph)*C'*lambda^(-l-1));

lmiterm([11 1 4 0],zeros(size(A,1),size(A,1)));
```

```
lmiterm([11 2 2 Qeta],-1,1);
lmiterm([11 2 2 0],mu_tau*eye(size(A,1)));
lmiterm([11 2 2 0],mu*eye(size(A,1)));

lmiterm([11 2 3 Peta],-lambda^(l),C');

lmiterm([11 2 4 0],zeros(size(A,1),size(A,1)));

lmiterm([11 3 3 K3],-2,1);
lmiterm([11 3 3 0],lambda^(-l-2));
lmiterm([11 3 3 0],mu*eye(size(C,1)));

lmiterm([11 3 4 0],C*lambda^(-1-l));

lmiterm([11 4 4 R],-1,lambda^(-1));
lmiterm([11 4 4 0],mu_h*lambda^(-1)*eye(size(A,1)));
lmiterm([11 4 4 0],mu*eye(size(A,1)));

LMIs_theo12 = getlmis;
[TMIN_theo12, XFEAS_theo12]= feasp(LMIs_theo12 );

Qx= dec2mat(LMIs_theo12,XFEAS_theo12,Qx)
Px= dec2mat(LMIs_theo12,XFEAS_theo12,Px)

Qeta= dec2mat(LMIs_theo12,XFEAS_theo12,Qeta)
Peta= dec2mat(LMIs_theo12,XFEAS_theo12,Peta)

R= dec2mat(LMIs_theo12,XFEAS_theo12,R)

K3= dec2mat(LMIs_theo12,XFEAS_theo12,K3)
```

CHAPTER 7

List of Publications

7.1 Journal Publications

- Q. T. Dam, R. E. H. Thabet, S. Ahmed Ali and F. Guérin, "Delay Estimation for Nonlinear Systems With Unknown Output Delay," in IEEE Control Systems Letters, vol. 7, pp. 1219-1224, 2023, doi: 10.1109/LCSYS.2022.3232407.

- Q. T. Dam, R. E. H. Thabet, S. A. Ali and F. Guerin, "Observer design for a class of uncertain nonlinear systems with sampled-delayed output using High-Gain Observer and low-pass filter: Application for a quadrotor UAV," in IEEE Transactions on Industrial Electronics, doi: 10.1109/TIE.2023.3247786.

7.2 Conference Publications

- Q. T. Dam, R. El Houda Thabet, S. A. Ali, F. Guérin and A. Hugo, "Continuous–Discrete Time High Gain Observer Design for State and Unknown Inputs Estimations Of Quadrotor UAV," 2021 European Control Conference (ECC), 2021, pp. 1181-1186, doi: 10.23919/ECC54610.2021.9655232.

- Q. T. Dam, R. El Houda Thabet, S. A. Ali and F. Guérin, "Filtered High-Gain Observer Design For a Class of Nonlinear Systems subject to Delayed Measurements: Application to a Quadrotor UAV," 2022 American Control Conference (ACC), 2022, pp. 4050-4055, doi: 10.23919/ACC53348.2022.9867537.

- Q. T. Dam, R. El Houda Thabet, S. A. Ali and F. Guérin, "Time-Delay estimation for a drift-observable nonlinear system with unknown output delay," American Control Conference (ACC), 2023. Accepted.

- Q. T. Dam, R. El Houda Thabet, S. A. Ali and F. Guérin, "A High-gain Observer Design for Nonlinear System with Delayed Measurements: Application to A Quadrotor UAV" 22nd IFAC World Congress, 2023. Accepted.

7.3 Paper in preparation

- Q. T. Dam, R. El Houda Thabet, S. A. Ali, F. Guérin and A. Hugo, "Filtered Adaptive High gain observer for a class of nonlinear systems with sampled-delayed outputs: Application to attitude estimation of a rigid body and state estimation of a quadrotor UAV.

- Q. T. Dam, R. El Houda Thabet, S. A. Ali, F. Guérin and Y. Tang "Adaptive Neural Network-Based Sliding Mode Controller For Trajectory Tracking of A Quadrotor UAV.

- Q. T. Dam, R. El Houda Thabet, S. A. Ali, F. Guérin and Y. Tang "Adaptive Sliding mode Control for a Quadrotor UAV based on Radial Basis Function Neural Network and Fuzzy Logic.

Bibliography

A. E. Rodríguez-Mata G. Flores, A. H. Martínez-Vásquez-Z. D. Mora-Felix R. Castro-Linares and L. E. Amabilis-Sosa (Sept. 2018). "Discontinuous High-Gain Observer in a Robust Control UAV Quadrotor: Real-Time Application for Watershed Monitoring". In: *Mathematical Problems in Engineering*. URL: http://doi.org/10.1155/2018/4940360.

Abaunza, H., P. Castillo, and R. Lozano (2018). "Quaternion Modeling and Control Approaches". In: *Handbook of Unmanned Aerial Vehicles*. Ed. by Kimon P. Valavanis and George J. Vachtsevanos. Cham: Springer International Publishing, pp. 1–29. ISBN: 978-3-319-32193-6. DOI: 10.1007/978-3-319-32193-6_179-1. URL: https://doi.org/10.1007/978-3-319-32193-6_179-1.

Abdelkader, Mohamed et al. (2014). "Optimal multi-agent path planning for fast inverse modeling in UAV-based flood sensing applications". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 64–71. DOI: 10.1109/ICUAS.2014.6842239.

Abhinav, Kumar and S.J Mija (2021). "Observer Based Sliding Mode Control for 3 DOF Helicopter System". In: *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1–4. DOI: 10.1109/CONIT51480.2021.9498495.

Abraham Efraim Rodríguez-Mata Ivan González-Hernández, Jesus Gabriel Rangel-Peraza Sergio Salazar and Rogelio Lozano Leal (Sept. 2018). "Wind-gust Compensation Algorithm based on High-gain Residual Observer to Control a Quadrotor Aircraft: Real-time Verification Task at Fixed Point". In: *International Journal of Control, Automation and Systems*. URL: http://doi.org/10.1007/s12555-016-0771-6.

Acevedo, J. J. et al. (2013). "Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications". In: *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 565–574. DOI: 10.1109/ICUAS.2013.6564734.

Adil, A. et al. (2020a). "High-Gain Observer Design for Nonlinear Systems with Delayed Outputs". In: *IFAC-PapersOnLine* 53.2. 21th IFAC World Congress, pp. 5057–5062. ISSN: 2405-8963.

Adil, A. et al. (2020b). "High-Gain Observer Design for Nonlinear Systems with Delayed Outputs". In: *IFAC-PapersOnLine* 53.2. 21th IFAC World Congress, pp. 5057–5062. ISSN: 2405-8963.

Ahmadzadeh, Ali et al. (2006). "Multi-UAV Cooperative Surveillance with Spatio-Temporal Specifications". In: *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5293–5298. DOI: 10.1109/CDC.2006.377157.

Ahmed-Ali, Tarek et al. (2019). "Observer design for a class of parabolic systems with large delays and sampled measurements". In: *IEEE Transactions on Automatic Control*.

Ajmera, Juhi and V. Sankaranarayanan (2016). "Point-to-Point Control of a Quadrotor: Theory and Experiment". In: *IFAC-PapersOnLine* 49.1. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016, pp. 401–406. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.03.087. URL: https://www.sciencedirect.com/science/article/pii/S2405896316300878.

Ali, Arslan et al. (2021). "A Decentralized Pattern Formation and Navigation Approach For Swarms: Multi-UAV Perspective". In: *2021 International Conference on Robotics and Automation in Industry (ICRAI)*, pp. 1–6. DOI: 10.1109/ICRAI54018.2021.9651460.

Ali, Sofiane Ahmed and N. Langlois (May 2017). "Sampled data's observer design with time varying correction gain for Electro Hydraulic Actuator Systems". In: *American Control Conference*, pp. 3276–3281. URL: https://doi.org/10.23919/ACC.2017.7963452.

Ali, Sofiane Ahmed et al. (2014). "Sampled-data disturbance observer for a class of nonlinear systems". In: *19th World Congress The International Federation of Automatic Control*, pp. 3346–3351.

Ali, Sofiane Ahmed et al. (2016a). "c32ontinuous–Discrete Time-Observer Design for State and Disturbance Estimation of Electro-Hydraulic Actuator Systems". In: *IEEE Transactions on Industrial Electronics*.

— (2016b). "Continuous–Discrete Time-Observer Design for State and Disturbance Estimation of Electro-Hydraulic Actuator Systems". In: *IEEE Transactions on Industrial Electronics* 32, pp. 975–994.

Alqaisi, Walid Kh. et al. (2019). "Adaptive Sliding mode Control Based on RBF Neural Network Approximation for Quadrotor". In: *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pp. 1–7. DOI: 10.1109/ROSE.2019.8790423.

Andrade, Fabio A. A. et al. (2022). "Unmanned Aerial Vehicles Motion Control with Fuzzy Tuning of Cascaded-PID Gains". In: *Machines* 10.1. ISSN: 2075-1702. DOI: 10.3390/machines10010012. URL: https://www.mdpi.com/2075-1702/10/1/12.

Anguelova, Milena and Bernt Wennberg (2008). "State elimination and identifiability of the delay parameter for nonlinear time-delay systems". In: *Automatica* 44.5, pp. 1373–1378. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2007.10.013. URL: https://www.sciencedirect.com/science/article/pii/S0005109807004529.

Araar, Oualid and Nabil Aouf (2014). "Quadrotor control for trajectory tracking in presence of wind disturbances". In: *2014 UKACC International Conference on Control (CONTROL)*, pp. 25–30. DOI: 10.1109/CONTROL.2014.6915110.

Assche, V. Van et al. (2011a). "High gain observer design for nonlinear systems with time varying delayed measurements". In: *IFAC Proceedings Volumes* 44.1. 18th IFAC World Congress, pp. 692–696. ISSN: 1474-6670.

— (2011b). "High gain observer design for nonlinear systems with time varying delayed measurements". In: *IFAC Proceedings Volumes* 44.1. 18th IFAC World Congress, pp. 692–696. ISSN: 1474-6670.

Astolfi, D. and L. Marconi (2015). "A high-gain nonlinear observer with limited gain power". In: *IEEE Transactions on Automatic Control* 60.11, pp. 3059–3064.

Astolfi, Daniele, Marc Jungers, and Luca Zaccarian (2018). "Output Injection Filtering Redesign in High-Gain Observers". In: *2018 European Control Conference (ECC)*, pp. 1957–1962.

Astolfi, Daniele, Luca Zaccarian, and Marc Jungers (2021). "On the use of low-pass filters in high-gain observers". In: *Systems Control Letters* 148, p. 104856.

Astolfi, Daniele et al. (2016). "Sensitivity to High-Frequency Measurement Noise of Nonlinear High-Gain Observers". In: *IFAC-PapersOnLine* 49.18. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016, pp. 862–866. ISSN: 2405-8963.

Aïda Feddaoui Nicolas Boizot, Eric Busvelle-Vincent Hugel (2018). "High-gain extended Kalman filter for continuous-discrete systems with asynchronous measurements". In: *International Journal of Control,* URL: https://doi.org/10.1080/00207179.2018.1539525.

Ball, Alexis A. and Hassan K. Khalil (Dec. 2008). "High-Gain Observers in the Presence of Measurement Noise: A Nonlinear Gain Approach". In: *IEEE Conference on Decision and Control*. URL: https://doi.org/10.1109/CDC.2008.4739408.

Belkoura, L. et al. (2004). "Identifiability and identification of linear systems with delays". In: *In Advances in Time-delay Systems, LNCSE, Springer*. 38, 123–136.

Boizot N Busvelle E, Gauthier JP-Sachau J (2007). "Aplication to a series-connected DC motor". In: *Conference on Systems and Control*. URL: https://www.researchgate.net/publication/228552188_Adaptive_Gain_Extended_Kalman_Filter_Application_to_a_Series-connected_DC_Motor.

Borrelli, F., T. Keviczky, and G.J. Balas (2004). "Collision-free UAV formation flight using decentralized optimization and invariant sets". In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Vol. 1, 1099–1104 Vol.1. DOI: 10.1109/CDC.2004.1428839.

Briat, C., O. Sename, and J.-F. Lafay (2011). "Design of LPV observers for LPV time-delay systems: an algebraic approach". In: *International Journal of Control* 84.9, pp. 1533–1542.

Broek, T. Avan den et al. (2011). "A virtual structure approach to formation control of unicycle mobile robots using mutual coupling". In: *International Journal of Control* 84, pp. 1886–1902.

BUSVELLE, ERIC and JEAN-PAUL GAUTHIER (Feb. 2005). "Observation and identification tools for non-linear systems: application to a fluid catalytic cracker". In: *International Journal of Control* 78. URL: https://doi.org/10.1080/00207170500036027.

Cacace, F., A. Germani, and C. Manes (2010a). "An observer for a class of nonlinear systems with time varying observation delay". In: *Systems Control Letters* 59.5, pp. 305–312. ISSN: 0167-6911.

— (2010b). "An observer for a class of nonlinear systems with time varying observation delay". In: *Systems Control Letters* 59.5, pp. 305–312. ISSN: 0167-6911.

Cacace, F. et al. (2016). "Delay identification for a class of nonlinear systems". In: *International Journal of Control* 89.11, pp. 2350–2359.

Cacace, Filippo et al. (2017). "Joint State Estimation and Delay Identification for Nonlinear Systems With Delayed Measurements". In: *IEEE Transactions on Automatic Control* 62.9, pp. 4848–4854. DOI: 10.1109/TAC.2017.2689503.

CAI, Zhihao et al. (2020). "Virtual target guidance-based distributed model predictive control for formation control of multiple UAVs". In: *Chinese Journal of Aeronautics* 33.3, pp. 1037–1056. ISSN: 1000-9361. DOI: https://doi.org/10.1016/j.cja.2019.07.016. URL: https://www.sciencedirect.com/science/article/pii/S1000936119302833.

Chen, Fuyang et al. (2016). "Robust Backstepping Sliding-Mode Control and Observer-Based Fault Estimation for a Quadrotor UAV". In: *IEEE Transactions on Industrial Electronics* 63.8, pp. 5044–5056. DOI: 10.1109/TIE.2016.2552151.

Cheng, Xing and Zhi-Wei Liu (2019). "Robust Tracking Control of A Quadcopter Via Terminal Sliding Mode Control Based on Finite-time Disturbance Observer". In: *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1217–1222. DOI: 10.1109/ICIEA.2019.8834345.

Cowling, Ian D. et al. (2007). "A prototype of an autonomous controller for a quadrotor UAV". In: *2007 European Control Conference (ECC)*, pp. 4001–4008. DOI: 10.23919/ECC.2007.7068316.

Cuny, Fabien et al. (2020). "Sampled-data observer design for delayed output-injection state-affine systems". In: *International Journal of Control* 93.12, pp. 2949–2959.

Dam, Quang Truc et al. (2021). "Continuous–Discrete Time High Gain Observer Design for State and Unknown Inputs Estimations Of Quadrotor UAV". In: *2021 European Control Conference (ECC)*, pp. 1181–1186. DOI: 10.23919/ECC54610.2021.9655232.

Dam, Quang Truc et al. (2022). "Filtered High-Gain Observer Design For a Class of Nonlinear Systems subject to Delayed Measurements: Application to a Quadrotor UAV". In: *2022 American Control Conference (ACC)*, pp. 4050–4055. DOI: 10.23919/ACC53348.2022.9867537.

Dam, Quang Truc et al. (2023a). "Delay Estimation for Nonlinear Systems With Unknown Output Delay". In: *IEEE Control Systems Letters* 7, pp. 1219–1224. DOI: 10.1109/LCSYS.2022.3232407.

Dam, Quang Truc et al. (2023b). "Observer design for a class of uncertain nonlinear systems with sampled-delayed output using High-Gain Observer and low-pass filter: Application for a quadrotor UAV". In: *IEEE Transactions on Industrial Electronics*, pp. 1–10. DOI: 10.1109/TIE.2023.3247786.

Deng, Yang et al. (2021). "Predictor-Based Control of LTI Remote Systems With Estimated Time-Varying Delays". In: *IEEE Control Systems Letters* 5.1, pp. 289–294. DOI: 10.1109/LCSYS.2020.3001671.

Ding, Xu Chu, Amir R. Rahmani, and Magnus Egerstedt (2010). "Multi-UAV Convoy Protection: An Optimal Approach to Path Planning and Coordination". In: *IEEE Transactions on Robotics* 26.2, pp. 256–268. DOI: 10.1109/TRO.2010.2042325.

Dong, Xiwang et al. (2014). "Formation Control for High-Order Linear Time-Invariant Multiagent Systems With Time Delays". In: *IEEE Transactions on Control of Network Systems* 1.3, pp. 232–240. DOI: 10.1109/TCNS.2014.2337972.

Dong, Xiwang et al. (2015). "Time-Varying Formation Control for Unmanned Aerial Vehicles: Theories and Applications". In: *IEEE Transactions on Control Systems Technology* 23.1, pp. 340–348. DOI: 10.1109/TCST.2014.2314460.

Dong, Xiwang et al. (2017). "Time-Varying Formation Tracking for Second-Order Multi-Agent Systems Subjected to Switching Topologies With Application to Quadrotor Formation Flying". In: *IEEE Transactions on Industrial Electronics* 64.6, pp. 5014–5024. DOI: 10.1109/TIE.2016.2593656.

Dong, Xiwang et al. (2019). "Theory and Experiment on Formation-Containment Control of Multiple Multirotor Unmanned Aerial Vehicle Systems". In: *IEEE Transactions on Automation Science and Engineering* 16.1, pp. 229–240. DOI: 10.1109/TASE.2018.2792327.

Dong J, He B (2018). "Novel Fuzzy PID-Type Iterative Learning Control for Quadrotor UAV". In: *Sensors* 19.1, pp. 401–406.

Doukhi, Oualid and Deok Jin Lee (2019). "Neural Network-based Robust Adaptive Certainty Equivalent Controller for Quadrotor UAV with Unknown Disturbances". In: *International Journal of Control, Automation and Systems* 17, 2365–2374.

Doyle, J. C. and G. Stein (1978). "Robustness with observers". In: *IEEE Conference on Decision and Control*. URL: https://doi.org/10.1109/cdc.1978.267883.

Drakunov, S. et al. (2006). "Delay identification in time-delay systems using variable structure observers". In: *Annual Reviews in Control* 30.2, 143–158.

Eltayeb, Ahmed et al. (2019). "Adaptive Fuzzy Gain Scheduling Sliding Mode Control for quadrotor UAV systems". In: *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, pp. 1–5. DOI: 10.1109/ICMSAO.2019.8880438.

Eskandarpour, Abolfazl and Inna Sharf (2020). "A constrained error-based MPC for path following of quadrotor with stability analysis". In: *Nonlinear Dynamics* 99, 899–918. DOI: 10.1007/s11071-019-04859-0.

F. Deza E. Busvelle, J.P. Gauthier and J. Rako-topara (Apr. 1992). "High gain estimation for nonlinear systems". In: *Systems and Control Letters* 18, pp. 295 –299.

Falcón, Romeo, Héctor Ríos, and Alejandro Dzul (2019). "Comparative analysis of continuous sliding-modes control strategies for quad-rotor robust tracking". In: *Control Engineering Practice* 90, pp. 241–256. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2019.06.013. URL: https://www.sciencedirect.com/science/article/pii/S0967066119300875.

Farza, Mondher et al. (2018). "Cascade observer design for a class of uncertain nonlinear systems with delayed outputs". In: *Automatica* 89, pp. 125–134. ISSN: 0005-1098.

Fridman, Emilia (2014a). In: *Introduction to Time-Delay Systems*.

— (2014b). In: *Introduction to Time-Delay Systems*.

Gauthier, J. P., H. Hammouri, and S. Othman (1992a). "A simple observer for nonlinear systems applications to bioreactors". In: *IEEE Transactions on Automatic Control* 37.6, 875–880.

Gauthier, J. P. and I. A. Kupka (1994). "Observability and observers for nonlinear systems". In: *SIAM Journal on Control and Optimization* 32.4, 975–994.

Gauthier, J.P., H. Hammouri, and S. Othman (1992b). "A simple observer for nonlinear systems applications to bioreactors". In: *IEEE Transactions on Automatic Control* 37.6, pp. 875–880.

González-Hernández, Iván et al. (2017). "Enhanced Robust Altitude Controller via Integral Sliding Modes Approach for a Quad-Rotor Aircraft: Simulations and Real-Time Results". In: *Journal of Intelligent Robotic Systems volume* 88, 313–327.

Gozzini, Giovanni et al. (2020). "Air-to-Air Automatic Landing of Unmanned Aerial Vehicles: A Quasi Time-Optimal Hybrid Strategy". In: *IEEE Control Systems Letters* 4.3, pp. 692–697. DOI: 10.1109/LCSYS.2020.2991701.

Grandi, Raffaele, Riccardo Falconi, and Claudio Melchiorri (2012). "A Navigation Strategy for Multi-Robot Systems Based on Particle Swarm Optimization Techniques". In: *IFAC Proceedings Volumes* 45.22. 10th IFAC Symposium on Robot Control, pp. 331–336. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20120905-3-HR-2030.00060. URL: https://www.sciencedirect.com/science/article/pii/S1474667016336321.

Guezmil, Amal et al. (2015). "High order sliding mode and an unknown input observers: Comparison with integral sliding mode control for induction machine drive". In: *2015 7th International Conference on Modelling, Identification and Control (ICMIC)*, pp. 1–6. DOI: 10.1109/ICMIC.2015.7409478.

Harik, El Houssein Chouaib et al. (2017). "Fuzzy logic controller for predictive vision-based target tracking with an unmanned aerial vehicle". In: *Advanced Robotics* 31.7, pp. 368–381. DOI: 10.1080/01691864.2016.1271500.

Hernández-González, Omar et al. (2019). "High gain observer for a class of nonlinear systems with coupled structure and sampled output measurements: application to a quadrotor". In: *International Journal of Systems Science* 50.5, pp. 1089–1105.

Hu, Zhongjun and Xu Jin (2022). "An Adaptive Formation Control Architecture for A Team of Quadrotors with Performance and Safety Constraints". In: *2022 American Control Conference (ACC)*, pp. 3400–3405. DOI: 10.23919/ACC53348.2022.9867857.

Hua, Chang-Chun, Kuo Li, and Xin-Ping Guan (2019). "Semi-global/global output consensus for nonlinear multiagent systems with time delays". In: *Automatica* 103, pp. 480–489. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2019.02.022. URL: https://www.sciencedirect.com/science/article/pii/S000510981930072X.

Huong, D. C., V. T. Huynh, and H. Trinh (2019). "Integral Outputs-Based Robust State Observers Design for Time-Delay Systems". In: *SIAM Journal on Control and Optimization* 57.3, pp. 2214–2239.

Jalili, Sadegh, Behrooz Rezaie, and Zahra Rahmani (2018). "A novel hybrid model predictive control design with application to a quadrotor helicopter". In: *Optimal Control Applications and Methods* 39.4, pp. 1301–1322. DOI: https://doi.org/10.1002/oca.2411. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.2411. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2411.

Kahelras, Mohamed et al. (2016). "Observer Design for Triangular Nonlinear Systems Using Delayed Sampled-Output Measurements". In: *IEEE Conference on Decision and Control*, pp. 1447–1451.

Karafyllis, I. and C. Kravaris (2009). "From Continuous-Time Design to Sampled-Data Design of Observers". In: *IEEE Transactions on Automatic Control* 54.9, pp. 2169–2174. DOI: 10.1109/TAC.2009.2024390.

Kayacan, Erdal and Reinaldo Maslim (2017). "Type-2 Fuzzy Logic Trajectory Tracking Control of Quadrotor VTOL Aircraft With Elliptic Membership Functions". In: *IEEE/ASME Transactions on Mechatronics* 22.1, pp. 339–348. DOI: 10.1109/TMECH.2016.2614672.

Khalil, Hassan K. and Laurent Praly (2013). "High-gain observers in nonlinear feedback control". In: *International Journal of Robust and Nonlinear Control* 24, pp. 993–1015.

Khalil, Hassan K. and Stephanie Priess (2016). "Analysis of the Use of Low-Pass Filters with High-Gain Observers". In: *IFAC-PapersOnLine* 49.18. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016, pp. 488–492. ISSN: 2405-8963.

Kokotovic, P.V. (1992). "The joy of feedback: nonlinear and adaptive". In: *IEEE Control Systems Magazine* 12.3, pp. 7–17. DOI: 10.1109/37.165507.

Koshkouei, Ali J. and Alan Zinober (2002). "Sliding mode observers for a class of nonlinear systems". In: *American Control Conference* 3, pp. 2106 –2111.

Krstic, Miroslav (2010). "Lyapunov Stability of Linear Predictor Feedback for Time-Varying Input Delay". In: *IEEE Transactions on Automatic Control* 55.2, pp. 554–559. DOI: 10.1109/TAC.2009.2038196.

Kumar Singh, Shiv Santosh, Arkdev, and Mrinal Kanti Sarkar (2018). "Load Frequency Control: Higher Order Sliding Mode Observer Based Integral Higher Order Sliding Mode Controller with Stochastic Perturbation". In: *2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)*, pp. 1–6. DOI: 10.1109/EPETSG.2018.8658555.

L'Afflitto, Andrea, Robert B. Anderson, and Keyvan Mohammadi (2018). "An Introduction to Nonlinear Robust Control for Unmanned Quadrotor Aircraft: How to Design Control Algorithms for Quadrotors Using Sliding Mode Control and Adaptive Control Techniques [Focus on Education]". In: *IEEE Control Systems Magazine* 38.3, pp. 102–121. DOI: 10.1109/MCS.2018.2810559.

Lei, Wenchao, Chanying Li, and Michael Z. Q. Chen (2019). "Robust Adaptive Tracking Control for Quadrotors by Combining PI and Self-Tuning Regulator". In: *IEEE Transactions on Control Systems Technology* 27.6, pp. 2663–2671. DOI: 10.1109/TCST.2018.2872462.

Levant, Arie (2003). "Higher-order sliding modes, differentiation and output-feedback control". In: *International Journal of Control*, pp. 924–941. DOI: 10.1080/0020717031000099029.

Li, Norman and Hugh Liu (2009). "Multiple UAVs Formation Flight Experiments Using Virtual Structure and Motion Synchronization". In: *AIAA Guidance, Navigation, and Control Conference*.

Li, Yue, Jun Yang, and Ke Zhang (2019). "Distributed Finite-Time Cooperative Control for Quadrotor Formation". In: *IEEE Access* 7, pp. 66753–66763. DOI: 10.1109/ACCESS.2019.2915594.

Liaquat, Muwahida, Muhammad Azeem Javaid, and Muhammad Saad (2017). "A nonlinear high-gain observer for n-link robot manipulator which has measurement noise in a feedback control framework". In: *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pp. 755–759. DOI: 10.23919/ICCAS.2017.8204329.

Liu, Qixing, Zongyang Lv, and Yuhu Wu (2019). "Design of Sliding Mode Controller Based on High-Gain Observer of Inverted Pendulum On a Cart". In: *2019 Chinese Control Conference (CCC)*, pp. 846–851. DOI: 10.23919/ChiCC.2019.8865212.

Loianno, Giuseppe and Vijay Kumar (2018). "Cooperative Transportation Using Small Quadrotors Using Monocular Vision and Inertial Sensing". In: *IEEE Robotics and Automation Letters* 3.2, pp. 680–687. DOI: 10.1109/LRA.2017.2778018.

Luenberger, David (May 1964). "Observing the State of a Linear System". In: *IEEE Transactions on Military Electronics* 32, pp. 74–80. URL: http://doi.org/10.1109/TME.1964.4323124.

López-Gutiérrez R., Rodriguez-Mata A.E.-Salazar S. et al (Dec. 2017). "Robust Quadrotor Control: Attitude and Altitude Real-Time Results". In: *J Intell Robot Syst* 88, pp. 299–312. URL: https://doi.org/10.1007/s10846-017-0520-y.

Maneekittichote, Thanakrit and Teeranoot Chanthasopeephan (2020). "Mobile Robot Swarm Navigation and Communication Using LoRaWan". In: *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pp. 22–25. DOI: 10.1109/ICMRE49073.2020.9064973.

McLain, T.W. et al. (2001). "Cooperative control of UAV rendezvous". In: *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*. Vol. 3, 2309–2314 vol.3. DOI: 10.1109/ACC.2001.946096.

M.D.shuster and S.D.Oh (1981). "Three-axis attitude determination from vector observations". In: *Journal of Guidance and Control* 4.1, pp. 70–77.

Mechali, Omar et al. (2022). "Theory and practice for autonomous formation flight of quadrotors via distributed robust sliding mode control protocol with fixed-time stability guarantee". In: *Control Engineering Practice* 123, p. 105150. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2022.105150. URL: https://www.sciencedirect.com/science/article/pii/S0967066122000570.

Mellinger, Daniel et al. (2013). "Cooperative Grasping and Transport Using Multiple Quadrotors". In: *Distributed Autonomous Robotic Systems: The 10th International Symposium*. Ed. by Alcherio Martinoli et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 545–558. ISBN: 978-3-642-32723-0. DOI: 10.1007/978-3-642-32723-0_39. URL: https://doi.org/10.1007/978-3-642-32723-0_39.

Miladi, Nadia et al. (2018). "Robust State Estimation of a Quadrotor Based on High-Gain and Sliding-Mode Observers". In: *2018 15th International Multi-Conference on Systems, Signals Devices (SSD)*, pp. 1166–1171. DOI: 10.1109/SSD.2018.8570646.

Minh, Ly Dat and Cheolkeun Ha (2010). "Modeling and control of quadrotor MAV using vision-based measurement". In: *International Forum on Strategic Technology 2010*, pp. 70–75. DOI: 10.1109/IFOST.2010.5668079.

Mofid, Omid and Saleh Mobayen (2018). "Adaptive sliding mode control for finite-time stability of quadrotor UAVs with parametric uncertainties". In: *ISA Transactions* 72, pp. 1–14. ISSN: 0019-0578. DOI: https://doi.org/10.1016/j.isatra.2017.11.010. URL: https://www.sciencedirect.com/science/article/pii/S0019057817306171.

Moreno-Valenzuela, Javier et al. (2018). "Nonlinear PID-Type Controller for Quadrotor Trajectory Tracking". In: *IEEE/ASME Transactions on Mechatronics* 23.5, pp. 2436–2447. DOI: 10.1109/TMECH.2018.2855161.

Msaddek, Abdelhak, Abderraouf Gaaloul, and Faouzi Msahli (2015). "High gain observer based higher order sliding mode control: Application to an induction motor". In: *2015 IEEE 12th International Multi-Conference on Systems, Signals Devices (SSD15)*, pp. 1–7. DOI: 10.1109/SSD.2015.7348193.

Nasr, Mohamed et al. (2018). "A comparitive study on the control of UAVs for Trajectory tracking by MPC, SMC, Backstepping, and Fuzzy Logic controllers". In: *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 1–6. DOI: 10.1109/ICVES.2018.8519511.

Nguyen, Cuong M., Chee Pin Tan, and Hieu Trinh (2021). "State and delay reconstruction for nonlinear systems with input delays". In: *Applied Mathematics and Computation* 390, p. 125609. ISSN: 0096-3003.

Nicolas Boizot, Eric Busvelle and Jean-Paul Gauthier (Apr. 2015). "Adaptive-Gain Extended Kalman Filter: Extension to the Continuous-Discrete Case". In: *European Control Conference*. URL: https://doi.org/110.23919/ECC.2009.7075121.

Nigam, Nikhil et al. (2012). "Control of Multiple UAVs for Persistent Surveillance: Algorithm and Flight Test Results". In: *IEEE Transactions on Control Systems Technology* 20.5, pp. 1236–1251. DOI: 10.1109/TCST.2011.2167331.

Oh, Kwang-Kyo, Myoung-Chul Park, and Hyo-Sung Ahn (2015). "A survey of multi-agent formation control". In: *Automatica* 53, pp. 424–440. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2014.10.022. URL: https://www.sciencedirect.com/science/article/pii/S0005109814004038.

Orlov, Y. et al. (2002). "On identifiability of linear time-delay systems". In: *IEEE Transactions on Automatic Control* 47.8, pp. 1319–1324. DOI: 10.1109/TAC.2002.801202.

— (2003). "Adaptive identification of linear time-delay systems". In: *International Journal of Robust and Nonlinear Control* 13.9, 857–872.

Osorio-Gordillo, Gloria L. et al. (2016). "New dynamical observers design for linear descriptor systems". In: *IET Control Theory & Applications* 10.17, pp. 2223–2232. DOI: https://doi.org/10.1049/iet-cta.2016.0431. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-cta.2016.0431. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cta.2016.0431.

Osorio-Gordillo, Gloria-Lilia, Mohamed Darouach, and Carlos-Manuel Astorga-Zaragoza (2015). "H dynamical observers design for linear descriptor systems. Application to state and unknown input estimation". In: *European Journal of Control* 26, pp. 35–43. ISSN: 0947-3580. DOI: https://doi.org/10.1016/j.ejcon.2015.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0947358015000862.

Pérez-Estrada, A-J et al. (2019). "Adaptive observer design for LPV systems". In: *IFAC-PapersOnLine* 52.28, pp. 140–145.

Petrlík, Matěj, Vojtěch Vonásek, and Martin Saska (2019). "Coverage optimization in the Cooperative Surveillance Task using Multiple Micro Aerial Vehicles". In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 4373–4380. DOI: 10.1109/SMC.2019.8914330.

Phattanasak, Matheepot et al. (2016). "Lyapunov-based control and observer of a boost converter with LC input filter and stability analysis". In: *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC)*, pp. 1–6. DOI: 10.1109/ESARS-ITEC.2016.7841388.

Polyakov, Andrey, Denis Efimov, and Wilfrid Perruquetti (2015). "Finite-time and fixed-time stabilization: Implicit Lyapunov function approach". In: *Automatica* 51, pp. 332–340. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2014.10.082. URL: https://www.sciencedirect.com/science/article/pii/S0005109814004634.

Pounds P.E.I., Bersak-D.R. Dollar-A.M (2012). "Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control". In: *Autonomous Robots* 33, pp. 129–142.

Pérez-Estrada, A.-J. et al. (2018). "Generalized dynamic observers for quasi-LPV systems with unmeasurable scheduling functions". In: *International Journal of Robust and Nonlinear Control* 28.17, pp. 5262–5278. DOI: https://doi.org/10.1002/rnc.4309. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.4309. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.4309.

Qingzheng Xu, Zhisheng Wang and Ziyang Zhen (2019). "Adaptive neural network finite time control for quadrotor UAV with unknown input saturation". In: *Nonlinear Dynamics volume* 98, 1973–1998. DOI: 10.1007/s11071-019-05301-1.

Raff, T. and F. Allgower (2006). "An EKF-based observer for nonlinear time-delay systems". In: *2006 American Control Conference*, 4 pp.–.

Raghavan, Sekhar and J. Karl Hedrick (1994). "Observer design for a class of nonlinear systems". In: *International Journal of Control* 59.2, 515–528.

Rahimilarki, Reihane et al. (2019). "Robust Neural Network Fault Estimation Approach for Nonlinear Dynamic Systems With Applications to Wind Turbine Systems". In: *IEEE Transactions on Industrial Informatics* 15.12, pp. 6302–6312.

Robles-Magdaleno, J.L. et al. (2020). "A filtered high gain observer for a class of non uniformly observable systems – Application to a phytoplanktonic growth model". In: *Journal of Process Control* 87, pp. 68–78. ISSN: 0959-1524.

Rodriguez-Mata, A.E., M. Farza, and M M'Saad (2019). "Altitude control of quadrator UVAs using high gain observer-based output feedback high gain regulator". In: *2019 8th International Conference on Systems and Control (ICSC)*, pp. 147–152. DOI: 10.1109/ICSC47195.2019.8950554.

Safeer Ullah Adeel Mehmood, Qudrat Khan-Sakhi Rehman and Jamshed Iqbal (2020). "Robust Integral Sliding Mode Control Design for Stability Enhancement of Under-actuated Quadcopter". In: *International Journal of Control, Automation and Systems* 18, pp. 1671–1678. DOI: 10.1007/s12555-019-0302-3.

Sajadi, Morteza, Mohammad Ataei, and Mohsen Ekramian (2018). "Cascade high-gain observers for a class of nonlinear systems with large delayed measurements". In: *International Journal of Systems Science* 49.12, pp. 2558–2570.

Santana, Lucas Vago, Alexandre Santos Brandão, and Mário Sarcinelli-Filho (2021). "On the Design of Outdoor Leader-Follower UAV-Formation Controllers From a Practical Point of View". In: *IEEE Access* 9, pp. 107493–107501. DOI: 10.1109/ACCESS.2021.3100250.

Scherer, Jürgen and Bernhard Rinner (2020). "Multi-UAV Surveillance With Minimum Information Idleness and Latency Constraints". In: *IEEE Robotics and Automation Letters* 5.3, pp. 4812–4819. DOI: 10.1109/LRA.2020.3003884.

Sebesta, Kenneth D. and Nicolas Boizot (Jan. 2013). "A Real-Time Adaptive High-Gain EKF, Applied to a Quadcopter Inertial Navigation System". In: *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 61. URL: https://doi.org/10.1109/TIE.2013.2253063.

Senanayake, Madhubhashi et al. (2016). "Search and tracking algorithms for swarms of robots: A survey". In: *Robotics and Autonomous Systems* 75, pp. 422–434. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2015.08.010. URL: https://www.sciencedirect.com/science/article/pii/S0921889015001876.

Shen, Yanjun, Daoyuan Zhang, and Xiaohua Xia (2017). "Continuous observer design for a class of multi-output nonlinear systems with multi-rate sampled and delayed output measurements". In: *Automatica* 75, pp. 127–132. ISSN: 0005-1098.

Shi, Peng and Bing Yan (2021). "A Survey on Intelligent Control for Multiagent Systems". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.1, pp. 161–175. DOI: 10.1109/TSMC.2020.3042823.

Shin, Donghoon et al. (2016). "Observer based nonlinear control using barrier Lyapunov function for position tracking of Sawyer motor under yaw constraint". In: *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1440–1445. DOI: 10.1109/ICCAS.2016.7832493.

Talebi, Soroush, Mohammad Ataei, and Pierdomenico Pepe (2020). "An observer for a class of nonlinear systems with multiple state and measurement delays: A differential geometry-based approach". In: *European Journal of Control* 56, pp. 132–141. ISSN: 0947-3580.

Tréangle, Clément, Mondher Farza, and Mohammed M'Saad (2019a). "Filtered high gain observer for a class of uncertain nonlinear systems with sampled outputs". In: *Automatica* 101, pp. 197–206.

— (2019b). "Observer design for a class of disturbed nonlinear systems with time-varying delayed outputs using mixed time-continuous and sampled measurements". In: *Automatica* 107, pp. 231–240.

Utkin, Vadim I (1992). "Sliding modes in optimization and control problems". In:

Vaclavek, Pavel and Petr Blaha (2009). "Lyapunov function based design of PMSM state observer for sensorless control". In: *2009 IEEE Symposium on Industrial Electronics Applications*. Vol. 1, pp. 331–336. DOI: 10.1109/ISIEA.2009.5356456.

Viel F., Busvelle E. and J. P. Gauthier (Dec. 1995). "Stability of polymerization reactors using I/O linearization and a high-gain observer". In: *Automatica* 31, pp. 971 –984.

Wang, Chunyan et al. (2019). "Predictor-Based Extended-State-Observer Design for Consensus of MASs With Delays and Disturbances". In: *IEEE Transactions on Cybernetics* 49.4, pp. 1259–1269. DOI: 10.1109/TCYB.2018.2799798.

Wang, Dong et al. (2017). "A PD-Like Protocol With a Time Delay to Average Consensus Control for Multi-Agent Systems Under an Arbitrarily Fast Switching Topology". In: *IEEE Transactions on Cybernetics* 47.4, pp. 898–907. DOI: 10.1109/TCYB.2016.2532898.

Wang, Jianan et al. (2020). "Cascade structure predictive observer design for consensus control with applications to UAVs formation flying". In: *Automatica* 121, p. 109200. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2020.109200. URL: https://www.sciencedirect.com/science/article/pii/S0005109820303988.

Wang, Yu, Zunshui Cheng, and Min Xiao (2020). "UAVs' Formation Keeping Control Based on Multi–Agent System Consensus". In: *IEEE Access* 8, pp. 49000–49012. DOI: 10.1109/ACCESS.2020.2979996.

Xia, Mao-Dong, Cheng-Lin Liu, and Fei Liu (2018). "Formation-containment Control of Second-order Multi-agent Systems with Sampled Data and Intermittent Communication". In: *2018 IEEE International Conference on Information and Automation (ICIA)*, pp. 1204–1209. DOI: 10.1109/ICInfA.2018.8812541.

Yadaiah, N. et al. (2007). "Neural network algorithm for parameter identification of dynamical systems involving time delays". In: *Applied Soft Computing* 7.3, 1084–1091.

Yang, Bin et al. (2015). "Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis". In: *Robotics and Autonomous Systems* 72, pp. 83–92. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2015.05.001. URL: https://www.sciencedirect.com/science/article/pii/S0921889015000937.

Yang, Yue, Yang Xiao, and Tieshan Li (2021). "A Survey of Autonomous Underwater Vehicle Formation: Performance, Formation Control, and Communication Capability". In: *IEEE Communications Surveys Tutorials* 23.2, pp. 815–841. DOI: 10.1109/COMST.2021.3059998.

Younes Al Younes Ahmad Drak, Hassan Noura Abdelhamid Rabhi and Ahmed El Hajjaji (2016). "Robust Model-Free Control Applied to a Quadrotor UAV". In: *Journal of Intelligent Robotic Systems volume* 84, 37–52. DOI: 10.1007/s10846-016-0351-2.

Yu, Gan et al. (2019). "Nonlinear Backstepping Control of a Quadrotor-Slung Load System". In: *IEEE/ASME Transactions on Mechatronics* 24.5, pp. 2304–2315. DOI: 10.1109/TMECH.2019.2930211.

Yun, B., B.M. Chen, and K.Y Lum (2010). "Design and implementation of a leader-follower cooperative control system for unmanned helicopters". In: *Journal of Control Theory and Applications volume* 8, pp. 61–68.

Zemouche, Ali et al. (2018). "High-Gain Nonlinear Observer With Lower Tuning Parameter". In: *IEEE Transactions on Automatic Control* 64.8, pp. 3194–3209.

Zhang, Guangjie et al. (2016). "High-gain observer-based model predictive control for cross tracking of underactuated autonomous underwater vehicles". In: *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, pp. 115–120. DOI: 10.1109/USYS.2016.7893920.

Zhang, Huaipin et al. (2019). "Data-Driven Distributed Optimal Consensus Control for Unknown Multi-agent Systems With Input-Delay". In: *IEEE Transactions on Cybernetics* 49.6, pp. 2095–2105. DOI: 10.1109/TCYB.2018.2819695.

ZHANG, Qinghua and Aiping XU (July 2001). "Implicit Adaptive Observers for a Class of Nonlinear Systems". In: *American Control Conference*, pp. 1551–1556. URL: https://doi.org/10.1109/ACC.2001.945947.

Zheng, Gang, Andrey Polyakov, and Arie Levant (2018). "Delay estimation via sliding mode for nonlinear time-delay systems". In: *Automatica* 89, pp. 266–273. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2017.11.033. URL: https://www.sciencedirect.com/science/article/pii/S0005109817305757.

Zhi, Hui et al. (2021). "Leader–Follower Affine Formation Control of Second-Order Nonlinear Uncertain Multi-Agent Systems". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 68.12, pp. 3547–3551. DOI: 10.1109/TCSII.2021.3072652.

Zhou, Yan, Xiwang Dong, and Yisheng Zhong (2016). "Time-varying formation tracking for UAV swarm systems with switching interaction topologies". In: *2016 35th Chinese Control Conference (CCC)*, pp. 7658–7665. DOI: 10.1109/ChiCC.2016.7554571.

Zhou, Yan et al. (2014). "Time-varying formation control for unmanned aerial vehicles with switching interaction topologies". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1203–1209. DOI: 10.1109/ICUAS.2014.6842376.