Robust planning and control of unmanned aerial vehicles

The Hung Pham

To cite this version:

The Hung Pham. Robust planning and control of unmanned aerial vehicles. Automatic Control Engineering. Université Paris-Saclay, 2021. English. NNT : 2021UPASG003 . tel-03179970

HAL Id: tel-03179970
https://theses.hal.science/tel-03179970v1

Submitted on 24 Mar 2021

Robust planning and control of unmanned aerial vehicles

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Automatique
Unité de recherche: Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France.
Référent: Université d'Évry Val-d'Essonne.

Thèse présentée et soutenue à Evry, le 15/01/2021, par

The Hung PHAM

Composition du jury:

Mohamed Djemai Professeur, Université Polytechnique Hauts-de-France	Président
Benoit Marx Maître de conférences, HDR, Université de Lorraine	Rapporteur
Pedro Castillo Chargé de recherche, HDR, CNRS/Université de Technologie de Compiègne	Rapporteur
Cristina Maniu Professeur, CentraleSupélec - Université Paris-Saclay	Examinatrice
Said Mammar Professeur, UEVE - Université Paris-Saclay	Directeur de thèse
Dalil Ichalal Professeur, UEVE - Université Paris-Saclay	Co-Directeur de thèse

Thèse de doctorat

NNT: 2021UPASG003

Abstract

The objective of this thesis is to realize the modeling, trajectory planning, and control of an unmanned helicopter robot for monitoring large areas, especially in precision agriculture applications. Several tasks in precision agriculture are addressed. In pest surveillance missions, drones will be equipped with specialized cameras. A trajectory will be researched and created to enable unmanned aircraft to capture images of entire crop areas and avoid obstacles during flight. Infected areas will be then identified by analyzing taken images. In insecticides spraying, the aircraft must be controlled to fly in a pre-programmed trajectory and spray the insecticide over all the infected crop areas.

In the first part, we present a new complete coverage path planning algorithm by proposing a new cellular decomposition which is based on a generalization of the Boustrophedon variant, using Morse functions, with an extension of the representation of the critical points. This extension leads to a reduced number of cells after decomposition. Genetic Algorithm (GA) and Travelling Salesman Problem (TSP) algorithm are then applied to obtain the shortest path for complete coverage. Next, from the information on the map regarding the coordinates of the obstacles, non-infected areas, and infected areas, the infected areas are divided into several non-overlapping regions by using a clustering technique. Then an algorithm is proposed for generating the best path for a Unmanned Aerial Vehicle (UAV) to distribute medicine to all the infected areas of an agriculture environment which contains non-convex obstacles, pest-free areas, and pests-ridden areas.

In the second part, we study the design of a robust control system that allows the vehicle to track the predefined trajectory for a dynamic model-changing helicopter due to the changes of dynamic coefficients such as the mass and moments of inertia. Therefore, the robust observer and control laws are required to adapt the changes in dynamic parameters as well as the impact of external forces. The proposed approach is to explore the modeling techniques, planning, and control by the Linear Parameter Varying (LPV) type technique. To have easily implantable algorithms and adaptable to changes in parameters and conditions of use, we favor the synthesis of Linear Parameter Varying (LPV) Unknown Input Observer (UIO), LPV state feedback, robust state feedback, and static output feedback controllers. The observer and controllers are designed by solving a set of Linear Matrix Inequality (LMI) obtained from the Bounded Real Lemma and LMI regions characterization.

Finally, to highlight the performances of the path planning algorithms and generated control laws, we perform a series of simulations in MATLAB Simulink. Simulation results are quite promising. The coverage path planning algorithm suggests that the generated trajectory shortens the flight distance of the aircraft but still avoids obstacles and covers the entire area of interest. Simulations for the LPV UIO and LPV controllers are conducted with the cases that the mass and moments of inertias change abruptly and slowly. The LPV UIO is able to estimate state variables and the unknown disturbances and the estimated values converge to the true values of the state variables and the unknown disturbances asymptotically. The LPV controllers work well for various reference signals (impulse, random, constant, and sine) and several types of disturbances (impulse, random, constant, and sine).

Acknowledgements

During the completion of this thesis, I have received tremendous support from many people, including my family, my teachers, my friends, and colleagues.

First of all, I would like to express my greatest gratitude toward the late Professor Yasmina BESTAOUI-SEBBANE, Professeure des universités à l'UFR Sciences et Technologies de l'Université d'Evry-Vald'Essonne, the teacher whom I respect enormously both professionally and personally. Her knowledge, patience, and kindness have always been a source of inspiration. Without her, this thesis could never have begun and could not have gotten this far.

My deepest gratitude goes to my supervisors: Professor Said Mammar and Professor Dalil Ichalal. I have been so fortunate to get to know and work with them. I am grateful for their guidance, concern, support, and sympathy. They have shown and reminded me often that hard work is always the right thing to do. They are, above all, outstanding champions of merit and excellence.

My deepest thanks go to the dissertation committee members. I would like to thank Mr. Castillo Pedro, Professor at Université de Technologie de Compiègne, and Mr. Benoit Marx, Maître de conférences, HDR at Université de Lorraine, for the honor they have given me by accepting to be reporters of this thesis. I would also express my gratitude to Mrs. Cristina Maniu, Professor at CentraleSupélec, Université Paris-Saclay, and Mr. Mohamed Djemai, Professor at the Université Polytechnique Hauts-de-France, who kindly accepted to be examiners.

I am grateful to my current and past lab-mates in the IBISC (Informatique, Bio-Informatique et Systèmes Complexes) Laboratory: El Mehdi Zareb, Redouane Ayad, Yasser Bouzid, Rayane Benyoucef, Eslam abouselima, and Sushil Sharma for the fun we had together in many occasions in the lab and outside during all this time.

Finally, I would like to thank my parents for their support in all my life. Words cannot express how grateful I am to my parents for all of their support and sacrifices. They have been and will always be the model for my life. A special thanks go to my brother The Anh. They have been a tremendous emotional and psychological support to me throughout these years and for that, I am eternally thankful.

Last but not least, I owe special thanks to my wife, Lan Phuong, and my daughters Phuong Linh, Phuong Chi, to which this dissertation is dedicated. Their smiles and jokes are always of great fun and support for me, on all occasions. I am truly grateful to my wife for her sacrifices while I was not around. She did all the hard works to raise my kids and encourages me to overcome hardships, her encouragement was a great comfort to me and contributed to the success of this work.

Contents

List of Figures

List of Tables

List of Abbreviations

LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
UAV	Unmanned Aerial Vehicle
IMU	Inertial Measurement Unit
CoG	Center of Gravity
DOF	Degrees of Freedom
GPS	Global Positioning System
DGPS	Differential Global Positioning System
VTOL	Vertical Take-Off and Landing
CT LPV	Continuous Time Linear Parameter Varying
DT LPV	Discrete Time Linear Parameter Varying
LTI	Linear Time Invariant
UIO	Unknown Input Observer
PA	Precision Agriculture
IoT	Internet of Things
ICT	Information and Communication Technology
CPP	Coverage Path Planning
TSP	Travelling Salesman Problem
GA	Genetic Algorithm
PID	Proportional Integral Derivative
PI	Proportional Integral
PD	Proportional Derivative
DOB	Disturbance Observer Based
LQR	Linear Quadratic Regulator

LQG	Linear Quadratic Gaussian
SMC	Sliding Mode Control
STA	Super Twisting Algorithm
MPC	Model Predictive Control
RRT	Rapidly-exploring Random Tree
PRM	Probabilistic RoadMap
CD	Cellular Decomposition
TS	Takagi-Sugeno
KF	Kalman Filter
EKF	Extended Kalman Filter
DCM	Direction Cosine Matrix
OKF	Optimal Kalman Filter
CGAL	Computational Geometry Algorithms Library

List of Symbols

$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}$	The set of integers
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of non-negative real numbers
$\mathbb{R}_{>0}$	The set of positive real numbers
$\mathbb{R}^n$	n-dimensional vector space over the field of the real numbers
$eig\left(A\right)$	The set of eigenvalues of square matrix $A \in \mathcal{R}^{n\times n}$
$\sec\left(x\right)$	The secant function $\frac{1}{\cos x}$
$\succ$	Positive definite symbol
$\prec$	Negative definite symbol
$Re\left(*\right)$	The real part of complex number *

1

Introduction

Chapter abstract

This chapter briefly introduces the needs of using robots, especially Unmanned Aerial Vehicle (UAV) in Precision Agriculture (PA) for increasing crop yields. Firstly, sustainable food production challenges are given in section 1.1. Then, the definition of PA is introduced in section 1.2. The uses of the mobile robot and UAV in PA are studied respectively in subsections 1.3 and 1.4. Our motivations and the objectives of the thesis are highlighted in section 1.5 and 1.6 respectively. The main contributions are listed in sections 1.7. Finally, the structure of the thesis document is outlined in section 1.8.

This Chapter contains:

1.1 Sustainable food production

According to the Food and Agriculture Organization (FAO, 2009a), it is estimated that by 2050 the world's population will reach 9 billion and this population change mainly occurs in developing countries [68]. Consequently, ensuring a sustainable supply of food for the world's fast-growing population is a major challenge. Added to the challenge is that sustainable food products need to be nutrient-dense to allow people to have a diverse diet that contains a balanced and adequate combination of energy and nutrients to support good health [240].

Sustainable food production is "a method of production using processes and systems that are non-polluting, conserve non-renewable energy and natural resources, are economically efficient, are safe for workers, communities and consumers, and do not compromise the needs of future generations"[76].

Although the area of land for agriculture is declining, water resources for cultivation are increasingly scarce, climate change [169], global warming [178], more pests appear, more food still have to be produced. These negative effects on agriculture can be offset to some degree by improving pest control technologies, implementing crop rotations, soil and water conservation, altering crop varieties, employing other sound ecological technologies for resource use in agriculture, and machinery applications in agricultural production.

1.2 Precision Agriculture (PA)

Precision Agriculture (PA) can be defined as "the application of modern information technologies to provide, process and analyze multi source data of high spatial and temporal resolution for decision making and operations in the management of crop production"[47]. The main goal of PA is the process of gathering, processing, and analyzing temporal, spatial, and individual data to support management decision making. This decision greatly affects improved resource use efficiency, productivity, quality, profitability and sustainability of agricultural production

Autonomous robots can be applied in a variety of field operations of PA. The use of Autonomous robots facilitate the processes of capturing and processing high quantities of data. They also provide the required capabilities for operating not only at the individual plant level but also at the complete field level. In [26], authors propose autonomous platforms which can be used for cultivation and seeding, weeding, scouting, application of fertilizers, irrigation, and harvesting.

For the reason that economic benefits should be easily achieved without requiring the integration of additional components or decision support systems, several advanced robotic technologies have been widely applied in PA such as autonomous guidance, autosteer, remote sensing, and controls.

The most important abilities of automatic agricultural robots can be categorized into four categories [40]:

- Guidance: the robot has to self-locate its position in the agricultural environment.
- Detection: the robot should have the ability to detect the biological features from the environment.

- Action: Based on the information on the first two categories, the robot should execute the task for which the vehicle was designed, such as spraying, monitoring, or harvesting.
- Mapping: the robot could construct the map of the agricultural field.

For accomplishing all the previously mentioned activities, the devices (sensors) such as Real-Time Kinetics Global Positioning System (RTK-GPS), camera, and Light Detection and Ranging (LiDAR) have to retrieve information from the environment and send them to the processing unit of the robot. This information can be the states of the robot(location, velocity,...) or the information related to the agriculture area. Then, a decision will be made based on the gathered information to accomplish the predefined tasks.

There are several types of agriculture robots. However, they can be defined as two main groups: mobile robot and aerial robot. In the next two subsections, the use of mobile robot and aerial robot in PA will be considered.

1.3 The use of mobile robot in PA

Mobile robots (Fig. 1.1) have been widely used in precision agriculture to reduce human labor and increase productivity. Farmers can use the tractors, harvesters, and they are self-guided by GPS, Robots can automate operations like pruning, thinning, mowing, spraying, and weed removal, Sensor technology is used to manage the pests and diseases that affect crops. Some examples of the application of mobile robot are autonomous targeted spraying for pest control in commercial greenhouses [176], to optimum manipulator design for autonomous de-leafing process of cucumber plants [95], simultaneous localization and mapping techniques for plant trimming [34], automated harvesting of valuable fruits (for example on sweet pepper[184][94], oil palm [206], mango [209], cucumber [229], almond [227], apple [161][101], strawberry [91], cherry fruit [215], citrus [147], vineyard [246], and tomato [182][181][233]).

(a) Hortibot robot[105]
source: `www.technologyreview.com`

(b) Kongskilde Robotti robot[88]
source: `conpleks.com`

FIGURE 1.1: Examples of mobile robots in PA

Besides mobile robots, flying robots are also widely used in PA. The applications of UAVs will be introduced in the next subsection.

1.4 The use of UAV in PA

The combination of modern technologies such as the Internet of Things (IoT), Information and Communication Technology (ICT), control theory, image processing, and especially the accelerated development of the UAV technology provides a promising solution to PA [212] to deal with the challenges of food sustenation problem. UAVs have been widely used in many applications of PA such as monitoring, weed mapping and management, crop spraying, etc. The specific applications of UAV in PA will be covered in this subsection.

1.4.1 Monitoring

A very common application of PA is to determine the crop health situation by using Remote Sensing (RS) techniques and image analysis. Images can be captured by satellites [22][156]or UAVs [18][144]. However, satellite images are not good choice due to the low spatial resolution of acquired images, restrictions of the temporal resolutions as satellites are not always available to capture the necessary images, the quality of the images depends on the weather condition and camera costs. Another method that gives better quality of the images is using the manned aircraft. Though, this method is still costly.

Crops health is a very important factor that determines the quality and quantity of the crop. Therefore, the crops health and diseases detection must be monitored [51][185][155][116] continuously to detect disease promptly and avoid spreading problems. Normally, crop health is often monitored by agricultural experts. Nevertheless, this is a very time-consuming task, as it requires several weeks or even months for inspecting an entire crop and preventing the potentials of "continuous" disease spreading. Another method that can be applied to prevent crop disease is to spray pesticides at fixed times during the plant's growth. This method has been applied and is relatively popular in developed countries, however, it is high cost and also increases the likelihood of groundwater contamination as pesticide residues in the products and soil.

The applicable of UAV in PA for monitoring the crop's health conditions provide excellent possibilities to obtain field data in an easy, fast, efficient, and cost-effective way compared to the aforementioned methods. UAVs equipped with specialized cameras are used for gathering photos of the entire plant area. The changes in plant biomass and their health can be determined by using special algorithms for analyzing the crop imaging information. Thus, diseases can be determined in their early stages enabling farmers to intervene in order to reduce losses. For this purpose,UAVs can be used in the two different stages of disease control as: (i) UAVs are used to collect crop images. Analyze the gathered images for detecting a possible infection before visual indications appear. Map infected areas for treatment; and (ii) during the treatment of infection, the UAV is used to spray insecticides to precisely the locations of the infected plants that were mapped in the previous period.

1.4.2 Weed mapping and management

Another application of UAV in PA is weed mapping [99][174][98][200][19]. Weeds are non-desirable crops, they grow simultaneously with the crop in agricultural areas and cause a number of serious problems. They use the same resources such as soil, water, nutrients, and space on agricultural land as crops and affect crop growth. Traditionally, herbicides are used for controlling the weeds. The same amounts of herbicides are sprayed over the entire field, even within the weed-free areas. Such a strategy incurs a waste of herbicides, potentially contaminating soil resources. Moreover, overuse of herbicides can lead to the evolution of herbicide-resistant weeds and it can affect crop growth and productivity. To overcome the above problems the agriculture area is divided into zones that each one receives customized management because as usual weeds spread through only a few spots of the field. Consequently, it is necessary to generate an accurate weed cover map for precise spraying of herbicide. UAVs are used to gathering images. Then, the gathered images are analyzed to generate a precise weed cover map depicting the spots where the chemicals are needed and the needed amount of herbicide for each spot.

1.4.3 Crop spraying

The traditional spraying equipment used in conventional farming is the manual air-pressure or battery-powered knapsack sprayers. However, this traditional spraying equipment causes a non-minor loss of pesticides due to the mechanical structure of the injector and the spraying is done manually. Furthermore, the operators have to be present when spraying, resulting in contact with operators and possibly affecting human health.

Therefore, the use of UAVs to spray pesticides [217][235][69][81][244] can reduce the operator's exposure to chemicals, reduce pesticide losses, and save time. An uncrewed helicopter, which is equipped with small pesticide tanks, was developed and introduced in Japan in the 1980s [212] for crop spraying. By analyzing image data, we can determine exactly and pest status of the crop on the whole area of agriculture. Because the level of plant pests and diseases varies from region to region, the amount of insecticide to be sprayed must also be different to ensure uniform plant growth. Path planning algorithms based on crop data not only create the optimal motion trajectory for the UAV, but also optimize the amount of needed agrochemical products.

1.4.4 Irrigation management

Currently, the process of irrigation for crops consumes about 70% of the water in the world. [38][196], while water resources are increasingly exhausted. Therefore, crop irrigation management [190][12][168][183] is a very important task in PA. Precision irrigation techniques must be improved and monitored more strictly to lead to the more efficient use of water: (i) in the right place; (ii) at the right time; and (iii) in correct quantities.

Typically, water is evenly irrigated over the entire crop area, although the amount of water actually needed for different areas varies. Therefore, determining the appropriate amount of water to irrigate in different areas can help farmers save time and water resources. Moreover, such precise farming techniques can lead to increased yields and crop quality. Dividing agricultural land into different irrigation areas, for precise management of resources, can be done easily by using UAV. Special sensors equipped on the UAVs give them the ability to classify and identify areas where plants need more water. Besides, based on data collected from UAV, we can create specialized maps that illustrate the morphology of the soil for more efficient irrigation, suitable for each type of crop and each specific area.

TABLE 1.1: The use of autonomous robot in PA

Publication	Robot type	Operation	Technique
[180]	Aerial robot	Monitoring Process - Monitoring vegetation state	Multispectral camera, GPS system, FlightCTRL, NaviCtTRL, First person view platform, GSM modem, Magnetic compass
[13]	Aerial robot	Monitoring Process -Detecting drainage pipes	VIS Camera, NIR Camera, Thermal Camera
[110]	Aerial robot	Monitoring Process -Evaluating water stress and vegetation state	Thermal camera, Multispectral camera, Single-board computer, GPS system, Stabilization mechanism
[249]	Aerial robot	Monitoring Process - Estimating nitrogen state	Hyperspectral camera
[49]	Aerial robot	Spraying Process - Spraying fruits and trees	Spraying Device, Multispectral camera, IMU, Magnetometer. Barometer, Servos
[193]	Aerial robot	Investigating computational resources during a monitoring process	IMU, GPS system
[198]	Aerial robot	Evaluating water stress	FlightCtrl, NaviCtrl, 3-axis accelerometer, Thermal Camera, Storing device, Pressure sensor, Digital compass, GPS system
[228]	Mobile robot	Coverage path planning	Harmony Search (HS) algorithm for finding complex coverage trajectories
[192]	Mobile robot	Field mapping	3D terrain maps, stereo camera, location sensor, IMU
[199]	Mobile robot	Weed and pest control	Image processing, decision control algorithms
[217]	Mobile robot	Weeding	Automatic computer vision, differential spraying to control the weeds.

1.4.5 Vegetation growth monitoring and yield estimation

UAVs are also often applied in monitoring vegetation growth and providing estimates of productivity [250][230][106][92]. One of the main major difficulties of increasing agricultural productivity and quality is the lack of means to systematically monitor the process of cultivation. Therefore, collecting and analyzing information related to crop growth regularly using the UAV provide an increased opportunity to track crop growth and observe changes in some parameters of the field.

Many works have been focusing on observing the biomass and nitrogen status of crops besides the yield estimation. Biomass of crops and nitrogen content status are the main parameters to determine the need for additional fertilizer or other actions. Furthermore, various parameters, such as crop height, the distance between rows or between plants, and the index Leaf Area Index (LAI) can be estimated by using the three-dimensional digital maps of the crop which is created by the gathered images by the UAVs. By analyzing the information gathered about crops by UAVs, farmers can make plans to control crop management, decide when to irrigate, spray pesticides, or harvest.

Table 1.1 provides examples of the developments of autonomous mobile and aerial robots in PA.

1.5 Motivations

We are motivated, in this thesis, by the coverage path planning scenario and control of Unmanned Aerial Vehicle (UAV) in Precision Agriculture (PA) for the two tasks, crop inspection and crop spraying.

Although the application of Unmanned Aerial Vehicle (UAV) in Precision Agriculture (PA) has been being studied intensively, however, many challenges are still open with respect to the coverage path planning scenarios as the requirements to visit all the predefined points while avoiding obstacles and minimize some objective functions.

In the control aspect, the UAV has to follow as most accurately as possible according to the given trajectory under the influence of disturbances such as wind or changes of dynamic parameters such as mass or moments of inertia during flying time.

1.6 Objectives

The objective of this thesis is to realize the modeling, planning, and control of an unmanned helicopter robot for monitoring large areas, especially in PA applications. Several tasks in PA are addressed such as monitoring, insecticides spraying.

- **Path planning objective**: In pest surveillance missions, drones will be equipped with specialized cameras. In order to monitor the whole agriculture area, we need to capture images by specialized cameras. A trajectory will be researched and generated to enable the UAV to capture images of entire crop areas and avoid obstacles during flight. Infected areas will be then identified by analyzing gathered images.
- **Control objective**: In insecticides spraying, the UAV must be controlled to fly in a pre-programmed trajectory, which has been generated in the path planing task, and spray the insecticide over all the infected crop areas. The UAV is equipped with pesticide tank and a mechanical or hydraulic system for pesticide spraying. During the spraying of insecticides, the mass of UAV will decrease over time, resulting in changes

in the dynamic parameters of UAV such as the moments of inertia. In addition, during the mission, UAV is affected by disturbances such as wind gusts. The changes in the kinetic parameters and the external disturbances cause instability of the UAV. Consequently, a robust controller has to be designed to stabilize the UAV and to follow exactly the predefined trajectory while keep stabilizing the UAV under the changes in the kinetic parameters and the external disturbances.

1.7 Contribution of the thesis

The objective of this thesis is to realize the modeling, planning, and control of an unmanned UAV robot for monitoring large areas, especially in PA[1] applications. Several tasks in PA are addressed. In pest surveillance missions, drones will be equipped with specialized cameras. A trajectory will be researched and created to enable unmanned aircraft to capture images of entire crop areas and avoid obstacles during flight. Infected areas will be then identified by analyzing taken images. In insecticides spraying, the aircraft must be controlled to fly in a pre-programmed trajectory and spray the insecticide over all the infected crop areas.

In the first part, we present a new complete coverage path planning algorithm by proposing a new cellular decomposition which is based on a generalization of the Boustrophedon variant, using Morse functions, with an extension of the representation of the critical points. This extension leads to a decrease in the number of cells after decomposition. GA[2] and TSP[3] algorithm are then applied to obtain the shortest path for complete coverage. Next, from the information on the map regarding the coordinates of the obstacles, non-infected areas, and infected areas, the infected areas are divided into several non-overlapping regions by using a clustering technique. Then an algorithm is proposed for generating the best path for a UAV to distribute medicine to all the infected areas of an agriculture environment which contains non-convex obstacles, pest-free areas, and pests-ridden areas.

In the second part, we study the design of a robust control system that allows the vehicle to track the predefined trajectory for a dynamic model-changing UAV due to the changes of dynamic coefficients such as the mass and moments of inertia. Therefore, the robust observer and control laws are required to adopt the changes in dynamic parameters as well as the impact of external forces. The proposed approach is to explore the modeling techniques, planning, and control by the LPV type technique. To have easily implantable algorithms and adaptable to changes in parameters and conditions of use, we favor the synthesis of LPV[4] UIO[5], LPV quadratic state feedback, dynamic ouptut feedback, and static output feedback controllers. The observers and controllers are designed by solving a set of Linear Matrix Inequality (LMI) obtained from the Bounded Real Lemma and LMI region characterization.

Finally, to highlight the performance of the path planning algorithms and generated control laws, we perform a series of simulations in MATLAB Simulink environment. Simulation results are quite promising. The coverage path planning algorithm suggests that the generated trajectory shortens the flight distance of the aircraft but still avoids obstacles and covers the

---

[1] Precision Agriculture (PA)

[2] Genetic Algorithm (GA)

[3] Travelling Salesman Problem (TSP)

[4] Linear Parameter Varying (LPV)

[5] Unknown Input Observer (UIO)

entire area of interest. Simulations for the LPV UIO and LPV controllers are conducted with the cases that the mass and moments of inertia change abruptly and slowly. The LPV UIO is able to estimate state variables and the unknown disturbances and the estimated values converge to the true values of the state variables and the unknown disturbances in a short time. The LPV controllers work well for various reference signals (impulse, random, constant, and sine) and several types of disturbances (impulse, random, constant, and sine).

1.8 Structure of the thesis

The manuscript of the thesis is divided into 7 chapters:

- **Chapter 1 - Introduction**: Introduces the applications and necessity of robots, especially UAVs in PA. Moreover, it highlights our motivations, the specific challenges, objectives, and the main contributions of the thesis.
- **Chapter 2 - State of the art**: offers a short state of the art for UAVs and more particularly the multirotors as typical vehicle for coverage path planning mission. Then, it also reviews the state of the art of the coverage path planning algorithms and control laws for quadcopters.
- **Chapter 3 - System modeling:** introduces the mathematical differential equations of quadcopter dynamics.
- **Chapter 4 - Robust path planning**: proposes a new approach for maximizing the coverage path planning while minimizing the path length of an aerial robot in agriculture environment with concave obstacles. For resolving this problem, we propose a new cellular decomposition which is based on a generalization of the Boustrophedon variant, using Morse functions, with an extension of the representation of the critical points. This extension leads to a decrease of the number of cells after decomposition. The results show that this new cellular decomposition works well even with several concave obstacles inside the environment. Furthermore, for path planning, the cells are divided again into two classes, leading to have a cell set better suited for use of the TSP to get complete coverage. GA and TSP algorithm are applied to obtain the shortest path. Then, an approach is also proposed to maximize the scanned area on the working area with obstacles. Then, we focus on generating the best path for an UAV to distribute medicine to all the infected areas of an agriculture environment which contains non-convex obstacles, pest-free areas and pests-ridden areas. The algorithm for generating this trajectory can save the working time and the amount of medicine to be distributed to the whole agriculture infected areas.

  From the information on the map regarding the coordinates of the obstacles, non-infected areas, and infected areas, the infected areas are divided into several non-overlapping regions by using a clustering technique. There is a trade-off between the number of classes generated and the area of all the pests-ridden areas. After that, a polygon will be found to cover each of these infected regions. However, obstacles may occupy part of the area of these polygons that have been created previously. Each polygon that is occupied in part by obstacles can be further divided into a minimum number of obstacle-free convex polygons. Then, an optimal path length of boustrophedon trajectory will be created for each convex polygon that has been created for the UAV to follow. Finally, this chapter deals with the process of creating a minimal path for the UAV to move between all the constructed convex polygons and generation of the final trajectory for the UAV which ensures that all the infected agriculture areas will be covered by the medicine.

- **Chapter 5 - Control:** During the spraying of insecticides, the mass of UAV will decrease over time, resulting in changes in the dynamic parameters of UAV such as the moments of inertia. In addition, during the mission, UAV is affected by disturbances such as wind. The changes in the kinetic parameters and the external disturbances cause instability of the UAV. Therefore, several robust control laws for quadcopter under disturbances and changes of the dynamic parameters have been developed such as LPV dynamic output feedback controller, LPV static output feedback controller, and LPV state feedback controller. We also design a Linear Parameter Varying (LPV) Unknown Input Observer (UIO) for quadcopters.
- **Chapter 6 - Simulation and experiments:** The simulations are conducted for both path planning and control tasks. This chapter also gives some comments on the simulation results and will discuss the advantages and performances of the proposed approaches.
- **Chapter 7 - Conclusion and perspectives:** Gives some concluding remarks as well as perspectives.

1.9 Publications

This work results in several publications in international conferences proceedings. Journal papers are submitted and are under review.

Published:

- T. H. Pham, Y. Bestaoui and S. Mammar, "Aerial robot coverage path planning approach with concave obstacles in precision agriculture," 2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Linkoping, 2017, pp. 43-48, doi: 10.1109/RED-UAS.2017.8101641.

- T. H. Pham and S. Mammar, "Quadrotor LPV Control using Static Output Feedback," 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 2019, pp. 74-79, doi: 10.1109/ICNSC.2019.8743181.

- T. H. Pham, D. Ichalal, S. Mammar (2019). "LPV and Nonlinear-based control of an Autonomous Quadcopter under variations of mass and moment of inertia". IFAC-PapersOnLine, 52(28), pp. 176 – 183. doi:10.1016/j.ifacol.2019.12.371.

- T. H. Pham, D. Ichalal, S. Mammar (2020). , "LPV state-feedback controller for Attitude/Altitude stabilization of a mass-varying quadcopter," 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea (South), 2020, pp. 212-218, doi: 10.23919/ICCAS50221.2020.9268310.

- T. H. Pham, D. Ichalal, S. Mammar (2020). "Complete coverage path planning for pests-ridden in precision agriculture using UAV," 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), Nanjing, China, 2020, pp. 1-6, doi: 10.1109/ICNSC48988.2020.9238122.

- T. H. Pham, D. Ichalal, S. Mammar (2020). "LPV Unknown Input Observer for Attitude of a mass-varying quadcopter". 16th International Conference on Control, Automation, Robotics and Vision, ICARCV 2020, Shenzhen, China, December 13 - 15, 2020, pp. 917 – 924.

To be submitted:

- T. H. Pham, D. Ichalal, S. Mammar. "Discrete LPV state feedback preview control design for Attitude/Altitude of a mass-varying quadcopter".

- T. H. Pham, D. Ichalal, S. Mammar. "Attitude/Altitude LPV UIO-based controller design of a mass-varying quadcopter".

- T. H. Pham, D. Ichalal, S. Mammar. "Complete coverage path planning using UAV applied in precision agriculture under the consideration of dynamic constraints.

2

State of the art

Chapter abstract

This chapter offers a brief state of the art of Unmanned Aerial Vehicle (UAV). Firstly, a short review of UAV such as UAV classification, applications, advantages, and disadvantages of UAV will be given in Section 2.1.1. Due to their special characteristics, which allow their use in a wide range of applications, we focus on particular Vertical Take-Off and Landing (VTOL) types of UAVs and especially the quadrotor configuration in Section 2.2. Then the state of the arts of complete coverage of UAV and control of quadrotors are given in Section 2.3 and 2.4 respectively.

This Chapter contains:

2.1 Unmanned Aerial Vehicles

2.1.1 Overview of unmanned aerial vehicles

An Unmanned Aerial Vehicle (UAV) or drone is an aircraft without passenger or pilot onboard which can fly autonomously based on pre-programmed flight plans, through the help of complex dynamic automation systems, or be controlled remotely by a pilot at a ground control station. Depending on its mission, endurance, and size, each UAV can carry different payloads.

UAVs are equipped with multiple sensors (such as Global Positioning System (GPS), Inertial Measurement Unit (IMU), Differential Global Positioning System (DGPS), accelerometer, magnetometer, pito tube, etc) and are connected in real time to many other systems at the ground control station. This allows the UAV to gain important information such as speed, position, heading, altitude, movement direction, temperature, speed and wind direction, the amount of fuel or remaining energy to perform the task, etc. UAV can perform a variety of tasks in military, civilian, and entertainment applications. Moreover, since they are free of screw on board, they can be designed to be smaller, can perform tasks that require greater flexibility, and can carry more payloads.

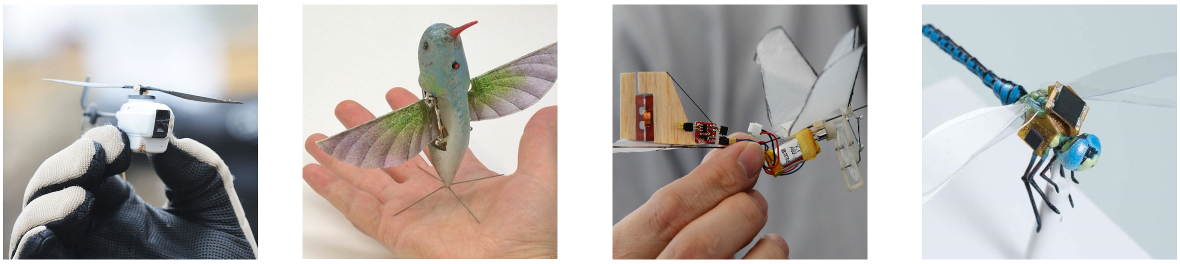
2.1.2 UAVs classification

There are many characteristics for classifying UAVs [107][138] based on their variety intended use.

Classification according to size

- Very small UAVs: are the unmanned aircraft with dimensions ranging from the size of a large insect to $30\,[cm] - 50\,[cm]$ long. The very small UAVs can be sub-divided into
  - Micro or nano UAVs: insect size is up to $15\,[cm]$.
  - Mini UAVs: insect size is between $15\,[cm]$ and $50\,[cm]$.

  The representative of very small UAVs is insect-like UAVs. Their popular micro design always has flapping or rotary wings. They are extremely small in size, very lightweight, and can be used for spying and biological warfare. Some of nano UAVs are shown in Fig. 2.12 Examples of mini UAVs are Xbird 250 quadro, Parrot bebop, Parrot Disco, and Eagle VTOL Mapping in Fig. 2.10.



(a) Black Hornet **(b)** Hummingbird **(c)** Hummingbird **(d)** Dragonfly robot

FIGURE 2.1: Nano UAV robots

(a) Xbird 250 quadro **(b)** Parrot bebop **(c)** Parrot Disco **(d)** Eagle VTOL Mapping

FIGURE 2.2: Mini UAV robots

- Small UAVs: are the UAVs class that have at least one dimension greater than 50 cm and no larger than 2 meters. Most of them are hand-launched by throwing in the air while some other are of fixed-wing model. Examples of small UAVs are RQ 11 Raven with a wingspan of 1.4 m, RQ 7 Shadow, RS 16 is a crossover UAV between a small and a medium sized system, and Turkish Bayraktar which weight is 5 kg and has a range of 20 km in Fig. 2.3.

(a) RQ 11 Raven **(b)** RQ 7 Shadow **(c)** RS 16 **(d)** Turkish Bayraktar

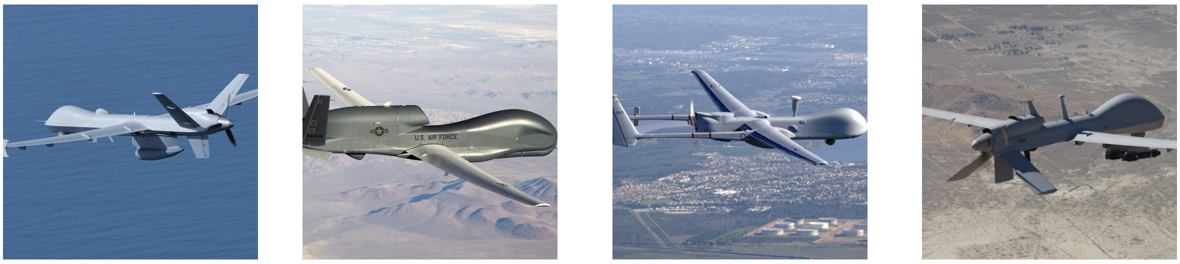
FIGURE 2.3: Small UAV robots

- Medium UAVs: are the class of UAVs that are too heavy to be carried by one person but are still smaller than a light aircraft. They usually have a wingspan of about 5-10 m and can carry payloads of 100 to 200 kg. Some medium fixed-wing UAVs are shown in Fig. 2.4. There are also numbers of medium sized UAVs in the form of rotary-based.

(a) UK Watchkeeper **(b)** RQ 2 Pioneer **(c)** RS 20 **(d)** Eagle Eye

FIGURE 2.4: Medium UAV robots

- Large UAVs: applies to the large UAVs and used mainly by the military. Some of large UAVs such as Atomics MQ-1 Predator, Global Hawk, Harfang, and MQ 1C Warrior are shown in Fig. 2.5.



(a) Atomics MQ-1 Predator **(b)** Global Hawk **(c)** Harfang **(d)** MQ 1C Warrior

FIGURE 2.5: Laege UAV robots

Classification of UAVs according to its range of action

- High-altitude long-endurance (HALE): this types of UAVs can flight over 15000 $[m]$ altitude and more than 24 $[hr]$ of endurance. These UAVs are usually used for surveillance mission. Two example of HALE UAVs are Northrop Grummans Global Hawk and Agilis HALE UAV in Fig. 2.6.

(a) Northrop Grummans Global Hawk **(b)** Agilis HALE UAV

FIGURE 2.6: High-attitude long-endurance (HALE) UAV robots

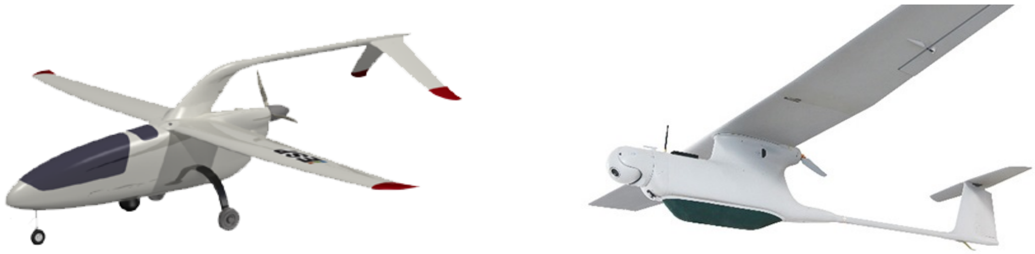
- Medium-altitude long-endurance (MALE): the altitude for this types of UAVs is about 5000 $[m]$ to 15000 $[m]$ with 24 $[hr]$ maximum of endurance. These UAVs are also used for surveillance mission but with shorter ranges. Two example of MALE UAVs are Persistent UAS Platforms and Agilis HALE UAV in Fig. 2.7.
- Medium-Range or tactile UAV (TUAV): the range of flight for this type of UAVs is between 100 $[km]$ and 300 $[km]$. The TUAVs IAI Heron (Machatz-1) and AAI (Textron) RQ-7 Shadow are shown in Fig. 2.8.
- Close-range UAV: The close range UAVs can operate within the range less than 100 $[km]$. This type of UAVs is mainly used for traffic monotoring, powerline inspection, surveillance, or crop-spraying in precision agriculture.The close range UAV Optimus UAV and RemoEye 006 are shown in Fig. 2.9.

(a) Persistent UAS Platforms **(b)** WK-450 Watchkeeper

FIGURE 2.7: Medium-attitude long-endurance (HALE) UAV robots

(a) IAI Heron (Machatz-1) **(b)** AAI (Textron) RQ-7 Shadow

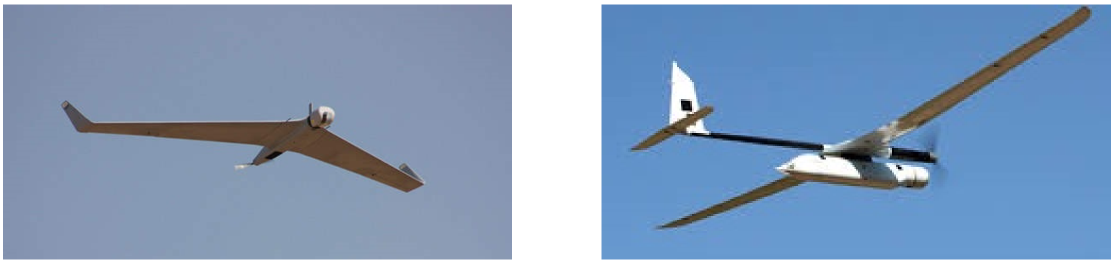
FIGURE 2.8: Medium-Range or tactile UAV (TUAV) robots



(a) Optimus UAV **(b)** RemoEye 006

FIGURE 2.9: Close-Range UAV

- Mini UAV (MUAV): The operation range for mini UAVs is up to $30\,[km]$ and their weight is less than $20\,[kg]$. The mini UAVs Orbiter Mini UAV and Skylark 1 LE Mini are shown in Fig. 2.10.



(a) Orbiter Mini UAV **(b)** Skylark 1 LE Mini

FIGURE 2.10: Mini UAV robots

- Micro UAV (MAV): the wingspan is less than 150 $[mm]$. This type of UAVs are mainly used in urban environments or for indoor applications. Due to the small size, the micro UAVs are very vulnerable to the wind.


(a) Micro UAV NX70 **(b)** Honeywell RQ-16 T-Hawk

FIGURE 2.11: Mini UAV robots

- Nano Air Vehicle (NAV): the size is of 10 $[mm]$. The Nano UAVs The Hornet 2-b (Prox Dynamics) and Nano UAV Hornet 3 Prox Dynamics in Fig. 2.12a are complete with camera and video transmitter.

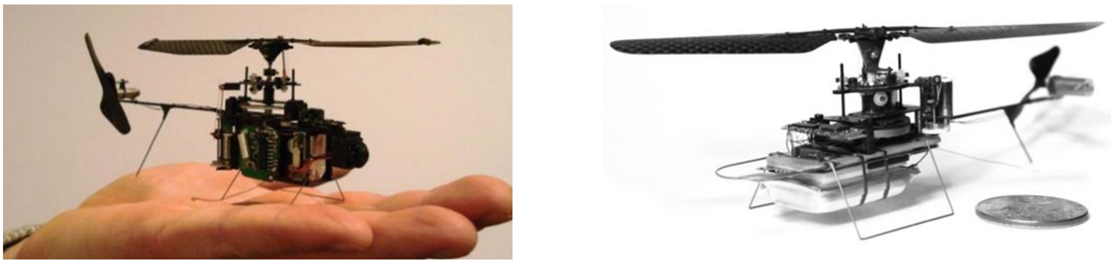

(a) The Hornet 2-b (Prox Dynamics) **(b)** Nano UAV Hornet 3 Prox Dynamics

FIGURE 2.12: Nano UAV robots

- Remotely piloted helicopter (RPH): are UAVs that are capable of vertical take-off and landing automatically with pre-programmed programs. This type of aircraft is often used in missions that require hovering flight such as inspection or surveillance. Remotely piloted helicopters Long Endurance Fuel Power Fixed Wing gasoline VTOL UAV and Terrafugia TF-2 Tiltrotor are shown in Fig. 2.13


(a) Long Endurance Fuel Power Fixed Wing gasoline VTOL UAV **(b)** Terrafugia TF-2 Tiltrotor

FIGURE 2.13: Nano UAV robots

2.1.3 Applications of UAVs

Nowadays, the uses of UAVs with diverse capabilities for both civilian and military applications are growing very fast. Moreover, there is a significant interest in the development of novel drones that can operate autonomously in different types of complicated environments and locations. UAVs can perform various missions in complicated environments both outdoor and indoor. Drones can be equipped with various sensors and cameras. Consequently, UAVs are capable of performing accurate and reliable tasks of intelligence, surveillance, and reconnaissance missions. The missions of UAVs can be categorized as for the military or civil, type of the flight zones (outdoor or indoor), and type of the environments (air/space) as shown in Figure 2.14.

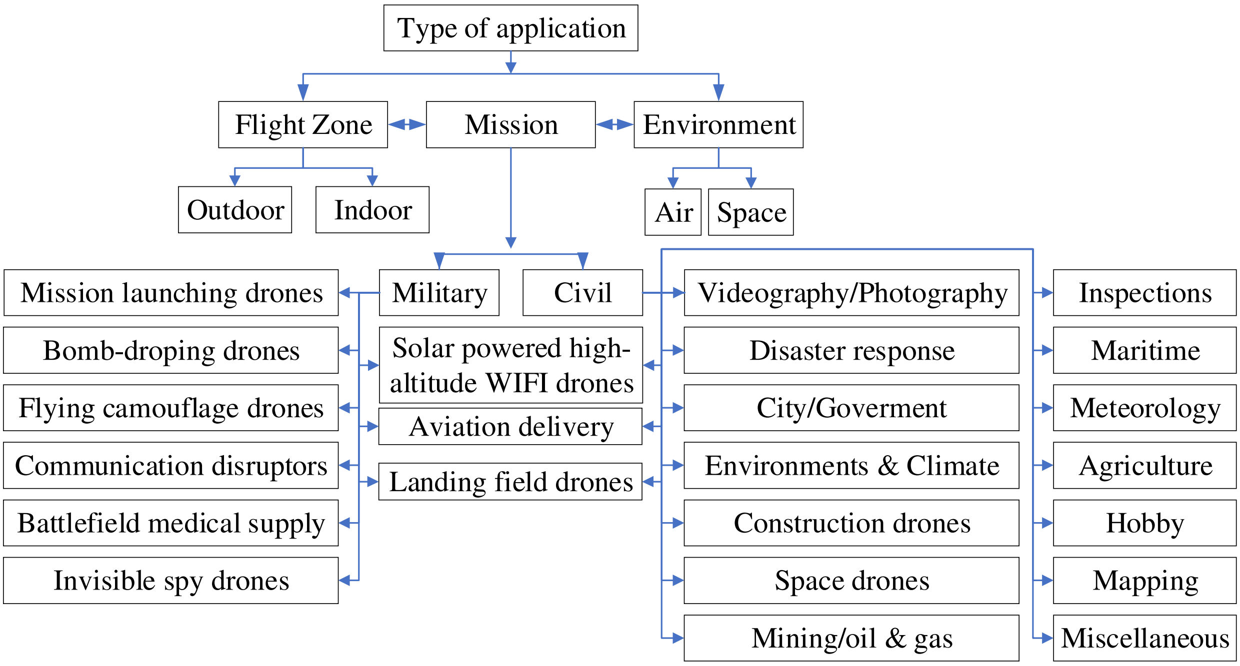


FIGURE 2.14: Classification of drones' applications[158]

Search and rescue missions

A really important application of UAV is to search and rescue missions [231]. A search and rescue UAV is usually used by emergency services, such as police officers, firefighters, or rescue teams. Every second is vital in search and rescue operations. In order to most effectively deploy rescue missions, we need to understand and get a quick overview of the situation. While manned aerial vehicles and helicopters require time to be ready for operating the mission, drones can be operated instantaneously without any preparation time. Unmanned aerial vehicles UAVs can provide real-time visual information and data in the aftermath of an earthquake or hurricane. They can also become an eye in the sky to locate a lost person in the mountain for example. Thanks to high mobility, quick deployment, able to operate in complex environments including small environments such as caves, this type of UAVs is ideal for searching over vast areas for missing persons in need of rescue and in any environment.

Environmental protection

Besides rescue or military applications, UAV is also increasingly applied in environmental management such as managing agricultural lands, real-time weather condition and forecast, observing the effects of climate change, monitoring the biodiversity of different ecosystems from rain forests to the oceans, monitoring crops pest, and tracking wildlife in different areas [219].

Mailing and delivery

Unmanned delivery service is attracting the attention of big companies in the world such as Amazon, Google [224][87]. More and more companies are using drones to deliver products to their customers. To accomplish this, the UAV is equipped with the ability to vertically take off and land.

Drone in Precision Agriculture (PA)

With increasing populations, the demand for food is increasing. The latest advanced technologies should be applied in agriculture to meet this growing demand. Several types of robots have been designed for using in PA such as mobile robots for harvesting, crops monitoring. With the introduction of low-cost drones with advanced capabilities, the use of UAVs in precision agriculture is growing really fast. Drones are playing a significant role in optimizing agriculture operations. Thanks to the help of drones, agricultural activities such as crop management, crop counting, crop health monitoring, and spraying pesticides can be operated more cost-effectively and more quickly as compared to conventional methods, which typically rely on the use of piloted aircraft.

2.1.4 Advantage, disadvantages, and typical uses of UAVs

In the previous subsection, we have examined the classification and applications of UAVs. In this chapter, we examine the advantages and disadvantages of UAVs for the fixed-wing UAV, single rotor UAV, and multirotor UAV.

2.1.4.1 Fixed Wing UAV

Fixed-wing UAVs use a wing like a normal aeroplane to provide the lift by the aircraft's forward motion. Fixed-wing UAVs can be self-propelled, pure gliders (vehicles whose free flight does not rely on a method of propulsion) or a mixture of the two. Fixed-wing UAVs can be distinguished to the other types of UAVs that they cannot stay hovering in one place with vertical lift rotors in the air.

Fixed wing UAVs are well known in the military, as they are often used when manned flight is considered too risky or difficult. They are also used in the commercial industry, monitoring, etc.

Advantages:

- The operating time of fixed-wing UAVs can be a couple of hours and can up to 16 hours or more.

- Fixed-wing UAVs can fly at a high altitude.
- Fixed-wing UAVs have the ability to carry more weight.

Disadvantages:

- Fixed-wing UAVs are usually expensive.
- Training is usually required for flying.
- Launcher or long runway are needed to get a fixed-wing UAVs into the air.
- It is required the runway, parachute, or special method for landing (recovering) the UAVs.
- Fixed-wing UAVs can't hover in the air.

Typical uses:

- The fixed-wing UAVs are usually used for commercial purposes such as aerial mapping, inspection, agriculture, construction, security, and surveillance due to the ability of long endurance, high altitude, and long operating fly time.

2.1.4.2 Single Rotor UAV

While a multi-rotor UAV has many different rotors to hold it up, a single rotor UAV has just one rotor, plus a tail rotor to control its heading.

Advantages:

- A single rotor UAVs have the benefit of much greater efficiency over a multi-rotor,
- Long flying time if they are powered by a gas motor.
- Single rotor UAVs are able to hover vertically in the air.
- Single Rotor UAV are strong and durable.
- Heavy payload capability.

Disadvantages:

- Single Rotor UAVs are harder to fly than multi-rotor UAV types.
- Single rotor UAVs can be expensive.
- Single rotor UAVs have a higher complexity.
- Single rotor UAVs can be dangerous because of their heavy spinning blade.

Typical uses:

- Surveying, research, Aerial LIDAR laser scan.

2.1.4.3 Multirotors UAVs

A multirotor UAV is one of the robots that has the ability to achieve agile motion in the air. Unlike fixed-wing UAV, multirotor UAVs rely on the rotation of rotors to generate power, which will generate powerful airflow. This is the popular choice for aerial photography, film making and surveillance. It can also be used by professionals and hobbyists alike because of its small size and ready to fly out of the box capabilities. The multirotors UAVs classification is shown in Figure 2.15.

Multirotor UAVs carry several rotors on their body and can be further classified based on the number of them on the platform of the drone. There are tricopters (3 rotors as in Figure 2.15a, 2.15b), quadcopters (4 rotors as in Figure 2.15c, 2.15d), hexacopters (6 rotors as in Figures 2.15e, 2.15f, 2.15g, 2.15h, 2.15i) and octocopters (8 rotors as in Figures 2.15j, 2.15k, 2.15l, 2.15m, 2.15n, 2.15o). The most commonly used of Multirotor UAVs are quadcopters.

Advantages:

- Simpler rotor mechanics required for flight control, easy for maintenance.
- Multirotor UAVs are easy control and maneuver.
- Multirotor UAVs have the ability to hover in the air.
- Multirotor UAVs can take off and land vertically.
- Multirotor UAVs are very stable.

Disadvantages:

- Low flying time (usually 15-30 minutes).
- Multirotor UAVs are limited on payload capability.
- Multirotor UAVs's energy is spent on fighting gravity and stabilizing in the air.

Typical uses:

- Aerial photography and video aerial inspection, leisure, agriculture, construction, security.

2.2 Quadrotors

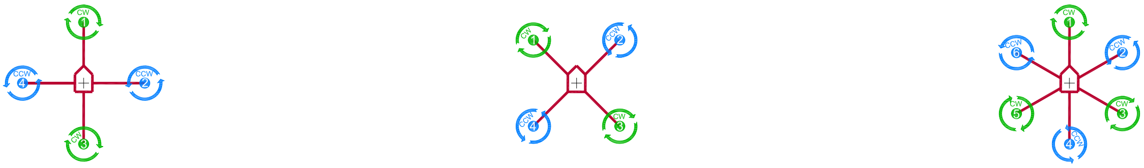
Based on the analysis of the advantages and disadvantages of different types of UAVs as detailed in the previous subsection, we choose quadrotor as the research object in the content of this thesis. The main features of the quadrotor will be considered in more detail in this subsection.

A quadcopter is a helicopter that consists of a rigid cross-frame equipped with four rotors M_1, M_2, M_3, and M_4 as shown in Figure 2.16. There are two main configurations for quadrotor: "+" configuration as in Figure 2.16a, and "X" configuration as in Figure 2.16b. The motors are equidistant from the center of the quad by about L.

Each rotor M_i (for $i = 1,..,4$) generates thrust f_i for $i = 1,...,4$. The direction of rotation of the rotors are fixed and f_i is always a positive quantity. The main thrust is the sum of all the thrusts which are generated by the rotors. The detailed mathematical equations of quadrotor will be studied in Chapter 3.

(a) Tri rotor Normal

(b) Tri rotor Reversed Multi



(c) Quadrotor plus

(d) Quadrotor X

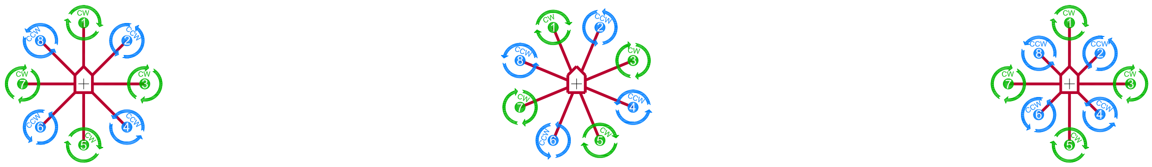
(e) Hexa Plus

(f) Hexa X

(g) Hexa H

(h) Hexa Coax Y6

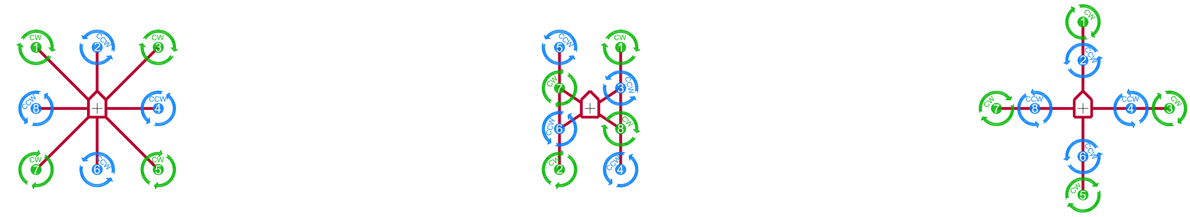
(i) Hexa Coax Reversed



(j) Octo Plus

(k) Octo X

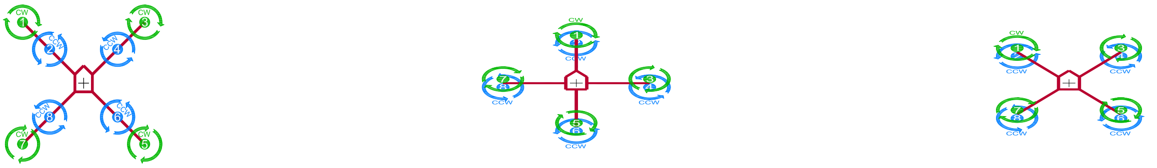
(l) Octo Square Plus



(m) Octo Square X

(n) Octo H

(o) Octo Colinear Plus



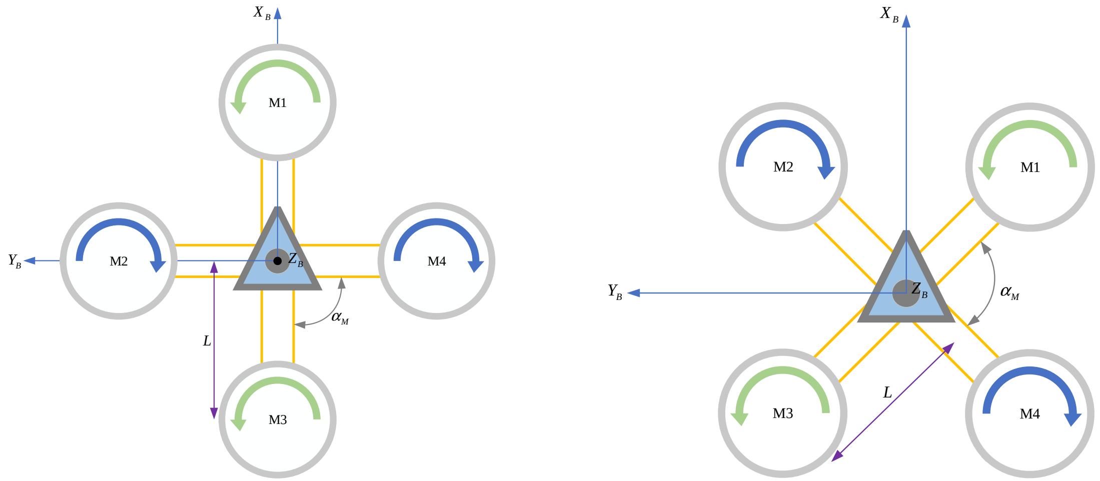
(p) Octo Colinear X

(q) Octo Coax Plus

(r) Octo Coax X X8

FIGURE 2.15: Multirotors classification according to the principle of flight

In order to avoid the yaw drift due to the reactive torques, the quadrotor aircraft is configured



(a) Quadrotor + configurations **(b)** Quadrotor X configurations

Figure 2.16: Quadrotor configurations

such that the set of rotors M_2, M_4 (left-right) revolves in clockwise (CW) direction, while the pair of rotors M_1, M_3 (front-rear) rotates in counterclockwise (CCW) direction.

Advantages:

- Rotor mechanics are simplified as it depends on four fixed pitch rotors unlike the variable pitch rotor in the helicopter.
- Due to the symmetry in the configuration, the gyroscopic effects are reduced leading to simpler control.
- Stationary hovering can be more stable in quadrotors than in helicopters due to the presence of four propellers providing four thrust forces shifted a fixed distance from the center of gravity instead of only one propeller centered in the middle as in the helicopters structure.
- More advantages are the vertical take-off and landing capabilities, better maneuverability and smaller size due to the absence of a tail, these capabilities make quadrotors useful in small area monitoring and buildings exploration.

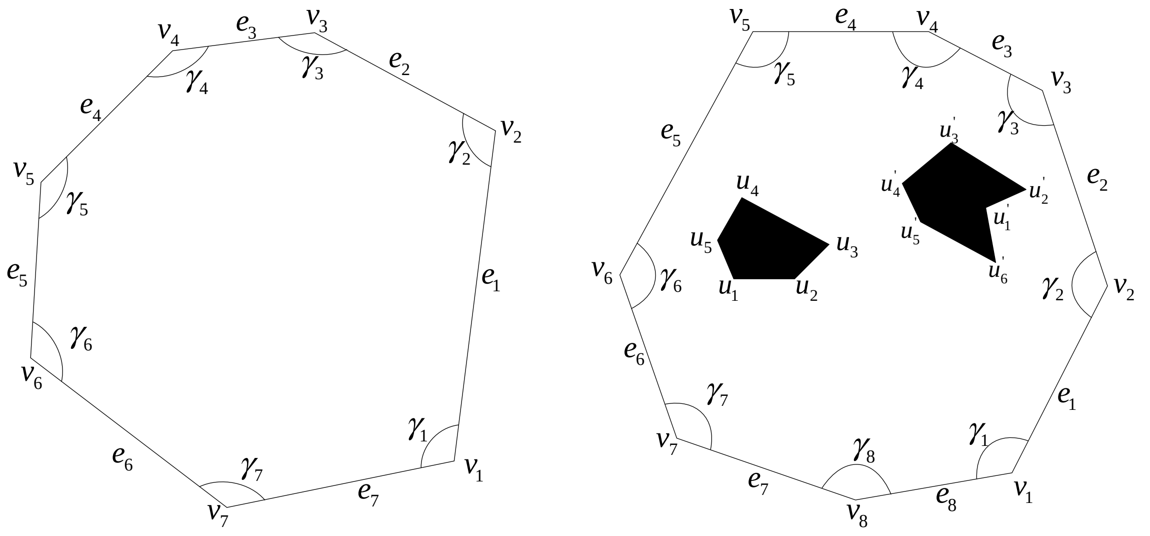
Disadvantages:

- Quadrotors consume a lot of energy due to the presence of four separate propellers.
- Quadrotors have a large size and heavier than some of their counterparts again to the fact that there is four separate propellers.

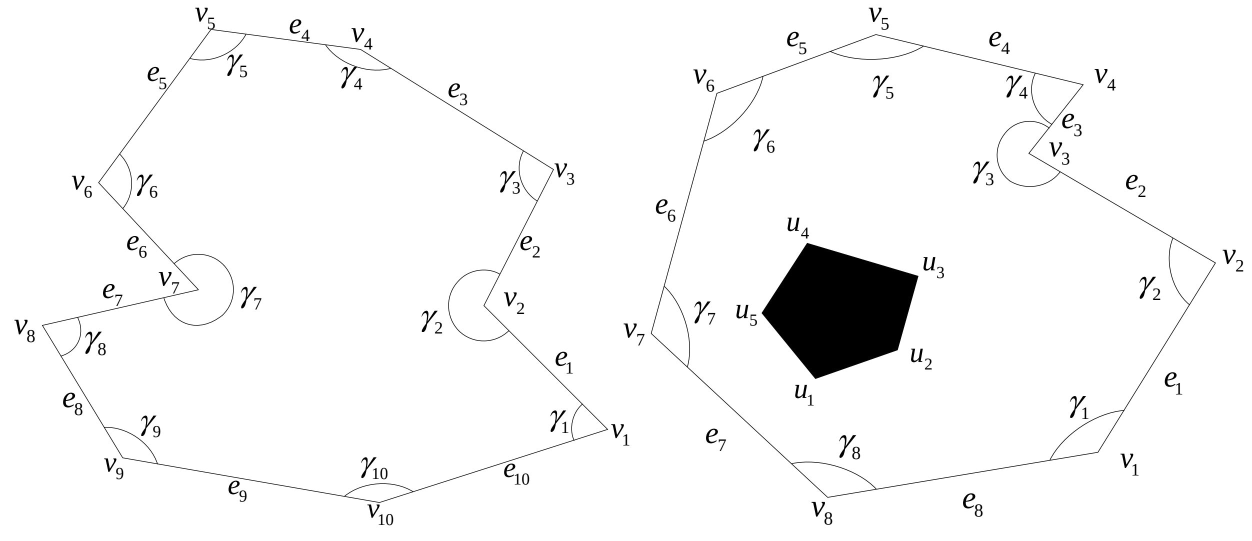
2.3 Coverage Path planning methods used for UAV

CPP is one of the most important tasks for robot motion and missions. Consider an area of interest composed by the robot's free space and its boundaries. CPP is the determination of a path that a robot must take in order to pass over each point in an environment while avoiding obstacles such that the entire target environment is covered.

Based on the path that has been generated, robots can accomplish their predefined tasks such as cleaning, painting, window cleaning, and inspection of complex structures. CPP has been extensively studied in recent years for applications such as vacuum cleaning robots [245], painter robots [17], path planning for autonomous underwater vehicles [171], demining robots [3], lawn mowers [28]. For the aerial context, the obstacles can be represented by no-flying zones (NFZ) that the UAV should avoid during its operation.



(a) Convex polygon without holes **(b)** Convex polygon with holes



(c) Concave polygon without holes **(d)** Concave polygon with holes

FIGURE 2.17: Regions of interest

In [36], the authors defined the requirements for the robot to perform a coverage operation. Although the target application for CPP problems in the aforementioned paper is mentioned for mobile robots in a 2-dimensional environment, the same requirements are used for other coverage scenarios as follows:

- the robot is required to go through all the points in the target area covering it completely

- the robot is required to fill the region of interest without overlapping paths.
- the robot must operate continuously and sequentially without any repetition of paths
- the robot is required to avoid all the obstacles of the region of interest
- simple motion trajectories such as straight lines or circles should be used for the purpose of simplicity in control
- the use of an "optimal" path is desired under available conditions

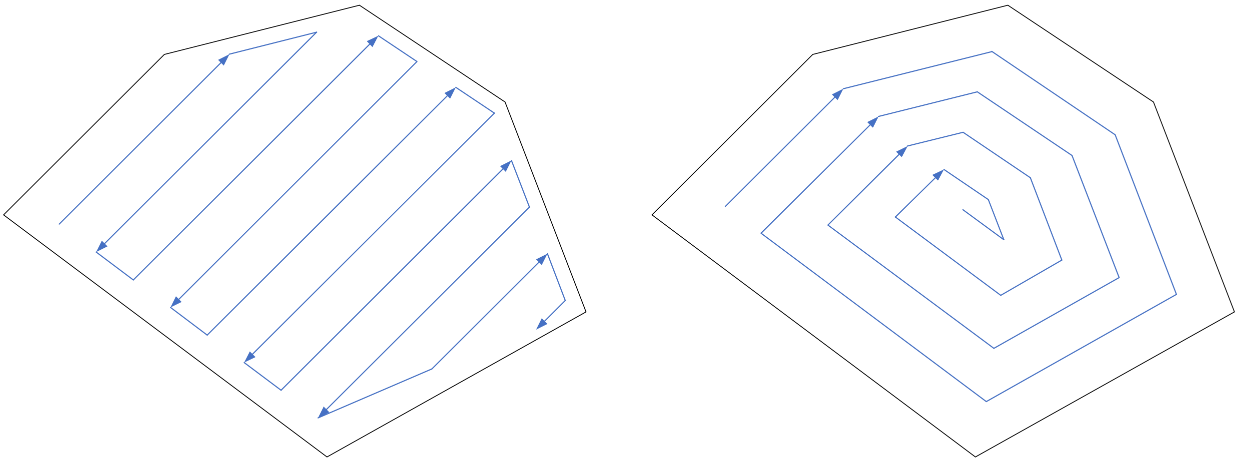
However, due to the complexity of the working region with obstacles, it is really hard to satisfy all the aforementioned requirements. As a result, some requirements need to be made with a higher priority than others.

Region of interest

The region of interest to be completely covered by a robot can be represented by a polygon which contains n vertices $v_1,...,v_n$. Each vertex v_i contains a pair of coordinates $(v_x(i), v_y(i))$, and the internal angle at i-th vertex can be referred by γ_i. The edge between two vertices v_i and v_{i+1} for $i = 1,...,n-1$ are e_i and the edge between vertex v_n and v_1 is e_n as in Figure 2.17. The shape of region of interest can be any type of polygon such as convex with no holes inside as in Figure 2.17a, concave with holes inside as in Figure 2.17d. These holes can be considered as the no-fly zones where the UAVs are not allowed to fly.

2.3.1 No Decomposition

If the shape of the region of interest is regular-shaped and non-complex, the complete coverage mission can be performed without decomposition. Simple geometric patterns are sufficient to explore such areas. The two popular patterns are back-and-forth (BF) as in Figure (2.18a) and the spiral (SP) as in Figure (2.18b).



(a) Back and Forth trajectory

(b) Spiral trajectory

FIGURE 2.18: Simple geometric patterns for coverage path planing

2.3.2 Cellular Decomposition (CD)

The environment is always divided into non-overlapping regions called cells using a decomposition technique in order to simplify the coverage [44]. The size of the cells depends on the type of decomposition method and an applied specific strategy to guarantee complete coverage. These cells typically have the same size as a robot (terrestrial coverage) or are proportional to the sensor's range (aerial coverage). Several motions are required to fully cover the large cell, while a single motion is enough for the smaller cells. There exist several different cellular decomposition methods, however, the most commonly used in CPP problems involving UAVs are exact and approximate cellular decomposition.

Exact cellular decomposition

Exact cellular decomposition is a decomposition method in which the regions of interest are divided into non-overlapping sub-areas (cells) whose reunion exactly occupies the target area. By using a simple motion as back-and-forth, each cell can be fully explored. Consequently, the CPP problem of the whole region of interest is now referred to as the motion planning problem from one cell to another [44]. The robot can move between two cells, which have a mutual border. The adjacency graph is proposed to represent the motion of the robot as in Figure. 2.19. In this adjacency graph, each cell represents a node, while the edge represents neighbour cells. The final complete coverage path is the combination of motions performed inside the cells and the connections between cells [80].

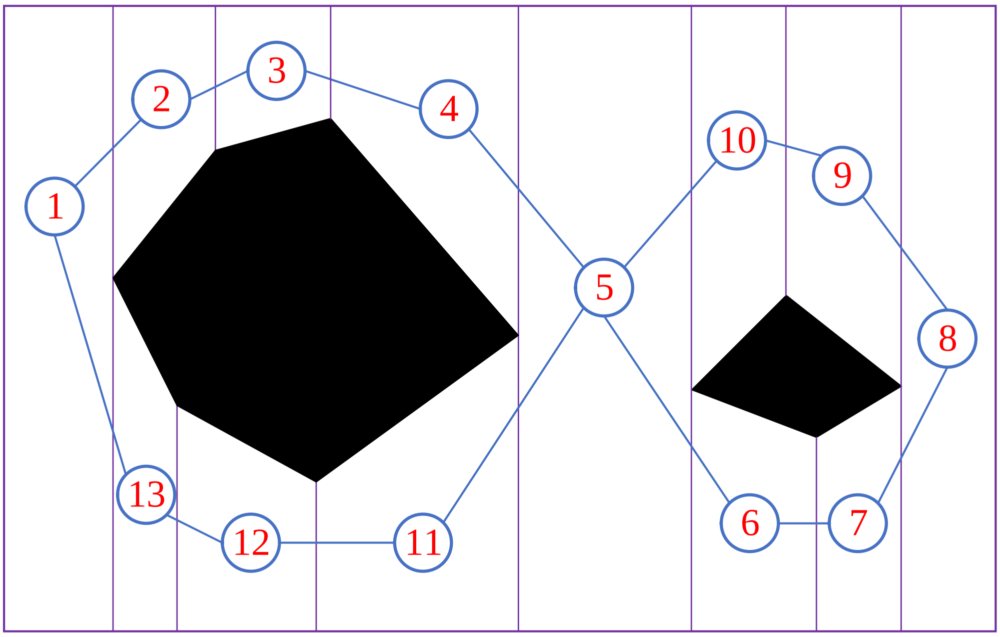
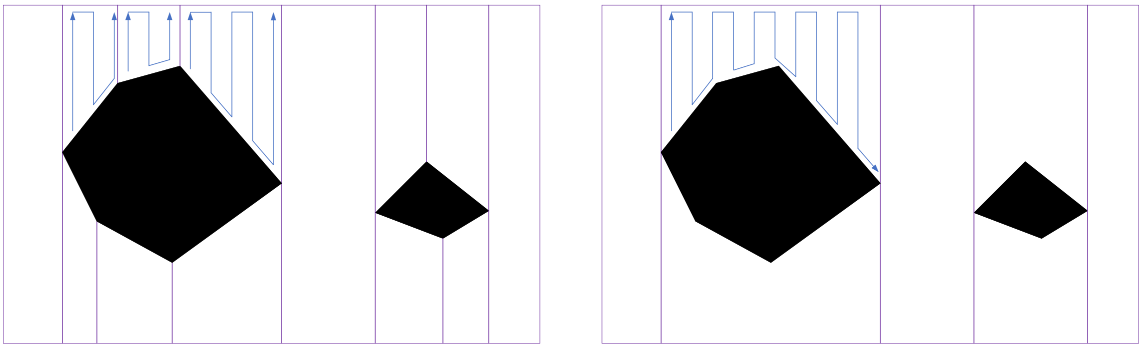


FIGURE 2.19: Exact cellular decomposition method and adjacency graph

Several cellular decomposition methods have been studied as trapezoidal decomposition technique [120], boustrophedon decomposition [45], and Morse-based cellular decomposition [225].

In the trapezoidal decomposition technique [120], the free spaces are divided into trapezoidal cells. There are two parallel sides on each cell, and each cell can be covered by simple back and forth motion parallel to either side. Therefore, the whole working area can be totally scanned by visiting each cell in the adjacency graph. A vertical line is swept from left to right through the working area of the robot. Each time the line encounters a vertex of a polygon, one can say that an event occurs. Three types of events are defined: IN, OUT, and MIDDLE. In an IN event, the current cell is closed and two new cells are opened. When an OUT event occurs, the current two cells are closed and a new cell is created. Finally, MIDDLE event is the event that a current cell is closed and a new single cell is initiated. Consequently, by using these three events, the environment can be totally divided into separated trapezoidal cells.

A drawback of the trapezoidal decomposition (shown in Figure 2.20a) is that it generates many cells that, intuitively, can be merged together to form bigger cells. This is clearly an inconvenient, as the more cells are present, the longer the final coverage path is. This happens because the trapezoidal decomposition creates only convex cells. However, non-convex cells can also be completely covered by simple motions. To overcome this limitation, the boustrophedon cellular decomposition was proposed. The boustrophedon decomposition (shown in Figure 2.20b) is similar to the trapezoidal decomposition introduced above, but it only considers vertices where a vertical segment can be extended both above and below the vertex. The vertices where this occurs are called critical points.



(a) Trapezoidal decomposition **(b)** Boustrophedon decomposition

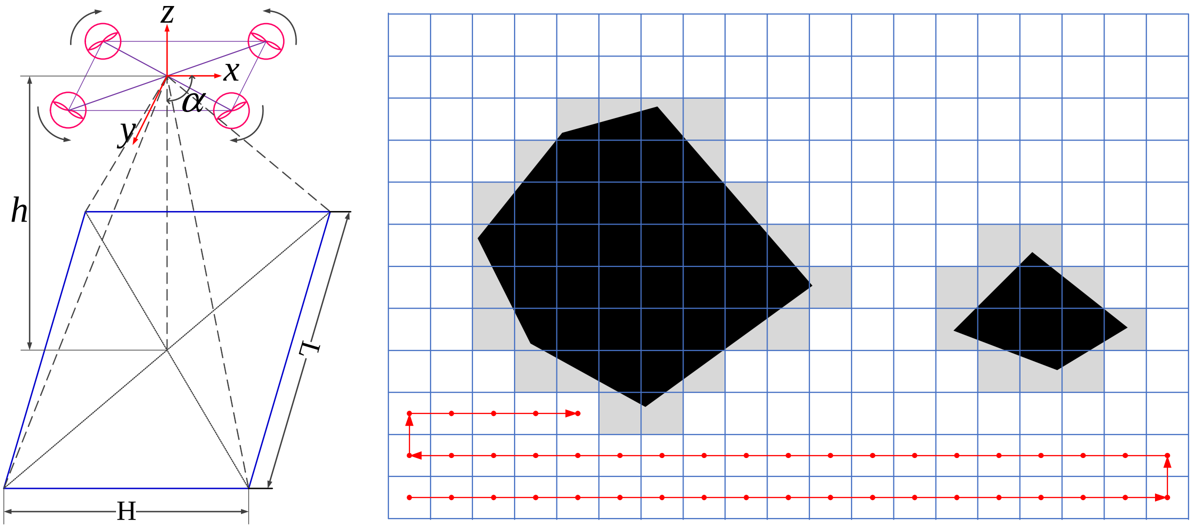
Figure 2.20: Exact cellular decomposition method

The boustrophedon decomposition [45] addresses the issue of redundant movements by merging together cells between successive IN and OUT movements. The major difference is in the case of a MIDDLE event, where no new cell is formed, but the current cell is updated in width. So cells are only created on change of connectivity, as a result the number of cells are less than in the case of trapezoidal decomposition. The point which changes the connectivity of the cell is termed as a critical point. In the implementation of this algorithm, the MIDDLE event is replaced by two new events called – FLOOR and CEILING. The FLOOR event corresponds to vertices on top of obstacles, while CEILING event corresponds to vertices that are at the bottom of obstacles. Both an IN event as well as an OUT event are associated with FLOOR and CEILING pointers. So for a given cell, CEILING and FLOOR pointers points to the top and bottom of a cell.

In [225], Acar *et al* generalized the boustrophedon decomposition by proposing a cellular decomposition approach based on critical points of Morse functions. In this paper, the authors show that the boustrophedon decomposition is a particular case of Morse decomposition. With respect to the original boustrophedon decomposition, the Morse-based decomposition has the advantage of handling also non-polygonal obstacles. By choosing different Morse functions, different cell shapes are obtained, e.g. circular or spiked cells. Theoretically, Morse-decomposition can be applied to any n-dimensional space. Moreover, they presented a method to perform coverage of planar spaces by detecting the critical points using sensory range information, and a motion-template-based algorithm that ensures to encounter all the critical points in the target area. Therefore, this method allows complete coverage on line.

Approximate cellular decomposition

For approximate cellular decomposition technique, the region of interest is divided into a set of regular cells [44]. The shape of regular cells can be a square, rectangular, triangle, or hexagonal. The size of the cells is proportional to the footprint of the camera in the UAV. The coverage path contains a set of n way-points $(w_1, ..., w_n)$ as in Figure 2.21. Each way-point w_i contains not only information such as latitude, longitude, and altitude, but also navigation command to the vehicle as take-off, landing, speed, the coordinate of the next point, or the command to move to the next point.



(a) UAV picture frame

(b) Grid and way-points

FIGURE 2.21: Approximate cellular decomposition method

2.4 State of the art on the control of quadcopters

The quadrotor UAV is a great platform for control systems research thanks to its nonlinearities and under-actuated configuration. Consequently, research on quadrotors UAV has attracted a lot of attention from researchers and industrials all over the world in recent years. Thanks to the wide development of modern control theory and technology, many advanced control algorithms have been researched and used successfully for quadrotors controlling.

The quadrotor UAV controllers can be classified into three main categories: (i) linear flight controllers, (ii) model-based nonlinear flight controllers, (iii) learning-based control methods.

2.4.1 Proportional Integral Derivative controller

Proportional Integral Derivative (PID) controller is one of the most successful and widely used among linear controllers. It is indeed the most applied controller in the industry [104]. The classical PID controller has the advantage that it has a simple structure, high stability, good robustness, and parameters are easy to adjust but needs experience.

In [58], a Disturbance Observer Based (DOB) PID controller for a quadrotor testbed has been developed which can stabilize the quadrotor and enhance the performance of the flight control under complex real-time flight scenarios. In [133], the authors propose a linear time-invariant controller consists of a Proportional Derivative (PD) controller and a robust compensator. This controller achieves the robust attitude control for uncertain quadrotors.

Another PID controller [127] is applied for regulating both the position and orientation of a quadrotor. The parameter gains of the controller are chosen manually. Small response time, almost zero steady-state error, and slight overshoot show the performance of the controller for attitude stabilization. In [197] a PID controller that can perform a steady flight of quadrotor from one point to another is designed while ignoring air resistance and external factors. Another Proportional Integral (PI)/PID controller which reduces the steady-state error is proposed in [86]. However, this controller is not reasonable and effective to compensate for Coriolis force.

A nonlinear PID controller is proposed in [211]. The structure of this nonlinear PID controller consists of a linear PID control module and an inertia torque compensation department module. The effectiveness of this controller for attitude stabilization and robustness of the unmodeled dynamic system is shown in the simulation results.

Although the PID controller has been successfully applied for controlling the quadcopter, there exists some disadvantages. The PID controller gains turning is a challenge as it must be conducted around the equilibrium point to give a good performance. Therefore, this is a time-consuming process. In addition, the classic PID controller might not really suit for quadrotor control when it operates in an unstable environment.

2.4.2 Linear Quadratic Regulator (LQR)/Gaussian-LQR/G

The Linear Quadratic Regulator (LQR) is a popular optimal control algorithm and have been successfully used for controlling quadrotors. The LQR operates a dynamic system in the space state form by minimizing a suitable quadratic objective function. By adjusting the system control gain, the LQR controller can adapt each control circuit performance coordinate with system requirements automatically.

The LQR algorithm is applied for quadcopter in [29]. The authors also compare its performance to the performance of PID controller. The two controllers developed on this paper provide average results. However, the conclusion about the better performance of the LQR controller than the PID is not indicated clearly.

In [48], the authors propose a simple path-following LQR controller on a full dynamic model quadrotor. The simulation results show that by using optimal real-time trajectories under the presence of wind and other disturbances, the quadcopter still tracks the predefined path. Nevertheless, the quadcopter loses path tracking after avoiding the obstacles.

However, these algorithms are suitable for linear systems and it is not convenient for nonlinear systems unless some hypotheses are made. In [252], an LQR is proposed for controlling the sensitivity and reaction speed of a quadrotor. The simulation results show that the performance of the linear LQR model is better than the nonlinear model for real-time and speed reaction.

The LQR algorithm becomes Linear Quadratic Gaussian (LQG) when it combines with Kalman Filter. The LQG algorithm is really effective for the systems with Gaussian noise or incomplete state information. In [150], an LQG with integral action is developed for quadrotor attitude stabilization and gives good results in hover mode.

2.4.3 Sliding Mode Control (SMC)

SMC is a nonlinear control algorithm that is robust to parameter variations and model uncertainties and is insensitive to external disturbances. SMC uses a discontinuous control signal to the system to make sure that the system slides along a prescribed path. The main advantage of SMC is that it does not simplify the dynamics through linearization and has good tracking. However, the main disadvantage of SMC is the chattering phenomenon.

A PID based sliding mode controller [141] is proposed for a micro quadrotor. This SMC is able to make the micro quadrotor accurately track the desired altitude and attitude. In [243], the quadrotor system was subdivided into the full-actuated and under-actuated systems. An SMC is designed for stabilizing the cascaded under-actuated systems. Simulation results show the robustness and good stability of the quadcopter.

In [53], a Super Twisting Algorithm (STA) based on the second-order sling mode technique is designed and implemented for attitude tracking of the quadrotor. The good performance of the proposed method in stabilization, reference tracking, and disturbance rejection cases are shown in experimental results. Beside, in [194], the authors propose a sliding mode controller based on the Lyapunov stability theory. The proposed SMC controller gains the ability to track the predefined path and drive the quadrotor to the desired position while injecting noises. An SMC controller is designed in [24] that improves the robustness to the external disturbances and model uncertainty of the system. This proposed SMC does not require to resort to high power gain and is able quickly to compensate for the changes of the external disturbances.

2.4.4 Backstepping Control

Backstepping control is a recursive algorithm that decomposes the controller into several steps and makes each step stable progressively. The advantage of backstepping control is that this algorithm converges fast thanks to less computational resources and it has the ability to reject disturbances well. However, its disadvantage is that its robustness is not good in the presence of modeling uncertainties.

A DOB based backstepping controller in [59] is designed for high-performance trajectory tracking of the quadrotor. This disturbance observer-based controller serves as a compensator, which effectively rejects external disturbances and model mismatches. In [142], the quadrotor system is divided into under-actuated, fully actuated, and propeller subsystems. Then, a backstepping control is designed for stabilizing the quadcopter. The simulation results show that good tracking was achieved for position and yaw angle. The roll and pitch angles are stabilized by using the Lyapunov stability theory.

Besides, a nonlinear controller using Lyapunov-based backstepping techniques for path following of quadrotor is designed in [35]. The controller in this approach guarantees the global convergence of the closed-loop path following error to zero under the presence of constant wind disturbances

In [71], an integrator is added and the traditional backstepping algorithm becomes integral backstepping control for increasing the robustness of the general backstepping controller for a quadcopter under external disturbances. The integrator in this integral backstepping controller is able to eliminate the steady-state errors of the system, reduces response time and restrains the overshoot of the control parameters.

2.4.5 Adaptive Control Algorithms

Adaptive control is a robust control technique which is aimed at adapting to parameter changes in the system to obtain the optimal control effect during operating time. The parameters are either uncertain or time-varying. Adaptation means a combination of online parameter estimation and control, whereby an appropriate controller is selected on the knowledge of the current estimate for the uncertain process.

In [54], a continuous time-varying adaptive controller is developed for quadcopter with known uncertainties in mass, moments of inertia, and aerodynamic damping coefficients. A decentralized adaptive controller is presented in [151]. This controller is designed with the improved Lyapunov based Model Reference Adaptive Control technique. Simulation results show the proposed controller works properly for a quadrotor with parametric and non-parametric uncertainties under various conditions during flight time.

2.4.6 Robust Control Algorithms

A robust controller is designed to deal with uncertainty in the system parameters or disturbances. This controller ensures the controller performance within acceptable disturbance ranges or un-modeled system parameters. The main disadvantage of the robust controller is poor tracking ability.

In [134], the authors develop a robust motion controller based on a nominal controller and robust compensator for attitude and position control of a quadrotor. The proposed robust compensator minimizes the influence of uncertainties, such as nonlinear dynamics, coupling, parametric uncertainties, and external disturbances in the rotational and translational dynamics.

Another robust tracking controller is designed in [221]. This controller guarantees asymptotic stability under the presence of parametric uncertainties and unknown nonlinear disturbances.

2.4.7 Optimal Control Algorithms

Several optimal controllers have been successfully applied for controlling the quadcopter, such as LQR, L_1, H_∞, and Kalman filter. The main disadvantage of optimal control is generally their poor robustness. An optimal controller for reference tracking of both attitude and heading of a quadcopter which is robust and L_1 optimal is proposed in [201]. The performance of the controller for error minimization and rejection of persistent disturbances is validated by the experiment.

In [186], the authors design an integral optimal predictive H_∞ controller for stabilization of the rotational movement of a quadrotor and for path following. The simulation shows that the controller has good robustness and is able to reject the external disturbances.

Another H_∞ controller is designed in [42] which is based on quasi-LPV. This controller can perform the quadrotor control of vertical speed and attitude. The loop shaping H_∞ is developed to combine robust control and classic loop shaping, which overcome the usual drawbacks of the standard H_∞. Some other H_∞ controller for quadcopter are also shown in [188] and [70].

2.4.8 Model Predictive Control (MPC)

MPC is a type of control that uses an explicit model of the plant to predict future output behavior while minimize the tracking error of future horizon by solving optimal control problems online [114]. The main advantage of MPC compared to Backstepping, feedback linearization, and also sliding mode, is that this controller is able to handle operational constraints prevalent in a control system explicitly. However, a typical disadvantage of MPC is the need for high computational power due to repetitive calculations online.

In [2], an efficient MPC was successfully implemented which deploys fewer prediction points and the less computational requirement is presented in order to control a quadrotor with limited airborne computational power.

2.4.9 Exact Feedback Linearization

Feedback linearization is a process of converting exactly, by nonlinear state or output feedback control a nonlinear system into a linear system so that the linear feedback control techniques can be applied. After that, the controller is designed by theory related to the linear control system to achieve the requirements of controlling the system. The main disadvantage of exact feedback linearization is the need of an exact model. This leads to the complexity in controller design.

In [154], the quadcopter dynamic is divided into the inner loop (for attitude) and the outer loop (for position). The authors ignore some factors such as air resistance and design feedback linearization controller to control the attitude of the quadcopter. Besides, the combination of feedback linearization and GH_∞ controllers are applied to design the controller for a quadcopter. This controller takes into account also the parameter uncertainties, noises, and external disturbances. Consequently, the designed controller gains a high quality of robustness.

2.5 Conclusion

In this chapter, a short review of UAVs, their classifications, applications, and advantages and disadvantages has been analyzed. Then, the state of the art of complete CPP[1] of UAVs and control of quadrotors are also reviewed.

---

[1] Coverage Path Planning (CPP)

3

System modeling

Chapter abstract

This chapter is dedicated to system modeling of quadrotor. In section 3.1, we offer a brief review of the previous methods for modeling the quadrotor. Then the concepts and generalities for writing the dynamic model of the quadrotor are given in the section 3.2. Next, the quadrotor kinematic is described in section 3.3. The knownledge about the forces and moments acting on the quadrotor is required for building the dynamic system differential equations of quadrotor. Therefore, the forces and moments acting on the quadrotor are discussed in section 3.4. Then, we get the dynamic model of the quadrotor by using the Euler-Lagrange formalism in section 3.5. The chapter concludes with small conclusion.

This Chapter contains:

3.1 Review of the multirotors modeling

Since there are 6 degrees of freedom (DOF) but controlled by only 4 input variables (the speeds of four propellers), the quadcopters are underactuated, highly dynamic, nonlinear, and coupled systems. In addition, the uncertainties, caused by the environment and induced by aerodynamic phenomena, lead to challenging modeling task. By using a number of assumptions, the factors that are too small and that have a negligible influence on quadcopters dynamics can be ignored. These simplified models can be used to approximate the vehicle dynamics requiring effective control strategies. Many models have been studied in the literature, with various levels of complexity and completeness.

Mesicopter project [72], started in 1999 and ended in 2001. It aimed to study the feasibility of a centimeter scale quadrotor. In this paper, the author derived a dynamic model for a small quadrotor called Mesicopter. The Mesicopter was the first proposed quadrotor model that is used for hovering. The Mesicopter model was generated by assuming that the dynamics are decoupled along the two planes of symmetry. The author has also explicitly given the expression of some aerodynamic forces such as thrust and hub.

In [62], the authors proposed a dynamic model for X-shaped quadrotor. In this paper, the attitude dynamics are modeled by double integrators and ZYX-Euler angles representation was used for describing the rotations of the quadcopter in the East-North-Up (ENU) standard of frames. Beside, a compact quadcopter model was proposed in [213]. The equations of motion of quadcopter were considered in the North-East-Down (NED) reference frame by using Newton's equations of motion of a rigid object. Several aerodynamic forces and moments are neglected such as the drag force were assumed in this model.

In [216], the author took into account the gyroscopic effect due to the rotation of propellers considered as rigid rotating disks. In [179], an extended model of the quadrotor is used by adding the dynamics of the rotor to the motion equation of the quadrotors. The authors related the applied inputs to the angular velocities of the rotors. The dynamics of the rotor are approximated by a first-order system. Then for control purposes, The final model of the quadrotor is simplified.

In [61], the drag forces that affect the translational dynamics were added to the equations of motion of the quadrotors. These forces were considered as functions of the velocities. The same model was also derived by using the Euler-Lagrange approach as in [167].

In the PhD thesis of Bouabdallah in [30] and in his paper [195], the dynamic model of a cross quadrotor was studied. This model included also the blade flapping effect. In this thesis, both Euler-Lagrange and Newton-Euler approaches are used for the modeling dynamic model of the quadcopter. However, the various effects are not well defined in the appropriate frames, which lead to some errors that are reported and rectified later in the thesis of Wang in [232].

Hoffmann's work in [79] showed that the flapping tips the thrust vector away from the horizontal plane. As a result, some residual forces influence attitude control as well as the total thrust force. After that, this phenomenon has been more extensively studied in [189]. Other aspects such as the non-rigidity of propellers [177], the ground effect [126][20], etc have also been further studied more.

In this chapter, we propose a dynamic model of the quadcopter based on Wang's model under the ideal flight conditions (i.e without wind). In our model, we consider the effect of the hub forces as well as the hub moment around Z_B-axis.

3.2 Concepts and Generalities

The Euler angles are the three angles provided by Leonhard Euler and are the basis of a method to represent the 3D orientation of an object with respect to a fixed coordinate system using a combination of three rotations around different axes. The reference frames used for describing the Euler angles are Inertial frame, Vehicle 1 frame, Vehicle 2 frame and body-fixed frame. The inertial frame axes are Earth-fixed, the body frame axes are aligned with the sensor, while the Vehicle-1 and Vehicle-2 are intermediary frames on the sequence of transform operations the inertial frame to the body frame.

The Inertial Frame

The inertial frame $R_E\left(O_E, X_E, Y_E, Z_E\right)$ as shown in Figure 3.1, is an Earth-fixed set of axes that is used as an unmoving reference.

The sequence of rotations used to represent a given orientation is first yaw, then pitch, and finally roll.

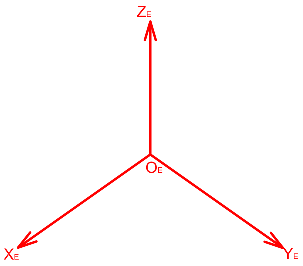


FIGURE 3.1: Inertial Frame

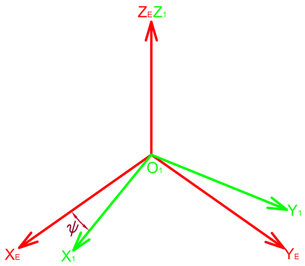


FIGURE 3.2: Vehicle-1 Frame

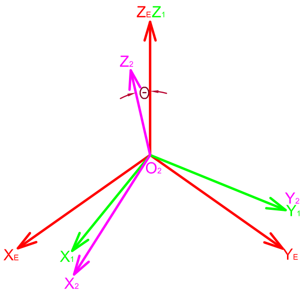


FIGURE 3.3: Vehicle-2 Frame

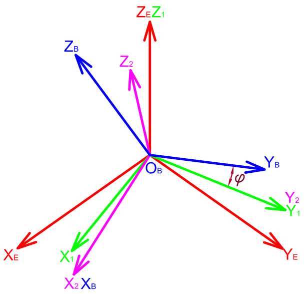


FIGURE 3.4: Body fixed Frame

The Vehicle-1 Frame (Yaw Rotation)

Yaw is the angle represents rotation about the inertial-frame $O_E Z_E$-axis by an angle ψ. The yaw rotation produces a new coordinate frame $R_1\left(O_1, X_1, Y_1, Z_1\right)$ where the $O_1 Z_1$-axis is aligned with the inertial frame and the x and y axes are rotated by the yaw angle ψ. The new coordinate frame $R_1\left(O_1, X_1, Y_1, Z_1\right)$ is called vehicle-1 frame and shown in Figure 3.2.

Rotation of a vector from the Inertial Frame R_E to the Vehicle-1 Frame R_1 can be performed by multiplying the vector by the rotation matrix

$$R_I^{v1}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The Vehicle-2 Frame (Yaw and Pitch Rotation)

The pitch angle quantifies the rotation of the vehicle 1 frame around the vehicle-1 frame O_1Y_1-axis by an angle θ. The pitch rotation produces a new coordinate frame $R_2\,(O_2, X_2, Y_2, Z_2)$ where the O_2Y_2-axis is aligned with the vehicle-1 O_1Y_1-axis and the x and z axes are rotated by the yaw angle θ. The new coordinate frame $R_2\,(O_2, X_2, Y_2, Z_2)$ is called vehicle-2 frame and shown in Figure 3.3. It is important to note that pitch is not rotation about the inertial-frame O_EY_E-axis.

The rotation matrix for moving from the vehicle-1 frame R_1 to the vehicle-2 frame R_2 is given by

$$R_{v1}^{v2}\,(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.2}$$

The rotation matrix for moving from the inertial frame R_E to the vehicle-2 frame R_2 is the multiplication of the yaw matrix and the pitch matrix:

$$R_I^{v2}\,(\theta,\psi) = R_I^{v1}\,(\theta)\,R_{v1}^{v2}\,(\psi) \tag{3.3}$$

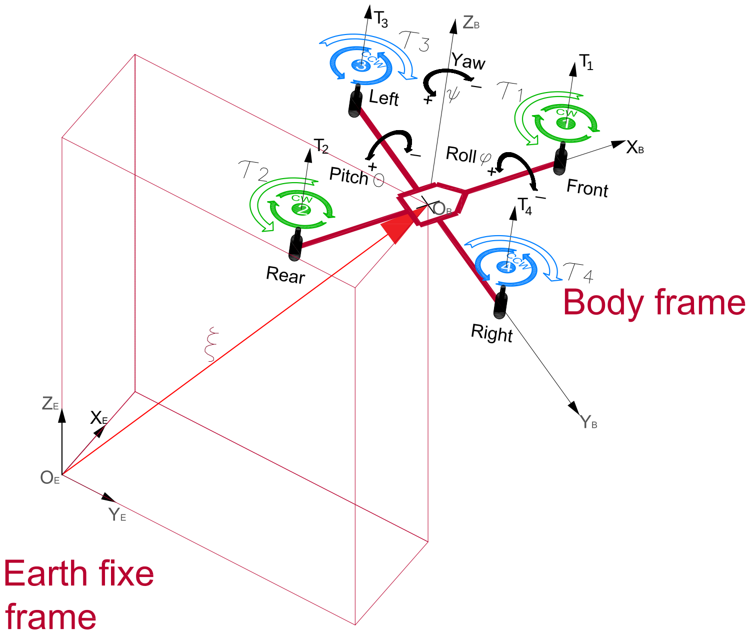


FIGURE 3.5: The Inertial Frame to Body frame

The Body Frame (Yaw, Pitch, and Roll Rotation)

The body reference frame $R_B\,(O_B, X_B, Y_B, Z_B)$ is produced from the rotation vehicle-2 frame R_2 about the vehicle-2 O_2X_2-axis by an angle ϕ as shown in Figure 3.4. The body frame is the coordinate system that is aligned with the body of the aircraft as in Figure 3.5. For the quadcopter, the body-frame O_BX_B-axis is the axis that passes from the center of the quadcopter O_B to the first motor, while the body-frame O_BY_B-axis is the axis that passes from the center of the quadcopter O_B to the fourth motor.

The rotation matrix for moving from the vehicle-2 R_2 frame to the body frame R_B is given by

$$R_{v_2}^{B}(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \tag{3.4}$$

Consequently, the complete rotation matrix for moving from the inertial frame R_E to the body frame R_B is given by

$$R_I^B(\varphi,\theta,\psi) = R_{v2}^B(\varphi)\, R_{v1}^{v2}(\theta)\, R_I^{v1}(\psi) \tag{3.5}$$

Performing the multiplication, the complete rotation from the inertial frame R_E to the body frame R_B is given by

$$R_I^B(\varphi,\theta,\psi) = \begin{bmatrix} \cos\psi\cos\theta & \cos\theta\sin\psi & -\sin\theta \\ \cos\psi\sin\varphi\sin\theta - \cos\varphi\sin\psi & \cos\varphi\cos\psi + \sin\varphi\sin\theta\sin\psi & \sin\varphi\sin\theta \\ \sin\varphi\sin\psi + \cos\varphi\cos\psi\sin\theta & \cos\varphi\sin\psi\sin\theta - \cos\psi\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \tag{3.6}$$

The rotation matrix for moving the opposite direction from the body frame R_B to the inertial frame R_E is given by

$$R_B^I(\varphi,\theta,\psi) = R_I^{v1}(-\psi)\, R_{v1}^{v2}(-\theta)\, R_{v2}^B(-\varphi) \tag{3.7}$$

Performing the multiplication, the complete rotation from the body frame R_B to the inertial frame R_E is given by

$$\begin{aligned} R_B^I(\varphi,\theta,\psi) &= R(x,\varphi)\, R(y,\theta)\, R(z,\psi) \\ &= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \sin\varphi\cos\psi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \end{aligned} \tag{3.8}$$

Note that all this is reverse the order of operations and reverse the direction of rotation.

Remark 3.1. *The Euler angles ϕ, θ, and ψ are bounded as follows*

$$\begin{array}{l} -\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2} \\ -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \\ -\pi \leq \psi \leq \pi \end{array} \tag{3.9}$$

$\square$

Based on the previous description, the transformation between the Earth-fixed frame and the Body-fixed frame can be explicitly expressed using the rotation matrix R. So, the velocity vector $V^B = \begin{bmatrix} u & v & w \end{bmatrix}^T \in \mathbb{R}^3$ of the vehicle, expressed in a Body-fixed frame, can be rotated into the Earth-fixed frame as follows

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{s}_\varphi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\varphi \mathrm{s}_\psi & \mathrm{c}_\varphi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\varphi \mathrm{s}_\psi \\ \mathrm{c}_\theta \mathrm{s}_\psi & \mathrm{s}_\varphi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\varphi \mathrm{c}_\psi & \mathrm{c}_\varphi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\varphi \mathrm{c}_\psi \\ -\mathrm{s}_\theta & \mathrm{s}_\varphi \mathrm{c}_\theta & \mathrm{c}_\varphi \mathrm{c}_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.10}$$

where $\dot{\xi} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \in \mathbb{R}^3$ is the velocity vector of the quadrotor in the Earth fixed frame R_E, c_* denotes $\cos *$ and s_* denotes $\sin *$.

3.2.1 Quadcopter model

A quadcopter is a helicopter that consists of a rigid cross-frame equipped with four rotors M_1, M_2, M_3, and M_4 which rotate at angular speed ω_1, ω_2, ω_3, and ω_4 respectively as shown in Fig. 3.6. Each rotor M_i (for $i = 1,..,4$) generates thrust f_i which is proportional to the square of the angular speed ω_i, means that $T_i = k\omega_i^2$. The direction of rotation of the rotors are fixed (i.e., $\omega_i \geq 0, i \in \{1,2,3,4\}$), consequently T_i is always a positive quantity.

The main thrust u is the sum of all the thrusts which are generated by the rotors, that is

$$u = \sum_{i=1}^{4} T_i \tag{3.11}$$

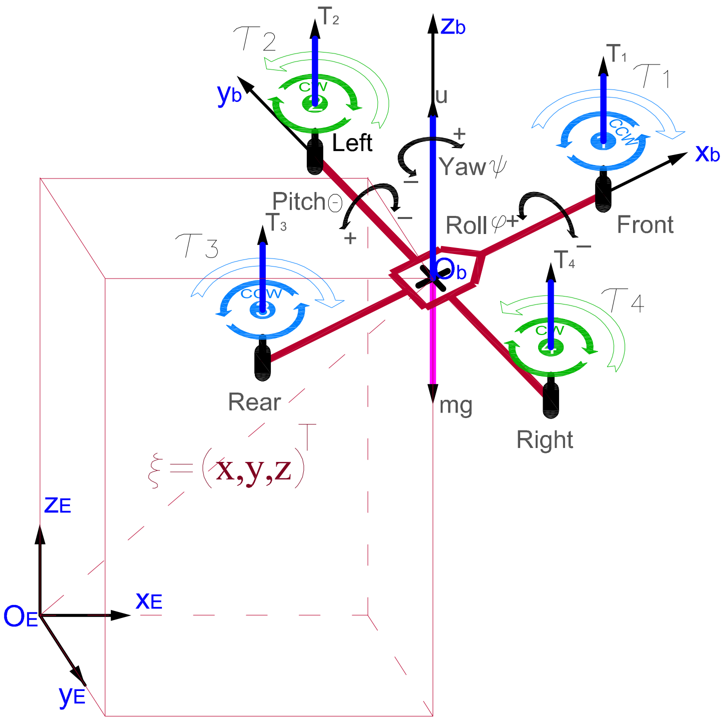


Figure 3.6: Quadcopter configuration with coordinate frames and forces

In order to avoid the yaw drift due to the reactive torques, the quadrotor aircraft is configured such that the set of rotors M_2, M_4 (left-right) revolves at angular speeds ω_1 and ω_2 in clockwise (CW) direction generating thrusts of T_1 and T_3, while the pair of rotors M_1, M_3 (front-rear) rotates at angular speeds ω_2 and ω_4 in counterclockwise (CCW) direction generating thrusts of T_2 and T_4.

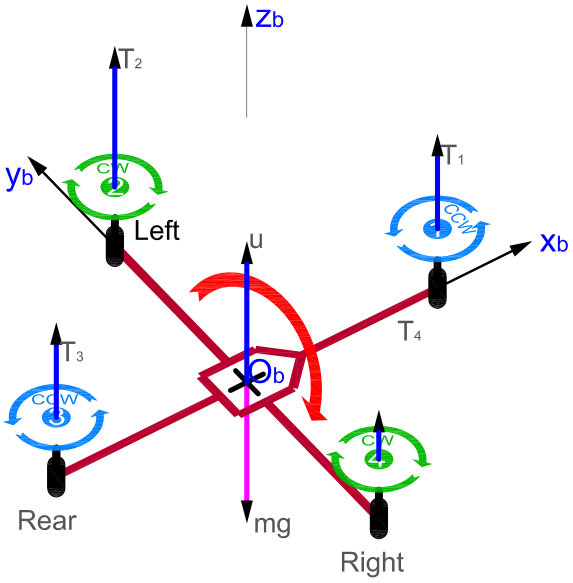


FIGURE 3.7:
Roll motion

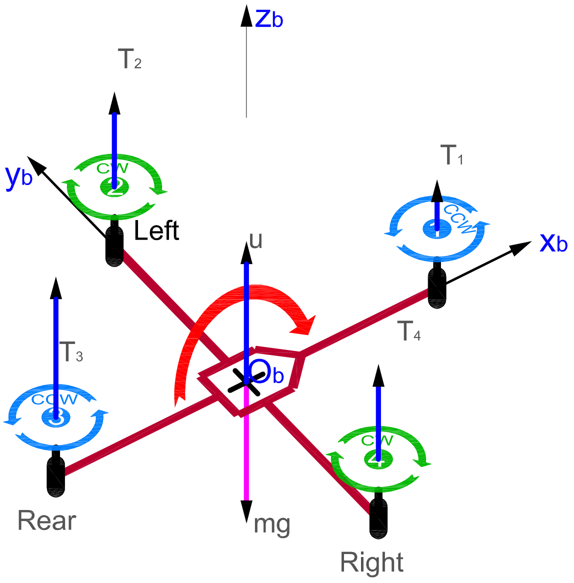


FIGURE 3.8:
Pitch motion

The quadcopter is controlled by varying the angular speeds $\omega_i, i = 1,2,3,4$ of the four rotors $M_i, i = 1,2,3,4$. The forward/backward, left/right, and the yaw motions are achieved through a differential control strategy of the thrust generated by each rotor.

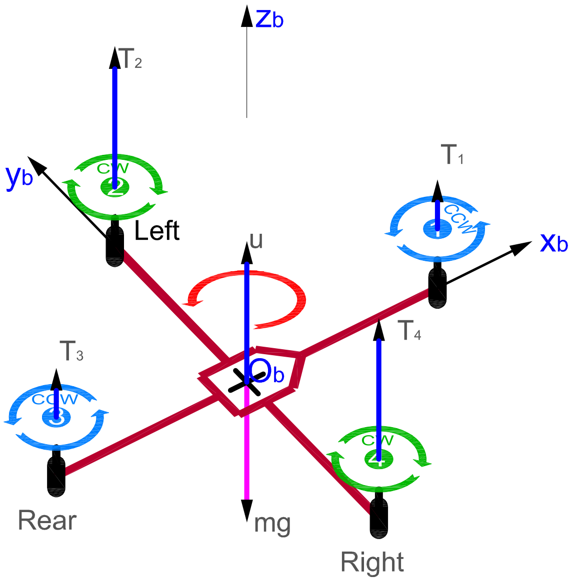


FIGURE 3.9:
Yaw motion

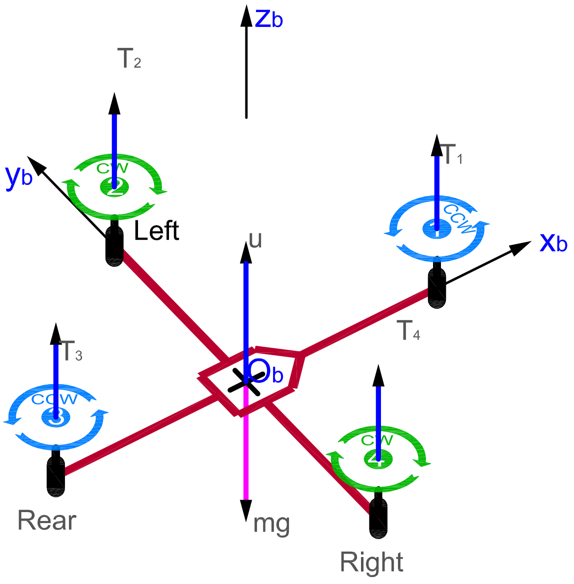


FIGURE 3.10:
Hovering motion

Forward pitch motion is obtained by increasing the speed of the rear motor M_3 while reducing the speed of the front morot M_1 as in Figure 3.8. Contrary, backward pitch motion is obtained by reducing the speed of the rear motor M_3 while increasing the speed of the front motor M_1. Similarly, the roll right motion is obtained by increasing the speed of the left motor M_2 while reducing the speed of the right motor M_4 as in Figure 3.7. Contrary, the roll left motion is obtained by reducing the speed of the left motor M_2 while increasing the speed of the right motor M_4. Yaw motion is obtained by increasing the torque of the front and rear motors (τ_1 and τ_3) while decreasing the torque of the lateral motors (τ_2 and τ_4) as in Figure 3.9 or inverse. Such motions can be performed while maintaining the total thrust constant.

The hovering motion is obtained by maintaining constant speed for all the motors such that the total thrust is equal to mg as in Figure 3.10 while compensating disturbances.

The Z motion is obtained by maintaining the same speed for all the motors. The quadrotor moves up if the total thrust is greater than mg as in Figure 3.11, and moves down if the total thrust is less than weight mg.

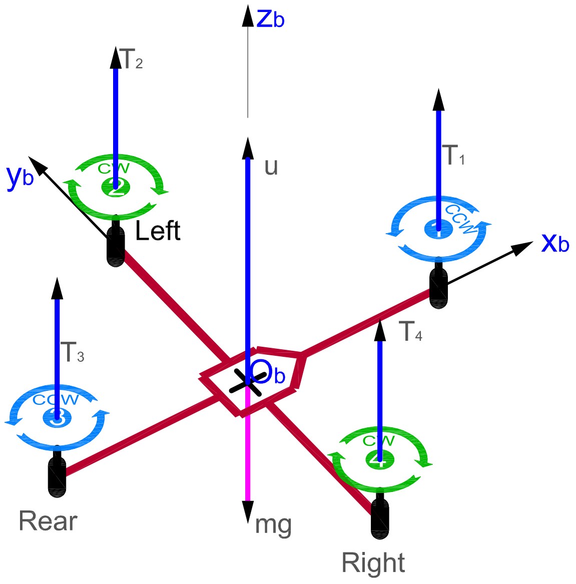


FIGURE 3.11: Z motion

3.3 Helicopter kinematics

Let **q** be the coordinate vector of the quadcopter which is expressed by the following equation

$$\mathbf{q} = (x, y, z, \varphi, \theta, \psi) \in \mathbb{R}^6 \tag{3.12}$$

where $\xi = (x, y, z) \in \mathbb{R}^3$ denotes the position vector of the center of mass of the quacopter relative to a fixed inertial reference frame R_E, while $\eta = (\varphi, \theta, \psi) \in \mathbb{R}^3$ is the vector of three Euler's angles which denotes the orientation of the quadcopter in R_E. The coordinates of the quadcopter is depicted in Fig. 3.5.

The linear velocities in the body frame are determined by V_B, and the angular velocities by Ω

$$V_B = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}; \quad \Omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.13}$$

The relationship between the angular velocity vector in the body frame and the generalized velocities $\dot{\eta}$ [85] is depicted as

$$\begin{aligned} \Omega &= W_\eta \dot{\eta} \\ \dot{\eta} &= W_\eta^{-1} \Omega \end{aligned} \tag{3.14}$$

where

$$W_\eta = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \cos\theta\sin\varphi \\ 0 & -\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \tag{3.15a}$$

$$W_\eta^{-1} = \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta \end{bmatrix} \tag{3.15b}$$

Remark 3.2. *The matrix W_η is invertible if*

$$\theta \neq (2k-1)\pi/2, k \in Z$$

□

The quadrotor is assumed to have symmetric structure with the four arms aligned with the body x- and y-axes. Thus, the inertia matrix is diagonal matrix I in which

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{3.16}$$

where I_x, I_y, and I_z are the moments of inertia with respect to the axis x, y, and z respectively.

3.4 Applied forces and moments on the quadcopter

3.4.1 Applied forces

The ground effects are neglected, then three main forces acting on the quadcopter are considered: gravity force, thrust forces, and hub forces.

- **Gravity**: Gravity is the force by which a planet or other body draws objects toward its center and denoted as

  $$G = -mg \tag{3.17}$$

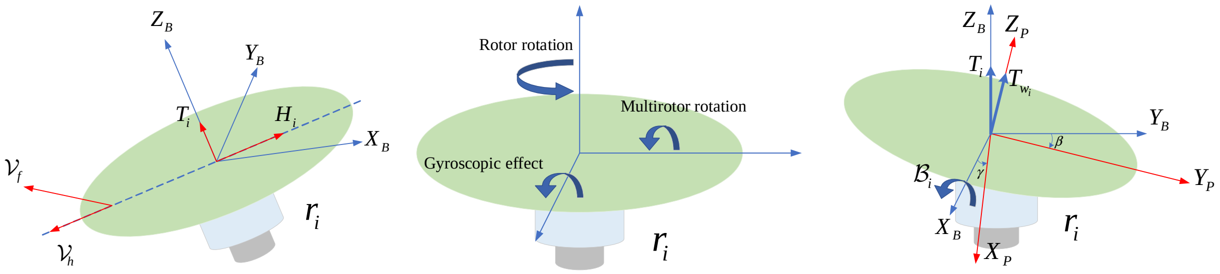
  Gravity is along the axis Z_E in the negative direction with g is the gravity acceleration.

- **Thrust**: Each rotor r_i produces the thrust T_i which depends on its angular velocity Ω_i. The thrust force T_i is along the Z_B axis in the positive direction. Thrust T_i is depicted in Figure 3.12a. The equation for thrust is

  $$T_i = C_T \rho A (\Omega_i R_r)^2 \tag{3.18}$$

  where C_T is the aerodynamic coefficient, $\rho = 1.293 kg/m^3$ is the air density, A is the effective propeller disk area, R_r is the propeller radius.

- **Hub forces**: Each rotor r_i also produces the hub force H_i which is also dependent on its angular velocity Ω_i. The hub force H_i is in the plane X_BY_B on the negative direction of the horizontal velocity $\mathcal{V}_h$ (the projection of the forwarding velocity $\mathcal{V}_f$ in the X_BY_B plane). Thus, the hub force can be decomposed into two components $\mathcal{H}_{xi}$ along X_B-axis and $\mathcal{H}_{yi}$ along Y_B-axis in the Body-fixed frame. The Hub force H_i and its two components $\mathcal{H}_{xi}$ and $\mathcal{H}_{yi}$ are depicted in Figure 3.12a. The equation for hub force is
$$H_i = C_H \rho A (\Omega_i R_r)^2 \tag{3.19}$$
where C_H is the aerodynamic coefficient.



(a) Forces induced by rotor **(b)** Rotor gyroscopic effect **(c)** Blade flapping

FIGURE 3.12: Aerodynamic phenomena [32]

Thus, the total thrust force T, and hub forces H_{xi}, H_{yi} on x and y axes acted on the quadcopter can be calculated as
$$T = \sum_{i=1}^{4} T_i \tag{3.20}$$
$$H_x = \sum_{i=1}^{4} H_{xi} \tag{3.21}$$
$$H_y = \sum_{i=1}^{4} H_{yi} \tag{3.22}$$

Once the main forces have been reviewed, the net external force vector acting on the vehicle is expressed in the Body-fixed frame as:
$$\begin{aligned} F^B &= \begin{bmatrix} H_x \\ H_y \\ T \end{bmatrix} + R_I^B \begin{bmatrix} 0 \\ 0 \\ G \end{bmatrix} \\ &= \begin{bmatrix} -\sum\limits_{i=1}^{4} H_{xi} \\ -\sum\limits_{i=1}^{4} H_{yi} \\ \sum\limits_{i=1}^{4} T_i \end{bmatrix} + R_I^B \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \end{aligned} \tag{3.23}$$

where R_I^B is the rotation matrix from the Earth-fixed frame R_E to the Body-fixed frame R_B as defined on 3.6.

3.4.2 Applied moments

In this section, the main external moments that are expressed in the Body-fixed frame and their directions will be presented.

- **Rolling and Pitching moments**: Roll moments $\mathcal{M}_\phi$ around X_B and pitch moment $\mathcal{M}_\theta$ around Y_B axes, are achieved by the difference in combined thrusts in the opposite sides of the vehicle. These moments are really important because they contribute to the control of the vehicle. The directions of these moments are decided according to the right hand rule. The Roll moments $\mathcal{M}_\varphi$ and pitch moment $\mathcal{M}_\theta$ can be defined as

$$\begin{aligned} \mathcal{M}_\varphi = & \ L \sum_{i=1}^{4} \sin \alpha_i T_i \\ \mathcal{M}_\theta = & -L \sum_{i=1}^{4} \cos \alpha_i T_i \end{aligned} \tag{3.24}$$

  where α_i denotes the angle between the i-th arm and X_B-axis.

- **Yawing moment**: Moment Q_i around the shaft of the i-th rotor is generated by the acting of the aerodynamic forces on the blade elements. Thus the yawing moment Q_i for each rotor is defined as:

$$Q_i = C_Q \rho A R_r (\Omega_i R_r)^2 \tag{3.25}$$

  where C_Q is the aerodynamic coefficient.

  The sum of these moments, considering all the rotors, induces the rotation of the quadrotor about Z_B axis. There are 4 rotors that are separated by different rotational directions (clockwise and counterclockwise) inducing moments in the different direction. Consequently, the total yawing moment is given by:

$$Q = \sum_{i=1}^{4} S_{pi} Q_i \tag{3.26}$$

  where $S_{pi} = 1$ if i-th rotor rotates in the clockwise direction and $S_{pi} = -1$ if i-th rotor rotates in the counter clockwise direction.

- **Hub moment**: As we have seen above, the hub force has two components H_x and H_y along X_B and Y_B axes respectively. These hub forces generate moments on the vehicle around the Z_B-axis. These two components contribute to the motion according to their locations $o_i, i = 1, ..., 2N_r$, and can be defined as.

$$\begin{aligned} \mathcal{H}_x &= L \sum_{i=1}^{4} \sin \alpha_i \mathcal{H}_{x_i} \\ \mathcal{H}_y &= -L \sum_{i=1}^{4} \cos \alpha_i \mathcal{H}_{y_i} \end{aligned} \tag{3.27}$$

  Therefore, the resulting hub moment is the sum of the moment of the two component $\mathcal{H}_x$ and $\mathcal{H}_y$, given by the following equation

$$\begin{aligned} \mathcal{H} = & \ \mathcal{H}_x + \mathcal{H}_y \\ = & \ L \sum_{i=1}^{4} \sin \alpha_i \mathcal{H}_{x_i} - L \sum_{i=1}^{4} \cos \alpha_i \mathcal{H}_{y_i} \end{aligned} \tag{3.28}$$

  where the moment $\mathcal{H}_x$ is generated by the force H_x and the moment $\mathcal{H}_y$ is generated by the force H_y.

- **Gyroscopic moment**: Rotations about two axes causes a third rotation about a third axis, which is perpendicular to the plane formed by the two former axes (Figure 3.12b). This effect is called the gyroscopic effect. Consequently, when the quadrotor rotates along the X_B axis, another moment along Y_B axis is generated on each spinning rotor. When the quadrotor rotates along Y_B axis, each rotor also has a moment along X_B axis. The direction of moments follows right hand rule. Therefore, the total gyroscopic components can be calculated as:

  $$\begin{aligned} \mathcal{G}_x = &\quad J_r q \sum_{i=1}^{4} S_{p_i}\Omega_i \\ \mathcal{G}_y = &\quad -J_r p \sum_{i=1}^{4} S_{p_i}\Omega_i \end{aligned} \tag{3.29}$$

  where J_r is the moment of inertia of the rotor (consider the moments of inertia of all the rotor are equal), $S_{p_i} = 1$ if i-th rotor rotates in the clockwise direction and $S_{p_i} = -1$ if i-th rotor rotates in the counter clockwise direction.

- **Blade flapping moment**: When the quadrotor is in the translational flight mode, there exists a difference in lift between the blades that advance and the blades that recede. This effect is called the blade flapping. The difference of the lifts generates a moment to the rotor disk as shown in Figure 3.12c). The flapping moment $\mathcal{B}_i$ is around an axis perpendicular to the plane formed by the rotor shaft and the forward velocity of the multirotor and is given by the following equation:

  $$\mathcal{B}_i = C_B \rho A R_r (\Omega_i R_r)^2 \tag{3.30}$$

  where C_B is the aerodynamic coefficient.

  The flapping moment $\mathcal{B}_i$ of each i-th rotor can be separated into two components $\mathcal{B}_{x_i}$ and $\mathcal{B}_{y_i}$ with respect to the x and y axes. Therefore, the total blade flapping moment of a quadrotor is:

  $$\begin{aligned} \mathcal{B}_x = &\quad \sum_{i=1}^{4} S_{p_i}\mathcal{B}_{x_i} \\ \mathcal{B}_y = &\quad \sum_{i=1}^{4} S_{p_i}\mathcal{B}_{y_i} \end{aligned} \tag{3.31}$$

  where $S_{p_i} = 1$ if the i-th rotor rotates in the clockwise direction and $S_{p_i} = -1$ if the i-th rotor rotates in the counter clockwise direction.

Based on the moments affecting the quadcopter as previously seen in (3.24), (3.25), (3.28), (3.29), (3.31), the total moment vector affecting the quadcopter is presented in the Body-fixed frame as:

$$\begin{aligned} \mathcal{M}^B = &\begin{bmatrix} \mathcal{M}_\varphi + \mathcal{G}_x + \mathcal{B}_x \\ \mathcal{M}_\theta + \mathcal{G}_y + \mathcal{B}_y \\ Q + \mathcal{H} \end{bmatrix} \\ = &\begin{bmatrix} l\sum_{i=1}^{4} \sin\alpha_i T_i + J_r q \sum_{i=1}^{4} S_{p_i}\Omega_i + \sum_{i=1}^{4} S_{p_i} B_{x_i} \\ -l\sum_{i=1}^{4} \cos\alpha_i T_i - J_r p \sum_{i=1}^{4} S_{p_i}\Omega_i + \sum_{i=1}^{4} S_{p_i} B_{y_i} \\ \sum_{i=1}^{4} S_{p_i} Q_i + l\sum_{i=1}^{4} \sin\alpha_i \mathcal{H}_{x_i} - l\sum_{i=1}^{4} \cos\alpha_i \mathcal{H}_{yi} \end{bmatrix} \end{aligned} \tag{3.32}$$

3.5 Modeling with Euler-Lagrange Formalism

Define the Lagrangian $L(\mathbf{q},\dot{\mathbf{q}})$ as the sum of the translational and rotational energies minus potential energy

$$L(\mathbf{q},\dot{\mathbf{q}}) = T_{trans} + T_{rot} - U \tag{3.33}$$

where T_{trans}, T_{rot}, and U are the translational kinetic energy, rotational kinetic energy, and potential energy respectively and expressed by the following equations

$$T_{trans} = \frac{m}{2}\dot{\xi}^T\dot{\xi} \tag{3.34a}$$

$$T_{rot} = \frac{1}{2}\mathbf{\Omega}^T I \mathbf{\Omega} \tag{3.34b}$$

$$U = mgz \tag{3.34c}$$

where z is the altitude, m is the mass, $\mathbf{\Omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is the vector of angular velocity, and I denotes the inertia matrix of the quadcopter, and g is the gravity acceleration.

Define the Jacobian matrix $J(\eta)$

$$J = J(\eta) = W_\eta^T I W_\eta \tag{3.35}$$

$$\mathrm{J} = \begin{bmatrix} \mathrm{I_x} & 0 & -I_x\sin\theta \\ 0 & I_y\cos^2\varphi + I_z\sin^2\varphi & (I_y - I_z)\sin\varphi\cos\varphi\cos\theta \\ -I_x\sin\theta & (I_y - I_z)\sin\varphi\cos\varphi\cos\theta & I_x\sin^2\theta + I_y\sin^2\varphi\cos^2\theta + I_z\cos^2\varphi\cos^2\theta \end{bmatrix} \tag{3.36}$$

From (3.34b), (3.14), and (3.35) one obtains

$$T_{rot} = \frac{1}{2}\dot{\eta}^T J\dot{\eta} \tag{3.37}$$

The full dynamic model of quadcopter can be obtained by using Euler-Lagrange equations with external forces as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \begin{bmatrix} F_\xi \\ \tau \end{bmatrix} \tag{3.38}$$

where $F_\xi = R(\varphi,\theta,\psi)\hat{F} \in \mathbb{R}^3$ is the translational force acting on the quadcopter, and $\tau \in \mathbb{R}^3$ are the yaw, pitch and roll moments around axis z, y, and x, the $R(\varphi,\theta,\psi) \in SO(3)$ is the rotational matrix in (3.8).

From (3.23), one can see that the force $\hat{F}$ can be written as

$$\hat{F} = \begin{bmatrix} -\sum\limits_{i=1}^{4} H_{xi} \\ -\sum\limits_{i=1}^{4} H_{yi} \\ \sum\limits_{i=1}^{4} T_i \end{bmatrix} + R_I^B \begin{bmatrix} 0 \\ 0 \\ G \end{bmatrix} \tag{3.39}$$

Consequently, the translational and rotational Lagrangians are defined respectively as

$$L_{trans} = \frac{m}{2}\dot{\xi}^T\dot{\xi} - mgz \tag{3.40a}$$

$$L_{rot} = \frac{1}{2}\dot{\eta}^T J\dot{\eta} \tag{3.40b}$$

The derivative of the translational Lagrangian is

$$\frac{d}{dt}\left(\frac{\partial L_{trans}}{\partial \dot{\xi}}\right) - \frac{\partial L_{trans}}{\partial \xi} = F_\xi \tag{3.41}$$

gives the following translational equation

$$m\ddot{\xi} + mgE_z = F_\xi \tag{3.42}$$

The derivative of the rotational Lagrangian is given by

$$\frac{d}{dt}\left(\frac{\partial L_{rot}}{\partial \dot{\eta}}\right) - \frac{\partial L_{rot}}{\partial \eta} = \tau \tag{3.43}$$

$$\frac{\partial L_{rot}}{\partial \dot{\eta}} = \frac{1}{2}\left(J + J^T\right)\dot{\eta} = J\dot{\eta} \tag{3.44}$$

$$J\dot{\eta} + \left(\dot{J} - \frac{1}{2}\frac{\partial}{\partial \eta}\left(\dot{\eta}^T J\right)\right)\dot{\eta} = \tau \tag{3.45}$$

$$C(\eta,\dot{\eta}) = \dot{J} - \frac{1}{2}\frac{\partial}{\partial \eta}\left(\dot{\eta}^T J\right) \tag{3.46}$$

where

$$\tau = \mathcal{M}^B = \begin{bmatrix} l\sum_{i=1}^{4}\sin\alpha_i T_i + J_r q\sum_{i=1}^{4} S_{p_i}\Omega_i + \sum_{i=1}^{4} S_{p_i}\mathcal{B}_{x_i} \\ -l\sum_{i=1}^{4}\cos\alpha_i T_i - J_r p\sum_{i=1}^{4} S_{p_i}\Omega_i + \sum_{i=1}^{4} S_{p_i}\mathcal{B}_{y_i} \\ \sum_{i=1}^{4} S_{p_i} Q_i + l\sum_{i=1}^{4}\sin\alpha_i\mathcal{H}_{x_i} - l\sum_{i=1}^{4}\cos\alpha_i\mathcal{H}_{yi} \end{bmatrix} \tag{3.47}$$

$$J\ddot{\eta} + C(\eta,\dot{\eta}) = \tau \tag{3.48}$$

with $C(\eta,\dot{\eta})$ defined as follows

$$C(\eta,\dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{3.49}$$

where

$$
\begin{aligned}
c_{11} &= 0\\
c_{12} &= (I_y - I_z)\left(\dot{\theta}\cos\varphi\sin\varphi + \dot{\psi}\sin^2\varphi\cos\theta\right) + (I_z - I_y)\,\dot{\psi}\cos^2\varphi\cos\theta - I_x\dot{\psi}\cos\theta\\
c_{13} &= (I_z - I_y)\,\dot{\psi}\cos\varphi\sin\varphi\cos^2\theta\\
c_{21} &= (I_z - I_y)\left(\dot{\theta}\cos\varphi\sin\varphi + \dot{\psi}\sin^2\varphi\cos\theta\right) + (I_y - I_z)\,\dot{\psi}\cos^2\varphi\cos\theta + I_x\dot{\psi}\cos\theta\\
c_{22} &= (I_z - I_y)\,\dot{\varphi}\cos\varphi\sin\varphi\\
c_{23} &= -I_x\dot{\psi}\sin\theta\cos\theta + I_y\dot{\psi}\sin^2\varphi\cos\theta\sin\theta + I_z\dot{\psi}\cos^2\varphi\sin\theta\cos\varphi\\
c_{31} &= (I_y - I_z)\,\dot{\psi}\cos\varphi\sin\varphi\cos^2\theta - I_x\dot{\theta}\cos\theta\\
c_{32} &= (I_z - I_y)\left(\dot{\theta}\cos\varphi\sin\varphi\sin\theta + \dot{\varphi}\sin^2\varphi\cos\theta\right) + (I_y - I_z)\,\dot{\varphi}\cos^2\varphi\cos\theta\\
&\quad + I_x\dot{\psi}\cos\theta\sin\theta - I_y\dot{\psi}\sin^2\varphi\cos\theta\sin\theta - I_z\dot{\psi}\cos^2\varphi\cos\theta\sin\theta\\
c_{33} &= (I_y - I_z)\,\dot{\varphi}\cos\varphi\sin\varphi\cos^2\theta - I_y\dot{\theta}\sin^2\varphi\cos\theta\sin\theta - I_z\dot{\theta}\cos^2\varphi\cos\theta\sin\theta\\
&\quad + I_x\dot{\theta}\cos\theta\sin\theta
\end{aligned}
\tag{3.50}
$$

3.5.1 Complete quadrotor simulation model

From equations (3.10), (3.42), (3.14), (3.48) the complete dynamic model of the quadrotor can be written as follows:

$$
\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}\mathrm{c}_\theta\mathrm{c}_\psi & \mathrm{s}_\varphi\mathrm{s}_\theta\mathrm{c}_\psi - \mathrm{c}_\varphi\mathrm{s}_\psi & \mathrm{c}_\varphi\mathrm{s}_\theta\mathrm{c}_\psi + \mathrm{s}_\varphi\mathrm{s}_\psi\\ \mathrm{c}_\theta\mathrm{s}_\psi & \mathrm{s}_\varphi\mathrm{s}_\theta\mathrm{s}_\psi + \mathrm{c}_\varphi\mathrm{c}_\psi & \mathrm{c}_\varphi\mathrm{s}_\theta\mathrm{s}_\psi - \mathrm{s}_\varphi\mathrm{c}_\psi\\ -\mathrm{s}_\theta & \mathrm{s}_\varphi\mathrm{c}_\theta & \mathrm{c}_\varphi\mathrm{c}_\theta\end{bmatrix}\begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{3.51a}
$$

$$
\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \frac{1}{m}\begin{bmatrix}\mathrm{c}_\theta\mathrm{c}_\psi & \mathrm{s}_\varphi\mathrm{s}_\theta\mathrm{c}_\psi - \mathrm{c}_\varphi\mathrm{s}_\psi & \mathrm{c}_\varphi\mathrm{s}_\theta\mathrm{c}_\psi + \mathrm{s}_\varphi\mathrm{s}_\psi\\ \mathrm{c}_\theta\mathrm{s}_\psi & \mathrm{s}_\varphi\mathrm{s}_\theta\mathrm{s}_\psi + \mathrm{c}_\varphi\mathrm{c}_\psi & \mathrm{c}_\varphi\mathrm{s}_\theta\mathrm{s}_\psi - \mathrm{s}_\varphi\mathrm{c}_\psi\\ -\mathrm{s}_\theta & \mathrm{s}_\varphi\mathrm{c}_\theta & \mathrm{c}_\varphi\mathrm{c}_\theta\end{bmatrix}\begin{bmatrix}-\sum\limits_{i=1}^{4} H_{xi}\\ -\sum\limits_{i=1}^{4} H_{yi}\\ \sum\limits_{i=1}^{4} T_i\end{bmatrix} + \begin{bmatrix}0\\ 0\\ G\end{bmatrix} \tag{3.51b}
$$

$$
\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \mathrm{s}_\varphi\tan\theta & \mathrm{c}_\varphi\tan\theta\\ 0 & \mathrm{c}_\varphi & -\mathrm{s}_\varphi\\ 0 & \mathrm{s}_\varphi\sec\theta & \mathrm{c}_\varphi\sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{3.51c}
$$

$$
\begin{bmatrix}I_x\dot{p}\\ I_y\dot{q}\\ I_z\dot{r}\end{bmatrix} = \begin{bmatrix}(I_y - I_z)\,qr\\ (I_z - I_x)\,pr\\ (I_x - I_y)\,pq\end{bmatrix} + \begin{bmatrix} l\sum\limits_{i=1}^{4}\sin\alpha_i T_i + J_r q\sum\limits_{i=1}^{4} S_{p_i}\Omega_i + \sum\limits_{i=1}^{4} S_{p_i}\mathcal{B}_{x_i}\\ -l\sum\limits_{i=1}^{4}\cos\alpha_i T_i - J_r p\sum\limits_{i=1}^{4} S_{p_i}\Omega_i + \sum\limits_{i=1}^{4} S_{p_i}\mathcal{B}_{y_i}\\ \sum\limits_{i=1}^{4} S_{p_i}Q_i + l\sum\limits_{i=1}^{4}\sin\alpha_i H_{x_i} - l\sum\limits_{i=1}^{4}\cos\alpha_i H_{yi}\end{bmatrix} \tag{3.51d}
$$

where $\sec *$ denotes $\frac{1}{\cos *}$.

3.5.2 Simplified quadrotor simulation model

The model of the quadcopter in this thesis assumes the following characteristics:

Assumption 3.1. *Assume that:*

- *The structure of the quadcopter and propellers are rigid and perfectly symmetrical.*
- *The Center of Gravity (CoG) of the quadcopter coincides with the origin of the body fixed frame.*
- *The thrust and drag are supposed to be proportional to the square of propeller's speed.*
- *The propellers are considered to be rigid and therefore blade flapping is negligible.*
- *The Earth is flat and non-rotating leads to the difference of gravity by altitude or the spin of the earth is negligible.*
- *Surrounding fluid velocities (wind) are negligible.*
- *Ground effect is negligible.*
- *The quadrotor is hovering over a point or flying with moderate speeds (speeds do not exceed* $5m/s$*).*

Under the above assumptions, we get

$$\begin{aligned}\dot{\varphi} &\approx p\\ \dot{\theta} &\approx q\\ \dot{\psi} &\approx r\end{aligned} \tag{3.52}$$

Therefore, the differential system equations describing the movement of quacopter without disturbances is obtained from (3.51a - 3.51d) as follows:

$$\ddot{x} = (\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi)\,\frac{U_1}{m} \tag{3.53a}$$

$$\ddot{y} = (\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi)\,\frac{U_1}{m} \tag{3.53b}$$

$$\ddot{z} = (\cos\varphi\cos\theta)\,\frac{U_1}{m} - g \tag{3.53c}$$

$$\ddot{\varphi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r}{I_x}\Omega_r\dot{\theta} + \frac{1}{I_x}U_2 \tag{3.53d}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\varphi}\dot{\psi} + \frac{J_r}{I_y}\Omega_r\dot{\varphi} + \frac{1}{I_y}U_3 \tag{3.53e}$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_4 \tag{3.53f}$$

3.6 Disturbance and parameters variations and their effect to quadcopters

Suppose that d_x, d_y, and d_z are the three disturbance forces acting on the quadrotor in the direction x, y, and z respectively. And there are also three disturbances moments d_φ, d_θ, and d_ψ acting on the quadcopter.

The differential system equations describing the movement of quacopter with disturbances can be rewritten from (3.53a - 3.53f) as follows:

$$\ddot{x} = (\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi)\frac{U_1}{m} + \frac{d_x}{m} \tag{3.54a}$$

$$\ddot{y} = (\cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi)\frac{U_1}{m} + \frac{d_y}{m} \tag{3.54b}$$

$$\ddot{z} = (\cos\varphi \cos\theta)\frac{U_1}{m} - g + \frac{d_z}{m} \tag{3.54c}$$

$$\ddot{\varphi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r}{I_x}\Omega_r\dot{\theta} + \frac{1}{I_x}U_2 + \frac{d_\varphi}{I_x} \tag{3.54d}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\varphi}\dot{\psi} + \frac{J_r}{I_y}\Omega_r\dot{\varphi} + \frac{1}{I_y}U_3 + \frac{d_\theta}{I_y} \tag{3.54e}$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_4 + \frac{d_\psi}{I_z} \tag{3.54f}$$

During operation, some dynamic parameters of quadcopter may change. For example, the mass of quadcopter changes in agricultural applications when quadcopters are used for spraying pesticides or for irrigation of crops. As the mass changes, the moments of inertia of the quadcopter also change. Therefore, the control laws for the quadcopter need to take into account the dynamics of the varying parameters for better stabilization. In the context of this thesis, the robust Linear Parameter Varying (LPV) control laws and LPV Unknown Input Observer (UIO) will be developed for controlling and stabilizing the quadcopter.

3.7 Conclusion

In this chapter, dynamic modeling of quadrotor has been described. First, six Degrees of Freedom (DOF) rigid-body model was developed using Euler-Largrange formalism. Next, the main acting forces and moments were included in the absence of wind. Finally, some simplifications have been considered in order to elaborate relatively simple control laws for a purpose of implementation by adopting the hierarchical control architecture.

4

Robust Path Planning

Chapter abstract

This chapter is devoted to the Coverage Path Planning (CPP) problem for quadrotor applications in Precision Agriculture (PA). First, the basic concepts of CPP for UAV will be covered in section 4.1. Then, section 4.2 will outline the CPP problems that will be studied in the content of this thesis. There are two CPP problems in this thesis: (i) CPP for detecting the infected areas of an agricultural area, and (ii) CPP for pesticide spraying the whole infected areas that have been detected in the first CPP problem. The next two sections, 4.3 and 4.4, will study in detail the two problems of CPP prementioned above. The final section, 4.5, will be devoted to conclusions and some comments on the obtained simulation results.

This Chapter contains:

4.1 Preliminary concepts

4.1.1 Path generation

The path-planning problem has been intensively studied for decades in robotics. In the beginning, path planners solved the problem by a deterministic search of a discrete workspace. Then, for applying the differential constraints to the path-planning problems, the randomization techniques were studied to sample the configuration space. There are a lot of research literature on path-planning methods. One of the most significant works is the textbook of LaValle [121]. In the comprehensive surveys [130] and [84], the authors compare some algorithms on a rigorous basis such as degree of soundness, completeness, optimality, and precision.

In this subsection, only basic information of well-known basic path-planning strategies such as Graph-based and Deterministic graph search methods with some illustrations are quickly reviewed.

4.1.1.1 Graph-based methods

4.1.1.2 Dijkstra Algorithm

Dijkstra's algorithm is an algorithm which is proposed by computer scientist Edsger W. Dijkstra in [55]. This algorithm finds the shortest paths between nodes in a graph, which may represent, for example, road networks. Its main feature is that the starting point is as the center to be extended to the end point. Each edge of the graph is formed to an ordered element pair by the two vertices. Each edge has its own value which is described by a weight function. Dijkstra's algorithm is working in two vertex sets named A and B. At the beginning, the set A is set to empty. When a vertex in B is moved to A, the selected vertex ensures the sum of all the edge weight from the starting point to the point is minimized. The efficiency of this algorithm is not high because it has to traverse numbers of nodes.

4.1.1.3 A* Algorithm

A* algorithm [172] was first proposed in 1968. The A* algorithm is developed on the basis of the Dijkstra algorithm. A* algorithm starts from a specific node then it updates the weighted value of the current child nodes. The child node with the smallest weighted value is used to update the current node until all nodes are traversed. The main feature of A* algorithm is that it establishes the evaluation function $f(n) = g(n) + h(n)$, where $g(n)$ is the actual cost from the initial node to the node n, and $h(n)$ is the estimated cost of the optimal path from node n to the target node in the state space.

In most cases, the value of $h(n)$ is the Euclidean distance between the two nodes. For the case that $g(n)$ is constant, $f(n)$ is only affected by the value of the function $h(n)$. When the node is close to the target node, the value of function $h(n)$ is small leads to the value of $f(n)$ is also small. Consequently, the search for the shortest path always proceeds in the direction of the target point. The A* algorithm is more efficient than the Dijkstra algorithm as it performs informed rather than uninformed search. In addition, A* considers additional information about the minimal distance to the target (the distance function), consequently the A* algorithm expands more promising vertices than Dijkstra.

4.1.1.4 D* Algorithm

The A* algorithm is mainly applied for global search in the static environment which means that the environment does not encompass any moving obstacles. However, in practical applications, the working environment for robots always contains dynamic obstacles. To overcome this inconvenient, the D* algorithm [210] was introduced in 1994. The D* algorithm is mainly used for robots to explore the path. This algorithm considers the working space as a series of states which represents the direction of the robot's position. Another researches on D* algorithm can be found in [74][108][115].

4.1.2 Deterministic graph search

4.1.2.1 PRM (Probabilistic RoadMap)

PRM planner is a very well-known sampling-based path planning strategies. In [112], an undirected graph G_p (called roadmap) is constructed by randomly sampling the space. For each vertex in G_p, connections to some k nearest neighboring vertices are considered. If the link between two vertices is in D_{free} (the space is not occupied or obstacle-free), the corresponding edge is added to the roadmap. As more vertices and more edges are added, the roadmap data structure approximates the work-space more accurately. Both initial and goal configurations are connected to the roadmap. In [111], authors discussed the theoretical properties of the PRM algorithm. This algorithm tries to connect two nodes p_{ini} and p_{goal} using a graph-search method like Dijkstra's algorithm. If there exist difficulties during the query phase, the PRM can turn back to the construction phase to adapt the size of the roadmap. A graphical representation of a roadmap is shown in Figure 4.1.

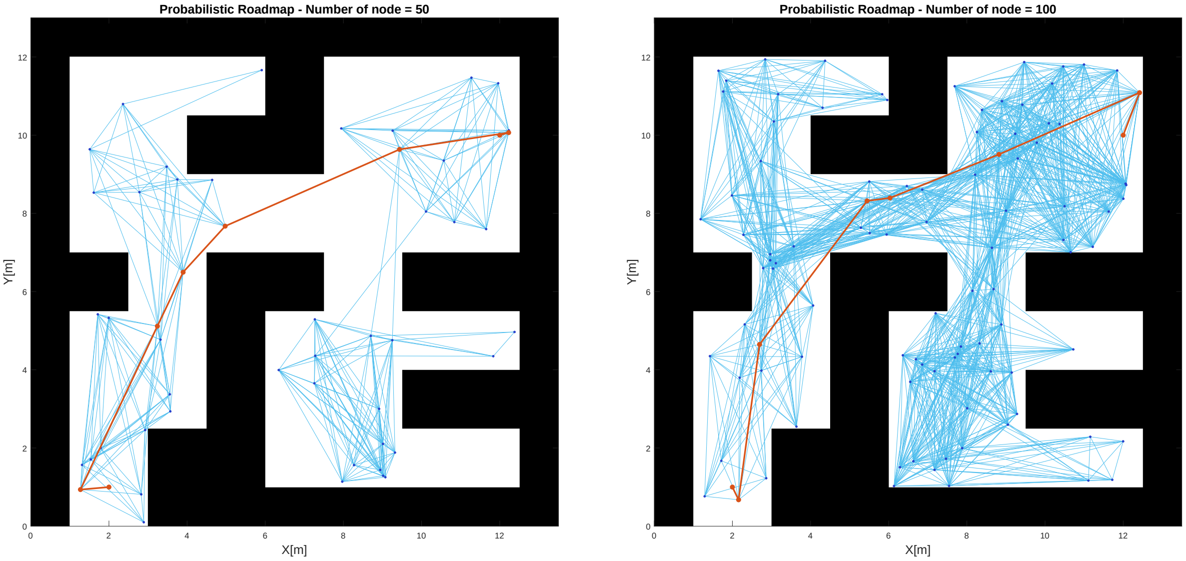


FIGURE 4.1: Path planning by PRM

If the environment is static, the same roadmap can be reused for further or for multiple queries at the same time. This feature gives PRM the ability to be suitable for trajectory planning of multi-agent systems. However, the convergence rate is slow, and the produced paths are not optimal because the produced paths are generated based on the random points, consequently, the produced paths are only optimal with respect to the points that have been generated. Furthermore, PRM has particular difficulty with solutions that pass through narrow passages where the performance of PRM planners can degrade as it may be less suitable for problems with significant dynamic constraints.

To overcome the shortcomings of PRM, several extension have been studied such as Lazy PRM in [27] or Medial Axis Probabilistic Roadmap planner (MAPRM) [97][157][237]. A version of PRM for dynamic environments is proposed in [33]. Some other PRM-based planners are in [208][162].

4.1.2.2 Rapidly-exploring Random Tree (RRT)

Beside PRM, another really good sampling-based planner is RRT. RRT planner can quickly find a feasible path from p_{ini} to p_{goal}. The first publication on RRT is described in [123], and followed by a more detailed report [122]. In these two papers, the vehicle is considered holonomic, neither dynamics nor kinematic constraints are considered. In [63], the author extended the RRT to dynamic systems. Then in [65], the same author gives more extensive description to the RRT planner and a tree T is created in D_{free}. In this paper, the author also proposed that collision-free configurations that can be reached are represented by the nodes and feasible collision-free paths between nodes are represented by edges. The root node of the tree is the initial configuration p_{ini}. The tree is growing and exploring D_{free} until a path is found, means that there exists a path between p_{ini} and p_{goal} as in Figure 4.2.

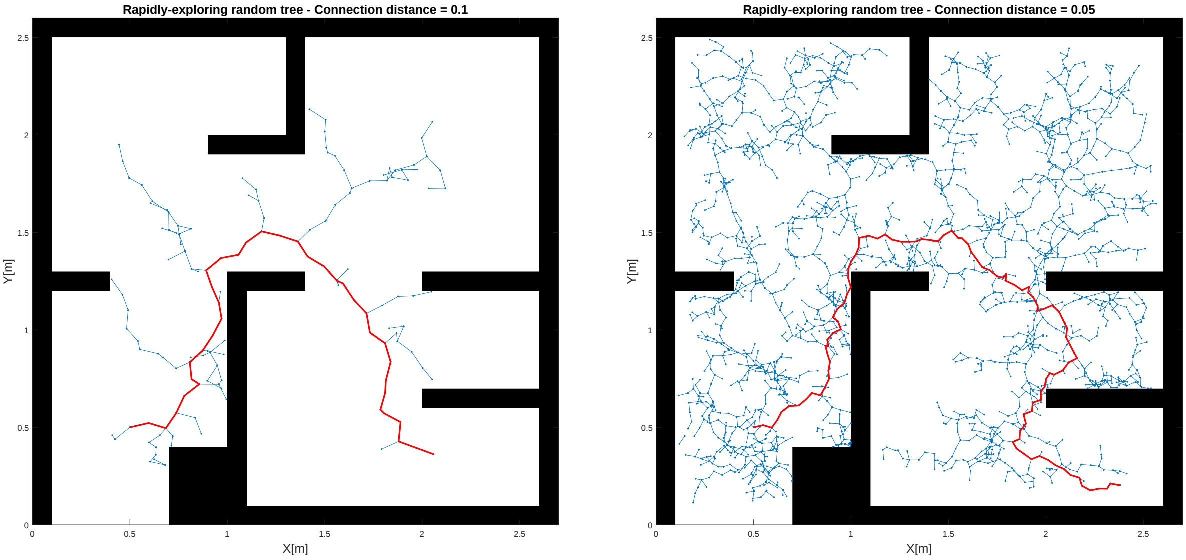


FIGURE 4.2: Path planning by RRT

4.1.2.3 RRT-connect

RRT-connect is a variant of RRT. It is designed specifically for path planning problems that involve no differential constraints. The RRT-Connect planner is based on two ideas: (i) the Connect heuristic that attempts to move over a longer distance, and (ii) the growth of RRTs from both p_{ini} and p_{goal}. With these extensions, the search tree converges rapidly to a solution. Nevertheless, this variant of RRT also has a limitation with narrow passages and high constrained workspaces [103]. Another approach [124] proposed a Selective Retraction-based RRT Planner (SR-RRT) that efficiently handles a wide variety of environments that have different characteristics.

4.2 Coverage path planing problems formulation

Suppose we have a rectangular agricultural area as shown in Figure 4.3. Note that this is not the restriction because if the agricultural area is not rectangular (polygon $ABCDEFG$ in Figure 4.4), we can add some no-fly-zone to the original agricultural area to get the new rectangular agricultural area (polygon $MNPQ$). The simplest way to get a rectangular agricultural area is to get the smallest rectangle containing the entire original agricultural area as in Figure 4.4.

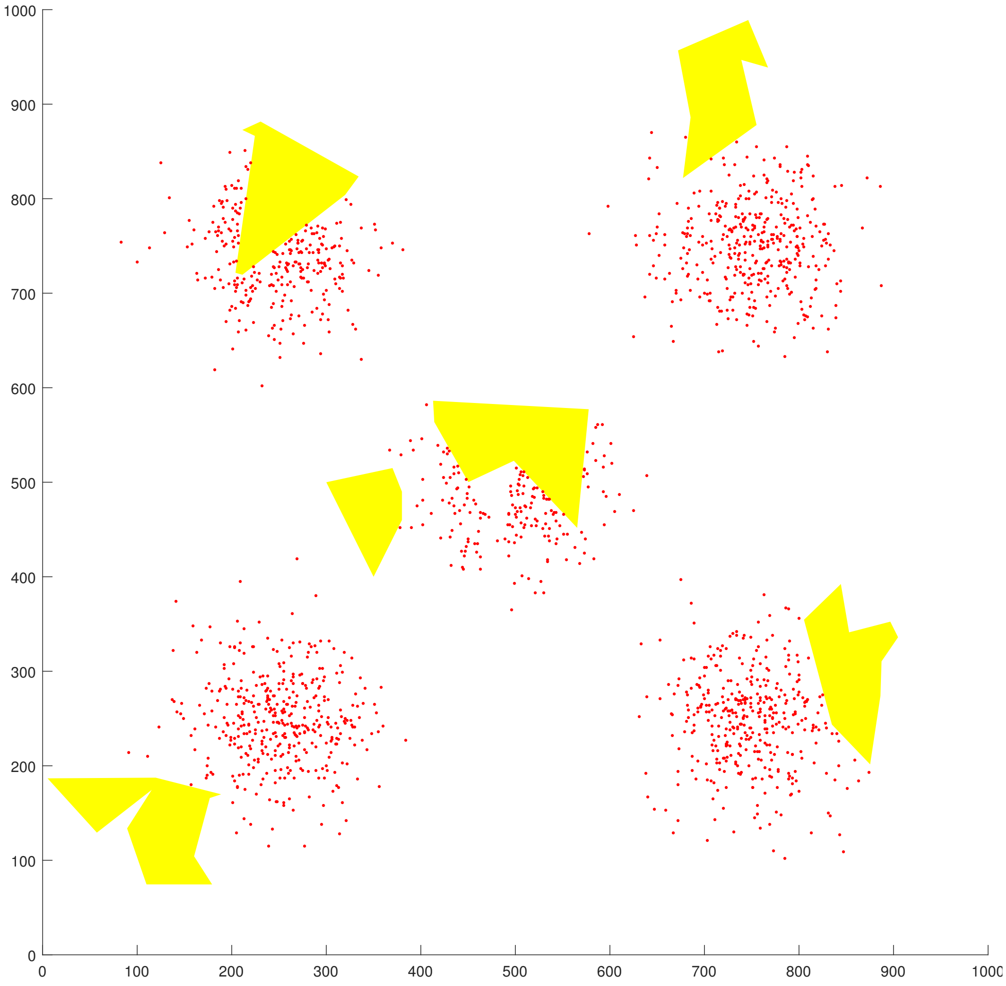


FIGURE 4.3: Coverage path planing problem formulation

The agricultural area (Figure 4.3) contains

- Obstacles, which are the yellow polygons. These obstacles can be convex or concave polygons, and they are the non-fly-zone,
- The white areas are areas where the plants are free of pests,

- The red dots are areas where crops are infected,
- The coordinates of the obstacles are assumed known prior, while the coordinates of the infected crops (red dots) are not known in prior.

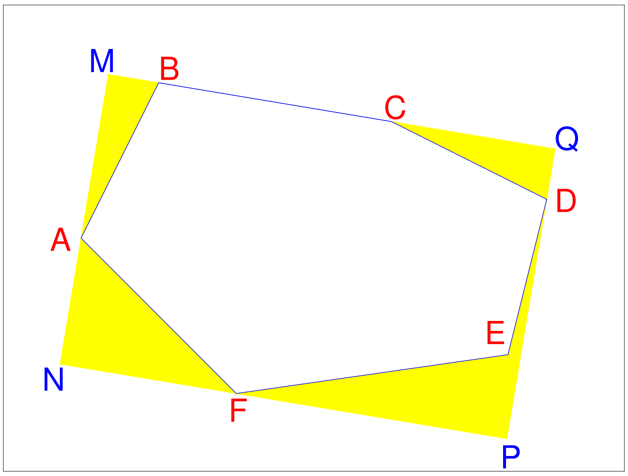
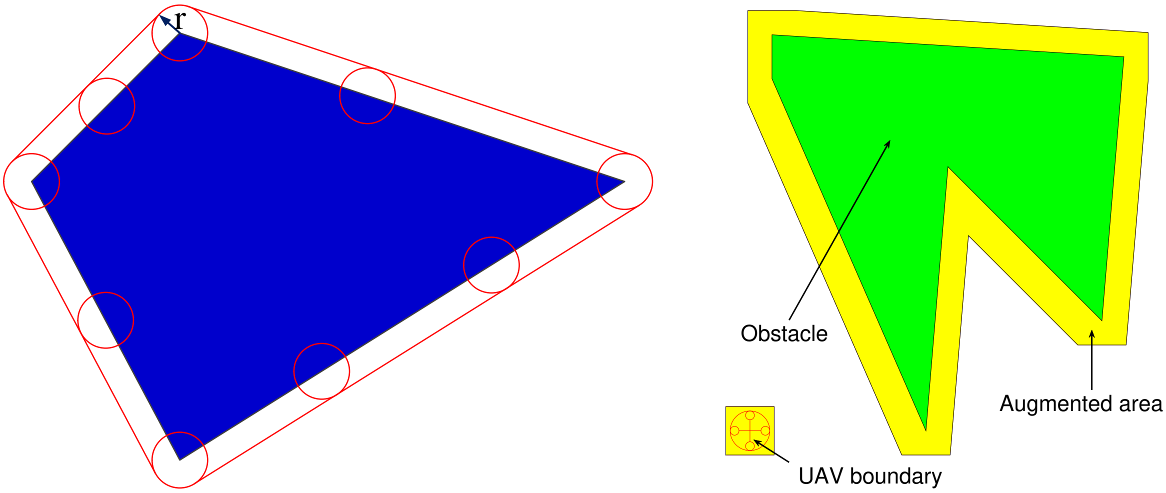


FIGURE 4.4: Coverage path planing problem formulation

Augmentation of obstacle sizes

In the process of performing its tasks, the UAV has to avoid the obstacles. Therefore, the distance from the center of the UAV to any point in the obstacles should be greater or equal to the radius r of the smallest circle which covers the UAV (Figure 4.5a). For simplicity, consider the UAV as a point, it is necessary to augment the size of the obstacles by a radius depending on the size of the UAV. At each border point of the obstacle, the operator draws a circle with radius r and creates the new obstacle inside the red curve (in Figure 4.5a). However, the augmented obstacle now is no longer a polygon.



(a) Augmented obstacle by using circle

(b) Augmented obstacle by using Minkowski sum

FIGURE 4.5: Augmentation of the size of obstacles

In this study, we use another solution to augment the size of obstacle such that the augmented obstacle is a polygon. Suppose that the shape of UAV is a square that its edge is greater than diameter of the UAV (Figure 4.5b). Let us denote the set of vertices of this square by A, and the set of vertices of the obstacles by B. The augmented obstacle is generated is the Minkowski sum [21] of A and B.

$$augmented\_obstacle = A \oplus B = \{x + y | x \in A, y \in B\}$$

It is easy to prove that in the worst case scenario, when the center of the UAV is on the edge of augmented obstacle, the distance from the center of the UAV to the real obstacle (green polygon in Fig 4.5b) is always greater than the radius of the UAV and the UAV will not collide with the obstacle.

Tasks to be solved

Our main tasks are:

1. **Infected areas detection**: We have to detect the coordinates of all the infected areas of crops by using UAV which is equipped with a special camera. It means that the coordinates of all the red dots in Figure 4.3 should be determined. Several requirements are needed to be satisfied while performing the infected areas detection task
    - UAV has to avoid the obstacles.
    - Maximize the infected areas that can be detected.
    - Minimize the path length for the UAV.
2. **Pesticide spraying**: With the information about the coordinates of all the infected areas (coordinates of all the red dots in Figure 4.3), we have to spray the pesticide to all the infected areas of crops by using UAV which is equipped with pesticide tank. Several requirements are needed to be satisfied while performing the infected areas detection task
    - UAV has to avoid the obstacles.
    - Pesticide should be sprayed to all the infected areas that have been detected from the Infected areas detection task
    - Minimize the path length for the UAV

4.3 Infected areas detection

4.3.1 Problem formulation

As we have discussed in Section 4.2, the main task of this section is to determine the coordinates of all the infected areas. To do this, the entire agricultural area must be taken by specialized cameras. The gathered images will be analyzed using specialized image processing algorithms. From there, the positions of an infected crop can be identified. Thus, the problem of detecting all the crops affected by pests will be converted into the problem CPP of all the agricultural areas.

Herein, we present a new approach for maximizing the coverage path planning while minimizing the path length of an aerial robot in agriculture environment with concave obstacles. For resolving this problem, we propose a new Cellular Decomposition (CD) which is based on a generalization of the Boustrophedon variant, using Morse functions, with an extension of the representation of the critical points. This extension leads to a decrease of the number of cells after decomposition. The results show that this new CD works well even with several concave obstacles inside the environment. Furthermore, for path planning, the cells are divided again into two classes, leading to have a cell set better suited for use of the TSP to get complete coverage. Genetic Algorithm (GA) and Travelling Salesman Problem (TSP) algorithms are applied to obtain the shortest path. Then, an approach is also proposed to maximize the scanned area on the working area with obstacles.

UAV platform

This section considers a quad-rotor equipped with a special spectral camera mounted on a gimbal stabilizer, which compensates for small rotation displacements that may be experienced by the UAV during navigation. The objective of the mission is to capture images of the whole area of plants by a dedicated spectral camera. Based on spectrum study of images that have been taken, one can have the information about the health and pest of the plant.

We assume that images are taken by special spectral camera which has the following parameters:

- Image frame is rectangle with the length and width are respectively L and H as depicted in Figure 4.6).
- Images have to be taken such that they have the same direction and the overlap regions on the horizontal and vertical sides as dL and dH (in Figure 4.7)

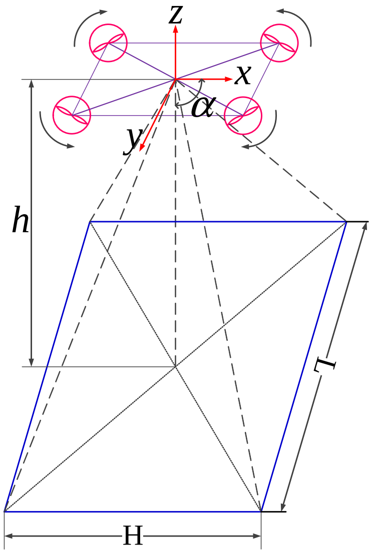


Figure 4.6: UAV with frame picture capture

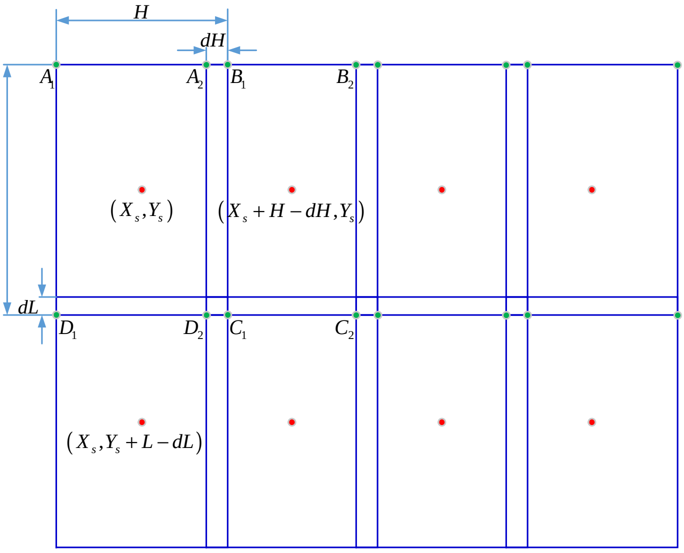


Figure 4.7: Picture frames

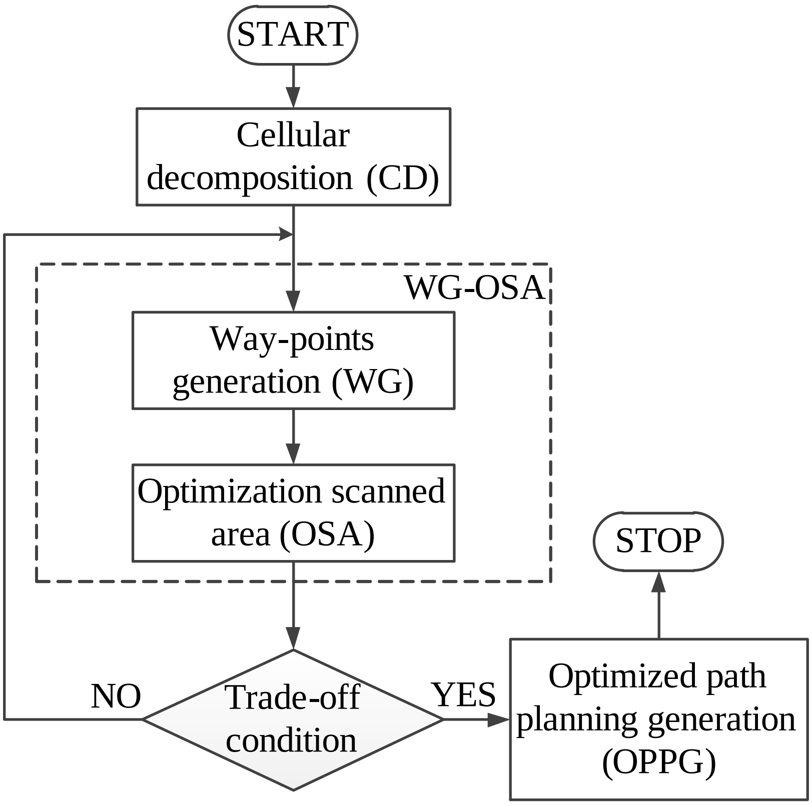


FIGURE 4.8: Proposed algorithm

4.3.2 Proposed algorithm for infected areas detection

The contribution of this section focuses on maximizing the coverage path planning and minimizing the path length of an aerial robot in agriculture environment with concave obstacles. The algorithm is depicted in Figure 4.8. It is a combination of distance minimization and profit maximization. For accomplishing this problem, one has to resolve three main tasks.

The first task is dedicated for CD (Figure 4.8). A new CD is based on a generalization of the Boustrophedon variant [45], using Morse functions [225], with an extension of the representation of the critical points. The feature of this extension is that the number of cells after decomposition is decreased compared to the decomposition methods in [45], [225]. With this new proposed approach, the environment can be divided into a number of separating obstacles and obstacle-free regions. In each obstacle-free region which can be called obstacle-free cell, the aerial robot can make back and forth motion to cover the whole working area.

Based on the cells that have been decomposed by CD, the second task is aiming to generate way-points in each cell (WG in Figure 4.8) and maximize the percentage of coverage (OSA in Figure 4.8). The trade-off condition between percentage of coverage and number of way-points is the condition for ending the algorithm.

Finally, in the third task, once the cells have been created by CD, the way-points have been created by WG and OSA, the cells are divided again into two classes, leading to have a cell set better suited for using the TSP algorithm to get complete coverage. GA and TSP algorithms with some modifications are applied to obtain the shortest path for the UAV to make complete coverage.

4.3.2.1 Cellular Decomposition (CD)

Several CD methods have been studied as trapezoidal decomposition technique [120], boustrophedon decomposition [45], and Morse-based CD [225].

In the trapezoidal decomposition technique [120], the free spaces are divided into trapezoidal cells. There are two parallel sides on each cell, and each cell can be covered by simple back and forth motion parallel to either side. Therefore, the whole working area can be totally scanned by visiting each cell in the adjacency graph. A vertical line is swept from left to right through the working area of the robot. Each time the line encounters a vertex of a polygon, one can say that an event occurs. Three types of events are defined: IN, OUT, and MIDDLE. In an IN event, the current cell is closed and two new cells are opened. When an OUT event occurs, the current two cells are closed and a new cell is created. Finally, MIDDLE event is the event that a current cell is closed and a new single cell is initiated. Consequently, by using these three events, the environment can be totally divided into separated trapezoidal cells.

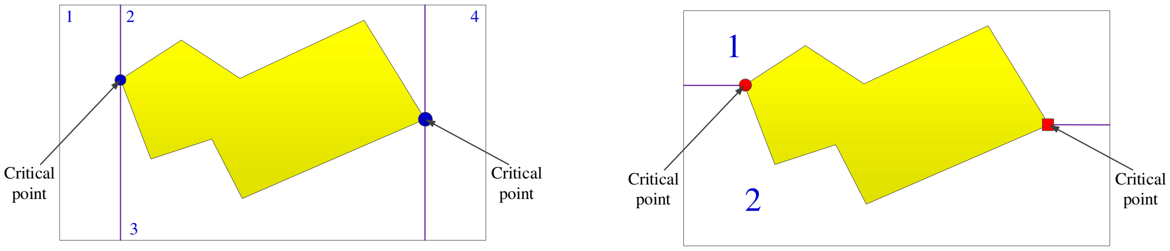
A drawback of the trapezoidal decomposition is that it generates many cells that, intuitively, can be merged together to form bigger cells. This is clearly an inconvenient, as the more cells are present, the longer the final coverage path is. This happens because the trapezoidal decomposition creates only convex cells. However, non-convex cells can also be completely covered by simple motions. To overcome this limitation, the boustrophedon CD was proposed. The boustrophedon decomposition is similar to the trapezoidal decomposition introduced above, but it only considers vertices where a vertical segment can be extended both above and below the vertex. The vertices where this occurs are called critical points.

The boustrophedon decomposition [45] addresses the issue of redundant movements by merging together cells between successive IN and OUT movements. The major difference is in the case of a MIDDLE event, where no new cell is formed, but the current cell is updated in width. So cells are only created on change of connectivity, as a result the number of cells are less than in the case of trapezoidal decomposition. The point which changes the connectivity of the cell is termed as a critical point. In the implementation of this algorithm, the MIDDLE event is replaced by two new events called – FLOOR and CEILING. The FLOOR event corresponds to vertices on top of obstacles, while CEILING event corresponds to vertices that are at the bottom of obstacles. Both an IN event as well as an OUT event are associated with FLOOR and CEILING pointers. So for a given cell, CEILING and FLOOR pointers points to the top and bottom of a cell.

In [225], Acar *et al* generalized the boustrophedon decomposition by proposing a CD approach based on critical points of Morse functions. In this paper, the authors show that the boustrophedon decomposition is a particular case of Morse decomposition. With respect to the original boustrophedon decomposition, the Morse-based decomposition has the advantage of handling also non-polygonal obstacles. By choosing different Morse functions, different cell shapes are obtained, e.g. circular or spiked cells. Theoretically, Morse-decomposition can be applied to any n-dimensional space. Moreover, they presented a method to perform coverage of planar spaces by detecting the critical points using sensory range information, and a motion-template-based algorithm that ensures to encounter all the critical points in the target area. Therefore, this method allows complete coverage on line.

A new approach for CD will be proposed here. Based on the Morse method, critical points can be determined. However, Morse method does not categorize the type of critical points. In this approach critical points are categorized into four groups as follows (Figure 4.9b):

- Critical point IN type 1 (red circle): the connectivity of a slice is increased by one and the obstacle is on the right.
- Critical point IN type 2 (blue circle): the connectivity of a slice is increased by one and the obstacle is on the left.
- Critical point OUT type 1 (red square): the connectivity of a slice is decreased by one and the obstacle is on the left.
- Critical point OUT type 2 (blue square): connectivity of a slice is decreased by one and the obstacle is on the right.



(a) Boustrophedon decomposition

(b) New approach for Cellular Decomposition (CD)

FIGURE 4.9: Cellular Decomposition (CD) methods

The algorithm for decomposing the cluttered environment to cells is shown as follows:

- Step 1: Use a vertical line and sweep it from the left to the right of the environment, and find all the critical points.
- Step 2:
  - At each critical point IN type 1, draw a horizontal line to the left till when that line touches the obstacle or the border of the area.
  - At each critical point OUT type 1, draw a horizontal line to the right till that line touches the obstacle or the border of the area.

The results of the original boustrophedon method and the CD of this approach are shown in Figure 4.9b and 4.9a.

Remark 4.1. *In Figure 4.9b and 4.9a, the vertical line sweep from the left to the right. In principle, however, the vertical line can be swept from the right to the left, from the bottom to the top, from the top to the bottom, or in any direction.* ☐

Simulation results analysis

This approach has been run and verified in Matlab. From Figure 4.9b and 4.9a, one can see that with our approach, the area has been splitted into two cells compared with four cells of the Boustrophedon decomposition method. Area with complicated forms (convex, non-convex, polygon, and arbitrary shape) (Figure 4.10) of obstacles has been divided to

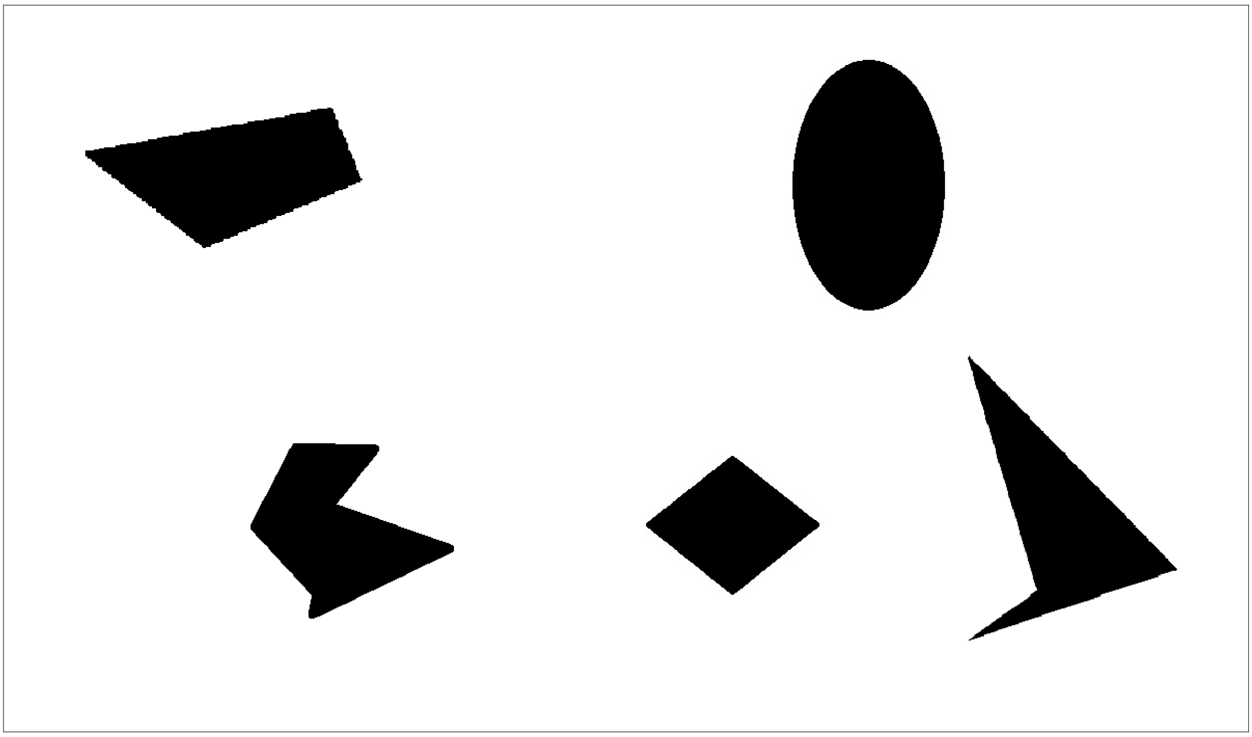


FIGURE 4.10: Agricultural area with convex and concave obstacles

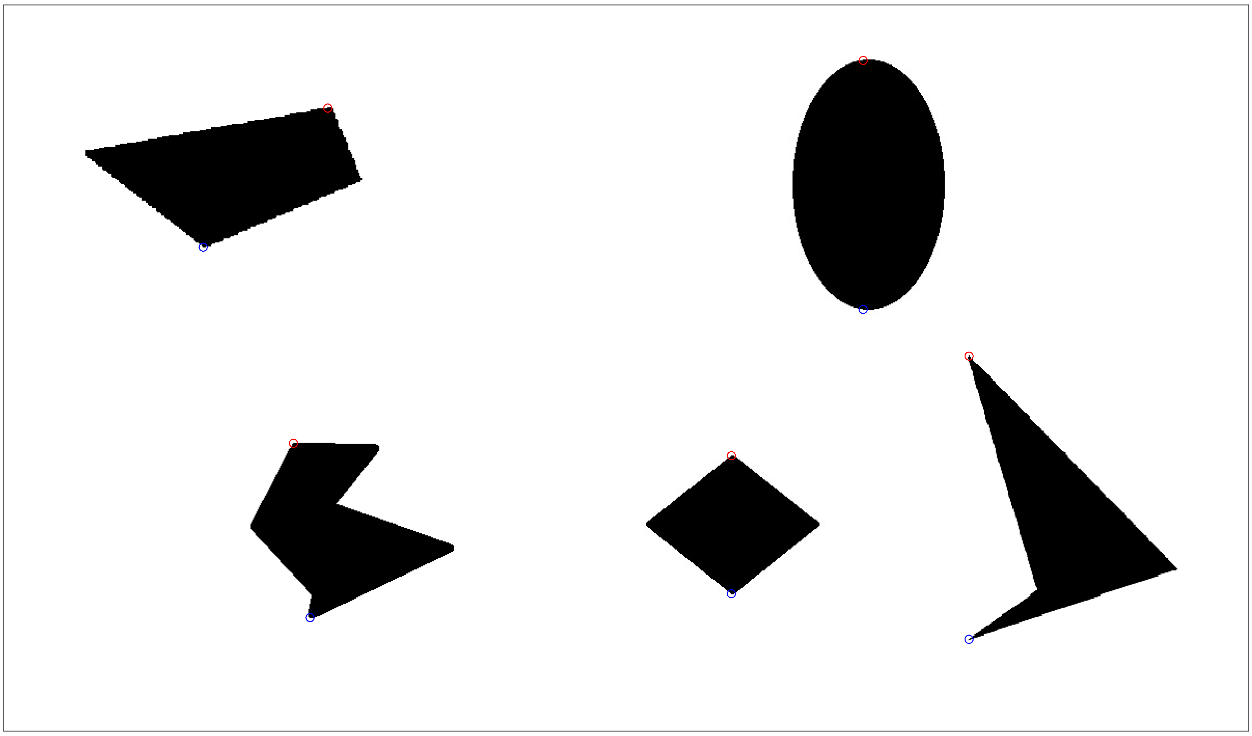


FIGURE 4.11: Critical points

only 6 cells (Figure 4.17) by using the new proposed approach, compared with more than 20 cells if using Boustrophedon method. In Figure 4.11 shows the critical points.

Remark 4.2. *In this simulation, the vertical line is swept from the top to the bottom to get fewer cell compared to the case that the vertical line is swept from the left to the right.* ☐

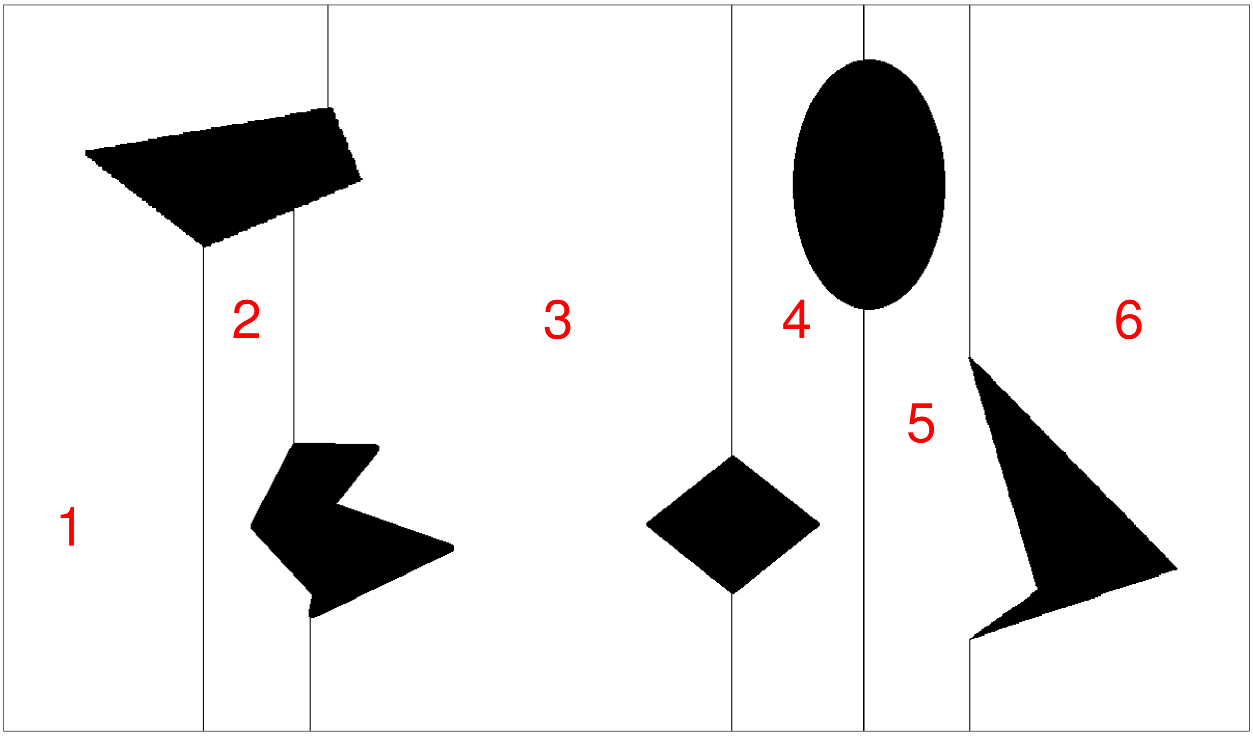


Figure 4.12: Cells after decomposition

4.3.2.2 Way-points generation (WG)

As described in section 4.3.1, picture frames which are taken by the camera mounted on the aerial robot are rectangular. Consequently, for gathering the whole area of the environment, frames need to be arranged as in Figure 4.7. Centers of these rectangles are the positions of the UAV, in which the pictures will be taken. We can see that the set of all these points is the set of way-points that the UAV has to visit.

Remark 4.3. *In rectangular picture frames, as shown in Figure 4.13, there are several rectangles that have a center (red star in Figure 4.14) belonging to the obstacles. These are the locations that UAV can not reach since they are located in the no-fly-zone regions.* ☐

Remark 4.4. *In many cases, the center of the rectangular frames on the right most and bottom may not be within the area of the agricultural area we are working with. Therefore, the algorithm needs to recalculate the positions of the rectangles so that they are within the agricultural land under consideration.* ☐

Way-points are generated as follows:

- The first rectangle on the top left corner has the center with coordinate (X_s, Y_s).
- Fill the whole working area with rectangles (Figure 4.13) from the left to the right and from the top to bottom such that two neighbor rectangles on horizontal direction have an overlapped rectangle area with dimension $dH \times L$, and two neighbor rectangles on vertical direction have an overlapped rectangle area with dimension $H \times dL$ (Figure 4.7).

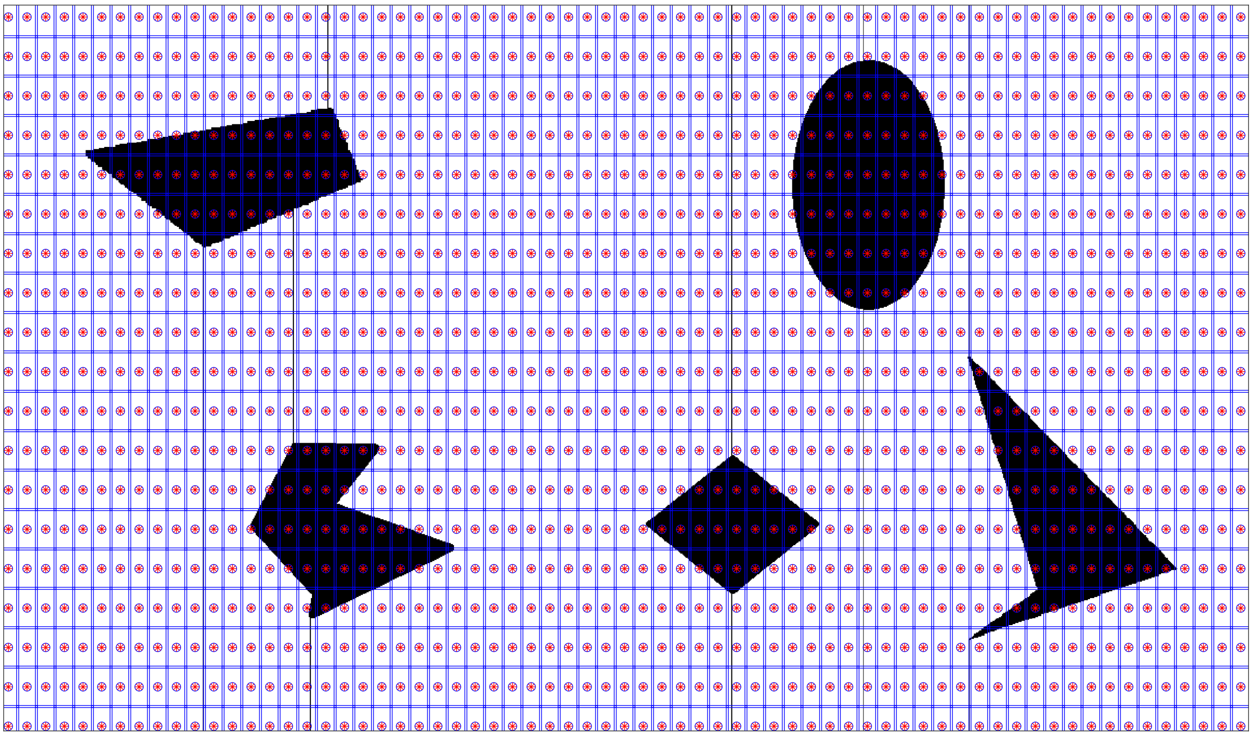


FIGURE 4.13: Picture frames

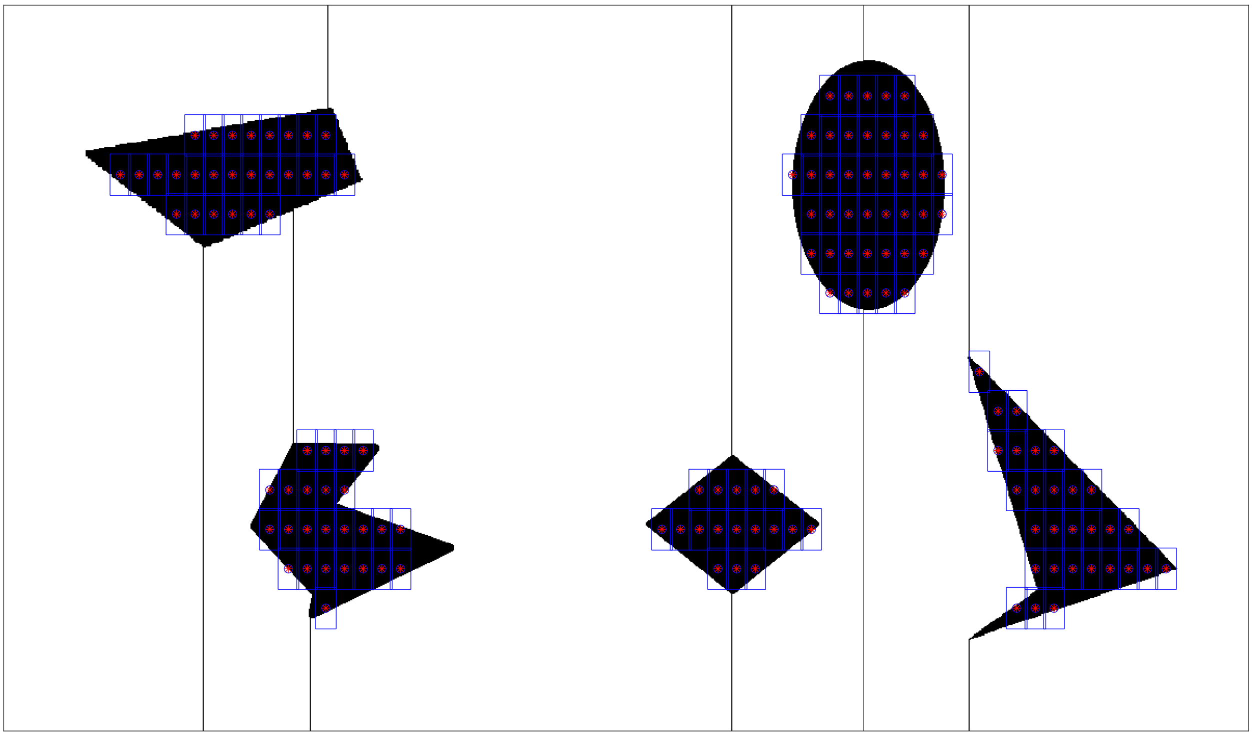


FIGURE 4.14: Rectangles with centers in obstacles

- Find coordinates of the set X that consists of all the centers of rectangles which are not located in the obstacle (Figure 4.15).

- From the coordinates of the picture frames in each cell, we have to generate the boustrophedon path as in Figure 4.16. In Figure 4.17 shows the boustrophedon trajectory in each cell and the picture frame in cell number 4.

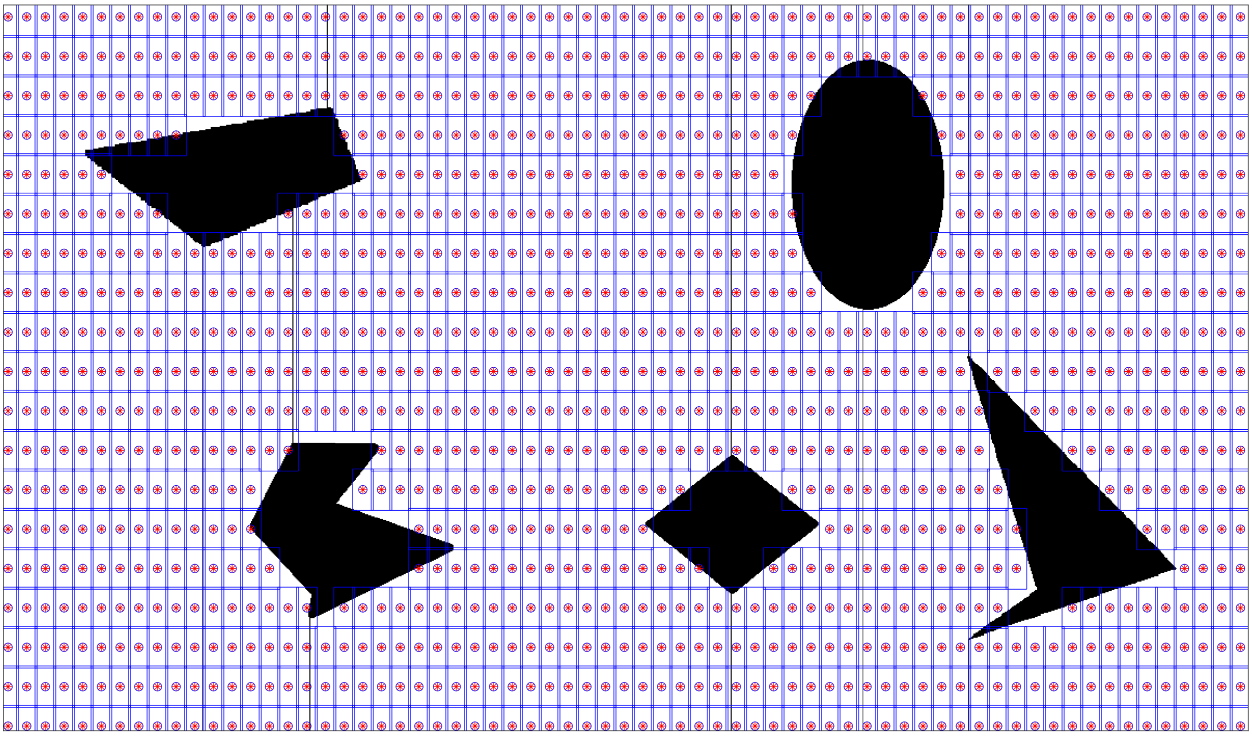


FIGURE 4.15: Picture frames that centers are not in obstacles

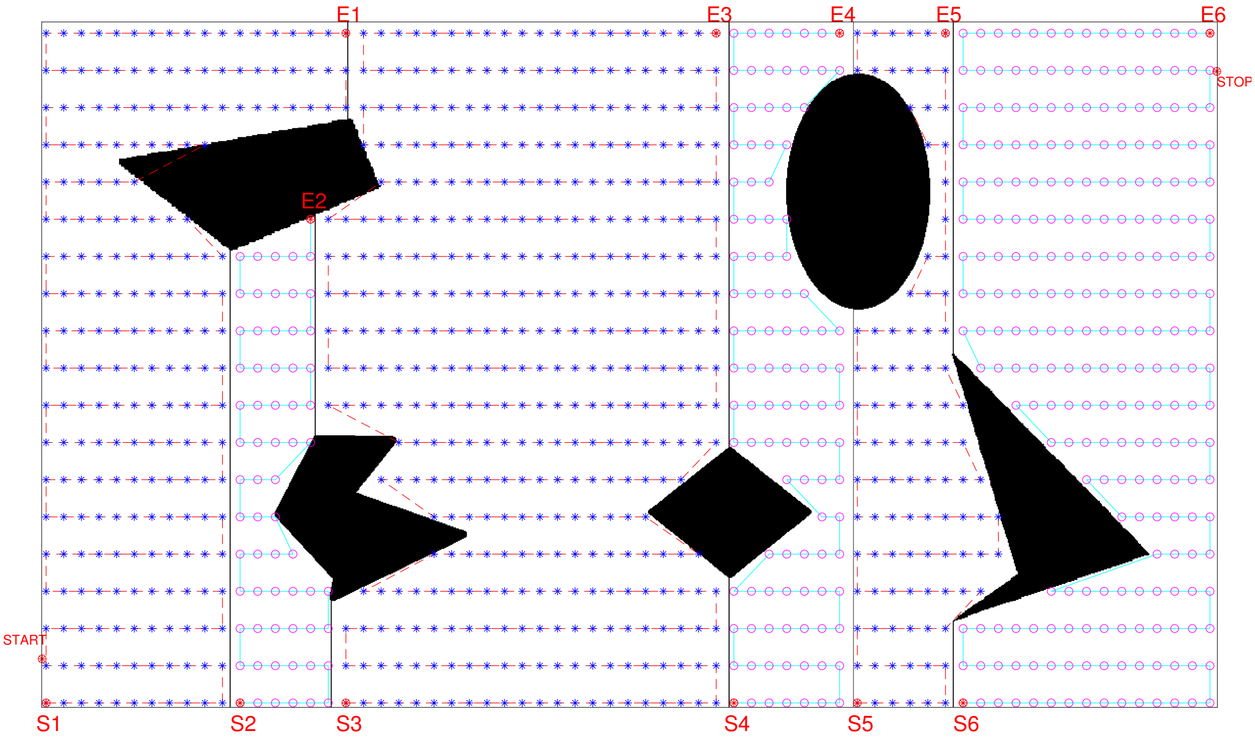


FIGURE 4.16: Boustrophedon path in each cell

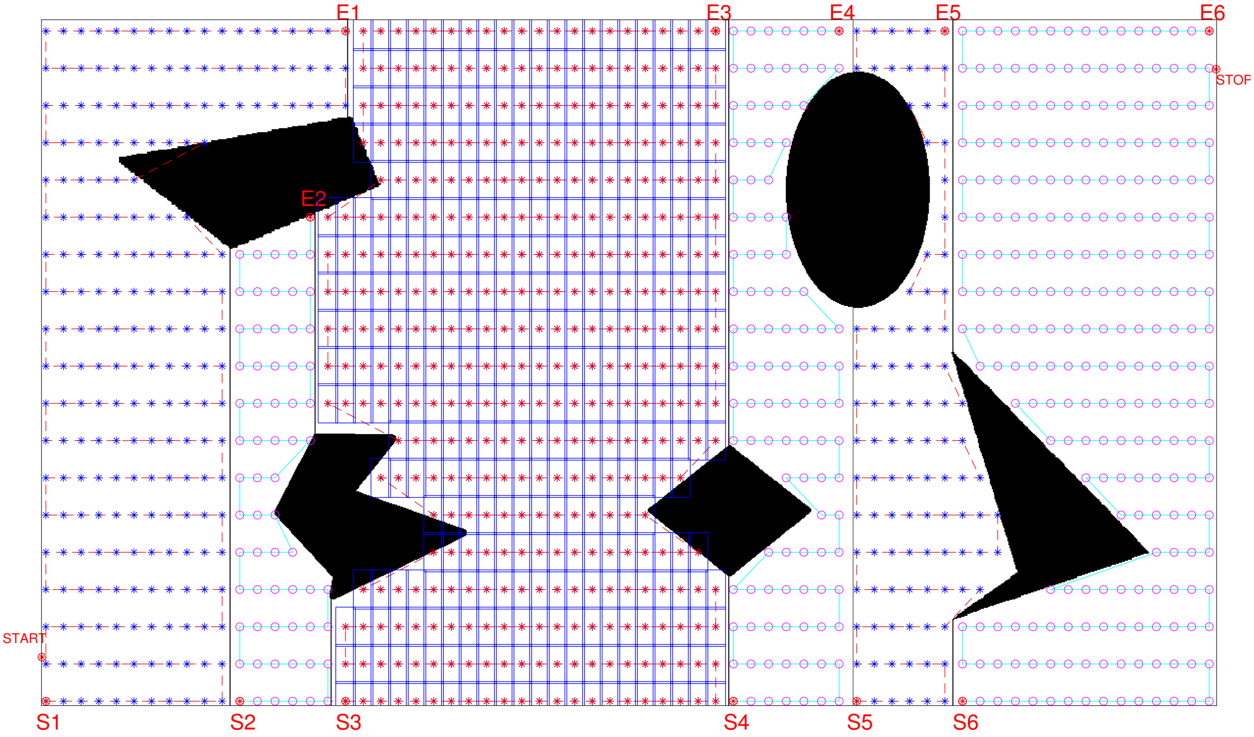


FIGURE 4.17: Cellular decomposition to cells

4.3.2.3 Optimization of the percentage of coverage (OSA)

Optimization process is processed as follows:

- Calculate the non-obstacle area of all rectangles that have been created with center in X.
- Find the percentage of coverage and number of images needed for the coverage.
- By varying the coordinate of the first image, means that by varying (X_s, Y_s) such that $(X_s, Y_s) \in \left[1 ... \frac{H}{2}; 1 ... \frac{L}{2}\right]$, one can have the maximal coverage percent or minimal number or images needed to be taken.

Simulation results analysis

Simulations in this section have the following parameters:

- Image dimension : 1200×700
- Picture frame dimension
  - $H = 40; \quad dH = 2$
  - $L = 80; \quad dL = 2$
- trade-off function: $coverage\_percentage \times (1 - \alpha) - no\_way\_point \times \alpha$ where $\alpha = 0.2$

Table 4.1: Some results for simulations in Matlab

X_s	Y_s	Number of way-points	% coverage	Trade-off function
20	40	230	95.0196	30.0085
20	38	229	94.6407	29.8860
19	39	228	94.3653	29.8922
1	1	275	96.3794	22.1035
2	10	266	96.4520	23.9616
3	13	266	96.3549	23.8839

Simulations in this section have the following parameters:

- Image dimension : 1200×700
- Picture frame dimension
  - $H = 20; \quad dH = 2$
  - $L = 40; \quad dL = 2$
- trade-off function: $coverage\_percentage \times (1 - \alpha) - no\_way\_po\text{int} \times \alpha$ where $\alpha = 0.2$

Table 4.2: Some results for simulations in Matlab

X_s	Y_s	Number of way-points	% coverage	Trade-off function
1	1	1044	97.5	-130.8
1	7	1048	97.8	-131.4
2	5	1030	97.9	-127.7
10	20	1026	97.5	-127.2
10	14	1031	97.9	-127.9
9	14	1029	97.8	-127.6

The simulation results given in table 4.1 show that, in the case when the dimension of images is large $(H = 40, dH = 2; L = 80, dL = 2)$ the number of way-points affects the percentage of coverage. However, with the same trade-off function, when the dimension of images is small $(H = 20, dH = 2; L = 40, dL = 2)$, the percentage of coverage seems not to be affected by the number of way-points.

4.3.2.4 Optimized path planning generation

Theoretically, it is possible to find the shortest traveling distance for all the way-points generated in the previous section. To do that, firstly, the distance between each two points needs to be calculated and the path between this two points has to avoid the obstacles. Several methods can perform that such as PRM or RRT[5]. For instance, if there are n way-points, the number of paths to be found is $\frac{n(n-1)}{2}$. After that we can use TSP for finding the shortest path between n way-points. However, the calculation time can be important.

For avoiding this disadvantage, this section provides another approach. Instead of applying TSP for all the way-points, first we will split the whole area to non-obstacle separated cell and each cell can be covered by simple back and forth motion (as has been proposed at the beginning of this section) by the UAV. Then, the TSP is applied only for finding the shortest path length for moving between cells.

In [45], the authors used the adjacency graph for finding the sequence of cells to be visited to cover the area. But this approach does not guarantee the order of visited cell gives the shortest path for changing between cells. In this section, we introduce a method for finding the cells' sequence with the shortest path length for changing cells.

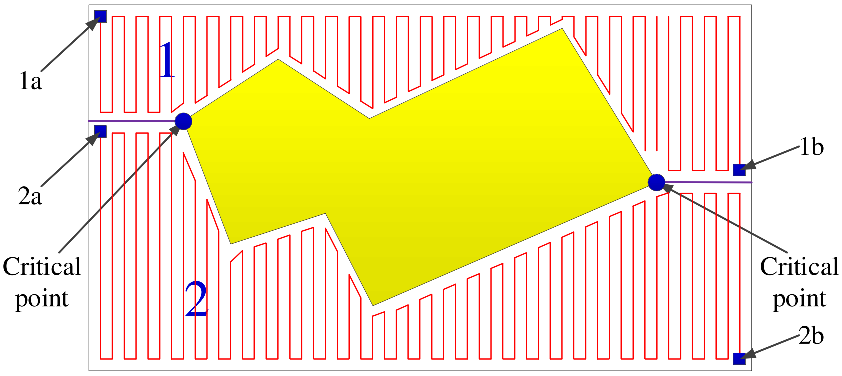


FIGURE 4.18: In/Out points in each cell

As seen in the previous section, the working area was splitted into obstacle-free cells, and the UAV can make the back and forth motion to cover cells. Consequently, for each cell, there will be two points: initial and final for the UAV coverage (Figure 4.18). For example, for $cell_k$, these two points are k_a and k_b. UAV can start covering the kth cell from point k_a to k_b or inverse from k_b to k_a.

Here each cell will form two nodes of the TSP such as two nodes $cell_{1_a}$ and $cell_{1_b}$ for $cell_1$. Consequently, if the UAV is in node $cell_{1_a}$, it has to go through node $cell_{1_b}$ and inverse. To do this, an additional constraint has to be added to the constraints of TSP for each cell:

$$x_{cellk_a\_cellk_b} + x_{cellk_b\_cellk_a} = 1 \qquad k = 1, 2, ..., 2n \tag{4.1}$$

The UAV starts at initial position (START), scan the whole area and go back to the final position (STOP) for charging the battery as shown in Figure 4.19.

To ensure that there will always exist a solution for this optimization problem, the initial population for GA has the form as shown in Figure 4.21. The flip, slide and swap operations are always operated on each pair of nodes x_a and x_b where $x \in [1, n]$. In each iteration, the random swap position of cell x_a and x_b is also processed.

The flip, slide and swap operations are described in Figures 4.20b, 4.20c, 4.20a i, j such as $i < j$.

The advantages of using proposed flip, slide, swap operations as depicted above will make the search for optimal path faster because this search is performed only on the set of paths that satisfy the given request. Furthermore, the solution is always found even though the number of cells is huge.

If proposed flip, slide, swap operations are not applied, the optimal path can only be found when the $number\_of\_cell \leq 16$, and the simulation time is always greater than 2000 seconds.

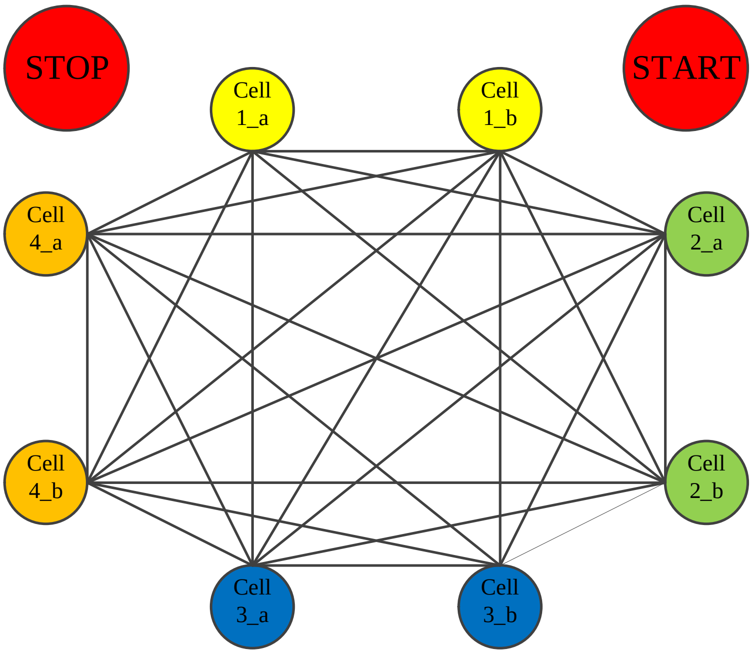


FIGURE 4.19: Travelling Salesman problems with additional constraints for this problem

Simulation results as showns in table 4.3 were performed on a Dell insprion 7559 with Core i7, 8Gb, Windows10, Matlab 2016b configurations specified in the table. The parameters for GA are: $population\_size = 1000$; $number\_of\_iteration = 1e4$; $number\_of\_cell = 100$;

TABLE 4.3: Time of simulation

Simu.	With proposed operations	Without proposed operations
1	732.8133 sec, solution found	$>$ 3000 sec, no solution
2	743.4828 sec, solution found	$>$ 3000 sec, no solution
3	742.2244 sec, solution found	$>$ 3000 sec, no solution
4	742.6994 sec, solution found	$>$ 3000 sec, no solution

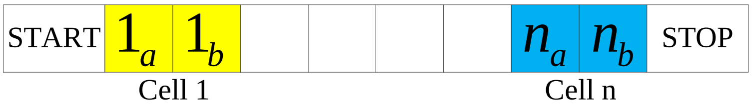
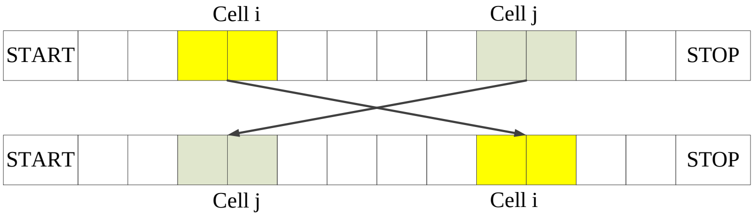
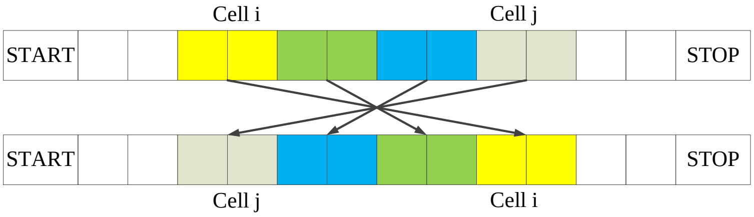


FIGURE 4.21: Genetic algorithm using in this approach

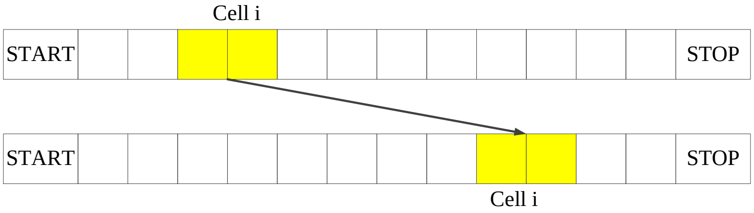
The algorithm is shown on Algorithm 1.



(a) Swap Operation



(b) Flip Operation



(c) Slide Operation

FIGURE 4.20: Swap, flip, and slide operations for GA

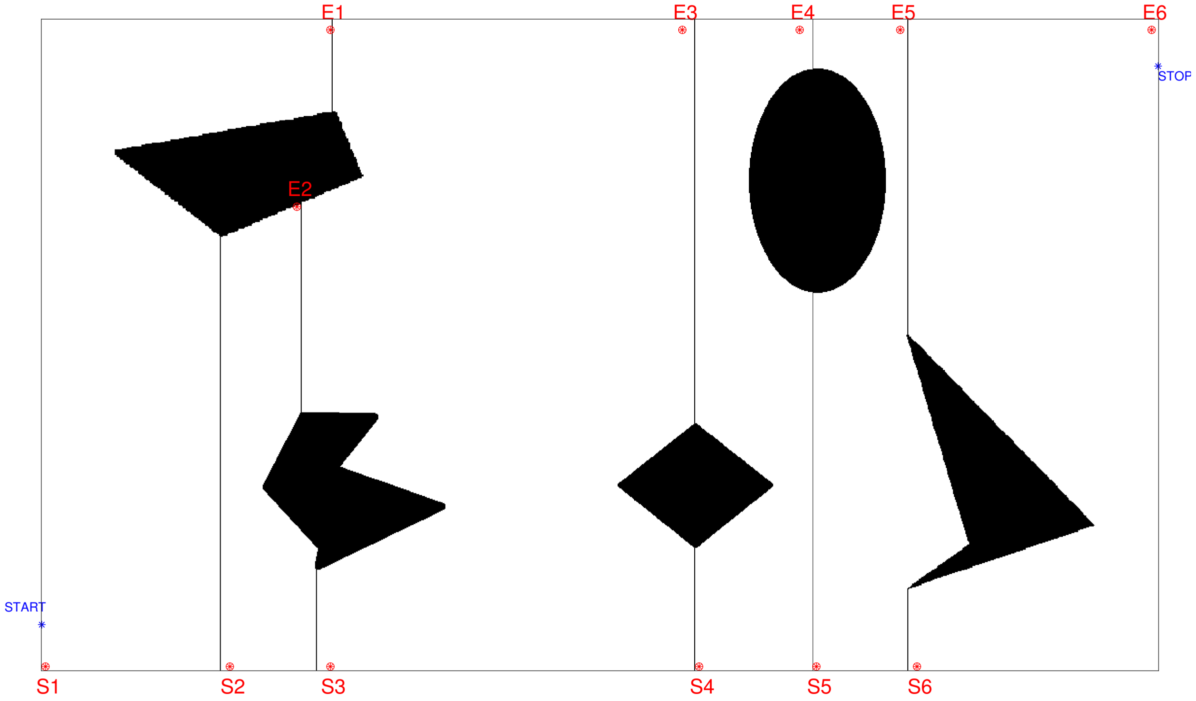


FIGURE 4.22: Cellular Decomposition (CD) in the proposed approach

```
Algorithm 1: Genetic Algorithm and Travelling salesman problem approach
input  : instance $\Pi$,
         population size: $4 \times \alpha$,
         number of iterations: $\delta$
output: Solution $X$
Generate $4 \times \alpha$ feasible solutions randomly with the form as in Figure 4.21;
Save them in the population $Pop$;
$globalMin = \infty$;
for $i \leftarrow 1$ to $\delta$ do
    // Update the best solution
    Select the best solution $minDist$ in $Pop$;
    if $minDist < globalMin$ then
        $globalMin = minDist$;
    // Mutation
    for $j \leftarrow 1$ to $\alpha$ do
        // Mutate the best to get three new routes
        Find the best solution $bs$ on 4 solutions $X_{4\times j}$; $X_{4\times j-1}$; $X_{4\times j-2}$; $X_{4\times j-3}$ of $Pop$;
        $bs_1$ is generated by flip $bs$ from random elements $I\,(node\,I_a\,and\,node\,I_b)$ to
          $J\,(node\,J_a\,and\,node\,J_b)$;
        $bs_2$ is generated by swap random element $I$ and $J$ in $bs$ ;
        $bs_3$ is generate by slide $bs$ from random element $I$ to $J$;
        Two gens in elements $I\,(I \in [1,..,n])$ of $bs_1$, $bs_2$, and $bs_3$ can be randomly
          swapped;
        Generate matrix $T_j$ from 4 columns matrix $bs$, $bs_1$, $bs_2$, and $bs_3$;
    // Updating
    Update $Pop = T_1 + T_2 + ... + T_\alpha$;
return the best solution $X$ in $Pop$;
```

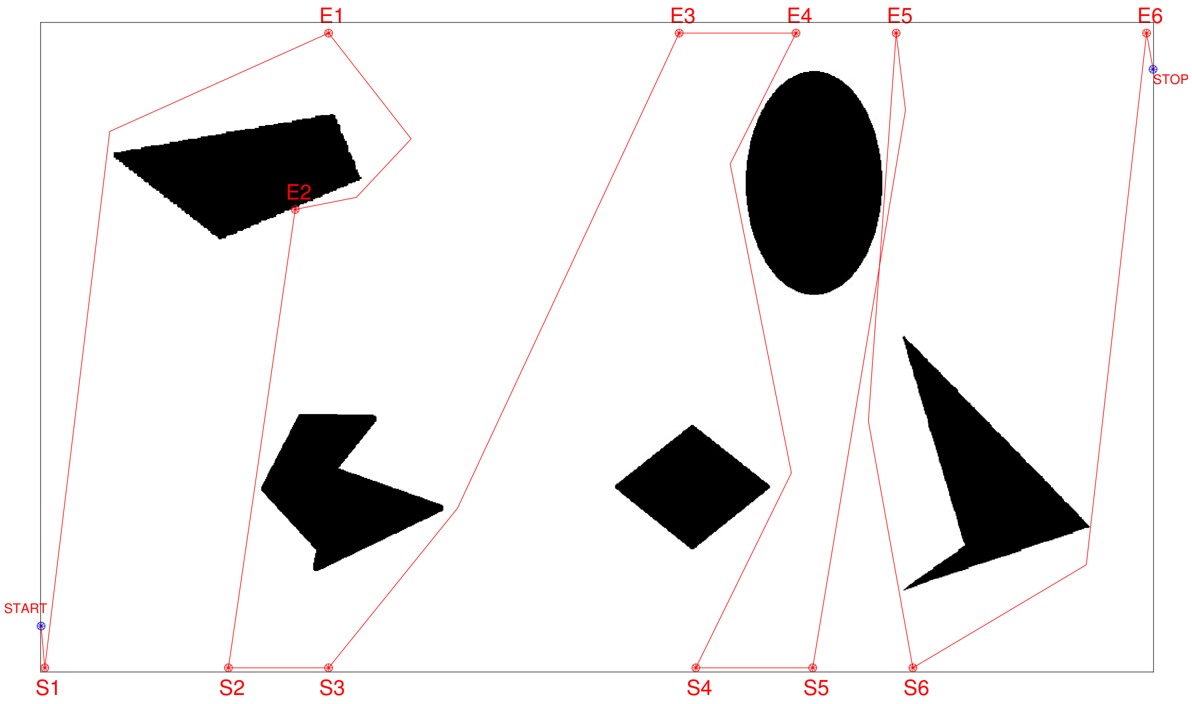


Figure 4.23: Genetic algorithm using in the proposed approach

4.4 Pesticide spraying

The contribution of this subsection focuses on generating the best path for an UAV to distribute medicine to all the infected areas of an agriculture environment which contains non-convex obstacles, pest-free areas and pests-ridden areas. The algorithm for generating this trajectory can save the working time and the amount of medicine to be distributed to the whole agriculture infected areas.

From the information on the map regarding the coordinates of the obstacles, non-infected areas, and infected areas, the infected areas are divided into several non-overlapping regions by using a clustering technique. There is a trade-off between the number of classes generated and the area of all the pests-ridden areas. After that, a polygon will be found to cover each of these infected regions. However, obstacles may occupy part of the area of these polygons that have been created previously. Each polygon that is occupied in part by obstacles can be further divided into a minimum number of obstacle-free convex polygons. Then, a Boustrophedon path of optimal length will be created for each convex polygon that has been created for the UAV to follow. Finally, this subsection deals with the process of creating a minimal path for the UAV to move between all the constructed convex polygons and generate the final trajectory for the UAV which ensures that all the infected agriculture areas will be covered by the medicine.

The algorithm of the proposed method has been tested on MATLAB and can be used in precision agriculture.

4.4.1 Problem statement and decomposition

To fix ideas, suppose that the agricultural area of interest is as depicted in Figure 4.24. In this figure, there are obstacles (the yellow areas), uninfected plants (the white areas), and infected plants (the dotted red area). One can see that the infected area are close to some obstacles and an UAV which wants to move from one area to another will meet an obstacle on its path. To tackle the stated problem, we propose the algorithm illustrated in Figure 4.25. In view of this algorithm, the problem could be subdivided into four main tasks.

- The first task is MAP DIVISION (MD). From the information of the map about the coordinates of obstacles and infected areas, we divide the infected areas into several non-overlapping regions (clusters) using a clustering technique [93]. Intuitively, the smallest shape covering each cluster is formed by the boundary of the cluster. Nevertheless, this shape does not necessarily minimize the trajectory of the UAVs. Hence we propose another approach for this cluster-covering step. In particular, we characterize the minimal convex polygon covering each cluster. Note that to ultimately determine the trajectory of the UAVs, we need to divide each covering polygon into convex regions. For this reason, the number of convex regions produced by the former approach might be larger than that of the later. In fact, which method is more efficient (i.e. producing fewer number of convex regions, smaller coverage region and shorter trajectory of the UAVs) is not known a prior; this is problem-specific. We provide an algorithm for the comparison in the simulation. However, the covering polygon of each cluster might be occupied by obstacles, which we eliminate using available coordinates data. Then, we divide each obstacles-free covering polygon into a minimum number of convex polygons.

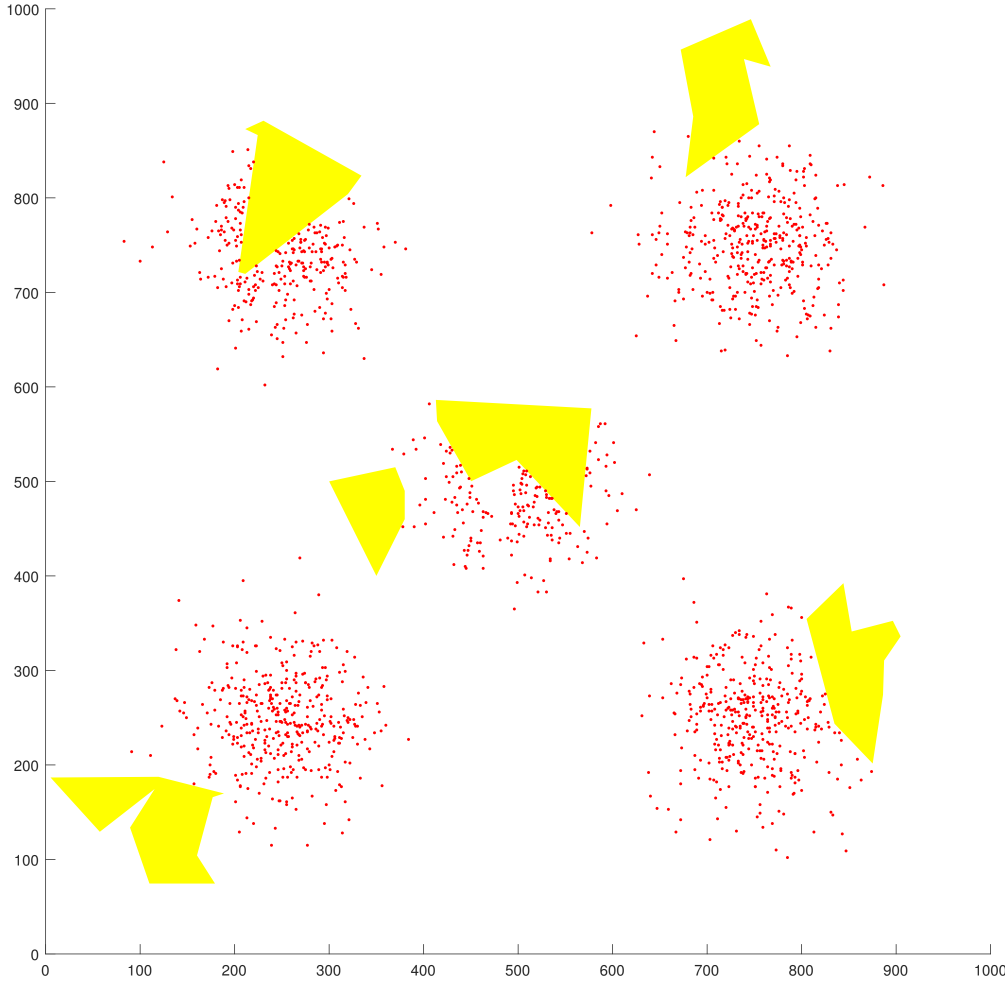


FIGURE 4.24: Problem definition

- The second task is WAY-POINTS INFECTED AREA (WIA). We create the way-points for each obstacle-free convex polygon (generated from the MD task) such that when the UAV follows these points, the whole area of the obstacle-free convex polygon will be covered.
- The third task is TRAJECTORY GENERATION (TG). This task aims at creating a boustrophedon trajectory for each convex polygon that has been created in MD. After that, by applying the TSP method, we can find the shortest path for the aerial robot to cover the whole polygon.
- The fourth task consists in finding the best path for the UAV to change between infected regions without colliding with obstacles.

4.4.1.1 UAV platform

This study deals with the spraying pesticide to plant by using UAV. Consequently, the UAV should be equipped with a mechanism that can spray the crop. We also assume that the distance from the UAV to the crop is $h(m)$ and a radial area of radius R (Figure 4.26) of the crop below the UAV will be covered by the pesticide.

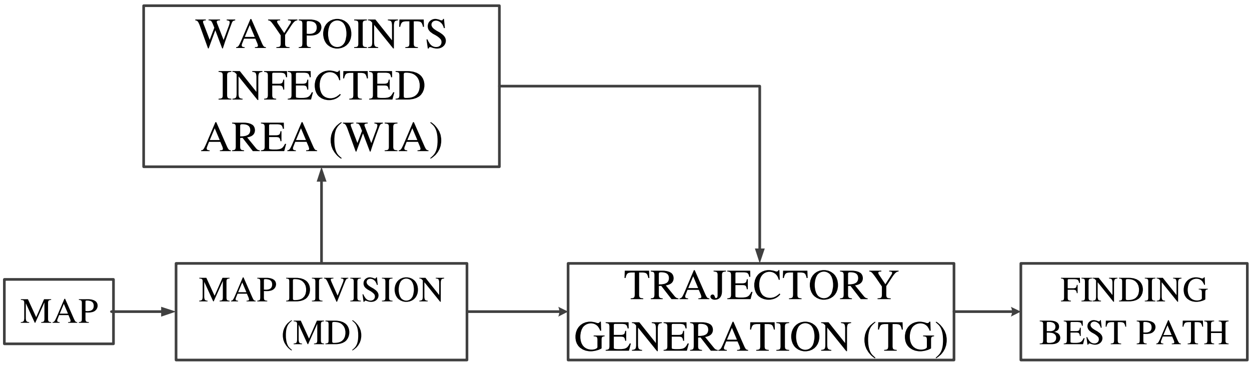


FIGURE 4.25: Path planning Algorithm

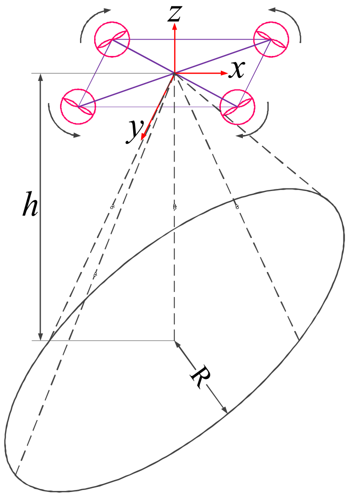


FIGURE 4.26: UAV with frame picture capture

4.4.2 Map division (MD)

The purpose of this section is to generate convex polygons which cover all the pest-ridden areas in the agriculture area with non-convex obstacles (Figure 4.24).

4.4.2.1 Classification of infected areas

The question now is that from the data of the coordinates of the pests, we need to group them into smaller non-intersecting areas. This task can be completed using clustering algorithms[93] [226]. Cluster analysis or clustering is a Machine Learning technique whose task is grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). Each data point of a given set of data points can be classified into a specific cluster. Data points in the same cluster should have analogous features or properties, while data points in different clusters should have highly dissimilar features or properties. This technique is widely applied in exploratory data mining, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

In this study, we adopt the K-means algorithm [102], the output of which is illustrated In Figure 4.27 where distinct clusters are represented by different colors.

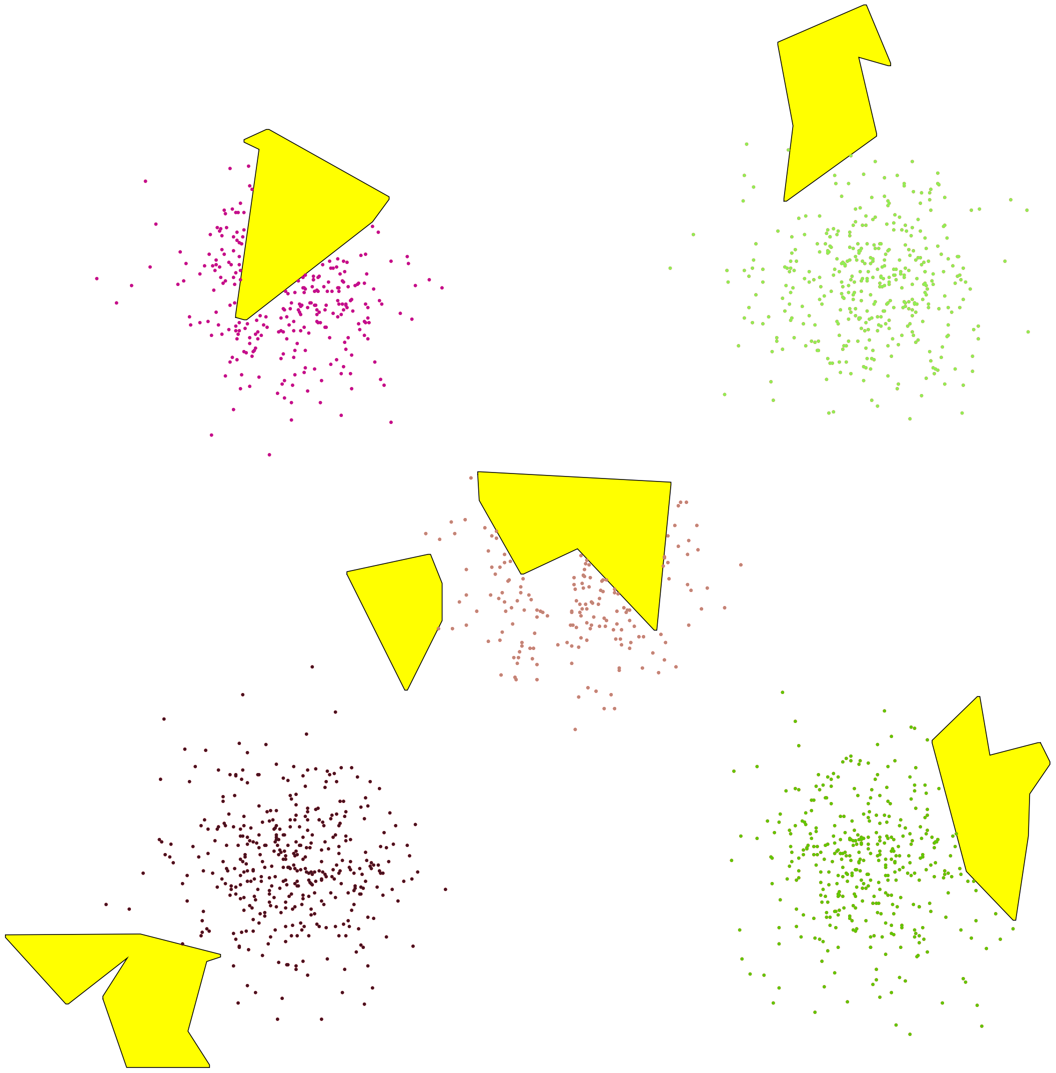


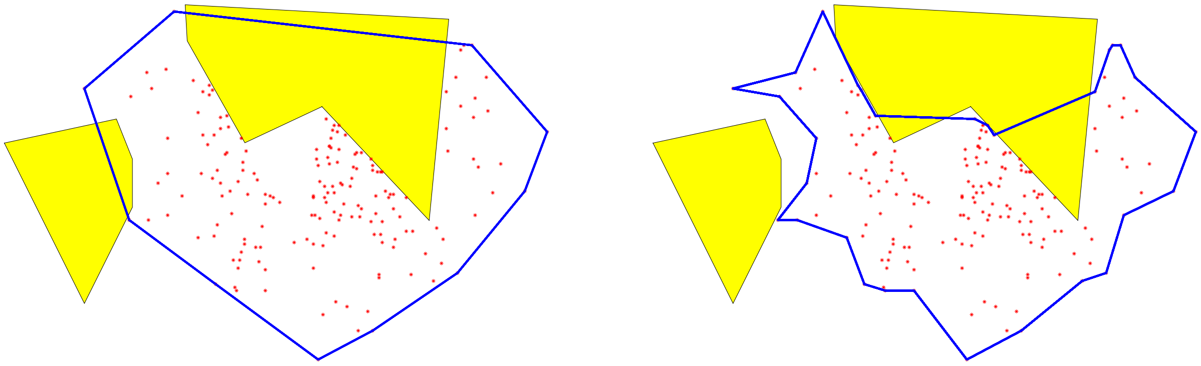
Figure 4.27: Division infected areas to several smaller regions

4.4.2.2 Calculation of polygons for covering all the infected areas

The aim of this task is to generate a polygon that contains all the infected points of each generated cluster. We propose two approaches: (i) find the minimal convex polygon which covers all the infected points in each cluster, and (ii) find the boundary polygon which covers all the infected points in each cluster. As mentioned earlier, an algorithm to compare the efficiency of the two methods will be presented in Section 4.4.3.4.

4.4.2.3 Minimal Convex Partitions

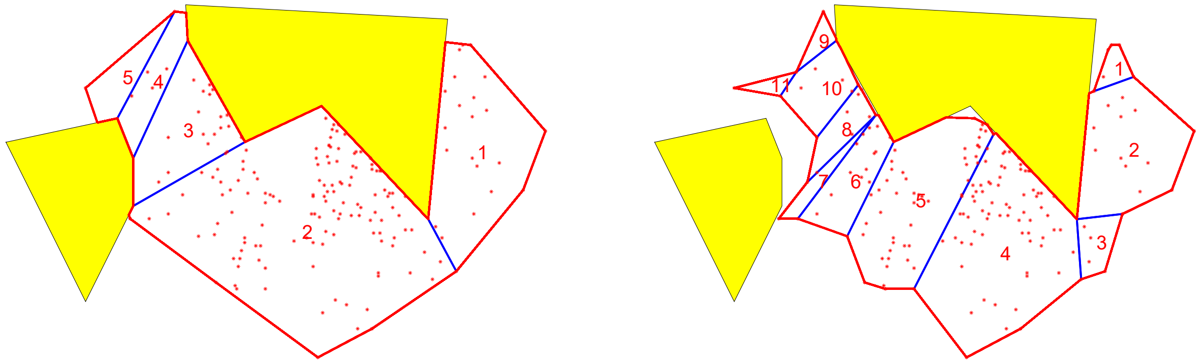
As shown in Figure 4.28, the polygon covering all infected points may occupy part of the obstacle. Therefore, a new obstacle-free polygon needs to be generated as shown in Figure 4.29.



(a) Infected points covered by miminal convex polygon method

(b) Infected points covered by boundary polygon method

FIGURE 4.28: Infected points covered by poygonal decompositions



(a) Minimal polygon decomposition for miminal convex polygon covers the infected points

(b) Minimal polygon decomposition for boundary polygon covers the infected points

FIGURE 4.29: Minimal convex polygon decomposition

A convex partition by segments of a polygon is a decomposition into convex polygons obtained by introducing arbitrary segments [89][75][7][6]. By applying Greene's dynamic programming algorithm [89], the non-convex polygon can be divided into a minimum number of convex polygons (Figure 4.29).

The algorithm that describes the process of dividing an infected area into obstacle-free convex polygons is shown in Figure 4.30.

4.4.3 Trajectory generation (TG)

The task of this section is to create a trajectory for the UAV to cover the entire infected area in a convex polygon. There exists two common trajectories for covering a convex polygon, the first is the Boustrophedon trajectory, and the second is spiral. In this thesis, we use the boustrophedon.

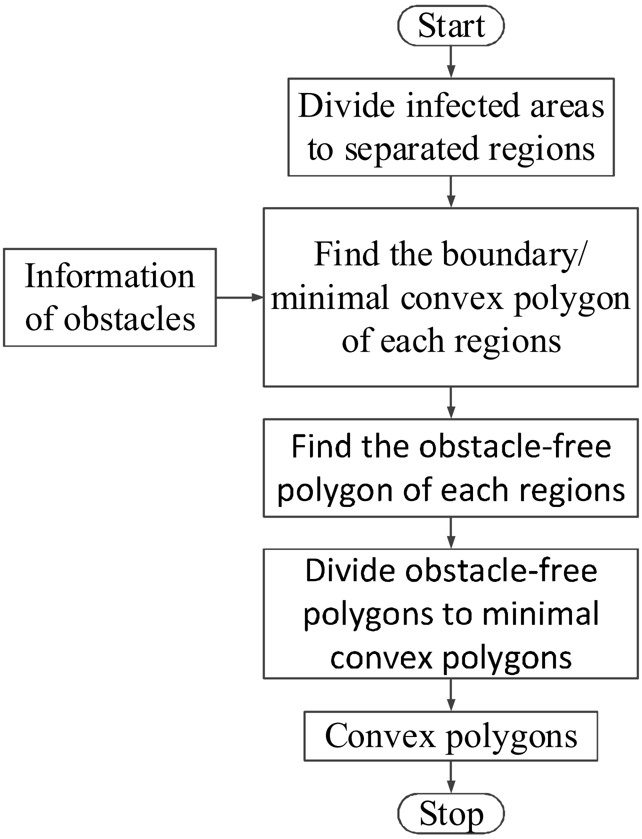


FIGURE 4.30: Map division

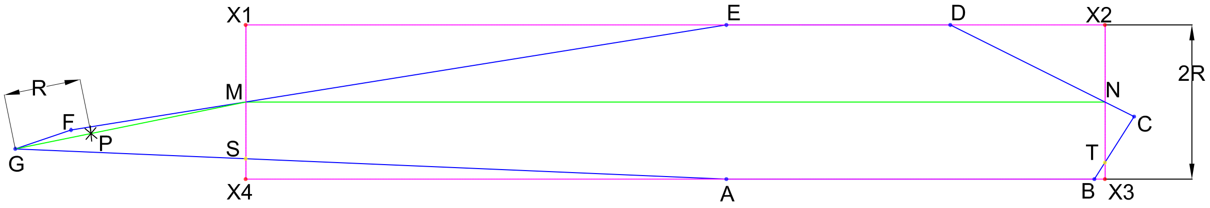


FIGURE 4.31: Trapezoid

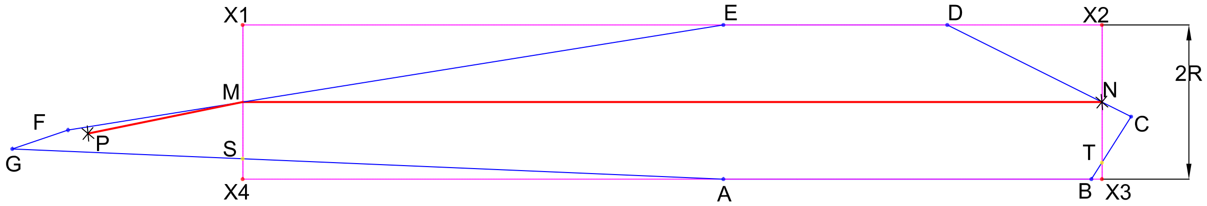
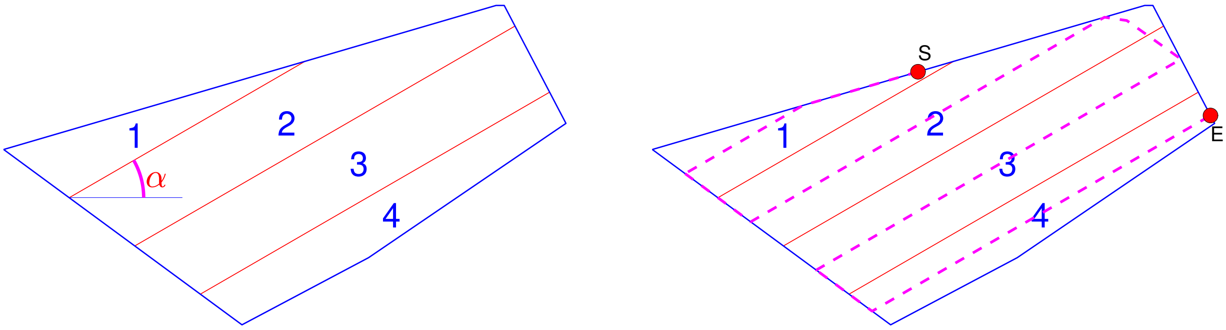


FIGURE 4.32: Way-points of trapezoid

4.4.3.1 Waypoints generation for convex polygon

Consider the convex polygon $ABCDEFG$ with two parallel sides AB and DE separated by a distance of $2R$ as shown (in Figure 4.31). Draw a parallel line at distance R to the edge AB. This line intersects the sides of the convex polygon $ABCDEFG$ at M and N. Form the rectangle $X_1X_2X_3X_4$ with two edges X_1X_2 and X_3X_4 parallel to edge AB and $X_1X_2 = X_3X_4 = MN$. When the UAV moves from M to N, overlapping region between the convex polygon $ABCDEFG$ and the rectangle $X_1X_2X_3X_4$ will be covered. It remains to find the path for the UAV in the uncovered regions of the $ABCDEF$ polygon. In Figure 4.31, these regions are made up of the triangle NTC and the quadrangle $MEGS$. Consider first the triangle NTC to the right of X_2X_3. Observe that the furthest vertex of this triangle to point N is less than R. Consequently, when the UAV is at N, it will cover the area of this triangle. For the quadrangle $MFGS$ to the left of X_1X_4, notice that its furthest vertex is G. Let P be a point on the segment GM at distance R from G. Since $ABCDEFG$ is a convex polygon, the point P and the segment PM belong to the polygon A. When the UAV moves from P to M, the whole area of $MFGS$ can be covered. Thus, PMN (the red lines in Figure 4.32) is the shortest path the UAV must follow to cover the entire area of polygon $ABCDEF$.

As calculated above, a moving trajectory can be calculated for convex polygons with two parallel sides of a distance $2R$ in between.



(a) Split convex polygon into several trapezoids with α angle

(b) Path way with α angle

FIGURE 4.33: Augmentation of the size of obstacles

The task now is to create the motion trajectory of the UAV over an entire convex polygon so that when the UAV moves in this orbit, its entire area can be covered. Consider a convex polygon that has been subdivided into smaller convex polygons as illustrated in Figure 4.33a by parallel lines at distance $2R$ from one another and at an angle α to the horizontal axis.

In this fashion, the optimal trajectory for each convex polygon $1, 2, 3, 4$ can be created. Combining these trajectories, we obtain the motion trajectory between the points S and E (Figure 4.33b). When the UAV moves along this trajectory, the entire area of the polygon can be covered.

4.4.3.2 Trajectory generation for each convex polygon

As shown in Figure 4.33a, the generated trajectory depends on the slope of the parallel lines. Therefore, we vary the angle of these parallel lines to numerically compute the shortest path. This algorithm is shown in Figure 4.34. There is an optimal α for each convex polygon.

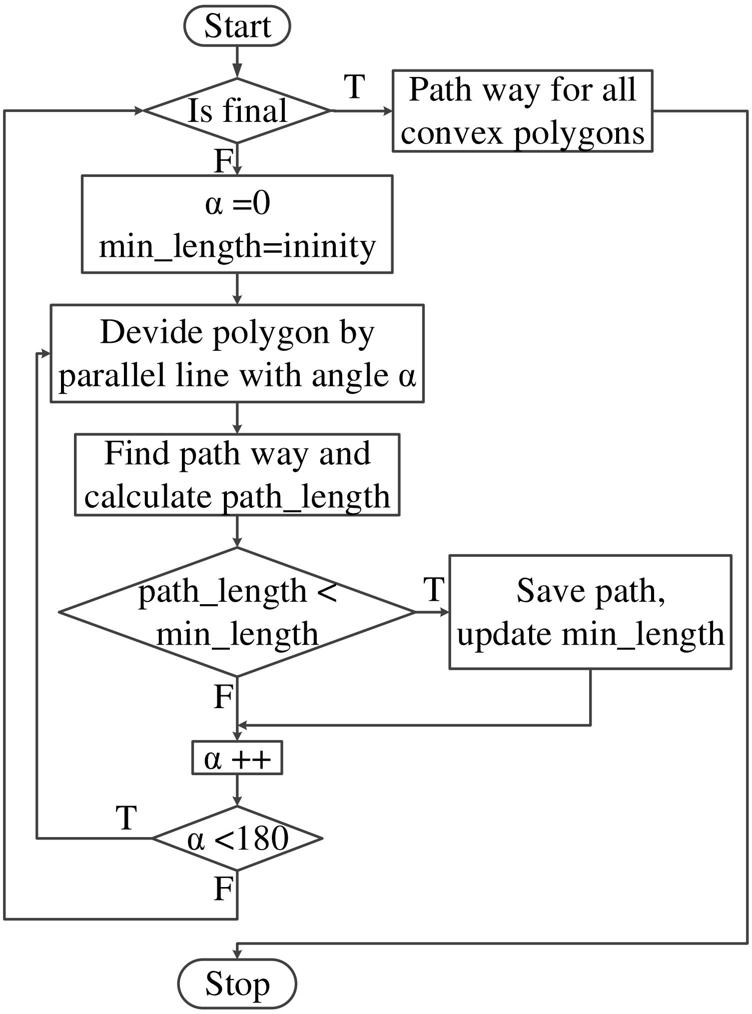


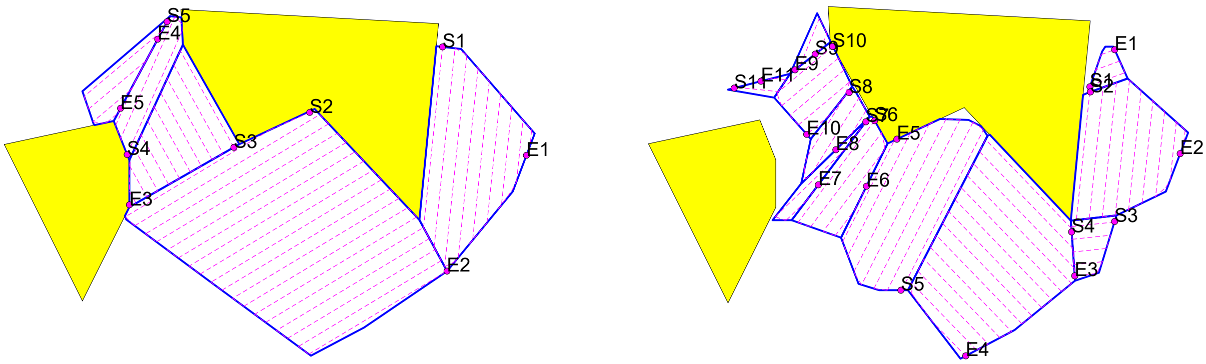
FIGURE 4.34: Algorithm for way-point of infected area

By using the algorithm in Figure 4.34, we can create the shortest trajectory for each convex polygon region so that when the UAV follows that path, the entire area of the convex polygon will be covered (Figure 4.35).

4.4.3.3 Trajectory generation for entire agriculture area

As a result of the previous section, the Boustrophedon trajectory was generated for each convex polygon. Each orbit of these convex polygons has two points S and E. This means that when UAV comes from point E, it goes to point S when moving on the Boustrophedon trajectory or vice versa from S to E, the entire area of that convex polygon will be covered. However, how the UAV can scan the entire area of a pest, also means that the UAV must move through all the generated Boustrophedon trajectory inside generated convex polygons.

An algorithm that allows finding the shortest path for UAV has been previously developed by the author [175]. By using GA with some modification on the swap, flip and slide operations and the Traveling Salesman Problem with some additional constraints, the shortest path for changing the cells has been developed. This algorithm guarantees that the solution always exists and the time for calculation is decreasing.



(a) Waypoints generation for infected areas covered by minimal convex polygon

(b) Waypoints generation for infected areas covered by boundary polygon

FIGURE 4.35: Waypoints generation for infected areas

4.4.3.4 Simulation results

The covered area and length of Boustrophedon trajectory

In this simulation, we compare the covered area and the trajectory length of generated Boustrophedon trajectories when each obstacle-free cluster is generated by minimal convex polygon (Figure 4.35a) and boundary polygon (Figure 4.35b). From the results in Table 4.4,

TABLE 4.4: Minimal convex polygon vs Boundary polygon

		Length(m)	Area(m^2)
Experiment 1	Minimal convex polygon	29860	8119
	Boundary polygon	26275	7482
Experiment 2	Minimal convex polygon	26089	7222
	Boundary polygon	23035	6841
Experiment 2	Minimal convex polygon	26146	7301
	Boundary polygon	22397	6664

we can see that the method for generating the obstacle-free by using boundary is better than the method for generating the obstacle-free by using minimal convex polygon. The area and the trajectory length for the later method are smaller than the former.

Number of classes vs covered area and length of Boustrophedon trajectory

In this simulation, we are going to compare the trajectory length when the pest-ridden areas of an agricultural area are divided into different numbers of classes. The result is shown in Figure 4.36.

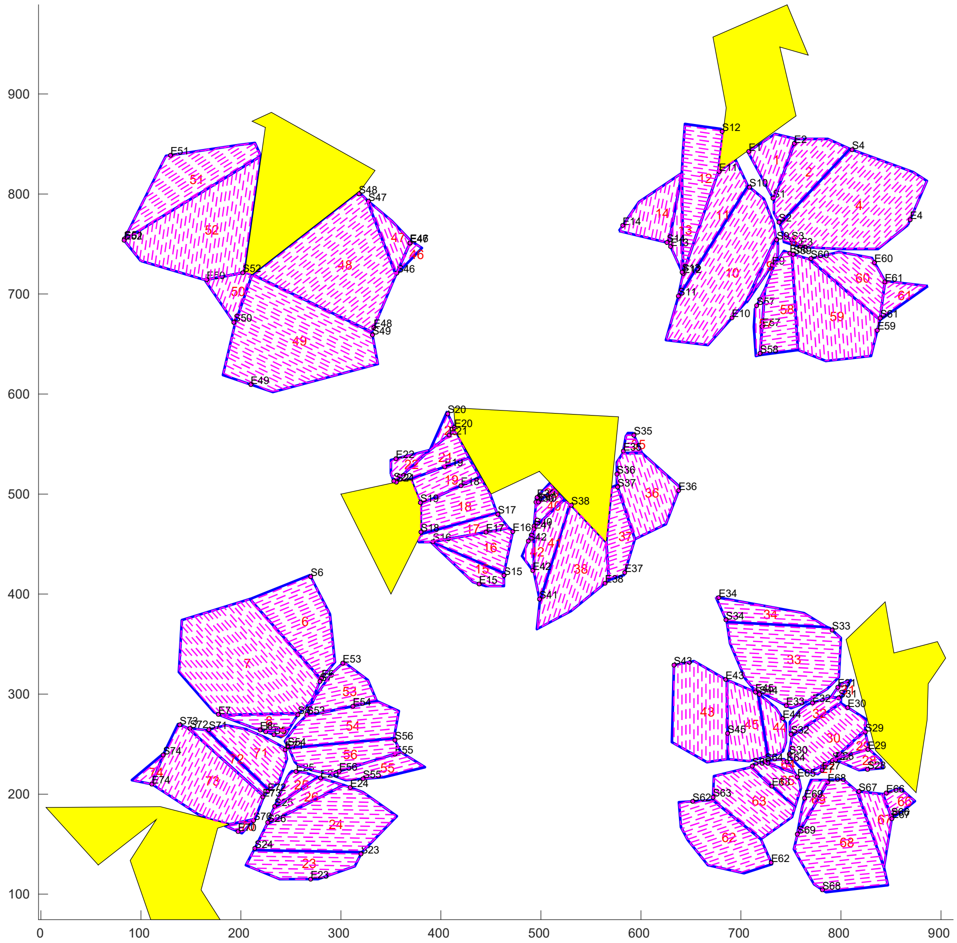


Figure 4.36: Simulation result

From the results in Table 4.5, we can see that the number of clusters affects the area of polygon needed to be covered and generated trajectory. When the number of cluster is higher, the area of polygon needed to be covered and generated trajectory are lower. Beside that, the method for generating the obstacle-free by using boundary polygon is better than the method for generating the obstacle-free by using minimal convex polygon. However, the number of divided convex polygons with the boundary polygon is much higher than the number of divided convex polygons with the minimal convex polygon decomposition. This number affects the length of trajectory for changing between divided convex polygons.

Table 4.5: Classes number vs Trajectory path length

No Cl.	Division method	No con. polygon	Area(m^2)	Trajectory length(m)
5	Min. con. polygon	14	240762	63357
	Boundary polygon	37	204158	59574
10	Min. con. polygon	20	227956	60764
		19	227681	60588
		19	229330	60965
		15	223351	59009
		18	229149	60893
	Boundary polygon	54	204158	56700
		55	206239	57272
		56	203441	56567
		56	206246	57423
		59	205353	57357
15	Min. con. polygon	24	218952	58573
		23	214078	57310
		23	208604	55935
		22	215226	57651
		26	221521	59350
	Boundary polygon	77	191593	54729
		74	191657	55345
		68	181859	50956
		79	193424	55266
		76	196747	54123

4.5 Conclusion

In this chapter, we have studied two tasks for the quadrotor to be used in PA:

- Infected area detection
- Pesticide spraying all the infected areas

For the first task, we have proposed a new approach for maximizing the coverage area and minimizing the path length of an aerial robot in agriculture environment with concave obstacles. Firstly, we propose a new approach for cellular decomposition. Based on the critical points as in [45] and [225], the extension of this approach is that the critical points have been categorized to several groups, and the cells have been created from these groups. This extension leads to a decrease of the number of cells after decomposition. The results show that this new cellular decomposition works well even with several concave obstacles inside the environment.

Secondly, we have proposed a method for maximizing the percentage of coverage and we have given some comments for the trade-off between the percentage of coverage and the number of way-points for the UAV. The percentage of coverage is up to 97.9%.

Finally, by using GA with some modification on the swap, flip and slide operations and the TSP with some additional constraints, the shortest path for changing the cells has been developed. This algorithm guarantees that the solution always exists and the time for calculation is decreasing.

For the second task, we have proposed a method for generating a trajectory which allows the UAV to put the medicine to the entire pest-ridden area of an agricultural area. First, the pest-ridden areas have been divided into several smaller areas (clusters) by using clustering technique. After that, each cluster is divided again to several obstacle-free convex polygons. Then the shortest Boustrophedon trajectory is created in each obstacle-free convex polygon. Finally, the shortest trajectory for changing between obstacle-free convex polygons is generated to form the final trajectory.

Several extensions from this section are possible. One might consider the recalculating the trajectory under the windy condition of the environment or trajectory generation for a team of UAV. Field tests are also subject of future work. The continuation of this work is also to add the UAV equations of motion as constraints on the path. Then the efficiency of the algorithm will be improved to be real-time usable.

5

Nonlinear robust control and state estimation of UAVs

Chapter abstract

Motion control design plays a crucial role in aerial autonomous vehicles for accomplishing the predefined tasks. The UAV faces a variety of disturbances when performing outdoor missions. Especially for the duty of spraying pesticides or transporting goods, the mass of the Unmanned Aerial Vehicle (UAV) changes. It is followed by the changes in dynamic parameters such as the moments of inertia. Therefore, developing a suitable control algorithm that is robustly stable under the external disturbances and the changes of parameters is essential. This chapter is organized as follows: in Section 5.1, we take a quick look at the control system of the quadcorter. Due to the changes of the mass and moments of inertia, their values should be calculated or estimated and will be presented in section 5.2. Then, the position/altitude control of quadcopter is shown in section 5.3. In the next section 5.4, the dynamic output feedback controller will be designed for the attitude subsystem of the quadcopter. Next, the Attitude/Altitude LPV Unknown Input Observer (UIO) of mass-varying quadcopter is designed in section 5.5. The Attitude/Altitude LPV H_∞ State feedback Controller and Static output feedback LPV for mass-varying controller are presented respectively in sections 5.6 and 5.7. Finally, the conclusions are given in the last section.

This Chapter contains:

5.1 Control system for quadcopter

For the purpose of convenience, the mathematical equations of the quadcopter model without disturbances in (3.53) can be written as follows:

$$\ddot{x} = (\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi)\frac{U_1}{m} \tag{5.1a}$$

$$\ddot{y} = (\cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi)\frac{U_1}{m} \tag{5.1b}$$

$$\ddot{z} = (\cos\varphi \cos\theta)\frac{U_1}{m} - g \tag{5.1c}$$

$$\ddot{\varphi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r}{I_x}\Omega_r\dot{\theta} + \frac{1}{I_x}U_2 \tag{5.1d}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\varphi}\dot{\psi} + \frac{J_r}{I_y}\Omega_r\dot{\varphi} + \frac{1}{I_y}U_3 \tag{5.1e}$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\varphi}\dot{\psi} + \frac{1}{I_z}U_4 \tag{5.1f}$$

The system can be rewritten in the state-space form

$$\dot{X} = f(X,U) \tag{5.2}$$

where U is the input vector and X denotes the state vector choosen as follows:

$$X = \begin{bmatrix} x & y & z & \varphi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & \dot{\varphi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \tag{5.3a}$$

$$U = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix}^T \tag{5.3b}$$

where inputs are mapped by

$$\begin{aligned} U_1 &= k_f\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \\ U_2 &= k_f\left(-\omega_2^2 + \omega_4^2\right) \\ U_3 &= k_f\left(\omega_1^2 - \omega_3^2\right) \\ U_4 &= k_z\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right) \end{aligned} \tag{5.4}$$

and

$$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{5.5}$$

where ω_i for $i = 1,2,3,4$ denotes the i-th rotor velocity, and T_i for $i = 1,2,3,4$ are the thrust generated by the i-th rotor and the thrust $T_i(t)$ is a function of the rotor speed defined by

$$T_i(t) = k_f\omega_i^2 \tag{5.6}$$

where k_f and k_z are the constant coefficients, and

$$f(X,U) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ u_x \frac{U_1}{m} \\ u_y \frac{U_1}{m} \\ (\cos\varphi \cos\theta) \frac{U_1}{m} - g \\ \frac{I_y - I_z}{I_x} \dot{\theta}\dot{\psi} - \frac{J_r \Omega_r}{I_x} \dot{\theta} + \frac{l}{I_x} U_2 \\ \frac{I_z - I_x}{I_y} \dot{\varphi}\dot{\psi} + \frac{J_r \Omega_r}{I_y} \dot{\varphi} + \frac{l}{I_y} U_3 \\ \frac{I_x - I_y}{I_z} \dot{\varphi}\dot{\theta} + \frac{1}{I_z} U_4 \end{bmatrix} \tag{5.7}$$

where

$$u_x = \cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi \tag{5.8a}$$
$$u_y = \cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi \tag{5.8b}$$

It is worthwhile to note in the latter system that the angles and their time derivatives do not depend on translation components. On the other hand, the translations depend on the angles. One can ideally imagine the overall system described by (5.7) as constituted of two subsystems, the angular rotations and the linear translations (see Figure. 5.1).

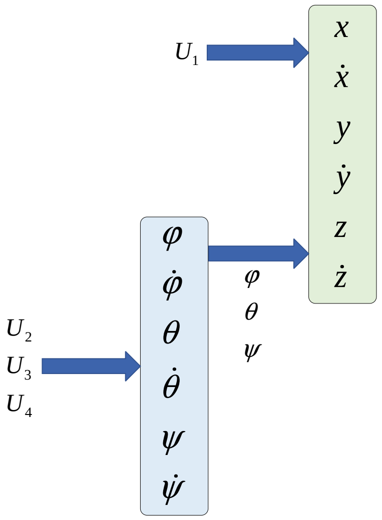


FIGURE 5.1: Connection between rotational and translational subsystems of the quadcopter

Therefore the control system of the quadcopter can be divided into subcontrollers such as position controller (for controlling x and y), altitude controller (for controlling z), and attitude controller (for controlling three Euleur angles φ, θ, and ψ) as described in Figure 5.2

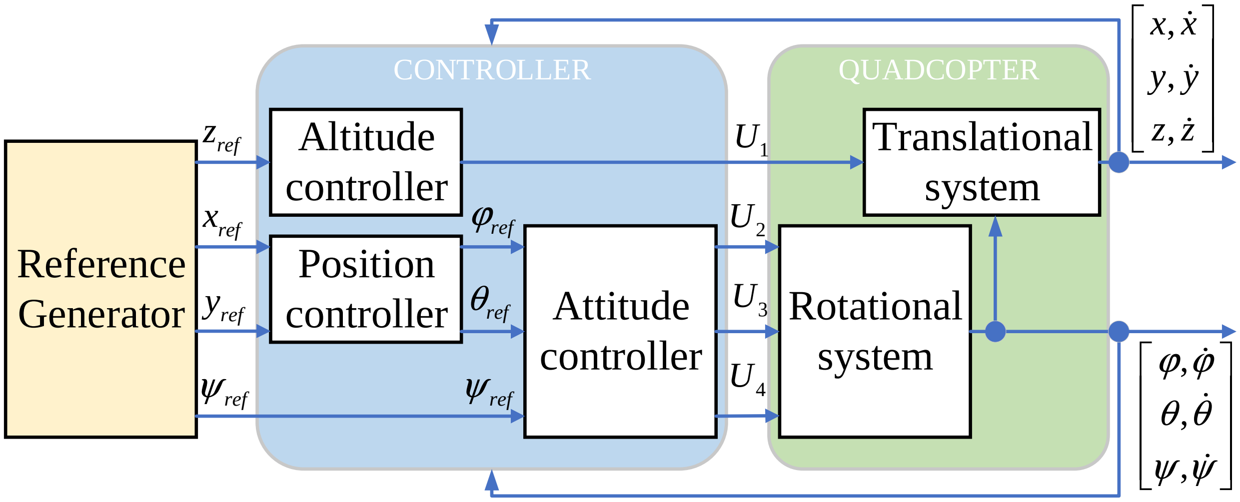


FIGURE 5.2: General Controller system structure for quadcopter

5.2 Quadrotor's states and parameters estimation

5.2.1 State estimation overview

Thanks to the maneuverability, low cost and ability to hover in place, the quadrotors have been widely applied in applications for a variety of dangerous or complex tasks, such as inspection [166] [247], search and rescue [149], exploration [220], interaction with the environment [218] [77], transportation [135], and mapping [136]. TO ensure the successful completion of the aforementioned tasks, precise control, and reliable navigation performance are essential. Therefore, the accurate knowledge of the vehicle's physical characteristics and its states plays a really critical role.

In [236], the state variables (position, velocities, and attitude) of an autonomous quadrotor UAV were estimated by using the Kalman Filter (KF) under the condition of sufficient measurements. In [66], the quadrotor's attitude was estimated by using two Extended Kalman Filters (EKF) along with a Direction Cosine Matrix (DCM) algorithm for a single low-cost IMU sensor. Also in this work, the authors use Extended Kalman Filter (EKF) for filtering a 3-axis gyro sensor to detect the angular rate, then Euler angles are calculated through the DCM algorithm using the filtered gyro and magnetometer sensor. Then, another EKF is used to enhance the estimation of the Euler angles. In [241], the authors use Optimal Kalman Filter (OKF) to estimate the state variables of the quadrotor under the presence of the white Gaussian process and measurement noises, which are caused by the actuator and sensor faults, respectively.

5.2.2 Quadrotor's parameters estimation

Besides the state variables, other dynamic parameters such as mass and moments of inertia are also important and need to be determined accurately.

In [129] and [113], the authors proposed approaches based on kinematic models. They identify the spatial-temporal poses of sensors such as IMU, monocular or stereo cameras. To estimate the inertia properties, the authors combine the IMU, pose sensor, and motor speed measurements. This not only allows the estimation of relative positions of the different sensor modules and the linear and rotational IMU biases but also the position of the Center of Mass, the Moment of Inertia and the mass.

In [148], the frequency-domain approach is used for determining the Moment of Inertia of the quadrotor. This work does not use a probabilistic framework to eliminate the noise and uncertainty associated with the sensor data. This approach also limits the motions of the system to one axis at a time and unable to estimate the position of the center of mass.

5.2.3 Calculation/estimation of mass and moments of inertia

When the UAV sprays pesticides or delivers packages, the mass of the UAV decreases gradually or abruptly. In both cases, the total mass of the UAV is the same and is assumed to contain several smaller cylinders. In the case of a gradual mass reduction, the number of cylinders is much larger than the case of a abrupt mass change.

In particular, we assume that there are n cylinders $m_1, m_2, ..., m_n$ attached to the UAV as shown in Fig. 5.3. Each cylinder $m_i, i = 1, ..., n$ has the height h_i, radius r, and mass m_i. To simulate the process of reducing the mass of the UAV, the cylinders $m_i, i = 1, ..., m$ will be sequentially $(m_n, ..., m_1)$ detached from the UAV over time. We call x_k, y_k, and z_k(Fig. 5.3) the three axes x, y, and z for calculation the moments of inertia of the mass which contains k cylinders from 1 to k, where $k = 1, ..., n$.

Due to the variation of the mass of the UAV over time, the moments of inertia will also change. Therefore, the moments of inertia of the whole system need to be recalculated at each change. Moments of inertia of mass $(m_1 + ... + m_k), k = 1, ..., n$ with respect to their x

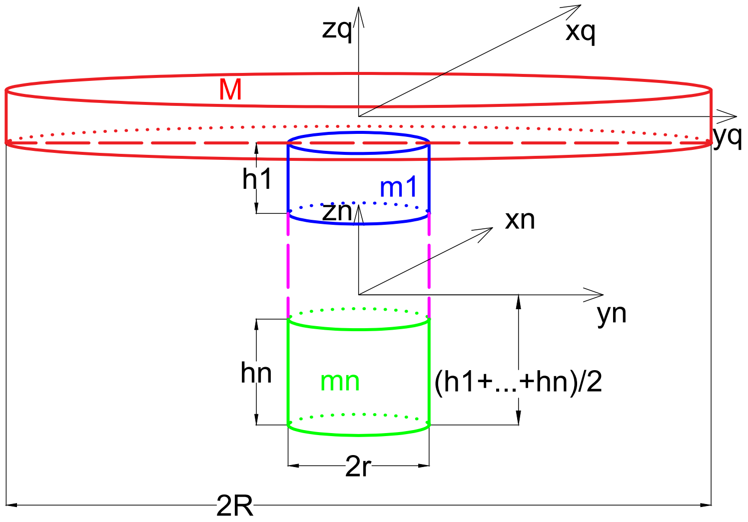


FIGURE 5.3: Moment of inertia of the systems quadcopter-mass

and y axes x_k and y_k respectively are:

$$\begin{aligned} J_{(m_1+...+m_k)/x_k} =& J_{(m_1+...+m_k)/y_k} \\ =& \left(\sum_{i=1}^{k} m_i\right)\frac{r^2}{4} + \left(\sum_{i=1}^{k} m_i\right)\frac{\left(\sum\limits_{i=1}^{k} h_i\right)^2}{12} \end{aligned} \tag{5.9}$$

Hence, moments of inertia of mass $(m_1 + ... + m_k), k = 1, ..., n$ with respect to the axes x_q and y_q of the quadcopter are:

$$\begin{aligned} J_{(m_1+...+m_k)/x_q} =& J_{(m_1+...+m_k)/y_q} \\ =& J_{(m_1+...+m_k)/x_k} + \left(\sum_{i=1}^{k} m_i\right)\frac{\left(\sum\limits_{i=1}^{k} h_i\right)^2}{2} \\ =& \sum_{i=1}^{n} m_i \left(\frac{r^2}{4} + \frac{7}{12}\left(\sum_{i=1}^{k} h_i\right)^2\right) \end{aligned} \tag{5.10}$$

Moments of inertia of mass $(m_1 + ... + m_k), k = 1, ..., n$ with respect to their z_k axes and the z_q axis of the quadcopter are:

$$J_{(m_1+...+m_k)/z_k} = J_{(m_1+...+m_k)/z_q} = \left(\sum_{i=1}^{k} m_i\right)\frac{r^2}{2} \tag{5.11}$$

Finally, moments of inertia of a system which contains the quadcopter and k cylinders $m_1, ..., m_k$ for $k = 1, .., n$ with respect to the three axis of the quadcopter x_q, y_q, and z_q are:

$$\begin{aligned} J_{[quad\_(m_1+...+m_k)]/x_q} =& J_{quad/x_q} + J_{(m_1+...+m_k)/x_q} \\ =& J_{quad/x_q} + \sum_{i=1}^{k} m_i \left(\frac{r^2}{4} + \frac{7}{12}\left(\sum_{i=1}^{k} h_i\right)^2\right) \end{aligned} \tag{5.12}$$

$$\begin{aligned} J_{[quad\_(m_1+...+m_k)]/y_q} =& J_{quad/y_q} + J_{(m_1+...+m_k)/y_q} \\ =& J_{quad/y_q} + \sum_{i=1}^{k} m_i \left(\frac{r^2}{4} + \frac{7}{12}\left(\sum_{i=1}^{k} h_i\right)^2\right) \end{aligned} \tag{5.13}$$

$$\begin{aligned} J_{[quad\_(m_1+...+m_k)]/z_q} =& J_{quad/z_q} + J_{(m_1+...+m_k)/z_q} \\ =& J_{quad/z_q} + \left(\sum_{i=1}^{k} m_i\right)\frac{r^2}{2} \end{aligned} \tag{5.14}$$

5.3 Position/Altitude control

In this section, the translational subsystem will be considered. The differential equations of dynamic translational subsystem is given by:

$$\begin{cases} \ddot{x} = (\sin\psi\sin\varphi + \cos\psi\sin\theta\cos\varphi)\frac{U_1}{m} \\ \ddot{y} = (\sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi)\frac{U_1}{m} \\ \ddot{z} = (\cos\theta\cos\varphi)\frac{U_1}{m} - g \end{cases} \tag{5.15}$$

The state vector is then defined by:

$$X = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T$$

5.3.1 Position control

As U_1 controls z, we can only define desired φ_{ref} and θ_{ref} to be computed to achieve that x_c and y_c go to desired x_{ref} and y_{ref}.

In this subsection, a simple PD controller is designed for calculating the required roll φ_{ref} and pitch θ_{ref}.

We define the position and velocity errors x and y as:

$$\begin{aligned} e_{px} &= x_{ref} - x \\ e_{vx} &= \dot{x}_{ref} - \dot{x} \\ e_{py} &= y_{ref} - y \\ e_{vy} &= \dot{y}_{ref} - \dot{y} \end{aligned}$$

And we want these errors to decay exponentially to 0.

If we take the error term and make it obey a second order linear differential equation with proper coefficients, we can guarantee that the error goes exponentially to 0.

Means:

$$\begin{aligned} \ddot{x}_{ref} - \ddot{x}_c + K_{dx}\dot{e}_{vx} + K_{px}e_{px} &= 0 \\ \ddot{y}_c - \ddot{y}_c + K_{dy}\dot{e}_{vy} + K_{py}e_{py} &= 0 \end{aligned}$$

where $\ddot{x}_c$ and $\ddot{y}_c$ are the commanded accelerations which are calculated by the controller.

Hence, the second derivative of commanded x_c and y_c are calculated as:

$$\begin{aligned} \ddot{x}_c &= \ddot{x}_{ref} + K_{p_x}(x_{ref} - x) + K_{d_x}(\dot{x}_{ref} - \dot{x}) \\ \ddot{y}_c &= \ddot{y}_{ref} + K_{p_y}(y_{ref} - y) + K_{d_y}(\dot{y}_{ref} - \dot{y}) \end{aligned} \tag{5.16}$$

Suppose that the quadcopter is around the hover position, we have:

$$\begin{aligned} U_1 &\approx mg \\ \theta &\approx 0 \\ \varphi &\approx 0 \\ \psi &\approx \psi_{ref} \end{aligned}$$

Therefore, the required φ_{ref} and θ_{ref} can be obtained as:

$$\begin{aligned} \varphi_{ref} &= \frac{1}{g}(\ddot{x}_c \cdot \sin\psi_{ref} - \ddot{y}_c \cdot \cos\psi_{ref}) \\ \theta_{ref} &= \frac{1}{g}(\ddot{x}_c \cdot \cos\psi_{ref} + \ddot{y}_c \cdot \cos\psi_{ref}) \end{aligned} \tag{5.17}$$

Coefficients of PD controller are selected to ensure stability and the responses to trajectory tracking as follows:

$$K_{P_x} = 20; \qquad K_{d_x} = 2; \qquad K_{P_y} = 25; \qquad K_{d_y} = 3$$

5.3.2 Altitude control

In this subsection, we design a integral back-stepping controller for altitude of the quadcopter which is able to track the reference. The error of the altitude z (state x_5), its derivative are defined as follows:

$$\begin{aligned} z_5 &= x_{5d} - x_5 \\ \dot{z}_5 &= \dot{x}_{5d} - x_6 \end{aligned} \tag{5.18}$$

For improving the system's robustness against modeling uncertainties and external disturbance, thus improving steady-state control accuracy, we define also the integral of state x_5 as

$$\xi_5 = \int_0^t z_5(\tau)\, d\tau \tag{5.19}$$

We study the stability of the altitude system via Lyapunov theory, the Lyapunov function and its derivative are:

$$\begin{array}{l} V(z_5) = \frac{1}{2} z_5^2 \\ \dot{V}(z_5) = z_5(\dot{x}_{5d} - x_6) \end{array} \tag{5.20}$$

We can see no direct control law in (5.20). Consequently, x_6 is defined as a virtual control. To make $\dot{V}(z_5)$ negative semi-definite, the desired virtual control is defined as:

$$x_{6d} = \dot{x}_{5d} + c_5 z_5 + \lambda_5 \xi_5 \tag{5.21}$$

where c_5 and λ_5 are positive numbers.

In order to make x_6 follows the stabilizing function x_{6d}, we define the error state z_6 as the deviation between x_{6d} and x_6:

$$z_6 = x_{6d} - x_6 \tag{5.22}$$

The virtual control x_6 and the derivative of z_6 are:

$$x_6 = \dot{x}_{5d} + c_5 z_5 + \lambda_5 \xi_5 - z_6 \tag{5.23}$$

$$\dot{z}_6 = \ddot{x}_{5d} + c_5 \dot{z}_5 + \lambda_5 \dot{\xi}_5 - \frac{\cos\varphi \cos\theta}{m} U_1 + g \tag{5.24}$$

The Lyapunov function is extended as

$$V_1 = \frac{1}{2} z_5^2 + \frac{1}{2} \lambda_5 \xi_5^2 \tag{5.25}$$

The time derivative of the Lyapunov function V_1 is:

$$\begin{aligned} \dot{V}_1 &= z_5 \dot{z}_5 + \lambda_5 \xi_5 \dot{\xi}_5 \\ &= z_5(\dot{x}_{5d} - x_6) + \lambda_5 \xi_5 z_5 \\ &= z_5(z_6 - c_5 z_5 - \lambda_5 \xi_5) + \lambda_5 \xi_5 z_5 \\ &= -c_5 z_5^2 + z_5 z_6 \end{aligned} \tag{5.26}$$

For making the time derivative of Lyapunov function negative, z_6 should be added to it. Define the new Lyapunov function as

$$V_2 = \frac{1}{2} z_5^2 + \frac{1}{2} \lambda_5 \xi_5^2 + \frac{1}{2} z_6^2 \tag{5.27}$$

Then, the derivative of the new Lyapunov function is

$$\dot{V}_2 = -c_5 z_5^2 + z_5 z_6 + z_6 \left(\ddot{x}_{5d} + c_5 \dot{z}_5 + \lambda_5 \dot{\xi}_5 - \frac{\cos\varphi \cos\theta}{m} U_1 + g \right) \tag{5.28}$$

For ensuring the negativity of $\dot{V}_2$, we choose U_1 as

$$U_1 = \frac{m}{\cos\varphi\cos\theta} \left(\ddot{x}_{5d} + \left(1 - c_5^2 + \lambda_5\right) z_5 - c_5 \lambda_5 \xi_5 + (c_5 + c_6) z_6 + g \right) \tag{5.29}$$

where c_6 is a positive number.

Then

$$\dot{V}_2 = -c_5 z_5^2 - c_6 z_6^2 \tag{5.30}$$

Coefficients of the Back-stepping controller are selected as:

$$\mathrm{c}_5 = 0.001; \quad \mathrm{c}_6 = 30; \quad \lambda_5 = 30$$

5.4 LPV H_∞ Attitude control

In this section, the rotational subsystem will be considered. In this section, we assume that the quadcopter is symmetric and $I_x = I_y$.

Therefore, the differential equations of rotational dynamic subsystem is given as:

$$\ddot{\varphi} = \frac{I_x - I_z}{I_x} \dot{\theta}\dot{\psi} - \frac{J_r \Omega_r}{I_x} \dot{\theta} + \frac{l}{I_x} U_2 \tag{5.31a}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_x} \dot{\varphi}\dot{\psi} + \frac{J_r \Omega_r}{I_x} \dot{\varphi} + \frac{l}{I_x} U_3 \tag{5.31b}$$

$$\ddot{\psi} = \frac{1}{I_z} U_4 \tag{5.31c}$$

5.4.1 Roll-pitch H_∞ controller

The dynamic of Roll-Pitch subsystem (5.31a - 5.31b) is rewritten in descriptor form for reducing the number of subsystems as

$$\begin{cases} I_x \ddot{\varphi} & = & (I_x - I_z) \dot{\theta}\dot{\psi} - J_r \Omega_r \dot{\theta} + l U_2 \\ I_x \ddot{\theta} & = & -(I_x - I_z) \dot{\varphi}\dot{\psi} + J_r \Omega_r \dot{\varphi} + l U_3 \end{cases} \tag{5.32}$$

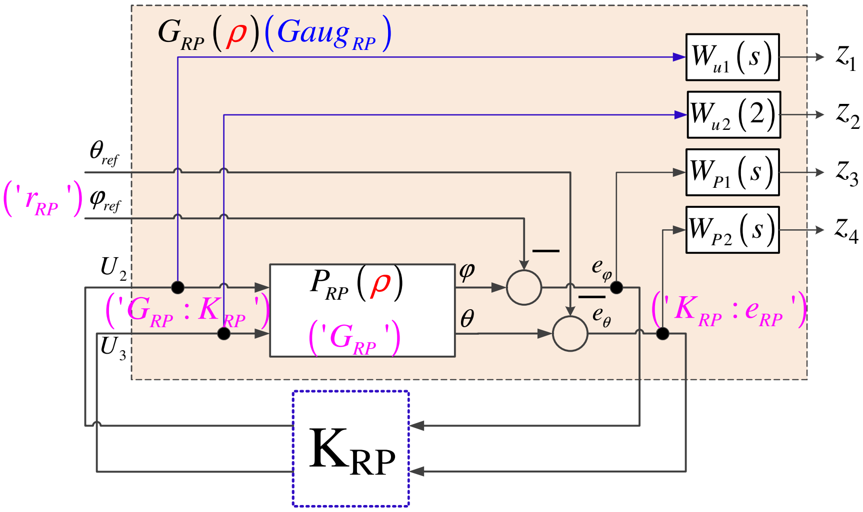


FIGURE 5.4: Roll Pitch H_∞ controller.

State vector is

$$X_{RP}^T = \begin{bmatrix} \varphi & \theta & \dot{\varphi} & \dot{\theta} \end{bmatrix}^T$$

The roll-pitch susbsystem can be written as:

$$\begin{array}{rcl} E_{RP}\dot{X} & = & A_{RP}X + B_{RP}u \\ y & = & C_{RP}X + D_{RP}u \end{array} \tag{5.33}$$

where:

$$E_{RP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \begin{array}{c} -(I_x - I_z)\dot{\psi} \\ +J_r\Omega_r \end{array} & I_x & 0 \\ \begin{array}{c} -(I_x - I_z)\dot{\psi} \\ -J_r\Omega_r \end{array} & 0 & 0 & I_x \end{bmatrix}$$

$$A_{RP} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad B_{RP} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ l & 0 \\ 0 & l \end{bmatrix}$$

$$C_{RP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \quad D_{RP} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The parameter ρ_{RP} that is varying is:

$$\begin{array}{rcl} \rho_{RP} & = & \begin{bmatrix} I_x & (I_x - I_z)\dot{\psi} & \Omega_r \end{bmatrix}^T \\ & = & \begin{bmatrix} \rho_{RP1} & \rho_{RP2} & \rho_{RP3} \end{bmatrix}^T \end{array} \tag{5.34}$$

Their varying ranges are in table 5.1.

TABLE 5.1: Parameters range for LPV H_∞ Roll-Pitch and Yaw controllers

Parameter	Description	Range
ρ_{RP_1}	I_x	$[0.0075, 0.0128]\, kg.m^2$
ρ_{RP_2}	$(I_x - I_z)\dot{\psi}$	$\left[-5\times10^{-4}, 0.0173\right]\, kg.m^2$
ρ_{RP_3}	Ω_r	$[-300, 300]\, rad.s^{-1}$
ρ_Y	I_z	$[0.0130, 0.0162]\, kg.m^2$

In order to express the system in a polytopic form, matrices $A_{RP}, B_{RP}, C_{RP}, D_{RP}, E_{RP}$ can be decomposed as:

$$\begin{array}{rcl} A_{RP} & = & A_{RP_0} + \rho_{RP_1}A_{RP_1} + \rho_{RP_2}A_{RP_2} + \rho_{RP_3}A_{RP_3} \\ B_{RP} & = & B_{RP_0} + \rho_{RP_1}B_{RP_1} + \rho_{RP_2}B_{RP_2} + \rho_{RP_3}B_{RP_3} \\ C_{RP} & = & C_{RP_0} + \rho_{RP_1}C_{RP_1} + \rho_{RP_2}C_{RP_2} + \rho_{RP_3}C_{RP_3} \\ D_{RP} & = & D_{RP_0} + \rho_{RP_1}D_{RP_1} + \rho_{RP_2}D_{RP_2} + \rho_{RP_3}D_{RP_3} \\ E_{RP} & = & E_{RP_0} + \rho_{RP_1}E_{RP_1} + \rho_{RP_2}E_{RP_2} + \rho_{RP_3}E_{RP_3} \end{array}$$

where:

$$A_0 = \left[\begin{array}{c|c} 0_2 & I_{22} \\ \hline 0_{22} & 0_{22} \end{array}\right]; A_1 = A_2 = A_3 = A_4 = [0_{44}]$$

$$B_0 = l\left[\begin{array}{c} 0_{22} \\ \hline I_{22} \end{array}\right]; B_1 = B_2 = B_3 = B_4 = [0_{42}]$$

$$C_0 = \left[\begin{array}{c|c} I_{22} & 0_{22} \end{array}\right]; C_1 = C_2 = C_3 = C_4 = [0_{24}]$$

$$D_0 = D_1 = D_2 = D_3 = D_4 = [0_{22}]$$

$$E_0 = \left[\begin{array}{cc} I_{22} & 0_{22} \\ 0_{22} & 0_{22} \end{array}\right]; E_1 = \left[\begin{array}{cc} 0_{22} & 0_{22} \\ 0_{22} & I_{22} \end{array}\right];$$

$$E_2 = \left[\begin{array}{ccc} \multicolumn{2}{c}{0_{22}} & 0_{22} \\ 0 & -1 & \multirow{2}{*}{0_{22}} \\ 1 & 0 & \end{array}\right]; E_3 = \left[\begin{array}{ccc} \multicolumn{2}{c}{0_{22}} & 0_{22} \\ 0 & -J_r & \multirow{2}{*}{0_{22}} \\ J_r & 0 & \end{array}\right]$$

Weight functions for Roll-Pitch H_∞ controller are chosen as following:

$$\begin{array}{l} W_{u1} = \frac{s}{s+400000}; W_{u2} = \frac{s}{s+400000} \\ W_{P1} = \frac{1}{s+0.1}; W_{P2} = \frac{1}{s+0.1} \end{array}$$

The norm of LPV H_∞ Roll-Pitch subsystem is $\gamma_{RP} = 0.544$

Remark 5.1. *In this simulation, LPV gain-scheduled H_∞ controller is synthesized by using the command $hinfgs$ of the robust control toolbox in Matlab. Basically, this controller is the dynamic output feedback controller. This controller minimizes the closed-loop quadratic H_∞ performance from reference signal $\begin{bmatrix} \varphi_{ref} & \theta_{ref} \end{bmatrix}^T$ to $\begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T$ signal.* □

5.4.2 Yaw H_∞ controller

The dynamic of Yaw susbsystem (5.31c) is rewritten in descriptor form as

$$I_z\ddot{\psi} = U_4 \tag{5.35}$$

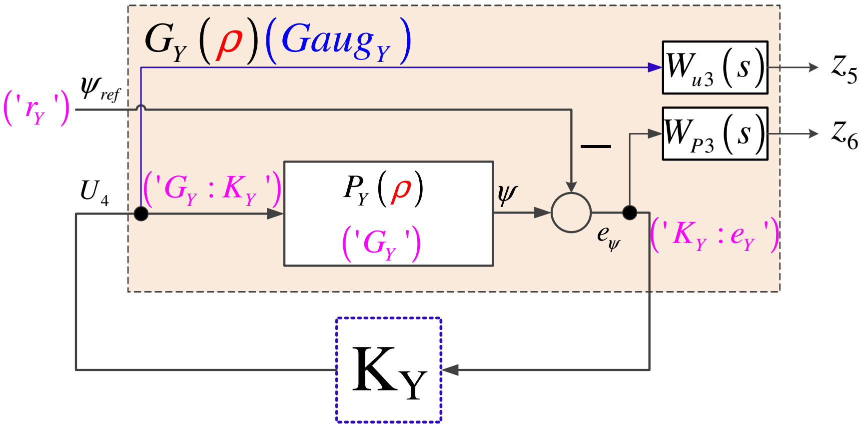


FIGURE 5.5: Yaw H_∞ controller.

The state vector is $X_Y^T = \begin{bmatrix} \psi & \dot{\psi} \end{bmatrix}^T$ and the control input is $u_Y = U_4$.

The roll-pitch susbsystem can be written as:

$$\begin{aligned} E_Y\dot{X}_Y &= A_YX + B_Yu_Y \\ y_Y &= C_YX + D_Yu_Y \end{aligned} \tag{5.36}$$

where:

$$A_Y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \quad B_Y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \quad C_Y = \begin{bmatrix} 1 & 0 \end{bmatrix};$$
$$D_Y = [0]; \qquad E_Y = \begin{bmatrix} 1 & 0 \\ 0 & I_z \end{bmatrix}$$

The parameter ρ_Y that is varying is: $\rho_Y = [I_z]$. Its varying range is in table 5.1.

In order to express the system in polytopic form, matrices A_Y, B_Y, C_Y, D_Y, E_Y can be decomposed as:

$$\begin{aligned} A_Y &= A_{Y_0} + \rho_Y A_{Y_1} \\ B_Y &= B_{Y_0} + \rho_Y B_{Y_1} \\ C_Y &= C_{Y_0} + \rho_Y C_{Y_1} \\ D_Y &= D_{Y_0} + \rho_Y D_{Y_1} \\ E_Y &= E_{Y_0} + \rho_Y E_{Y_1} \end{aligned}$$

where:

$$A_{Y_0} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; A_{Y_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; B_{Y_0} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; B_{Y_1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$C_{Y_0} = \begin{bmatrix} 1 & 0 \end{bmatrix}; \; C_{Y_1} = \begin{bmatrix} 0 & 0 \end{bmatrix}; \; D_{Y_0} = [0]; D_{Y_1}\,[0]$$
$$E_{Y_0} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; \; E_{Y1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Weight functions for Yaw H_∞ controller are chosen as following:

$$\begin{aligned} W_{u3} &= \frac{0.25}{s+0.2} \\ W_{P3} &= \frac{s^3+0.03s^2}{s^3+12000s^2+11300000s+1000} \end{aligned}$$

The norm of LPV H_∞ Yaw subsystem is $\gamma_Y = 0.0188$

Remark 5.2. *The controller is synthesized by using the command hinfgs of the robust control toolbox in Matlab as in Remark 5.1. This controller minimizes the closed-loop quadratic H_∞ performance from reference signal ψ_{ref} to $\begin{bmatrix} z_5 & z_6 \end{bmatrix}^T$ signal.* □

5.4.3 Simulation results and discussions

In this section, the trajectory for the quadcopter to follow is defined as follows: first, the quadcopter goes up $2m$ on altitude, and then follows a square of $8m$ of side on x and y (Fig. 5.13). The yaw angle ψ is not change during the flight.

The mass declines in two manners: gradually and abruptly. A test for the robustness of the proposed controller with respect to step and impulse disturbances is also considered. In particular, we consider wind as a source of disturbances.

The first type of disturbances involves a series of wind impulses with velocity $V_w = 7\cdot\mathbf{i}+7\cdot\mathbf{j}+7\cdot\mathbf{k}$ at $5s$, $15s$, $25s$, $35s$, $45s$, $55s$, $65s$, $75s$, $85s$, respectively. The second type of disturbances comes from two wind steps with velocity $V_w = 1\cdot\mathbf{i}+1\cdot\mathbf{j}+1\cdot\mathbf{k}$ from $15s$ to $25s$, and from $55s$ to $65s$, respectively. Both types of disturbances are demonstrated in the fourth plot of Fig. (5.6) . In this figure, the changes of mass, moments of inertia wrt to the three axis x, y, z are shown in the first, second, and third plot, respectively.

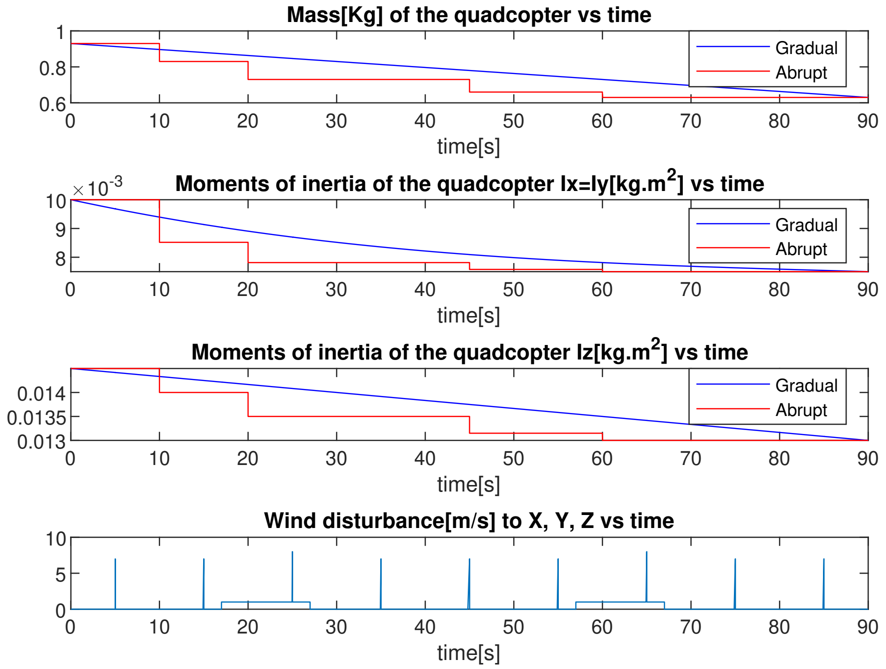


FIGURE 5.6: Variation of mass and moments of inertia and disturbances.

In the gradual mass reduction simulation, the mass reduces gradually from $0.93\,[kg]$ to $0.6\,[kg]$) in a period from $0s$ to $90s$, as shown in the first plot of Fig. (5.6). At each point

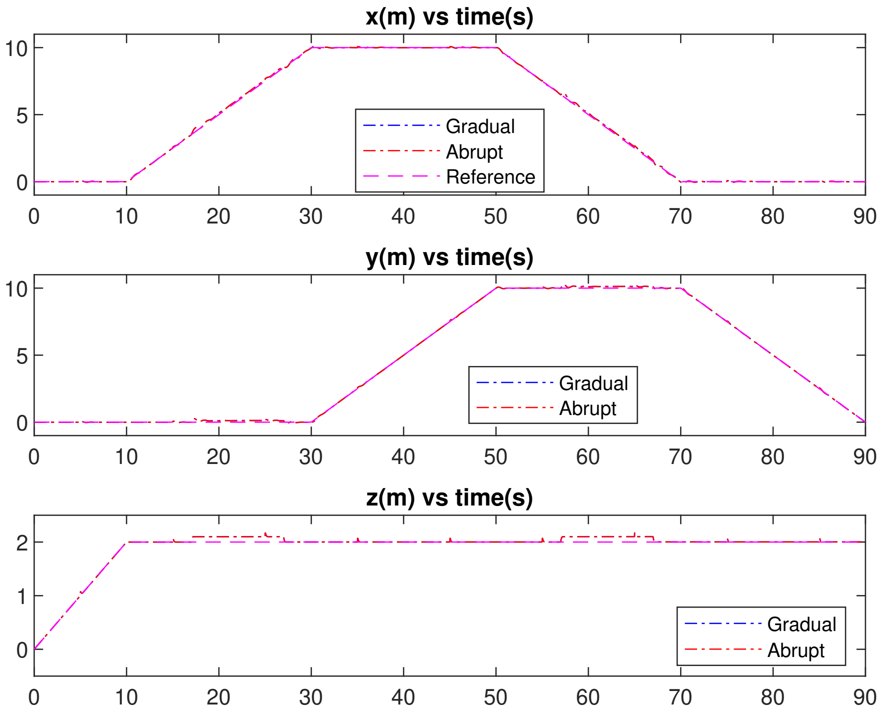


FIGURE 5.7: Translation coordinates: X, Y, Z.

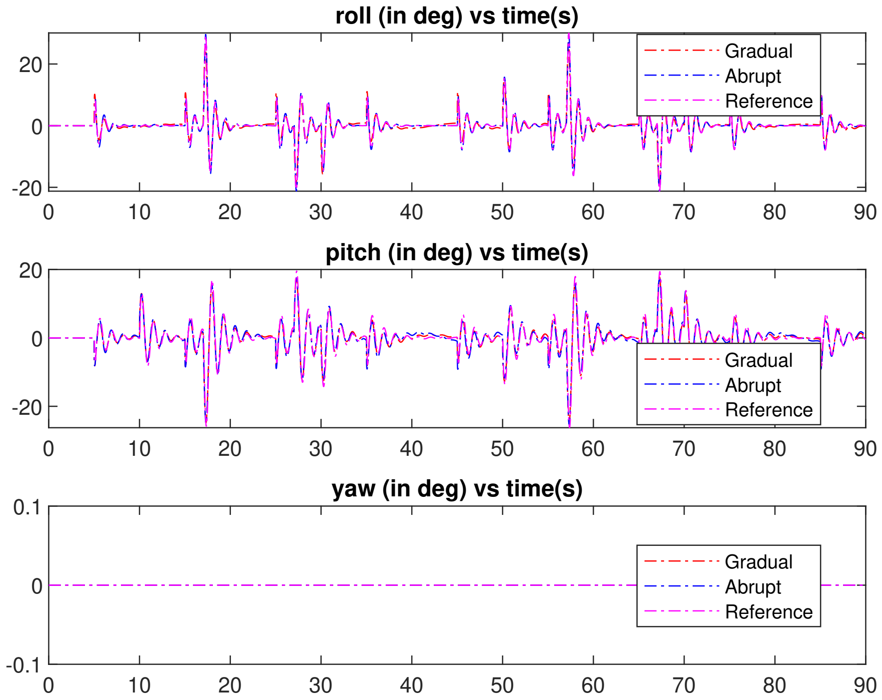


FIGURE 5.8: Orientation coordinates: φ, θ, ψ.

the mass changes, the recalculated moments of inertia are shown in the second and third plots of the same figure.

In the abrupt mass reduction simulation, the mass declines abruptly at $10s$ (from $0.93\,[kg]$ to $0.83\,[kg]$), at $20s$ (from $0.83\,[kg]$ to $0.73\,[kg]$), at $45s$ (from $0.73\,[kg]$ to $0.63\,[kg]$), and at $60s$ (from $0.63\,[kg]$ to $0.6\,[kg]$), as shown in the first plot of Fig. (5.6). At each point the

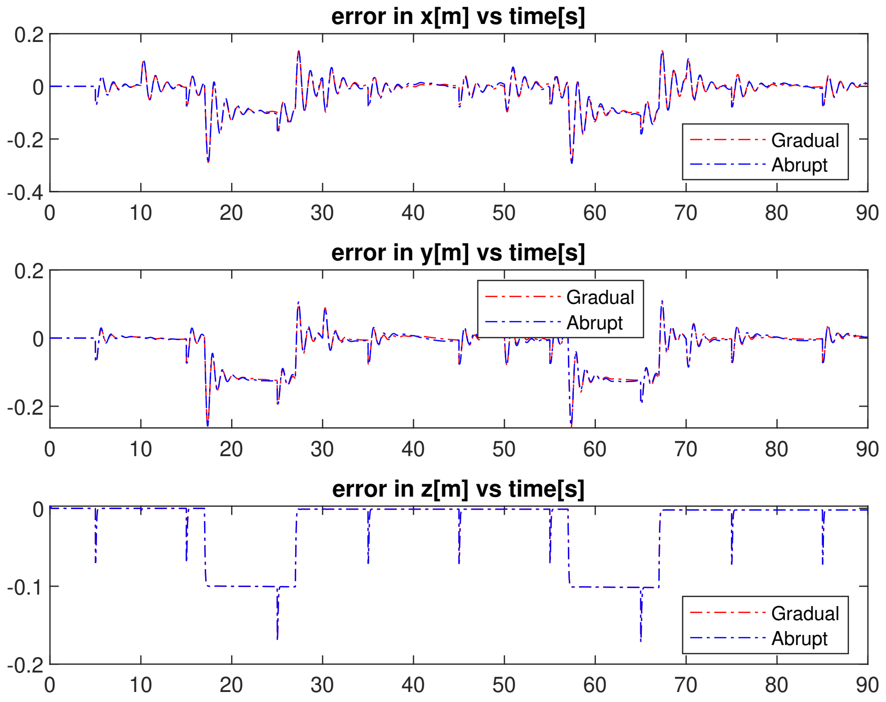


FIGURE 5.9: Error in X, Y, Z, ψ.

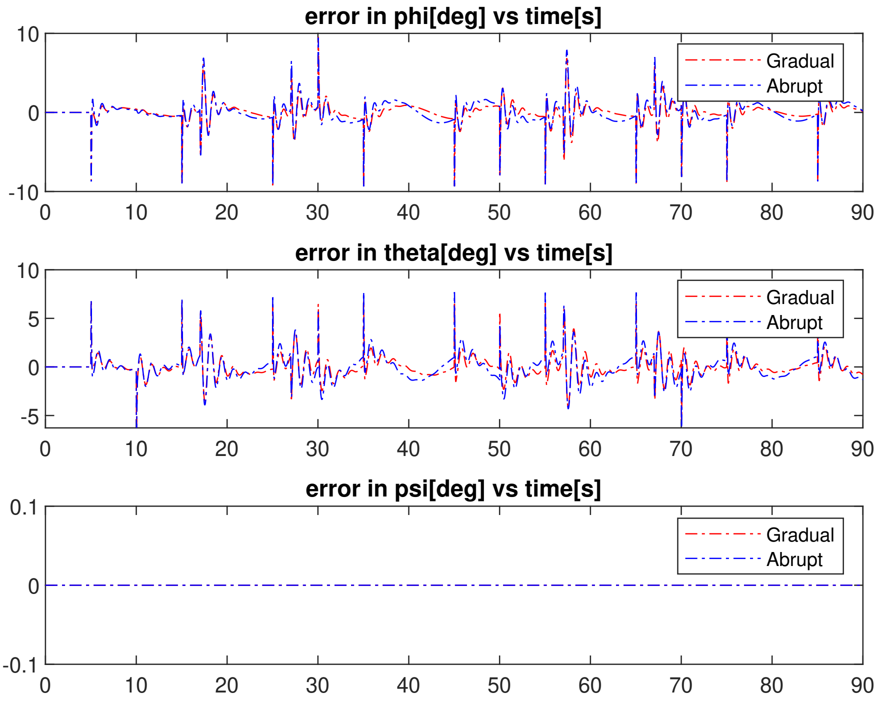


FIGURE 5.10: Error in ϕ, θ, and ψ.

mass changes, the recalculated moments of inertia are shown in the second and third plots of the same figure.

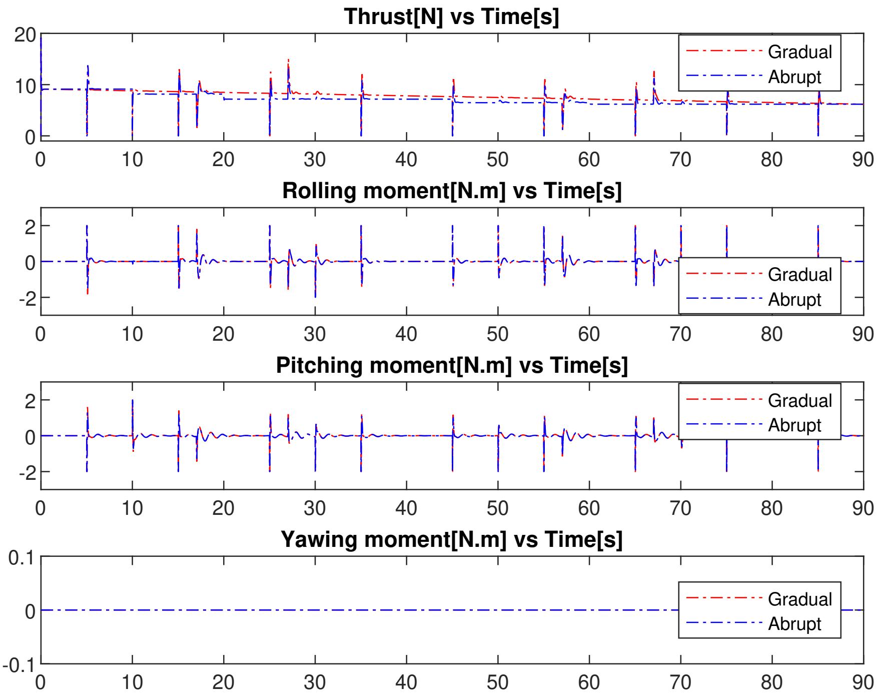


FIGURE 5.11: Input signals: U_1, U_2, U_3, U_4.

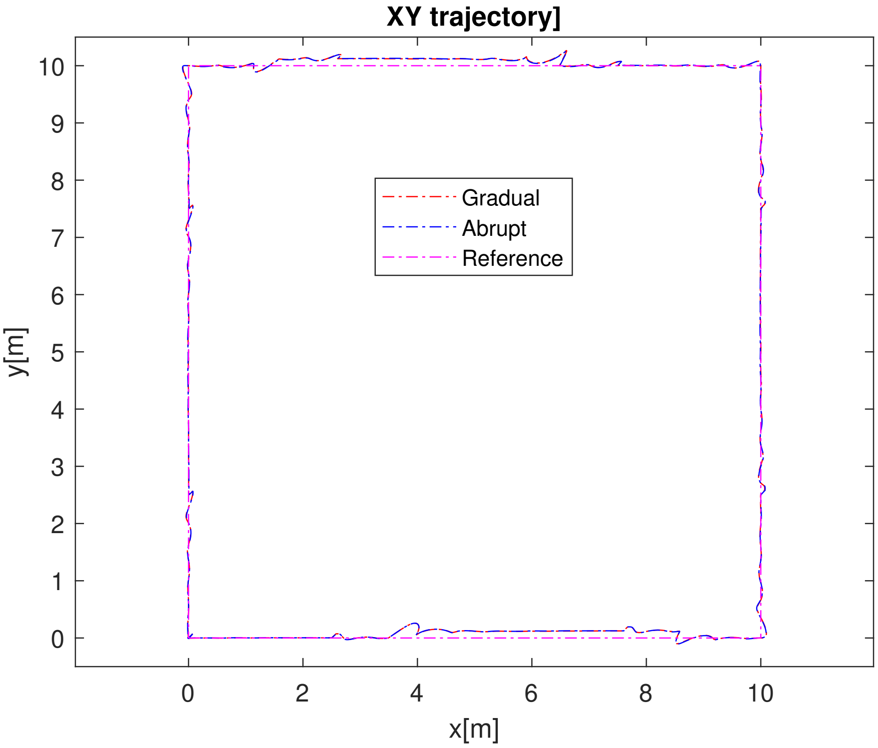


FIGURE 5.12: Horizontal trajectory.

5.4.4 Comments on the simulation results

From the simulation results, we can see that the quadcopter is stable and tracks the predefined trajectory well under the variation of dynamic parameters and the presence of disturbances.

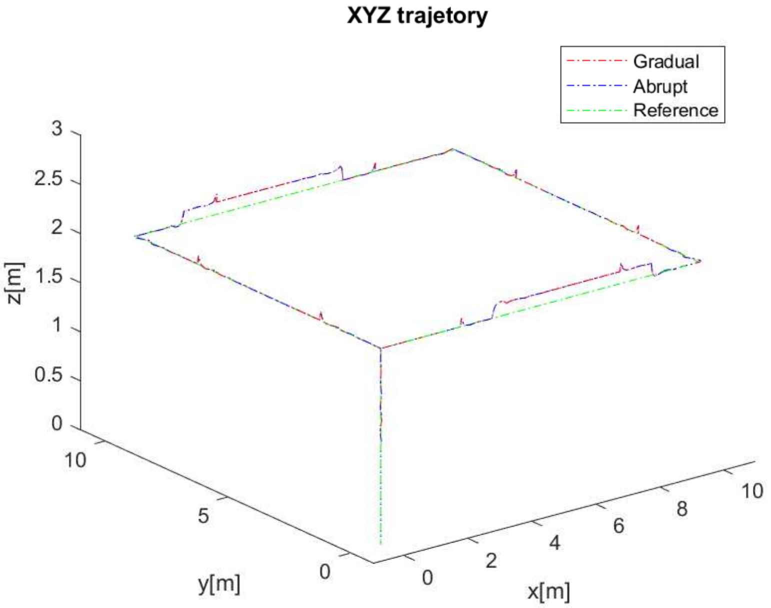


Figure 5.13: 3D trajectory.

Fig. (5.8) shows that the roll, pitch and yaw angles track their designated trajectories. The error of roll, pitch and yaw are shown in Fig. (5.10). The maximum error is about 10 degrees only when the system is disturbed by high impulse disturbances, but it quickly drops to near 0 degree.

Fig. (5.7) shows the references and responses in x, y, and z. Fig. (5.9) shows the errors with respect to x, y, and z and ψ.

In both instances of mass variation, the quadcopter still tracks the predefined trajectory even under step and impulse disturbances, as shown in Fig. (5.12) and Fig. (5.13).

In many cases, the states of the system are not unavaible or too expensive for measuring, the measured outputs are noisy, or some disturbances acting to the system are unable to be measured. Consequently, obsever is used to overcome this difficulties.

5.5 Attitude/Altitude Linear Parameter Varying (LPV) Unknown Input Observer (UIO)

The problems of observing the state and the unknown inputs of a linear/non-linear dynamic system is a challenging problem and have been studied since the 1970's [41][170][64]. An observer can play the aims of a virtual (software) sensor. This virtual sensor is aimed to estimate system parameters that are difficult or impossible to measure such as states or unknown inputs (faults, disturbances,...).

Several extensions for observer have been considered for linear and nonlinear systems after the result proposed by D.G. Luenberger in [140]. The state estimation error is expressed as a system which is free from any unknown input in decoupling approach. The full rank conditions ensure the necessary and sufficient conditions for the existence of the observer

[50][60]. For the second approach, both state vector and unknown input are estimated at the same time under the assumption that unknown inputs are part of the state vector [109][238].

In [223], a nonlinear disturbance observer-based backstepping controller is developed for attitude, altitude, and position control subject to some external disturbances. The stability analysis of the nonlinear disturbance observer is successfully done using the Lyapunov stability theory. In [9], a sliding mode control scheme is proposed for a quadrotor in the presence of an exogenous disturbance. The authors propose a disturbance observer to reject the effect of the unknown disturbance on the quadrotor by using a nonlinear sliding mode surface.

For a quadrotor, it is really important to determine the external forces and moments such as the force of the wind. However, forces and moments are really difficult to measure during the operation. Therefore, an alternative solution is to use an observer. Furthermore, the mass and moments of inertia of UAVs are important impacting parameters to take into account, however, for some reasons, the mass is varying. For example, in applications for spraying pesticides, the mass and moments of inertia of flying equipment will vary slowly over time. Meanwhile, with the job of transporting goods, the mass and moments of inertia of aircraft change abruptly. Mass change implies the changes of moments of inertia. Because of the changes of quadcopter's parameter (mass and moments of inertia), the fixed structure observer might not be working precisely. These issues can be addressed using LPV UIOs [100].

The aim of this section is to design an LPV Unknown Input Observer for the attitude of a mass-varying quadcopter, which can estimate the external torques. The feature of this proposed LPV UIO is that its structure can vary along with the changes of the mass and moments of inertia. To begin with, the design of UIO for the LPV system is considered. The existence conditions of the LPV UIO is studied, and then the gains of the UIO are calculated by resolving LMIs, which ensure the convergence to zero of the state estimation error and the unknown input estimation error. Then, from the dynamic of the attitude of the mass-varying quadcopter, the LPV altitude controller is generated. Finally, an LPV UIO is designed for the altitude system of the mass-varying quadcopter.

5.5.1 Problem formulation

This sub-section aims to design an LPV UIO for a LPV system which structure is expressed by

$$\left\{ \begin{array}{l} \dot{x}(t) = A(\rho(t))\, x(t) + B(\rho(t))\, u(t) + E(\rho(t))\, d(t) \\ y(t) = Cx(t) \end{array} \right. \tag{5.37}$$

where $x(t) \in R^n$, $u(t) \in R^{n_u}$, $d(t) \in R^{n_d}$, and $y(t) \in R^y$ are the state vector, the control input, the unknown input, and the output of the system, respectively. The matrices $A(\cdot)$, $B(\cdot)$, and $D(\cdot)$ are varying parameters with appropriate dimensions, and C is a matrix of constants. $\rho^T(t) = \left(\rho_1, ..., \rho_{n_\rho}\right)$ represents the vector of n_ρ time-varying parameters which are sufficiently smooth and bounded which means that $\rho(t)$ is an element of Θ, a hyper-rectangle defined by

$$\Theta = \left\{ \rho(t) \in R^{n_\rho} \middle| \rho_1 \in \left[\underline{\rho}_1, \overline{\rho}_1\right], ..., \rho_{n_p} \in \left[\underline{\rho}_{n_p}, \overline{\rho}_{n_p}\right] \right\} \tag{5.38}$$

where $\underline{\rho}_i$ and $\overline{\rho}_i$, $i = 1, ..., n_\rho$ define the lower and upper bounds of the varying parameter $\rho_i(t)$.

For sake of simplicity, in what follows we put $\Delta(\rho(t))$ as Δ_ρ where Δ is a matrix depending on the varying parameter $\rho(t)$. In addition, system (5.37) can be free of input by defining an auxiliary system in the form

$$\left\{\begin{array}{rcl} \dot{s}(t) & = & A_\rho s(t) + B_\rho u(t) \\ y_s(t) & = & Cs(t) \end{array}\right. \tag{5.39}$$

Define the errors $z(t) = x(t) - s(t)$ and $y_z(t) = y(t) - y_s(t)$, the new dynamics

$$\left\{\begin{array}{rcl} \dot{z}(t) & = & A_\rho z(t) + E_\rho d(t) \\ y_z(t) & = & Cz(t) \end{array}\right. \tag{5.40}$$

can be obtained from the systems in (5.37) and (5.39), which are free from the known input $u(t)$. Thus, after estimating the state $\hat{z}(t)$, the real state $\hat{x}(t)$ is obtained by the equation

$$\hat{x}(t) = \hat{z}(t) + s(t) \tag{5.41}$$

Consequently, the problem of designing the LPV UIO for LPV system in (5.37) is transformed to the problem of designing the LPV UIO for the LPV system without the known input depicted in (5.40).

Definition 5.1. *The system (5.37) is said to be uniformly strongly algebraically observable [100] with respect to the parameter $\rho(t)$ if there exists positive integers k_y , k_u and k_ρ such that $\rho^{(j)}(t) \in \Theta_j, j = 0, ..., k_\rho$ the state of the system (5.37) can be expressed as a vector function of the outputs, the inputs, the parameters and their time derivatives up to a finite orders*

$$x(t) = F\left(y(t), ..., y^{(k_y)}, u(t), ..., u^{(k_u)}, \rho(t), ..., \rho^{(k_\rho)}\right) \tag{5.42}$$

□

Assume also that the time derivatives of the parameters belong to the compact sets defined by

$$\Theta_j = \left\{\rho^{(j)} \in R^{n_\rho} \middle| \rho_1^{(j)} \in \left[\underline{\rho}_{1j}, \overline{\rho}_{1j}\right], ..., \rho_{n_p}^{(j)} \in \left[\underline{\rho}_{n_pj}, \overline{\rho}_{n_pj}\right]\right\} \tag{5.43}$$

where $\underline{\rho}_{ij}$ and $\overline{\rho}_{ij}$, $i = 1, ..., n_\rho$ define the lower and upper bounds of the jth derivative of the varying parameter $\rho_i(t)$.

Definition 5.2. *Given the system (5.37), the number r is called the relative degree [11] of the output $y(t)$ with respect to the unknown input $d(t)$ if the unknown input $d(t)$ appears in the equation of the r^{th} time derivative of the output $\left(y^{(r)}(t)\right)$ and not in $y^{(k)}$ for $k < r$..* □

5.5.2 LPV UIO design for LPV system

Let us consider the LPV system in (5.40) where $y_z(t) \in R^{n_y}$ and $d(t) \in R^{n_d}$. Suppose that, each output $y_{z_i}(t)$ has a relative degree r_i where $i = 1, ..., n_y$ with respect to the unknown inputs. Thus, the vector of relative orders is given by $\left\{r_1, r_2, ..., r_{n_y}\right\}$.

The matrices $E_\rho(\rho(t))$ and C can be rewritten as $E_\rho(\rho(t)) = \left[\begin{array}{cccc} E_\rho^1 & E_\rho^2 & ... & E_\rho^{n_d} \end{array}\right]$, $C = \left[\begin{array}{ccc} C_1 & \dots & C_{n_y} \end{array}\right]^T$

By differentiating r_i times the i^{th} output $y_{z_i}(t)$, ones gets

$$y_{z_i}(t) = C_i z(t) \tag{5.44a}$$

$$\dot{y}_{z_i}(t) = \underbrace{C_i A_\rho}_{M^i_{1\rho}} z(t) \tag{5.44b}$$

$$\ddot{y}_{z_i}(t) = \underbrace{\left(M^i_{1\rho} A_\rho + \dot{M}^i_{1\rho}\right)}_{M^i_{2\rho}} z(t) \tag{5.44c}$$

$$y^{(3)}_{z_i}(t) = \underbrace{\left(M^i_{2\rho} A_\rho + \dot{M}^i_{2\rho}\right)}_{M^i_{3\rho}} z(t) \tag{5.44d}$$

$$\vdots$$

$$y^{(r_i)}_{z_i}(t) = \underbrace{\left(M^i_{(r_i-1)\rho} A_\rho + \dot{M}^i_{(r_i-1)\rho}\right)}_{M^i_{r_i\rho}} z(t) + M^i_{(r_i-1)\rho} E^i_\rho d_i(t) \tag{5.44e}$$

From (5.44a) to (5.44e), the output time derivatives are obtained in matrix form as follows

$$\mathcal{Y}(t) = \mathcal{M}_\rho z(t) + \Gamma_\rho d(t) \tag{5.45}$$

where

$$\mathcal{Y}(t) = \begin{bmatrix} y^{(r_1)}_1(t) \\ y^{(r_2)}_2(t) \\ \vdots \\ y^{\left(r_{n_y}\right)}_{n_y}(t) \end{bmatrix}, \mathcal{M}_\rho(t) = \begin{bmatrix} M^1_{r_1\rho} \\ M^2_{r_2\rho} \\ \vdots \\ M^{n_y}_{r_{n_y}\rho} \end{bmatrix}$$

and

$$\Gamma_\rho = \begin{bmatrix} M^1_{(r_1-1)} E^1_\rho & M^1_{(r_1-1)} E^2_\rho & \cdots & M^1_{(r_1-1)} E^{n_d}_\rho \\ M^2_{(r_2-1)} E^1_\rho & M^2_{(r_2-1)} E^2_\rho & \cdots & M^2_{(r_2-1)} E^{n_d}_\rho \\ \vdots & \vdots & \ddots & \vdots \\ M^{n_y}_{\left(r_{n_y}-1\right)} E^1_\rho & M^{n_y}_{\left(r_{n_y}-1\right)} E^2_\rho & \cdots & M^{n_y}_{\left(r_{n_y}-1\right)} E^{n_d}_\rho \end{bmatrix}$$

Consequently, the observer for (5.40) is proposed in the form

$$\left\{ \begin{array}{lcl} \dot{\hat{z}}(t) & = & (A_\rho - Q_\rho \mathcal{M}_\rho - L_\rho C)\hat{z}(t) + Q_\rho \mathcal{Y}(t) + L_\rho y_z(t) \\ \hat{d}(t) & = & \Gamma^{-1}_\rho (\mathcal{Y}(t) - \mathcal{M}_\rho \hat{z}(t)) \end{array} \right. \tag{5.46}$$

where Q_ρ and L_ρ are to be determined.

Theorem 5.1. *The system (5.46) is an observer for the system (5.40) if the following conditions are satisfied:*

- *The matrix* Γ *is full column rank* $\forall \rho \in \Theta$
- *The pair* $(A_\rho - Q_\rho \mathcal{M}_\rho, C)$ *is detectable* $\forall \rho \in \Theta$*, where* $Q_\rho = E\Gamma^{-1}$ *if* $n_y = n_d$*, and* $Q_\rho = E\Gamma^\dagger$ *if* $n_y > n_d$*,* $\Gamma^\dagger$ *is the pseudo inverse of* Γ
- *The parameter varying matrix* $(A_\rho - Q_\rho \mathcal{M}_\rho - L_\rho C)$ *is stable* $\forall \rho \in \Theta$

Proof. The state estimation error is defined as $e(t) = z(t) - \hat{z}(t)$ and the unknown input estimation error $e_d(t) = d(t) - \hat{d}(t)$. The derivative of the error is

$$\begin{aligned}\dot{e}_z(t) &= \dot{z}(t) - \dot{\hat{z}}(t)\\ &= A_\rho z(t) + E_\rho d(t) - (A_\rho - Q_\rho \mathcal{M}_\rho - L_\rho C)\hat{z}(t) - Q_\rho \mathcal{Y}(t) - L_\rho y_z(t)\\ &= (A_\rho - Q_\rho \mathcal{M}_\rho - L_\rho C)e(t) + (E_\rho - Q_\rho \Gamma_\rho)d(t)\end{aligned} \tag{5.47}$$

Under the condition that the matrix Γ is full column rank, $\Gamma^\dagger$ (Γ^{-1} if Γ is square) exists and ensures that $E_\rho - Q_\rho \Gamma_\rho = 0$, the state estimation dynamics becomes

$$\dot{e}_z(t) = (A_\rho - L_\rho C - Q_\rho \mathcal{M}_\rho)e(t) \tag{5.48}$$

From the condition that the pair $(A_\rho - Q_\rho \mathcal{M}_\rho, C)$ is detectable $\forall \rho \in \Theta$, the gain matrix L_ρ can be computed in order to ensure the asymptotic stability of the system (5.48) which ensures that state of the observer converges asymptotically to the state of the system. The unknown input estimation can be computed as follows

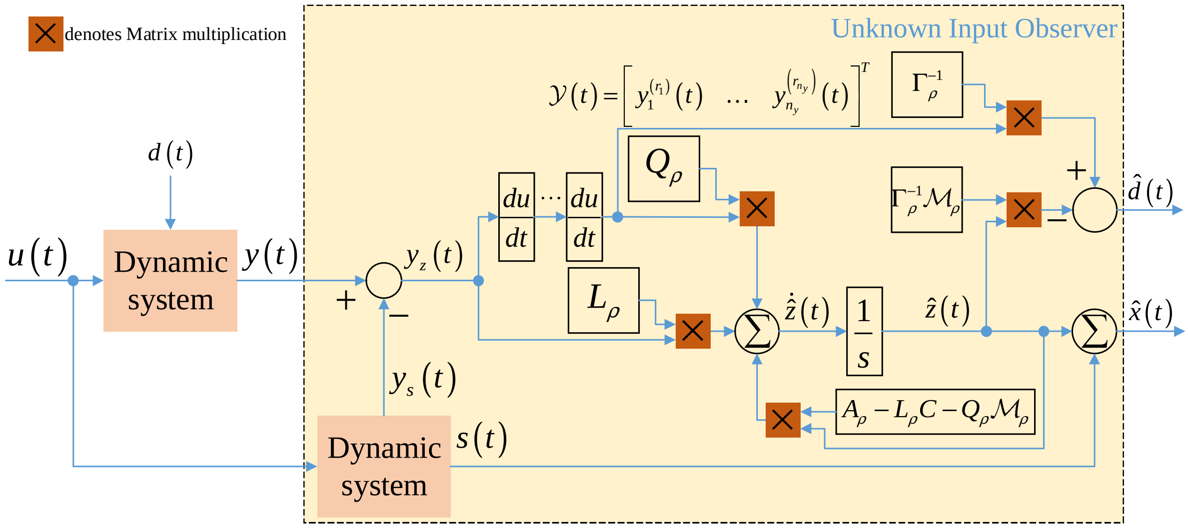


FIGURE 5.14: Unknown Input Observer for dynamic system

$$e_d(t) = d(t) - \hat{d}(t) = -\Gamma_\rho^{-1}\mathcal{M}_\rho e_z(t) \tag{5.49}$$

which ensures that e_d converges towards 0 when $e_z(t) \to 0$. The structure of UIO in (5.46) is depicted in figure 5.14. ■

5.5.3 Convergence analysis and LMI formulation

From (5.48) and (5.49), we have

$$\left\{ \begin{array}{rcl} \dot{e}_z(t) & = & (A_\rho - Q_\rho \mathcal{M}_\rho - L_\rho C)e_z(t) \\ e_d(t) & = & -\Gamma_\rho^{-1}\mathcal{M}_\rho e_z(t) \end{array} \right. \tag{5.50}$$

In order to derive LMI conditions for design purpose, we next transform the matrices A_ρ, Q_ρ, $\mathcal{M}_\subset$, and Γ_ρ in a polytopic form where the parameters $\rho(t) \in \Theta$, one obtains

$$\left\{ \begin{array}{ccl} A_\rho & = & \sum\limits_{i=1}^{2^{n_\rho}} \mu_i(\rho(t)) A_i \\ Q_\rho & = & \sum\limits_{i=1}^{2^{n_\rho}} \mu_i(\rho(t)) Q_i \\ \mathcal{M}_\rho & = & \sum\limits_{i=1}^{2^{n_\rho}} \mu_i(\rho(t)) \mathcal{M}_i \\ \Gamma_\rho & = & \sum\limits_{i=1}^{2^{n_\rho}} \mu_i(\rho(t)) \Gamma_i \end{array} \right. \tag{5.51}$$

$\forall \rho(t) \in \Theta$ and $\mu_i(\rho(t))$ satisfy the convex sum property

$$\sum_{i=1}^{2^{n_\rho}} \mu_i(\rho(t)) = 1, \;\; 0 \leq \mu_i(\rho(t)) \leq 1, \;\; i = 1, ...2^{n_\rho}, \;\; \forall \rho(t) \in \Theta \tag{5.52}$$

Thus, the gain matrix L_ρ can be determined as

$$L_\rho = \sum_{i=1}^{2^{n_\rho}} \mu_i(\rho(t)) L_i \tag{5.53}$$

Consequently, the state estimation error dynamics (5.48) can be expressed by

$$\dot{e}_z(t) = \sum_{i=1}^{2^{n_\rho}} \sum_{j=1}^{2^{n_\rho}} \mu_i(\rho(t)) \mu_j(\rho(t)) (A_j - Q_i \mathcal{M}_j - L_j C) e_z(t) \tag{5.54}$$

Standard LMI for stability can be obtained using a quadratic Lyapunov function in the form

$$V(e_z(t)) = e_z^T(t) X e_z(t), X = X^T > 0 \tag{5.55}$$

The time derivative of the Lyapunov function is

$$\dot{V}(e_z(t)) = \dot{e}_z^T(t) X e_z(t) + e_z^T(t) X \dot{e}_z(t) \tag{5.56}$$

Finally, the derivative of the Lyapunov function $V(e_z(t))$ in (5.56) can be rewritten as

$$\dot{V}(e_z(t)) = e_z^T(t) [A_\rho X + X A_\rho - (Q_\rho M_\rho)^T X + X Q_\rho M_\rho - C^T K_\rho^T - K_\rho C] e_z(t) \tag{5.57}$$

where $K_\rho = L_\rho C$.

After computing the time derivative of the Lyapunov function $V(e_z(t))$ and by using the state estimation error dynamics (5.54) and the convex sum property of the weighting functions in (5.52), sufficient LMI conditions ensuring asymptotic stability are obtained as follows

$$A_j^T X + X A_j - (Q_i M_j)^T X + X Q_i M_j - C^T K_j^T - K_j C < 0, \quad i, j = 1...2^{n_p} \tag{5.58}$$

where the gains of the observer are obtained from the equation $L_i = X^{-1} K_i$. This solution ensures the convergence to zero of the state estimation error $e_z(t)$, and the unknown input estimation error $e_d(t)$.

5.5.4 LPV UIO design

The proposed LPV Unknown input observer is based on Linear Matrix Inequality (LMI) methods. The resolvability of the resulting LMI conditions in this case is quite compromised due to the number of submodels. The more number of submodel is, the more is conservativeness of LMI conditions. In order to reduce this number, we adopt here a simpler model. We assume that the quadcopter is symmetric and $I_x = I_y$. We also assume that to the altittude sub-model of the quadcopter is disturbed the torques $d(t) = \begin{bmatrix} d_\varphi & d_\theta & d_\psi \end{bmatrix}^T$ allowing to write:

$$\begin{cases} \ddot{\varphi} & = & \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r\Omega_r}{I_x}\dot{\theta} + \frac{l}{I_x}U_2 + \frac{1}{I_x}d_\varphi \\ \ddot{\theta} & = & \frac{I_z - I_x}{I_x}\dot{\varphi}\dot{\psi} + \frac{J_r\Omega_r}{I_x}\dot{\varphi} + \frac{l}{I_x}U_3 + \frac{1}{I_x}d_\theta \\ \ddot{\psi} & = & \frac{1}{I_z}U_4 + \frac{1}{I_z}d_\psi \end{cases} \tag{5.59}$$

We can see that, the dynamic equation of the yaw angle does not contain Euler angles or their derivatives. Thus, the dynamic of quadcopter attitude can be decomposed into two subsystems, the roll-pitch subsystem is the first two equations and the yaw subsystem is the third equation of (5.59).

5.5.4.1 LPV UIO for Roll-Pitch

The system differential equations for φ and θ are rewritten from (5.59) as

$$\begin{cases} \dot{\varphi} & = & \dot{\varphi} \\ \dot{\theta} & = & \dot{\theta} \\ \ddot{\varphi} & = & \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r\Omega_r}{I_x}\dot{\theta} + \frac{l}{I_x}U_2 + \frac{1}{I_x}d_\varphi \\ \ddot{\theta} & = & \frac{I_z - I_x}{I_x}\dot{\varphi}\dot{\psi} + \frac{J_r\Omega_r}{I_x}\dot{\varphi} + \frac{l}{I_x}U_3 + \frac{1}{I_x}d_\theta \end{cases} \tag{5.60}$$

LPV system for roll pitch can be obtained from system differential equation (5.60) as

$$\begin{cases} \dot{x}_1(t) & = & A_{1_{\rho_1}} x_1(t) + B_{1_{\rho_1}} u_1(t) + E_{1_{\rho_1}} d_1(t) \\ y_1(t) & = & C_1 x_1(t) \end{cases} \tag{5.61}$$

where varying parameters, state, output, known input, and unknown input are respectively $\rho_1(t) = \begin{bmatrix} \rho_{11} & \rho_{12} \end{bmatrix}^T = \begin{bmatrix} \frac{1}{I_x} & \frac{I_y - I_z}{I_x}\dot{\psi} - \frac{J_r\Omega_r}{I_x} \end{bmatrix}^T$, $x_1(t) = \begin{bmatrix} \varphi & \theta & \dot{\varphi} & \dot{\theta} \end{bmatrix}^T$;$y_1(t) = \begin{bmatrix} \varphi & \theta & \dot{\varphi} & \dot{\theta} \end{bmatrix}^T$, $u_1(t) = \begin{bmatrix} U_2 & U_3 \end{bmatrix}$;$d_1(t) = \begin{bmatrix} d_\varphi & d_\theta \end{bmatrix}^T$ and the system matrices are:

$$\begin{gathered} A_{1_{\rho_1}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \rho_{12} \\ 0 & 0 & -\rho_{12} & 0 \end{bmatrix}; B_{1_{\rho_1}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ l\rho_{11} & 0 \\ 0 & l\rho_{11} \end{bmatrix} \\ C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; E_{1_{\rho_1}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \rho_{11} & 0 \\ 0 & \rho_{11} \end{bmatrix} \end{gathered} \tag{5.62}$$

Define an auxiliary system in the form

$$\begin{cases} \dot{s}_1(t) & = & A_{1_{\rho_1}} s_1(t) + B_{1_{\rho_1}} u_1(t) \\ y_{s_1}(t) & = & C_1 s_1(t) \end{cases} \tag{5.63}$$

Define the errors $z_1(t) = x_1(t) - s_1(t)$ and $y_{z_1}(t) = y_1(t) - y_{s_1}(t)$, the new dynamics

$$\left\{ \begin{array}{rcl} \dot{z}_1(t) & = & A_{1_{\rho_1}} z_1(t) + E_{1_{\rho_1}} d_1(t) \\ y_{z_1}(t) & = & C_1 z_1(t) \end{array} \right. \tag{5.64}$$

can be obtained from the systems in (5.61) and (5.63), which are free from the known input $u_1(t)$. Thus, after estimating the state $\hat{z}_1(t)$, the real state $\hat{x}_1(t)$ is obtained by the equation $\hat{x}_1(t) = \hat{z}_1(t) + s_1(t)$.

Take the derivative of the output $y_{z_1} = \begin{bmatrix} y_{z_{11}} & y_{z_{12}} & y_{z_{13}} & y_{z_{14}} \end{bmatrix}^T$ as in (5.44a) to (5.44e), one obtains:

$$\left\{ \begin{array}{rcl} y_{z_{11}}^{(2)} & = & \rho_{12}\tau_4 + \rho_{11} d_\varphi \\ y_{z_{12}}^{(2)} & = & -\rho_{12}\tau_3 + \rho_{11} d_\theta \\ y_{z_{13}}^{(1)} & = & \rho_{12}\tau_4 + \rho_{11} d_\varphi \\ y_{z_{14}}^{(1)} & = & -\rho_{12}\tau_3 + \rho_{11} d_\theta \end{array} \right. \tag{5.65}$$

Consequently, the relative degrees r_i of the i^{th} output, $i = 1,..,4$ respectively are $r_{11} = 2$, $r_{12} = 2$, $r_{13} = 1$, and $r_{14} = 1$.

Following the defined vector $\mathcal{Y}_1(t) = \begin{bmatrix} y_{z_{11}}^{(2)} & y_{z_{12}}^{(2)} & y_{z_{13}}^{(1)} & y_{z_{14}}^{(1)} \end{bmatrix}^T = \mathcal{M}_{1_{\rho_1}} z_1(t) + \Gamma_{1_{\rho_1}} d_1(t)$ in (5.45), one obtains

$$\Gamma_{1_{\rho_1}} = \begin{bmatrix} \rho_{11} & 0 \\ 0 & \rho_{11} \\ \rho_{11} & 0 \\ 0 & \rho_{11} \end{bmatrix} \tag{5.66}$$

which satisfies the full column rank condition and its pseudo-inverse is given by

$$\Gamma_{1_{\rho_1}}^{\dagger} = \begin{bmatrix} \frac{1}{2\rho_{11}} & 0 & \frac{1}{2\rho_{11}} & 0 \\ 0 & \frac{1}{2\rho_{11}} & 0 & \frac{1}{2\rho_{11}} \end{bmatrix} \tag{5.67}$$

The unknown input decoupling matrix $Q_{1_{\rho_1}}$ is then given by

$$Q_{1_{\rho_1}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \tag{5.68}$$

We can verify that the pair $\left(A_{1_{\rho_1}} - Q_{1_{\rho_1}} \mathcal{M}_{1_{\rho_1}}, C_1\right)$ is detectable, where the matrix $\mathcal{M}_{1_{\rho_1}}$ and $A_{1_{\rho_1}} - Q_{1_{\rho_1}} \mathcal{M}_{1_{\rho_1}}$ are respectively

$$\begin{array}{l} \mathcal{M}_{1_{\rho_1}} = \begin{bmatrix} 0 & 0 & 0 & \rho_{12} \\ 0 & 0 & -\rho_{12} & 0 \\ 0 & 0 & 0 & \rho_{12} \\ 0 & 0 & -\rho_{12} & 0 \end{bmatrix} \\ A_{1_{\rho_1}} - Q_{1_\rho} \mathcal{M}_{1_{\rho_1}} = \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix} \end{array} \tag{5.69}$$

Finally, the gain matrix L_{1_ρ} of the UIO can be calculated such that the parameter varying matrix $A_{1_{\rho_1}} - Q_{1_{\rho_1}} \mathcal{M}_{1_{\rho_1}} - L_{1_{\rho_1}} C_2$ is stable $\forall \rho_{11}, \rho_{12} \in \Theta_{\rho_1}$ by solving LMIs as in (5.58), where Θ_{ρ_1} is the hyper-rectangle for ρ_{11} and ρ_{12} which is defined in (5.38).

5.5.4.2 Linear Parameter Varying (LPV) Unknown Input Observer (UIO) for Yaw

The system differential equations for ψ are rewritten from (5.59) as

$$\left\{\begin{array}{lcl}\dot{\psi} & = & \dot{\psi}\\ \ddot{\psi} & = & \frac{1}{I_z}U_4+\frac{1}{I_z}d_\psi\end{array}\right. \tag{5.70}$$

LPV system for yaw can be obtained from system differential equation (5.70) as

$$\left\{\begin{array}{lcl}\dot{x}_2(t) & = & A_{2_{\rho_2}}x_2(t)+B_{2_{\rho_2}}u_2(t)+E_{2_{\rho_2}}d_2(t)\\ y_2(t) & = & C_2x_2(t)\end{array}\right. \tag{5.71}$$

where varying parameters, state, output, known input, and unknown input are respectively $\rho_2(t)=[\rho_{21}]=\left[\frac{1}{I_z}\right];x_2(t)=\left[\begin{array}{cc}\psi & \dot{\psi}\end{array}\right]^T;y_2(t)=\left[\begin{array}{cc}\psi & \dot{\psi}\end{array}\right]^T;\ u_2(t)=[U_4];d_2(t)=[d_\psi]$ and the system matrices are

$$A_{2_{\rho_2}}=\left[\begin{array}{cc}0 & 1\\ 0 & 0\end{array}\right];B_{2_{\rho_2}}=E_{2_{\rho_2}}=\left[\begin{array}{c}0\\ \rho_{21}\end{array}\right];C_2=\left[\begin{array}{cc}1 & 0\\ 0 & 1\end{array}\right] \tag{5.72}$$

Define an auxiliary system in the form

$$\left\{\begin{array}{lcl}\dot{s}_2(t) & = & A_{2_{\rho_2}}s_2(t)+B_{2_{\rho_2}}u_2(t)\\ y_{s_2}(t) & = & C_2s_2(t)\end{array}\right. \tag{5.73}$$

Define the errors $z_2(t)=x_2(t)-s_2(t)$ and $y_{z_2}(t)=y_2(t)-y_{s_2}(t)$, the new dynamics

$$\left\{\begin{array}{lcl}\dot{z}_2(t) & = & A_{2_{\rho_2}}z_2(t)+E_{2_{\rho_2}}d(t)\\ y_{z_2}(t) & = & C_2z_2(t)\end{array}\right. \tag{5.74}$$

can be obtained from the systems in (5.71) and (5.73), which are free from the known input $u_2(t)$. Thus, after estimating the state $\hat{z}_2(t)$, the real state $\hat{x}_2(t)$ is obtained by the equation $\hat{x}_2(t)=\hat{z}_2(t)+s_2(t)$.

Take the derivative of the output $y_{z_2}=\left[\begin{array}{cc}y_{z_{21}} & y_{z_{22}}\end{array}\right]^T$ as in (5.44a) to (5.44e), one obtains:

$$\left\{\begin{array}{lcl}y_{z_{21}}^{(2)} & = & \rho_{21}d_\psi\\ y_{z_{21}}^{(1)} & = & \rho_{21}d_\psi\end{array}\right. \tag{5.75}$$

Consequently, the relative degrees r_i of the $i^{th}, i=1,2$ output respectively are $r_{21}=2$, $r_{22}=1$.

Following the defined vector $\mathcal{Y}_2=\left[\begin{array}{cc}y_{z_{21}}^{(2)} & y_{z_{22}}^{(1)}\end{array}\right]^T$ in (5.45), one obtains

$$\Gamma_{2_{\rho_2}}=\left[\begin{array}{c}\rho_{21}\\ \rho_{21}\end{array}\right] \tag{5.76}$$

which satisfies the full column rank condition and the pseudo-inverse exists and given by

$$\Gamma_{2_{\rho_1}}^{\dagger}=\left[\begin{array}{cc}\frac{1}{2\rho_{21}} & \frac{1}{2\rho_{21}}\end{array}\right] \tag{5.77}$$

The unknown input decoupling matrix Q_{1_ρ} is then given by

$$Q_{2_{\rho_2}} = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{5.78}$$

We can verify that the pair $\left(A_{2_{\rho_2}} - Q_{2_{\rho_2}} \mathcal{M}_{2_{\rho_2}}, C_2\right)$ is detectable, where the matrix $\mathcal{M}_{2_{\rho_2}}$ and $A_{2_{\rho_2}} - Q_{2_{\rho_2}} \mathcal{M}_{2_{\rho_2}}$ are respectively

$$\mathcal{M}_{2_{\rho_2}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; A_{2_{\rho_2}} - Q_{2_{\rho_2}} \mathcal{M}_{2_{\rho_2}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{5.79}$$

Finally, the gain matrix $L_{2_{\rho_2}}$ of the UIO can be calculated such that the parameter varying matrix $A_{2_{\rho_2}} - Q_{2_{\rho_2}} \mathcal{M}_{2_{\rho_2}} - L_{2_{\rho_2}} C_2$ is stable $\forall \rho_{21} \in \Theta_{\rho_2}$ by solving LMIs as in (5.58), where Θ_{ρ_2} is the hyper rectangle for ρ_{21} which is defined in (5.38).

Remark 5.3. *As we can see the system matrices in (5.62) and (5.72), the state repeats exactly the output of the system. It means that we use the information of the measured outputs to estimate itself. This observer helps to recover the real state of the system efficiently in case of disturbed outputs. Furthermore, this observer allows us to estimate the unknown input that affects the system. The information about unknown inputs is really useful for improving the quality of quadcopter control.*

Remark 5.4. *Suppose that each actuator thrust Laplace transform is given by*

$$\mathcal{T}_i(s) = \frac{K_i}{1+\tau_i s} V_i(s), \quad i = 1,2,3,4 \tag{5.80}$$

where $\mathcal{T}_i$ *is the Laplace transform of the thrust* $T_i(t)$, V_i *is the pulse width modulation (PWM) voltage,* K_i *is the armature gain, and* τ_i *is the time constant of the* $i^{th}, i = 1,..,4$ *rotor. The corresponding differential equation is*

$$\dot{T}_i = -\frac{1}{\tau_i} T_i + \frac{K_i}{\tau_i} v_i \tag{5.81}$$

We also known that, the thrust $T_i(t)$ *is a function of the rotor speed*

$$T_i(t) = k_f \omega_i^2 \tag{5.82}$$

where k_f *is the constant coefficient.*

Consequently, based on the PWM applied to each rotor, the rotor speed ω_i *can be estimated. Thus, the residual speed* $\Omega_r = \omega_2 + \omega_4 - \omega_1 - \omega_3$ *can also be estimated.*

Remark 5.5. *Due to Remark 5.3 and Remark 5.4, one can see that all the varying parameters for LPV UI observers designed in 5.5.4.1 and 5.5.4.2 can be measured.*

5.5.4.3 LPV UIO for Altitude

The system differential equations for altitude z are rewritten from (3.54c) as

$$\left\{ \begin{array}{lcl} \dot{z} & = & \dot{z} \\ \ddot{z} & = & \frac{\cos\varphi \cos\theta}{m} U_1 - g + \frac{1}{m} d_z \end{array} \right. \tag{5.83}$$

LPV system for altitude can be obtained from system differential equation (5.83) as

$$\left\{\begin{array}{lcl} \dot{x}_3(t) & = & A_{3_{\rho 3}} x_3(t) + B_{3_{\rho 3}} u_1(t) + E_{3_{\rho 3}} d_3(t) \\ y_3(t) & = & C_3 x_3(t) \end{array}\right. \tag{5.84}$$

where varying parameters, state, output, known input, and unknown input are respectively $\rho_3(t) = \left[\frac{\cos\varphi\cos\theta}{m}\right]; x_3(t) = \left[\begin{array}{cc} z & \dot{z} \end{array}\right]^T; y_3(t) = \left[\begin{array}{cc} z & \dot{z} \end{array}\right]^T$; $u_3(t) = [U_1]; d_3(t) = [d_z]$ and the system matrices are

$$A_{3_{\rho 3}} = \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array}\right]; B_{3_{\rho 3}} = E_{3_{\rho 3}} = \left[\begin{array}{c} 0 \\ \rho_3 \end{array}\right]; C_3 = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right] \tag{5.85}$$

Define an auxiliary system in the form

$$\left\{\begin{array}{lcl} \dot{s}_3(t) & = & A_{3_{\rho 3}} s_3(t) + B_{3_{\rho 3}} u_3(t) \\ y_{s_3}(t) & = & C_3 s_3(t) \end{array}\right. \tag{5.86}$$

Define the errors $z_3(t) = x_3(t) - s_3(t)$ and $y_{z_3}(t) = y_3(t) - y_{s_3}(t)$, the new dynamics

$$\left\{\begin{array}{lcl} \dot{z}_3(t) & = & A_{3_{\rho 3}} z_3(t) + E_{3_{\rho 3}} d_3(t) \\ y_{z_3}(t) & = & C_3 z_3(t) \end{array}\right. \tag{5.87}$$

can be obtained from the systems in (5.84) and (5.86), which are free from the known input $u_3(t)$. Thus, after estimating the state $\hat{z}_3(t)$, the real state $\hat{x}_3(t)$ is obtained by the equation $\hat{x}_3(t) = \hat{z}_3(t) + s_3(t)$.

Take the derivative of the output $y_{z_3} = \left[\begin{array}{cc} y_{z_{31}} & y_{z_{32}} \end{array}\right]^T$ as in (5.44a) to (5.44e), one obtains:

$$\left\{\begin{array}{lcl} y_{z_{31}}^{(2)} & = & \rho_3 d_\psi \\ y_{z_{32}}^{(1)} & = & \rho_3 d_\psi \end{array}\right. \tag{5.88}$$

Consequently, the relative degrees r_i of the $i^{th}, i = 1,2$ output respectively are $r_{31} = 2$, $r_{32} = 1$.

Following the defined vector $\mathcal{Y}_3 = \left[\begin{array}{cc} y_{z_{31}}^{(2)} & y_{z_{32}}^{(1)} \end{array}\right]^T$ in (5.45), one obtains

$$\Gamma_{3_{\rho 3}} = \left[\begin{array}{c} \rho_3 \\ \rho_3 \end{array}\right] \tag{5.89}$$

which satisfies the full column rank condition and the pseudo-inverse exists and given by

$$\Gamma_{3_{\rho 3}}^{\dagger} = \left[\begin{array}{cc} \frac{1}{2\rho_3} & \frac{1}{2\rho_3} \end{array}\right] \tag{5.90}$$

The unknown input decoupling matrix Q_{3_ρ} is then given by

$$Q_{3_{\rho 3}} = \left[\begin{array}{cc} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{array}\right] \tag{5.91}$$

We can verify that the pair $\left(A_{3_{\rho 3}} - Q_{3_{\rho 3}} \mathcal{M}_{3_{\rho 3}}, C_3\right)$ is detectable, where the matrix $\mathcal{M}_{3_{\rho 3}}$ and $A_{3_{\rho 3}} - Q_{3_{\rho 3}} \mathcal{M}_{3_{\rho 3}}$ are respectively

$$\mathcal{M}_{3_{\rho 3}} = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array}\right]; A_{3_{\rho 3}} - Q_{3_{\rho 3}} \mathcal{M}_{3_{\rho 3}} = \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array}\right] \tag{5.92}$$

Finally, the gain matrix $L_{3_{\rho 3}}$ of the UIO can be calculated such that the parameter varying matrix $A_{3_{\rho 3}} - Q_{3_{\rho 3}} \mathcal{M}_{3_{\rho 3}} - L_{3_{\rho 3}} C_3$ is stable $\forall \rho_3 \in \Theta_{\rho_3}$ by solving LMIs as in (5.58), where Θ_{ρ_3} is the hyper-rectangle for ρ_3 which is defined in (5.38).

5.5.5 Simulation results

The quadcopter parameters for simulation are listed in the following table 5.2.

Table 5.2: Quadcopter parameters definition

Parameter	**Name**	**Value**	**Unit**
m	Quadcopter mass	2.0	Kg
l	Arm length	0.23	m
b	Thrust coefficient	7.73213×10^{-6}	$N \cdot s^2$
d	Drag coefficient	1.27513×10^{-7}	$N \cdot m \cdot s^2$
I_x, I_y	Inertia on x and y axis	0.0142	$Kg \cdot m^2$
I_z	Inertia on z axis	0.0267	$Kg \cdot m^2$
J_r	Rotor inertia	8.5×10^{-4}	$Kg \cdot m^2$
ω_i	Rotor speed	$[0, 500]$	$rad \cdot s^{-1}$

Based on quadcopter's parameters in table 5.2, and the definition of varying parameters in subsections 5.5.4.1 and 5.5.4.2, the ranges of varying parameters are shown in the table 5.3.

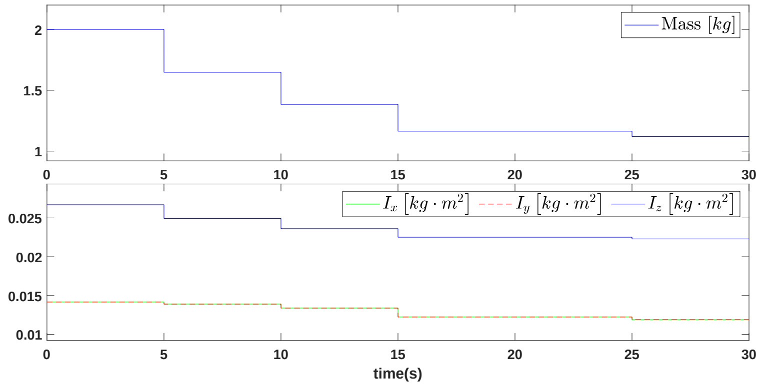


Figure 5.15: Variations of mass and moments of inertia

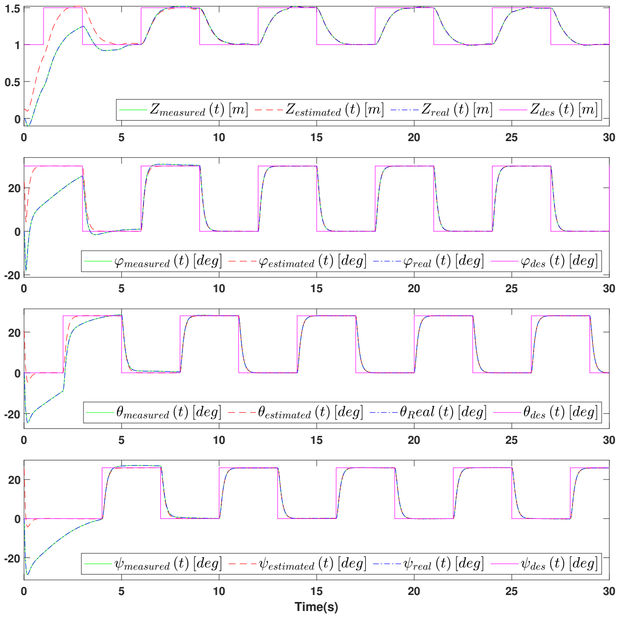


FIGURE 5.16: States z, φ, θ, ψ vs estimated states $\hat{z}, \hat{\varphi}, \hat{\theta}, \hat{\psi}$

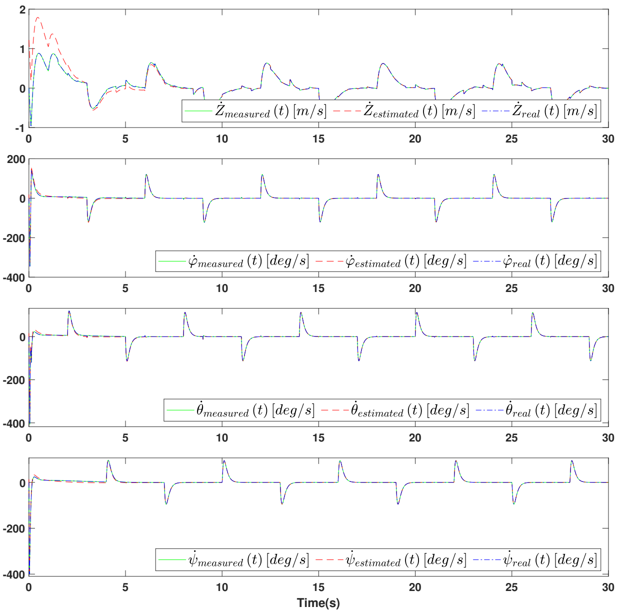


FIGURE 5.17: States $\dot{z}, \dot{\varphi}, \dot{\theta}, \dot{\psi}$ vs estimated states $\dot{\hat{z}}, \dot{\hat{\varphi}}, \dot{\hat{\theta}}, \dot{\hat{\psi}}$

TABLE 5.3: Variation ranges of varying parameters

	$\underline{\rho}_i$	$\overline{\rho}_i$
ρ_{11}	47.09580	84.0336
ρ_{12}	35.84230	44.8430
ρ_{21}	-74.1176	74.1176

For simulation, the mass of quadcopter is varying abruptly between $5s$ and $25s$ from $2\,(kg)$ to $1.12\,(kg)$. Along with the quadcopter's mass variation, the moments of inertia I_x, I_y, I_z around the three axes $I_x = I_y \in \left[\begin{array}{cc} 0.0119 & 0.0142 \end{array}\right] \left[kg \cdot m^2\right]$, and $I_z \in \left[\begin{array}{cc} 0.0223 & 0.0267 \end{array}\right] \left[kg \cdot m^2\right]$ abruptly change as in Fig. 5.15.

The actual and estimated states of the quadcopter for the simulation with the step reference trajectories of φ, θ, and ψ are shown in Fig. 5.16 and Fig. 5.17, while the actual and estimated disturbances (random disturbances) and their differences are shown in Fig. 5.18. The estimated state and disturbances converge to the actual values in about $7s$.

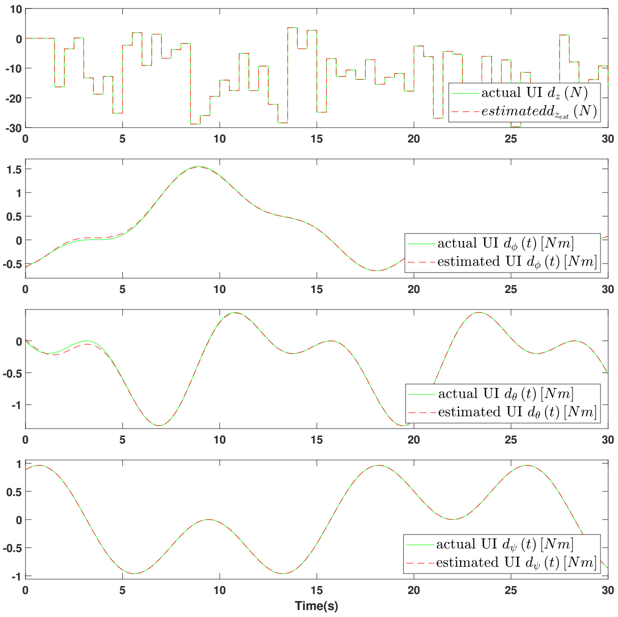


Figure 5.18: Unknown Inputs estimation

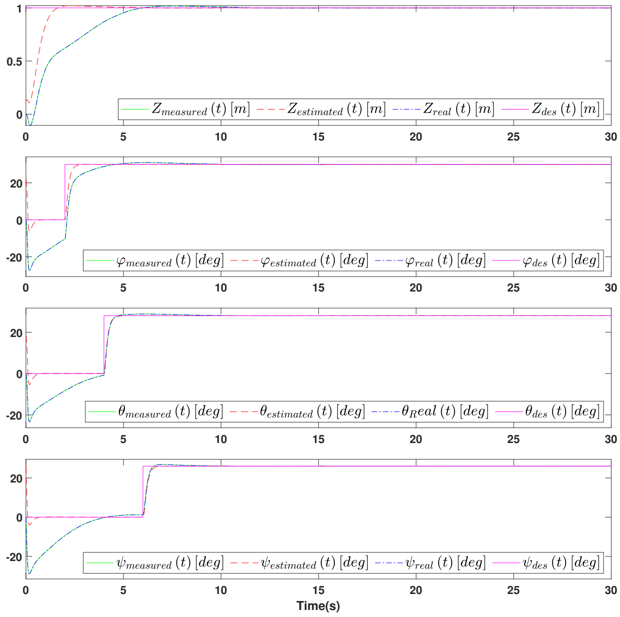


Figure 5.19: States z, φ, θ, ψ vs estimated states $\hat{z}, \hat{\varphi}, \hat{\theta}, \hat{\psi}$

In Fig. 5.19, 5.20, and 5.21 are shown the results for the case of step reference signals and constant disturbances.

Different simulations using multiple reference (impulse, step, random, and sin) and different types of disturbances (impulse, step, random, and sin) have shown a satisfactory performance of the proposed UIO.

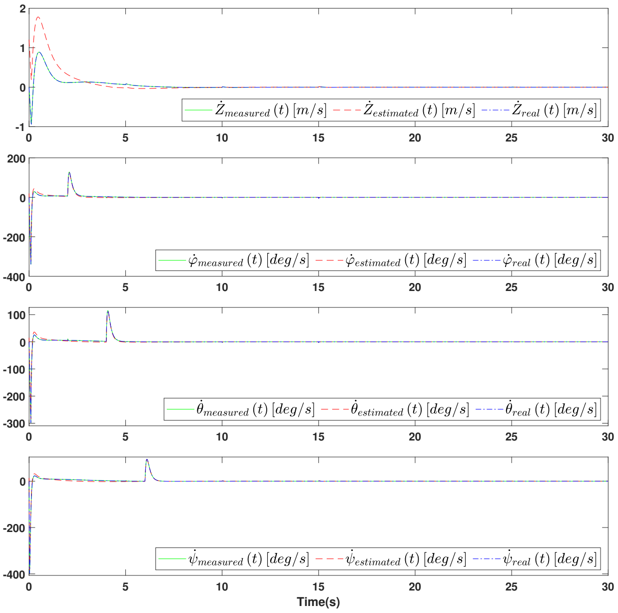


Figure 5.20: States $\dot{z}, \dot{\varphi}, \dot{\theta}, \dot{\psi}$ vs estimated states $\dot{\hat{z}}, \dot{\hat{\varphi}}, \dot{\hat{\theta}}, \dot{\hat{\psi}}$

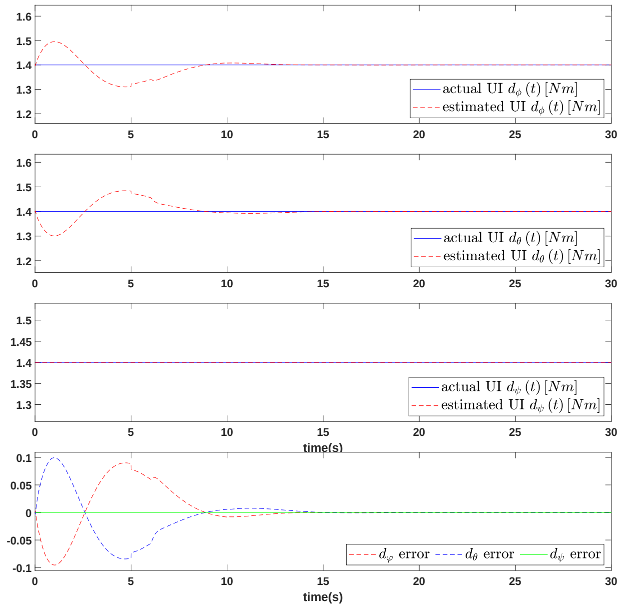


Figure 5.21: Unknown Inputs estimation

The same simulations for slow variation of mass are also conducted, the state and unknown inputs are also well estimated.

5.6 Attitude/Altitude LPV H_∞ State feedback Controller

5.6.1 System model and problem statement

The aim of this section is to propose a simple design procedure of LPV state feedback for the attitude/altitude stabilization problem of the mass-varying quadrotor aircraft. In our context, the main objective of the state feedback synthesis is to handle the mass, moments of inertia and rotors velocity variations which are assumed to be measured. This is achieved using a LPV formalism which allows to obtain a Takagi-Sugeno (TS) model with 16 sub-models depending on the extremal values of the varying parameters. The controller is then synthesized on the basis of a 16 sub-models system. It allows us to perform both a reference trajectory tracking and disturbance rejection for attitude/altitude control of a mass-varying quadcopter. First, an augmented state which includes the integration of the trajectory errors for improving tracking control is computed. Next, to penalize the control inputs of the attitude/altitude system, weight functions are also added to the previous augmented system. Then, an LPV H_∞ state-feedback controller is designed by solving a set of Linear Matrix Inequality (LMI) obtained from the Bounded Real Lemma and LMI region characterization. Simulations are conducted for several types of disturbances (sine, impulse, step, and random) and variations (slow and abrupt) of mass and moments of inertia. The reference path (sine, impulse, and step) is well-followed showing the ability of the design method to handle different performance objectives.

The remainder of the section is organized as follows: Subsection 5.6.1.1 presents the dynamical model of the quadcopter and some preliminary concepts for designing the objective of the multi-objective controller. Subsection 5.6.3 is dedicated for designing the LPV state feedback controller for the attitude/altitude of the quadcopter. The controller is practically synthesized in subsection 5.6.4, while simulation results are presented in subsection 5.6.5. Some remarks on the simulation results are shown in subsection 5.6.6.

5.6.1.1 Quadrotor model

A quadcopter is a helicopter which consists of a rigid cross frame equipped with four rotors as shown in Fig. 5.22. Its four rotors generate four independent thrusts. In order to avoid the yaw drift due to the reactive torques, the quadrotor aircraft is configured such that the set of rotors M_2, M_4 (left-right) revolve clockwise (CW) at angular speeds ω_2 and ω_4, respectively generating thrusts of τ_2 and τ_4, while the pair of rotors M_1, M_3 (front-rear) rotates at angular speeds ω_1 and ω_3 in counterclockwise (CCW) direction generating thrusts of τ_1 and τ_3. The direction of rotation of the rotors are fixed (i.e., $\omega_i \geq 0, i \in \{1,2,3,4\}$). The forward/backward, left/right and the yaw motions are achieved through a differential control strategy of the thrust generated by each rotor.

If a yaw motion is desired, the thrust of one set of rotors has to be reduced and the thrust of the other set are increased while maintaining the same total thrust to avoid an up (down) motion. Therefore, the yaw motion is then realized in the direction of the induced reactive torque. Besides, forward (backward) motion is achieved by pitching in the desired direction by increasing the rear (front) rotor thrust and decreasing the front (rear) rotor thrust to maintain the total thrust. Finally, a sideways motion is achieved by rolling in the desired direction by increasing the left (right) rotor thrust and decreasing the right (left) rotor thrust to maintain the total thrust.

Let $I = \{e_x, e_y, e_z\}$ denotes an inertial frame, and $A = \{e_1, e_2, e_3\}$ denotes a frame rigidly attached to the aircraft as shown in Fig. 5.22.

The mathematical model of the quadcopter was generated by the techniques of both Euler-Newton [30] and Euler-Lagrange [31], given as follows:

$$\begin{cases} \ddot{x}_c &= (\sin\psi\sin\varphi + \cos\psi\sin\theta\cos\varphi)\frac{U_1}{m} \\ \ddot{y}_c &= (\sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi)\frac{U_1}{m} \\ \ddot{z}_c &= (\cos\theta\cos\varphi)\frac{U_1}{m} - g \\ \ddot{\varphi} &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r\Omega_r}{I_x}\dot{\theta} + \frac{l}{I_x}U_2 \\ \ddot{\theta} &= \frac{I_z - I_x}{I_y}\dot{\varphi}\dot{\psi} + \frac{J_r\Omega_r}{I_y}\dot{\varphi} + \frac{l}{I_y}U_3 \\ \ddot{\psi} &= \frac{I_x - I_y}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_4 \end{cases} \tag{5.93}$$

where m denotes the mass of the quadcopter, (x_c, y_c, z_c) are the three positions of the center of mass, (φ, θ, ψ) are the three Euler angles, I_x, I_y, and I_z are the moments of inertia w.r.t the three axis x, y, and z respectively; J_r is the moment of inertia of the propellers, l represents the distance from the rotors to the center of mass of the quadrotor aircraft. Ω_r is the overall residual propeller angular speed, b and d are thrust and drag coefficients. The quadcopter's inputs are: the thrust force (U_1), three torques (roll torque (U_2), pitch torque (U_3), and yaw torque (U_4)). The force and torques are related to the rotor speed as follows:

$$\begin{cases} U_1 &= k_f\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) = \sum_{i=1}^{4} T_i \\ U_2 &= k_f\left(\omega_4^2 - \omega_2^2\right) = T_4 - T_2 \\ U_3 &= k_f\left(\omega_3^2 - \omega_1^2\right) = T_3 - T_1 \\ U_4 &= k_z\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right) \\ &= (T_2 + T_4) - (T_1 + T_3) \end{cases} \tag{5.94}$$

and

$$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{5.95}$$

where ω_i for $i = 1, 2, 3, 4$ denotes the i-th rotor velocity, and T_i for $i = 1, 2, 3, 4$ are the thrust generated by the i-th rotor and the thrust $T_i(t)$ is a function of the rotor speed defined by

$$T_i(t) = k_f\omega_i^2 \tag{5.96}$$

where k_f and k_z are the constant coefficients.

The first three equations of the system differential equations in (5.93) denote the translational movement, while the last three present the rotational movement of the quadcopter. We restrict the purpose of this section to the attitude tracking. Thus, the equations related to the longitudinal and lateral motions of the quadcopter in (5.93) are removed.

5.6.1.2 Actuator model

Adopting an actuator model has twofold. First of them is to reflect the low pass filtering of each actuator with a time constant $\kappa_i, i = 1,2,3,4$. The second allows us to prevent the B matrix of the obtained state-space representation to be parameter dependent. Each actuator thrust Laplace transform is given by

$$\mathcal{T}_i(s) = \frac{K_i}{1+\kappa_i s} V_i(s), \quad i = 1,2,3,4 \tag{5.97}$$

$\mathcal{T}_i$ is the Laplace transform of the thrust $T_i(t)$, V_i is the pulse width modulation (PWM) voltage applied to rotor i, and K_i is the armature gain.

The corresponding differential equation is

$$\dot{T}_i = -\frac{1}{\kappa_i} T_i + \frac{K_i}{\kappa_i} V_i \tag{5.98}$$

Remark 5.6. *Based on the PWM applied to each rotor, the rotor speed ω_i can be estimated. Thus, the residual speed $\Omega_r = \omega_2 + \omega_4 - \omega_1 - \omega_3$ can also be estimated.*

5.6.1.3 Simplified model

The previous model still exhibits too many parameters and its polytopic representation will involve at least 2^6 sub-models. If one considers control synthesis using LMI methods, the solvability of the resulting LMI conditions in this case is quite compromised due to conservativeness of conditions which will request the common stabilization of a huge number of sub-models. In order to reduce this number, we adopt here a simplified model. In particular, suppose that $I_x = I_y$ and φ and θ are small so that $cos\varphi cos\theta \approx 1$.

We also assume that the attitude/altitude subsystem of the quadcopter is affected by torques d_φ, d_θ, d_ψ, and force d_z allowing to write:

$$\left\{\begin{array}{lcl} \ddot{\varphi} & = & -\frac{J_r\Omega_r}{I_x}\dot{\theta} + \frac{l}{I_x}(T_4 - T_2) + \frac{1}{I_x}d_\varphi \\ \ddot{\theta} & = & \frac{J_r\Omega_r}{I_x}\dot{\varphi} + \frac{l}{I_x}(T_3 - T_1) + \frac{1}{I_x}d_\theta \\ \ddot{\psi} & = & \frac{(T_4+T_2)-(T_3+T_1)}{I_z} + \frac{1}{I_z}d_\psi \\ \ddot{z}_c & = & \frac{T_1+T_2+T_3+T_4}{m} + \frac{1}{m}d_z - g \end{array}\right. \tag{5.99}$$

Note that g is also the disturbance to the system. Then the disturbances can be written in the vector form as $d(t) = \begin{bmatrix} d_\varphi & d_\theta & d_\psi & d_z & g \end{bmatrix}^T$.

Then, the LPV model of attitude/altitude sub-system in (5.93) is performed as:

$$\dot{x}(t) = A(I_x, I_z, \Omega_r, m)\, x(t) + Bu(t) + E(I_x, I_z, \Omega_r, m)\, d(t) \tag{5.100}$$

where state $x = \left(\varphi, \theta, \psi, z_c, \dot{\varphi}, \dot{\theta}, \dot{\psi}, \dot{z}_c, T_1, T_2, T_3, T_4\right)^T$ has twelve components, the control input vector is composed of the four motor voltages $u = (v_1, v_2, v_3, v_4)^T$, and the system matrices $A(I_x, I_z, \Omega_r, m)$, $B(I_x, I_z, \Omega_r, m)$, and $E(I_x, I_z, \Omega_r, m)$ are

$$
\begin{aligned}
A(I_x, I_z, \Omega_r, m) &= \begin{bmatrix} A_{11} & A_{12} & A_{13} \end{bmatrix} \\
B(I_x, I_z, \Omega_r, m) &= \begin{bmatrix} 0_{8\times 4} \\ B_* \end{bmatrix} \\
E(I_x, I_z, \Omega_r, m) &= \begin{bmatrix} 0_{4\times 5} \\ E_* \\ 0_{4\times 5} \end{bmatrix}
\end{aligned}
$$

where

$$
A_{11} = [0_{12\times 4}]; \quad A_{12} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & \frac{I_x - I_z}{I_x}\dot{\psi} - \frac{J_r}{I_x}\Omega_r & 0 & 0 \\
-\frac{I_x - I_z}{I_x}\dot{\psi} + \frac{J_r}{I_x}\Omega_r & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

$$
A_{13} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & -l/I_x & 0 & l/I_x \\
-l/I_x & 0 & l/I_x & 0 \\
-l/I_z & l/I_z & -l/I_z & l/I_z \\
1/m & 1/m & 1/m & 1/m \\
-1/\kappa_1 & 0 & 0 & 0 \\
0 & -1/\kappa_2 & 0 & 0 \\
0 & 0 & -1/\kappa_3 & 0 \\
0 & 0 & 0 & -1/\kappa_4
\end{bmatrix}
$$

$$
B_* = \begin{bmatrix}
K_1/\kappa_1 & 0 & 0 & 0 \\
0 & K_2/\kappa_2 & 0 & 0 \\
0 & 0 & K_3/\kappa_3 & 0 \\
0 & 0 & 0 & K_4/\kappa_4
\end{bmatrix}; \quad E_* = \begin{bmatrix}
1/I_x & 0 & 0 & 0 & 0 \\
0 & 1/I_x & 0 & 0 & 0 \\
0 & 0 & 1/I_z & 0 & 0 \\
0 & 0 & 0 & 1/m & -1
\end{bmatrix}
$$

Note that, as described above, the system matrix B is time invariant.

Thus a TS model with 16 sub-models could be obtained depending on the extremal values of the parameters. This representation is called nonlinear sector approximation [214]. In fact, we define the varying parameters as follows

$$\begin{array}{l} \rho_1 = \frac{1}{I_x} \in \left[\begin{array}{cc} \frac{1}{\overline{I_x}} & \frac{1}{\underline{I_x}} \end{array} \right] = \left[\begin{array}{cc} \underline{\rho}_1 & \overline{\rho}_1 \end{array} \right] \\ \rho_2 = \frac{1}{I_z} \in \left[\begin{array}{cc} \frac{1}{\overline{I_z}} & \frac{1}{\underline{I_z}} \end{array} \right] = \left[\begin{array}{cc} \underline{\rho}_2 & \overline{\rho}_2 \end{array} \right] \\ \rho_3 = \frac{1}{m} \in \left[\begin{array}{cc} \frac{1}{\overline{m}} & \frac{1}{\underline{m}} \end{array} \right] = \left[\begin{array}{cc} \underline{\rho}_3 & \overline{\rho}_3 \end{array} \right] \\ \rho_4 = \frac{I_x - I_z}{I_x}\dot{\psi} - \frac{J_r}{I_x}\Omega_r \in \left[\begin{array}{cc} \underline{\rho}_4 & \overline{\rho}_4 \end{array} \right] \end{array} \tag{5.101}$$

The main objective of the control design procedure is to synthesize a state feedback controller that could be scheduled according to mass, moment of inertias and rotor speeds variations. As it can be seen from the simplified quadrotor model, it is linear in all the parameters. One can thus obtain a LPV model depending on four parameters, the moment of inertia with respect to x axis $I_x \in \left[\begin{array}{cc} \underline{I_x} & \overline{I_x} \end{array} \right]$, the moment of inertia with respect to z axis $I_z \in \left[\begin{array}{cc} \underline{I_z} & \overline{I_z} \end{array} \right]$, the residual velocity $\Omega_r \in \left[\begin{array}{cc} \underline{\Omega_r} & \overline{\Omega_r} \end{array} \right]$, and the mass $m \in \left[\begin{array}{cc} \underline{m} & \overline{m} \end{array} \right]$ of the quadcopter.

Then a sixteen sub-models TS system is achieved

$$\dot{x}(t) = \sum_{i=1}^{16} \mu_i \left(\bar{A}_i x(t) + B u(t) + \bar{E}_i d(t) \right) \tag{5.102}$$

where $\mu_i \geq 0, 1 \leq i \leq 16, \sum_1^{16} \mu_i = 1$ and

$$\begin{array}{ll} \mu_1(\rho(t)) = m_{11}m_{21}m_{31}m_{41}; & \mu_2(\rho(t)) = m_{11}m_{21}m_{31}m_{42} \\ \mu_3(\rho(t)) = m_{11}m_{21}m_{32}m_{41}; & \mu_4(\rho(t)) = m_{11}m_{21}m_{32}m_{42} \\ \mu_5(\rho(t)) = m_{11}m_{22}m_{31}m_{41}; & \mu_6(\rho(t)) = m_{11}m_{22}m_{31}m_{42} \\ \mu_7(\rho(t)) = m_{11}m_{22}m_{32}m_{41}; & \mu_8(\rho(t)) = m_{11}m_{22}m_{32}m_{42} \\ \mu_9(\rho(t)) = m_{12}m_{21}m_{31}m_{41}; & \mu_{10}(\rho(t)) = m_{12}m_{21}m_{31}m_{42} \\ \mu_{11}(\rho(t)) = m_{12}m_{21}m_{32}m_{41}; & \mu_{12}(\rho(t)) = m_{12}m_{21}m_{32}m_{42} \\ \mu_{13}(\rho(t)) = m_{12}m_{22}m_{31}m_{41}; & \mu_{14}(\rho(t)) = m_{12}m_{22}m_{31}m_{42} \\ \mu_{15}(\rho(t)) = m_{12}m_{22}m_{32}m_{41}; & \mu_{16}(\rho(t)) = m_{12}m_{22}m_{32}m_{42} \end{array}$$

with

$$\begin{array}{ll} m_{11}(\rho(t)) = \frac{\overline{\rho}_1 - \rho_1}{\overline{\rho}_1 - \underline{\rho}_1}; & m_{12}(\rho(t)) = 1 - m_{11} \\ m_{21}(\rho(t)) = \frac{\overline{\rho}_2 - \rho_2}{\overline{\rho}_2 - \underline{\rho}_2}; & m_{22}(\rho(t)) = 1 - m_{21} \\ m_{31}(\rho(t)) = \frac{\overline{\rho}_3 - \rho_3}{\overline{\rho}_3 - \underline{\rho}_3}; & m_{32}(\rho(t)) = 1 - m_{31} \\ m_{41}(\rho(t)) = \frac{\overline{\rho}_4 - \rho_4}{\overline{\rho}_4 - \underline{\rho}_4}; & m_{42}(\rho(t)) = 1 - m_{41} \end{array}$$

The matrices $\overline{A}_i, 1 \leq i \leq 16$ are obtained from

$$\begin{array}{ll} \bar{A}_1 = A\left(\underline{\rho}_1, \underline{\rho}_2, \underline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_2 = A\left(\underline{\rho}_1, \underline{\rho}_2, \underline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_3 = A\left(\underline{\rho}_1, \underline{\rho}_2, \overline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_4 = A\left(\underline{\rho}_1, \underline{\rho}_2, \overline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_5 = A\left(\underline{\rho}_1, \overline{\rho}_2, \underline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_6 = A\left(\underline{\rho}_1, \overline{\rho}_2, \underline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_7 = A\left(\underline{\rho}_1, \overline{\rho}_2, \overline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_8 = A\left(\underline{\rho}_1, \overline{\rho}_2, \overline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_9 = A\left(\overline{\rho}_1, \underline{\rho}_2, \underline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_{10} = A\left(\overline{\rho}_1, \underline{\rho}_2, \underline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_{11} = A\left(\overline{\rho}_1, \underline{\rho}_2, \overline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_{12} = A\left(\overline{\rho}_1, \underline{\rho}_2, \overline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_{13} = A\left(\overline{\rho}_1, \overline{\rho}_2, \underline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_{14} = A\left(\overline{\rho}_1, \overline{\rho}_2, \underline{\rho}_3, \overline{\rho}_4\right) \\ \bar{A}_{15} = A\left(\overline{\rho}_1, \overline{\rho}_2, \overline{\rho}_3, \underline{\rho}_4\right); & \bar{A}_{16} = A\left(\overline{\rho}_1, \overline{\rho}_2, \overline{\rho}_3, \overline{\rho}_4\right) \end{array} \tag{5.103}$$

The output vector $y = \begin{bmatrix} \varphi & \theta & \psi & z \end{bmatrix}^T$ is constituted by the quadrotor attitude/altitude position which are obtained from

$$y = Cx + Du \tag{5.104}$$

where

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

and

$$D = [0_{4\times 4}]$$

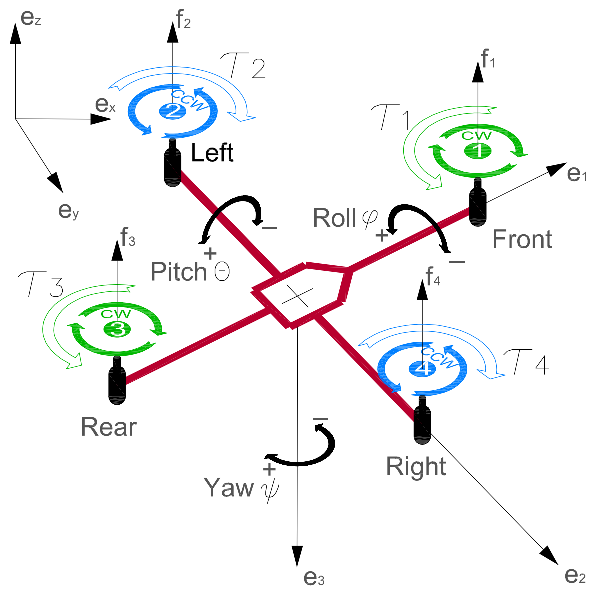


FIGURE 5.22: Quadcopter

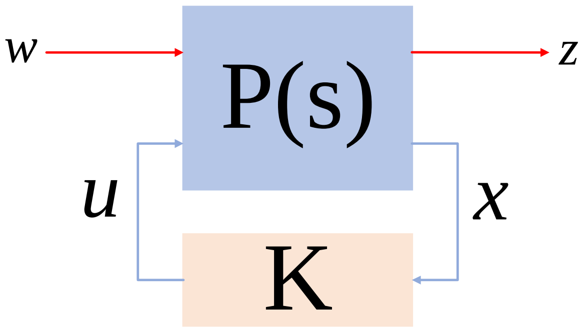


FIGURE 5.23: Control structure

Remark 5.7. *Suppose that the quadcopter is attached with* n *objects* $o_1,...,o_n$*, and the mass of quadcopter and objects are* m_q, $m_{o_1},...,m_{o_n}$ *respectively. Therefore, the mass of the system consists of the quadcopter and* n *objects can be easily calculated by the equation* $m = m_q + m_{o_1} + ... + m_{o_n}$. *When* o_i *- the* i*-th object is detached from the quadcopter for* $i = n,...,1$*, the remaining mass of the system can be recalculated.*

Depending on the mass and shape of each object, ones can calculate its moments of inertia around the axes passing through its center of mass. When attaching these objects to the quadcopter, based on their shapes and positions with respect to the center of gravity G *of the quadcopter, their the moments of inertia with respect to the three axes* I_x, I_y, I_z *of the quadcopter can be calculated. Thus the moment of inertia of the system which contains quadcopter and* n *objects* $o_1,...,o_n$ *relative to* I_x, I_y, I_z *can be calculated.*

Another online approach to estimate the geometric and inertia parameters of a multirotor aerial vehicle is already developed in [239].

Remark 5.8. *From Remarks 5.6 and 5.7 one can see that all the varying parameters can be estimated in real time.*

5.6.2 Preliminary concepts

Suppose the polytopic LPV system is of the form

$$\begin{array}{l} \dot{x}\left(t\right) = A\left(\rho\left(t\right)\right)x\left(t\right) + B_1\left(\rho\left(t\right)\right)w\left(t\right) + B_2u\left(t\right) \\ z\left(t\right) = C\left(\rho\left(t\right)\right)x\left(t\right) + D_{11}\left(t\right)w\left(t\right) + D_{12}u\left(t\right) \\ x\left(0\right) = x_0 \end{array} \tag{5.105}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, $w \in \mathbb{R}^p$ is the exogenous input, and $z \in \mathbb{R}^q$ is the controlled output. The ρ-parameter dependent system matrices is defined as

$$\begin{array}{ll} A(\rho(t)) = \sum\limits_{i=1}^{N} \mu_i(\rho(t))A_i; & B_1(\rho(t)) = \sum\limits_{i=1}^{N} \mu_i(\rho(t))B_{1_i} \\ C_1(\rho(t)) = \sum\limits_{i=1}^{N} \mu_i(\rho(t))C_{1_i}; & D_{11}(\rho(t)) = \sum\limits_{i=1}^{N} \mu_i(\rho(t))D_{11_i} \end{array}$$

while B_2 and D_{12} are constant matrices.

The purpose of this section is to design a LPV state-feedback control law

$$u\left(t\right) = \sum_{i=1}^{N} \mu_i\left(\rho\left(t\right)\right)K_i x\left(t\right) \tag{5.106}$$

such that the following conditions are satisfied:

- The H_∞ norm of the system (5.105) from w to z (as depicted in Fig. 5.23) is guaranteed to be smaller than some predefined value $\gamma > 0$ for tracking and disturbance rejection (robustness). This condition is guaranteed by the following Theorem 5.2 below
- Closed loop poles are placed in a predefined LMI region [202] for ensuring the ability of fast and well-damped transient response. The closed-loop poles satisfy the condition $Re\left(\lambda\left(A + B_2YX^{-1}\right)\right) < -\alpha$ for $\alpha > 0$

$$\begin{array}{l} Re\left(\lambda\left(A + B_2YX^{-1}\right)\right) < -\alpha, \quad \alpha > 0 \ \Leftrightarrow \exists X = X^T \succ 0 \\ s.t\ 2\alpha X + He\left(A_iX + B_2Y_i\right) \prec 0, i = 1, ..., N \end{array} \tag{5.107}$$

where $Re\left(*\right)$ defines the real part of the complex number $*$.

Theorem 5.2. *(Theorem 3.4.1 in [46]) The LPV system (5.105) is quadratically stabilizable using a state-feedback of the form (5.106) if there exist a matrix $X \in S^n_{\succ 0}$, matrices $Y_i \in {}^{m \times n}, i = 1, ..., N$, and a scalar $\gamma > 0$ such that the LMIs*

$$\begin{bmatrix} He\left(A_iX + B_2Y_i\right) & \left(*\right)^T & \left(*\right)^T \\ {B_{1_i}}^T & -\gamma I_p & \left(*\right)^T \\ C_iX + D_{12}Y_i & D_{11_i} & -\gamma I_q \end{bmatrix} \prec 0 \tag{5.108}$$

hold for all $i = 1, ..., N$. Moreover, the state-feedback control law given by (5.106) with the matrices $K_i = Y_iX^{-1}$ ensures that the L_2-gain of the transfer $w \to z$ is smaller than $\gamma > 0$ for all $\mu : R_{\geq 0} \to \Lambda_N$. ■

Then the state-feedback control law given by (5.106) with the matrices $K_i = Y_iX^{-1}$ satisfy Theorem 5.2 and equation (5.107) ensures that the L_2-gain of the transfer $w \to z$ is smaller than $\gamma > 0$ for all $\mu : R_{\geq 0} \to \Lambda_N$ and the poles of the close loop system satisfy condition $Re\left(\lambda\left(A + B_2YX^{-1}\right)\right) < -\alpha, \ \ \alpha > 0$.

5.6.3 LPV Attitude State feedback controller design

In this subsection, we aim to design a H_∞ LPV feedback control scheme for the attitude/altitude stabilization of the quadrotor aircraft.

First, the output $y = \begin{bmatrix} \varphi & \theta & \psi & z_c \end{bmatrix}^T$ of the system must track $r = \begin{bmatrix} \varphi_{ref} & \theta_{ref} & \psi_{ref} & z_{ref} \end{bmatrix}^T$, a reference trajectory . Therefore, to achieve these objectives, the outputs of the integrator are considered as extra state variables $x_e = \begin{bmatrix} x_\varphi & x_\theta & x_\psi & x_z \end{bmatrix}^T$ as

$$\begin{aligned} x_\varphi &= \int_0^t e_\varphi(\delta)\, d\delta, \quad e_\varphi = \varphi_{ref} - \varphi \\ x_\theta &= \int_0^t e_\theta(\delta)\, d\delta, \quad e_\theta = \theta_{ref} - \theta \\ x_\psi &= \int_0^t e_\psi(\delta)\, d\delta, \quad e_\psi = \psi_{ref} - \psi \\ x_z &= \int_0^t e_z(\delta)\, d\delta, \quad e_z = z_{ref} - z_c \end{aligned} \tag{5.109}$$

Define the error signal $e = y - r$. The error signal e can be rewritten in the matrix form as

$$e = y - r = Cx - I_4 r \tag{5.110}$$

Second, for penalizing the outputs U_1, U_2, U_3, U_4 of the system, the weight functions $W_{u_i}, i = 1,2,3,4$ are added to the system as depicted in Fig (5.24). The system matrices of weight functions $W_{u_i}, i = 1,2,3,4$ are A_{u_i}, B_{u_i}, C_{u_i}, and D_{u_i}.

Then, the dynamic of all the weighting functions W_{u_1}, W_{u_2}, W_{u_3} and W_{u_4} can be constituted as

$$\begin{cases} \dot{x}_u &= A_u x_u + B_u u \\ y_u &= C_u x_u + D_u u \end{cases} \tag{5.111}$$

where $x_u = \begin{bmatrix} x_{u_1} & x_{u_2} & x_{u_3} & x_{u_4} \end{bmatrix}^T$ is the state, $u = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix}^T$ represents the input, $y_u = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T$ is the outputs of weight functions, and the system matrices of the weight function in (5.111) can be deducted as follows:

$$\Delta_u = \begin{bmatrix} \Delta_{u_1} & 0 & 0 & 0 \\ 0 & \Delta_{u_2} & 0 & 0 \\ 0 & 0 & \Delta_{u_3} & 0 \\ 0 & 0 & 0 & \Delta_{u_4} \end{bmatrix}, \quad \Delta \in \{A, B, C, D\}$$

The augmented system with the new states, weight functions is depicted in Fig. 5.24.

Define $w = \begin{bmatrix} r^T & d^T \end{bmatrix}^T$, $z = \begin{bmatrix} {y_u}^T & e^T \end{bmatrix}^T$, and $\tilde{x} = \begin{bmatrix} x^T & {x_e}^T & {x_u}^T \end{bmatrix}^T$ respectively as the exogenous input, exogenous output, and state of the augmented affine parameter-dependent. The affine parameter-dependent of the system differential equations in (5.100) with augmented states and weight functions can be regathered from (5.100), (5.104), (5.110), and (5.111) as follows:

$$\begin{cases} \dot{\tilde{x}} &= \sum_{i=1}^{16} \mu_i(\rho(t)) \left(\tilde{A}_i \tilde{x} + \tilde{B}_{1_i} w + \tilde{B}_2 u \right) \\ z &= C_1 \tilde{x} + D_{11} w + D_{12} u \end{cases} \tag{5.112}$$

where

$$\tilde{A}_i = \begin{bmatrix} \bar{A}_i & 0 & 0 \\ -C & 0 & 0 \\ 0 & 0 & A_u \end{bmatrix}; \tilde{B}_{1_i} = \begin{bmatrix} 0 & \bar{E}_i \\ -I_4 & 0 \\ 0 & 0 \end{bmatrix}; \tilde{B}_2 = \begin{bmatrix} \bar{B}_i \\ 0 \\ B_u \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 0 & C_u \\ C & 0 & 0 \end{bmatrix}; D_{11} = \begin{bmatrix} 0 & 0 \\ -I_4 & 0 \end{bmatrix}; D_{12} = \begin{bmatrix} D_u \\ 0 \end{bmatrix}$$

The aim now is to design the LPV H_∞ optimal state-feedback controller of the form

$$u(t) = \left(\sum_{i=1}^{16} \mu_i(\rho(t)) K_i \right) \tilde{x}(t) \tag{5.113}$$

making the closed-loop system

$$\dot{\tilde{x}}(t) = \sum_{i=1}^{16} \mu_i(\rho(t)) \left(\left(\tilde{A}_i + \tilde{B}_2 K_i \right) \tilde{x}(t) + \tilde{B}_{1_i} w \right) \tag{5.114}$$

robustly asymptotically stable.

Define the LMIs for H_∞ optimal state-feedback controller for all TS sub model with common matrix X and each Y_i for each TS sub model based on Theorem 5.2 and poles location conditions in (5.107) as

$$\begin{aligned}
&\underset{\gamma, X, Y_1, \ldots, Y_{16}}{\text{minimize}}\ \gamma \\
&\text{subject to} \\
&X = X^T \succ 0 \\
&\begin{bmatrix} He\left(\overline{A}_i X + \overline{B}_2 Y_i\right) & (*)^T & (*)^T \\ \overline{B}_{1_i}^T & -\gamma I & (*)^T \\ \overline{C}_{1i} X + D_{12} Y_i & \overline{D}_{11_i} & -\gamma I \end{bmatrix} \prec 0 \\
&He\left(\overline{A}_i X + \overline{B}_2 Y_i\right) + 2\alpha X \prec 0; \quad i = 1..16
\end{aligned} \tag{5.115}$$

By solving the $LMIs$ in (5.115), the optimal H_∞ state feedback controller with the smallest attenuation level $\gamma > 0$ for the attitude/altitude sub system of the mass-varying quadcopter can be formulated as

$$K(\rho) = \sum_{i=1}^{16} \mu_i(\rho(t)) Y_i X^{-1} \tag{5.116}$$

5.6.4 Practical controller design

We consider Takagi-Sugeno (TS) model where the mass varies in the interval interval $[\underline{m}, \overline{m}]$ with $\underline{m} = 1.12\,(kg)$ and $\overline{m} = 2.0\,(kg)$. The moments of inertia $I_x = I_y$ varies in the interval $\left[\underline{I_x}, \overline{I_x}\right]$ with $\underline{I_x} = 0.0119\left(kg.m^2\right)$ and $\overline{I_x} = 0.0142\left(kg.m^2\right)$. The moments of inertia I_z varies in the interval $\left[\underline{I_z}, \overline{I_z}\right]$ with $\underline{I_z} = 0.0223\left(kg.m^2\right)$ and $\overline{I_z} = 0.0267\left(kg.m^2\right)$. The total residual angular speed $\mathbf{\Omega}_r$ of motors varies in the interval $\left[\underline{\mathbf{\Omega}_r}, \overline{\mathbf{\Omega}_r}\right]$ with $\underline{\mathbf{\Omega}_r} = -1000\left[rad \cdot s^{-1}\right]$ and $\overline{\mathbf{\Omega}_r} = 1000\left[rad \cdot s^{-1}\right]$. The controller is designed using the procedure developed above.

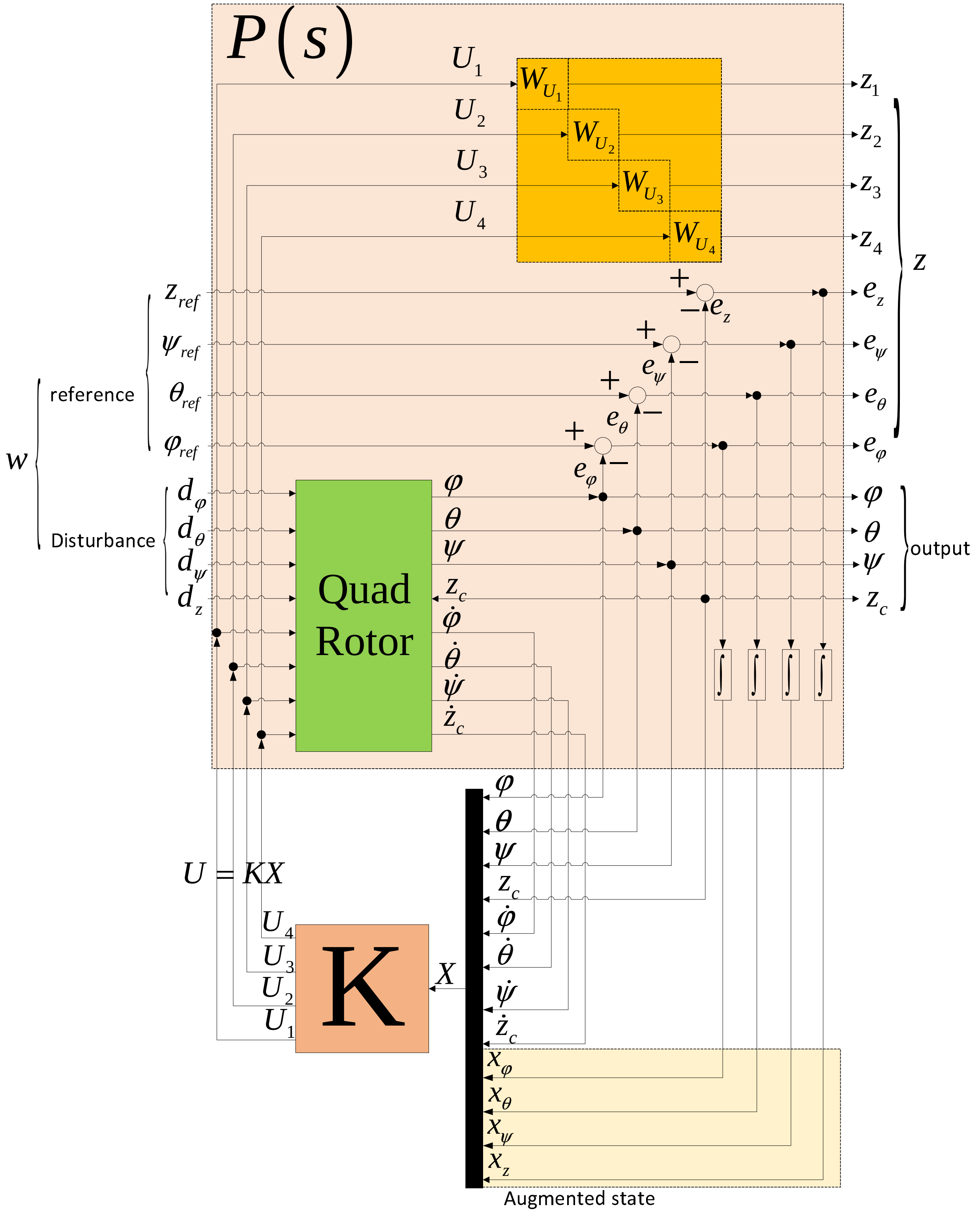


FIGURE 5.24: Block diagram of the attitude robust controller with augmented states and weight functions

The quadcopter parameters for simulation are listed in the following table 5.4. Based on the quadcopter's parameters in table 5.4, and the definition of varying parameters in subsections 5.6.1.3, the ranges of varying parameters are shown in the table 5.5.

TABLE 5.4: Quadcopter parameters definition

Parameters	**Name**	**Value**	**Unit**
m	Quad. mass	2.0	Kg
l	Arm length	0.23	m
b	Thrust coef.	7.73213×10^{-6}	$N.s^2$
d	Drag coef.	1.27513×10^{-7}	$N.m.s^2$
I_x, I_y	Inertia vs x, y	0.0142	$Kg.m^2$
I_z	Inertia vs z	0.0267	$Kg.m^2$
J_r	Rotor inertia	8.5×10^{-4}	$Kg.m^2$
ω_i	Rotor speed	$[0, 500]$	rad/s
τ_i	Rotor time const	15	rad/s
g	Gravity accel.	9.81	m/s^2

5.6.5 Testing scenario

In simulations, the mass of the quadcopter is varying abruptly between $5s$ and $25s$ from $2\,(kg)$ to $1.12\,(kg)$. Along with the quadcopter's mass variation, the moments of inertia I_x, I_y, I_z for $I_x = I_y \in \left[\begin{array}{cc} 0.0119 & 0.0142 \end{array}\right]$, and $I_z \in \left[\begin{array}{cc} 0.0223 & 0.0267 \end{array}\right] \left[kg \cdot m^2\right]$ also abruptly change as in Fig. 5.25. Fig. 5.26 shows the responses of φ, θ, ψ and z when the reference signals are impulses and the disturbances d_φ, d_θ, d_ψ, and d_z are impulses. Fig. 5.27 shows the responses of φ, θ, ψ and z when the reference signals are random, the z reference signal is impulse, and the disturbances d_φ, d_θ, d_ψ, and d_z are random. Fig. 5.28 shows the responses of φ, θ, ψ when the reference signals ϕ, θ, ψ are, the z reference signal is step, and the disturbances d_φ, d_θ, d_ψ, and d_z are also sine.

TABLE 5.5: Variation ranges of varying parameters

	$\underline{\rho}_i, i = 1,2,3,4$	$\overline{\rho}_i, i = 1,2,3,4$
ρ_1	47.09580	84.0336
ρ_2	35.84230	44.8430
ρ_3	0.5000	0.89290
ρ_4	-74.1176	74.1176

The simulation results suggest that the proposed controller works well for various reference signals (impulse, random, constant, and sine) and several types of disturbances (impulse, random, constant, and sine).

The same simulation for slow variation of mass is also conducted, the reference paths are also well-followed.

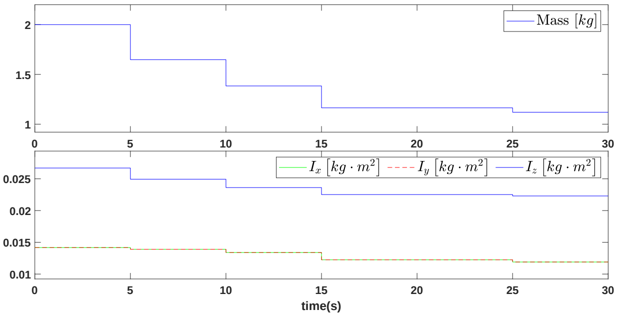


FIGURE 5.25: Variations of Mass, I_x, I_y, and I_z

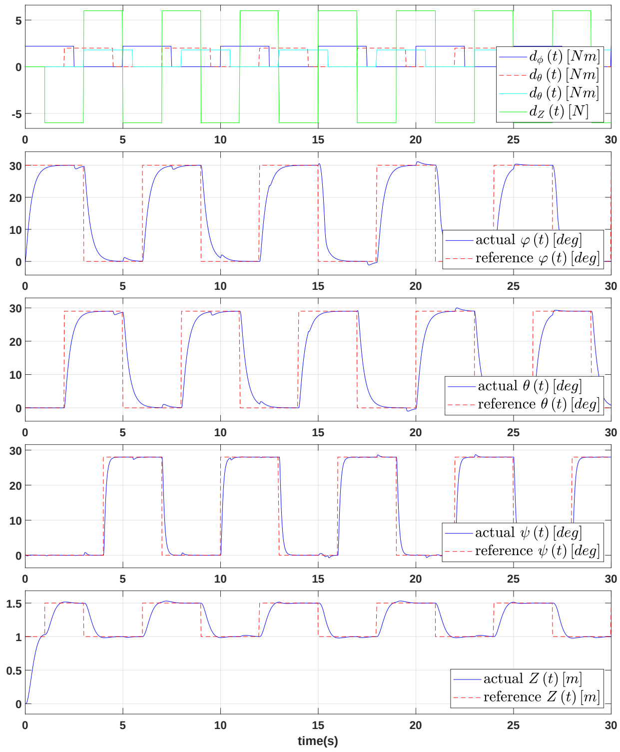


FIGURE 5.26: Impulse references φ, θ, ψ, impulse reference z, and impulse disturbances d_φ, d_θ, d_ψ, d_z

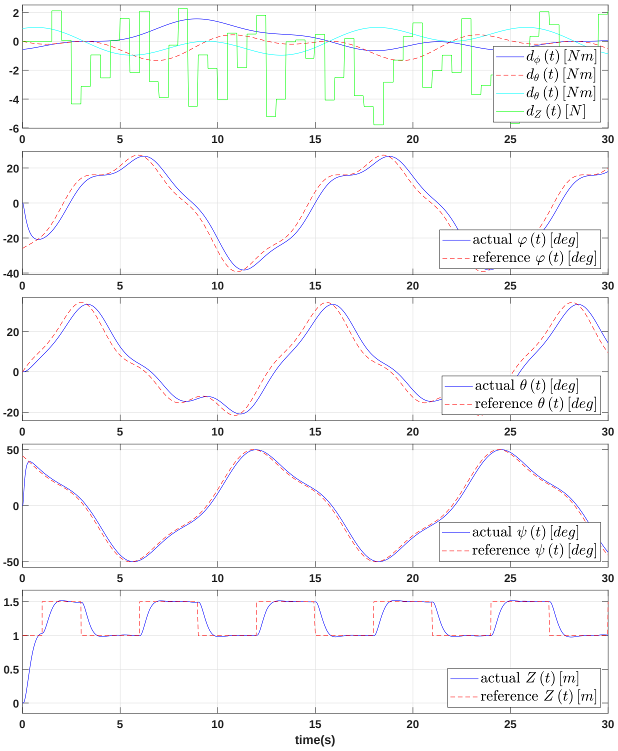


FIGURE 5.27: Random references φ, θ, ψ, impulse reference z, and random disturbances d_φ, d_θ, d_ψ, d_z

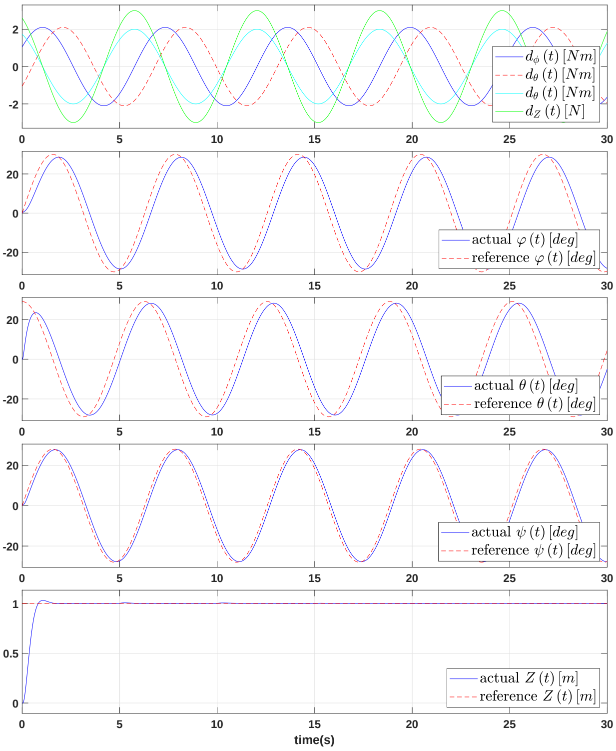


FIGURE 5.28: Sine references φ, θ, ψ, step reference z, and sine disturbances d_φ, d_θ, d_ψ, d_z

5.6.6 Remarks on simulation results

This section addresses the problem of attitude/altitude control of a quadcopter UAV. The focus is on handling mass, moments of inertia variation of the UAV according to the specific application of transporting different device types. By adding some additional state and weight functions, the linear parameter-dependent system is gathered. Thus the problem of reference tracking is formulated as H_∞ state feedback. It is solved using the LMI conditions framework. The obtained controller is found to be able to follow the prescribed trajectory with a high level of performance.

Future works of this section could concern the observer-based controller for a mass-varying quadcopter.

5.7 LPV Static output feedback controller for Attitude/Altitude

The aim of this section is to propose a simple design procedure of static output feedback for quadrotor. The use of static output feedback controller simplifies greatly control law reading and implementation [37], [57]. However, the synthesis is known as a hard problem but some relaxation procedures allow to reduce conservatism [56]. In our context, the main objective of the static output feedback synthesis is to handle the mass and rotors velocity variations which are assumed to be measured. This is achieved using a Takagi-Sugeno (TS) formalism which allows to obtain a linear parameter varying model in the form of a convex sum of submodels [214]. The controller is then synthesized on the basis of a 4 submodels system.

The remainder of this section is organized as follows. Subsection 5.7.1 presents the classical quadrotor model which is simplified in order to obtain a TS model suitable for control synthesis. Subsection 5.7.2 is devoted to the reference model tracking problem setting and model adaptation. Thus, the controller is synthesized in subsection 5.7.3 while simulation results are presented in section 5.7.4. Some remarks on simulation results are shown in subsection 5.7.5.

5.7.1 More simplified model

The mathematical differential equations of attitude/Altitude system of quadcopter are rewritten as

$$\left\{\begin{array}{rcl} \ddot{\varphi} & = & \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r\Omega_r}{I_x}\dot{\theta} + \frac{l}{I_x}U_2 \\ \ddot{\theta} & = & \frac{I_z - I_x}{I_y}\dot{\varphi}\dot{\psi} + \frac{J_r\Omega_r}{I_y}\dot{\varphi} + \frac{l}{I_y}U_3 \\ \ddot{\psi} & = & \frac{I_x - I_y}{I_z}\dot{\varphi}\dot{\theta} + \frac{1}{I_z}U_4 \\ \ddot{z}_c & = & (\cos\theta\cos\varphi)\frac{U_1}{m} - g \end{array}\right. \tag{5.117}$$

The previous model still exhibits too much parameters and its polytopic representation will involve 2^9 submodels. If one considers control synthesis using Linear Matrix Inequalities methods, the resolvability of the resulting LMI conditions in this case is quite compromised due to conservativeness of conditions which will request the common stabilization of a huge number of submodels. In order to reduce this number, we adopt here a more simplified model.

By using the model of the actuator as given in remark 5.4 and the simplification that velocities are small such that the product of the velocities can be assumed to be zero, allowing to write:

$$\left\{\begin{array}{lcl} \ddot{\varphi} & = & -\frac{J_r}{I_x}\dot{\theta}\Omega_r + l\frac{T_4-T_2}{I_x} \\ \ddot{\theta} & = & \frac{J_r}{I_y}\dot{\varphi}\Omega_r + l\frac{T_1-T_3}{I_y} \\ \ddot{\psi} & = & l\frac{(T_2+T_4)-(T_1+T_3)}{I_z} \\ \ddot{z}_c & = & -g + \frac{T_1+T_2+T_3+T_4}{I_z} \end{array}\right. \tag{5.118}$$

Gathering all the equations, one can establish the LPV model

$$\dot{x}(t) = A(m,\Omega_r)\,x(t) + Bv(t) + Ed(t) \tag{5.119}$$

where the state vector is has twelve components: $x = \left(\varphi,\dot{\varphi},\theta,\dot{\theta},\psi,\dot{\psi},z_c,\dot{z}_c,T_1,T_2,T_3,T_4\right)^T$, the control input vector is composed of the four motor voltages $v = (v_1,v_2,v_3,v_4)^T$, the disturbance input is the Earth gravity $d = g$, and the matrix $A(m,\Omega_r)$ is given by

$$A(m,\Omega_r) = \begin{bmatrix} A_{11} & A_{12} & A_{13} \end{bmatrix} \tag{5.120}$$

where

$$A_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{J_r}{I_x}\Omega_r \\ 0 & 0 & 0 & 1 \\ 0 & \frac{J_r}{I_y}\Omega_r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \; A_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \; A_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{l}{I_x} & 0 & \frac{l}{I_x} \\ 0 & 0 & 0 & 0 \\ \frac{l}{I_y} & 0 & -\frac{l}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{l}{I_z} & \frac{l}{I_z} & -\frac{l}{I_z} & \frac{l}{I_z} \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} \\ -\frac{1}{\tau_1} & 0 & 0 & 0 \\ 0 & -\frac{1}{\tau_2} & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_4} \end{bmatrix}$$

while

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{K_1}{\tau_1} & 0 & 0 & 0 \\ 0 & \frac{K_2}{\tau_2} & 0 & 0 \\ 0 & 0 & \frac{K_3}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{K_4}{\tau_4} \end{bmatrix}; \quad E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The main objective of the control design procedure is to synthesize a controller that could be scheduled according to mass, inertia and rotor speed variation. As it can be seen from the simplified quadrotor model, it is linear of both parameters. In addition, parametrizing the moment of inertia in the form $J_i = \kappa_i m, (i = x, y, z)$, one can thus obtain a LPV model depending on two parameters, the mass $m \in [m_m, m_M]$ and the velocity $\Omega_r \in [\Omega_m, \Omega_M]$.

Thus a Takagi-Sugeno model with fours submodels could be obtained depending on the extremal values of the parameters. This representation is called nonlinear sector approximation [214]. In fact, defining $\rho_1 = m \in [m_m, m_M]$ and $\rho_2 = \frac{\Omega_r}{m} \in \left[\frac{\Omega_m}{m}, \frac{\Omega_M}{m}\right]$ a four submodels TS system is achieved

$$\dot{x}(t) = \left(\sum_{i=1}^{4} \mu_i(\rho(t))\overline{A}_i\right) x(t) + \overline{B}u(t) + \overline{E}d(t) \tag{5.121}$$

where $\mu_i \geq 0, 1 \leq i \leq 4, \sum\limits_{i=1}^{4} \mu_i = 1$ and

$$\begin{array}{ll} \mu_1 = m_{11}m_{21}, & \mu_2 = m_{11}m_{21} \\ \mu_3 = m_{12}m_{21}, & \mu_3 = m_{12}m_{22} \end{array}$$

with

$$\begin{array}{ll} m_{11} = \frac{\rho_{1\max} - \rho_1}{\rho_{1\max} - \rho_{1\min}}, & m_{12} = 1 - m_{11} \\ m_{21} = \frac{\rho_{2\max} - \rho_2}{\rho_{2\max} - \rho_{2\min}}, & m_{22} = 1 - m_{21} \end{array}$$

The matrices $\overline{A}_i, 1 \leq i \leq 4$ are obtained from

$$\begin{array}{l} \overline{A}_1 = A(m_m, \Omega_m) \\ \overline{A}_2 = A(m_m, \Omega_M) \\ \overline{A}_3 = A(m_M, \Omega_m) \\ \overline{A}_4 = A(m_M, \Omega_M) \end{array}$$

Notice that the obtained model is an exact representation of the model of equation (5.119). The obtained model is a continuous time model. However for implementation aspects and when considering preview information, it is more convenient to consider it in discrete time domain. In fact the preview information is only available at some sensor sample times. Knowing that, the model is discretized using a simple Euler method, this leads to the discrete-time state-space model with sample time of $T = 0.05sec$.

$$\begin{cases} x(k+1) = \left(\sum\limits_{i=1}^{4} \mu_i(\rho(t))\overline{A}_i\right) x(k) + Bu(k) + Ed(k) \\ y(k) = Cx(k) \end{cases} \tag{5.122}$$

where $(A_i, B, E) = \left(I_{12} + T\overline{A}_i, T\overline{B}, T\overline{E}\right)$

The output vector is constituted by the quadrotor altitude and attitude position which are obtained from the matrix

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

5.7.2 Controller Design

The considered model has the gravity a constant disturbance input. In order to cancel it in the control design and to obtain a zero steady state error, one can take the difference operator on both sides of equation (5.122) which leads to:

$$\begin{cases} \Delta x(k+1) = \left(\sum\limits_{i=1}^{4} \mu_i(\rho(t))\overline{A}_i\right)\Delta x(k) + B\Delta u(k) \\ \Delta y(k) = C\Delta x(k) \end{cases} \tag{5.123}$$

We consider in this section the problem of reference signal tracking with preview information. The output signal $y(k)$ has to follow the predefined reference signal $r(k)$. Defining the error signal $e(k)$ as

$$e(k) = y(k) - r(k) \tag{5.124}$$

and writing the error dynamics, one can define the augmented plant with the augmented state vector $\tilde{x} = \left[e^T(k), \Delta x^T(k)\right]^T$ which reads

$$\begin{cases} \tilde{x}(k+1) = \sum\limits_{i=1}^{4} \mu_i(\rho(t))\left(\tilde{A}_i\tilde{x}(k) + \tilde{B}\Delta u(k) + G_p\Delta r(k)\right) \\ e(k) = \tilde{C}\tilde{x}(k) \end{cases} \tag{5.125}$$

where

$$\tilde{A}_i = \begin{bmatrix} I_4 & C \\ 0 & A_i \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ B \end{bmatrix}, \quad G_p = \begin{bmatrix} -I_4 \\ 0 \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} I_4 & 0 \end{bmatrix}$$

Suppose now that the reference signal values are known n_p samples ahead. Let us define the vector

$$x_r(k) = \begin{bmatrix} \Delta r^T(k) & \cdots & \Delta r^T(k+n_p) \end{bmatrix}^T \tag{5.126}$$

and the matrix

$$A_r = \begin{bmatrix} 0 & I_4 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & I_4 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}$$

The objective now is to embed the reference signal appearing in equation (5.125) into a state vector. This is achieved by defining the state vector

$$\hat{x}(k) = \begin{bmatrix} \tilde{x}^T(k) & \tilde{x}_r^T(k) \end{bmatrix}^T$$

which allows to obtain the augmented system

$$\begin{cases} \tilde{x}(k+1) = \sum\limits_{i=1}^{4} \mu_i(\rho(t))\left(\hat{A}_i\hat{x}(k) + \hat{B}\Delta u(k)\right) \\ y_p(k) = \hat{C}\hat{x}(k) \end{cases} \tag{5.127}$$

where

$$\hat{A}_i = \begin{bmatrix} \tilde{A}_i & G_p \\ 0 & A_r \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} I_q & & \\ & C & \\ & & I_{n_pq+q} \end{bmatrix}$$

The aim now is to design a static output feedback controller of the form

$$\Delta u(k) = \left(\sum_{i=1}^{4} \mu_i(\rho(t)) K_i\right) y_p(k) \tag{5.128}$$

making the closed-loop system

$$\hat{x}(k+1) = \sum_{i=1}^{4} \mu_i(\rho(t)) \left(\hat{A}_i + \hat{B} K_i \hat{C}\right) \hat{x}(k) \tag{5.129}$$

robustly asymptotically stable. Following the results provided in [128], given a scalar β and

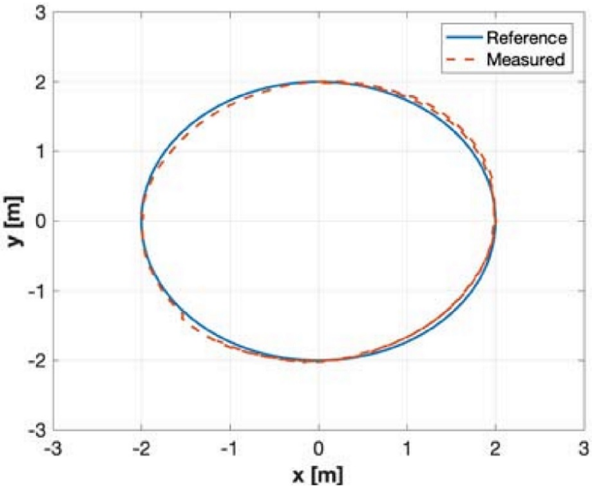


FIGURE 5.29: Measured and reference path

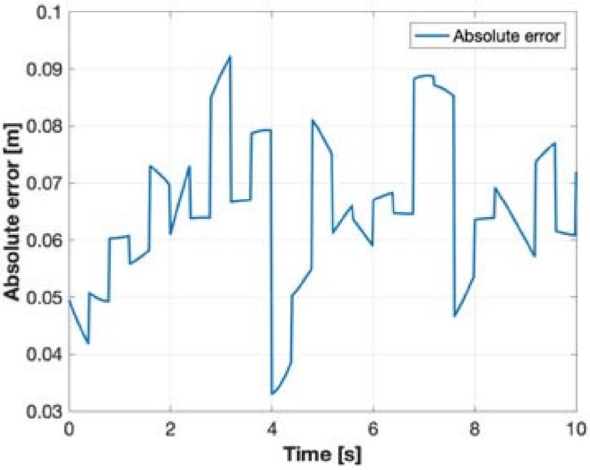


FIGURE 5.30: Absolute error on path following

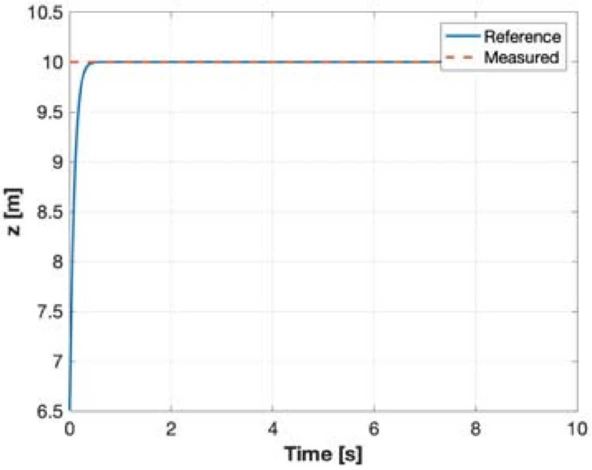


FIGURE 5.31: Measured and reference value for the altitude

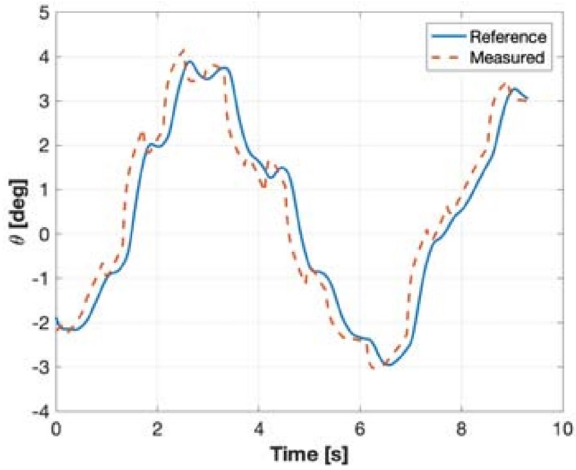


FIGURE 5.32: Measured and reference values for the pitch angle

matrices Q, W, if there exist $P_i > 0, U_i, L_i, G_i, 1 \leq i \leq 4$ such that

$$\Pi_i \prec 0, \quad 1 \leq i \leq 4 \tag{5.130}$$

where

$$\Pi_i = \begin{bmatrix} -G_i - G_i^T + P_i & * & * \\ \hat{A}_i G_i + \hat{B}_i L_i Q & -P_i & * \\ \hat{C} G_i - UQ & \beta W^T L_i^T \hat{B}^T & \Theta \end{bmatrix} \tag{5.131}$$

and $\Theta = -\beta UW - \beta W^T U^T$, then the system (5.129) is robustly asymptotically stable, and the gain matrix can be obtained by

$$K_i = L_i U^{-1} \tag{5.132}$$

One can notice that this matrix gain involves the following components

$$K_i = \begin{bmatrix} K_{ei} & K_{yi} & K_{ri}(0) & \dots & K_{ri}(n_p) \end{bmatrix} \tag{5.133}$$

thus the command increment is given by

$$\Delta u(k) = K_e e(k) + K_y \Delta y(k) + \sum_{j=0}^{n_p} K_r(j) \Delta r(k+j) \tag{5.134}$$

where

$$K_e = \sum_{i=1}^{4} \mu_i K_{ei}, \quad K_y = \sum_{i=1}^{4} \mu_i K_{yi}, \quad K_r(j) = \sum_{i=1}^{4} \mu_i K_{ri}(j)$$

and finally the control input is obtained from

$$u(k) = K_e \sum_{j=0}^{n_p} e(k) + K_y y(k) + \sum_{j=0}^{n_p} K_r(j) r(k+j) \tag{5.135}$$

5.7.3 Practical controller design

The TS model is considered assuming that the mass varies in the interval $[m_m, m_M]$ with $m_m = 1\,[kg]$ and $m_M = 2\,[kg]$. The total propeller velocity is varying in the interval $[\Omega_m, \Omega_M]$ with $\Omega_m = -0.2$ and $\Omega_M = 0.2$. The controller is designed using the procedure developed above. The preview horizon is taken up to $n_p = 5$ points.

5.7.4 Testing scenario

As stated in the introduction, this paper is concerned with the attitude/altitude control. It is then assumed that the reference values for the pitch, roll, yaw and attitude are provided by an external loop. The quadrotor is requested to follow a circular path while increasing its altitude from $5\,[m]$ to $10\,[m]$. The path is shown in Fig. 5.29. One can see that the reference path is well followed. This can be verified in the Fig. 5.30 where the absolute value does not exceed $0.1\,[m]$. The altitude reference is also well followed as presented in Fig. 5.31.This is also the case for the pith angle in Fig. 5.32. The reference value set for the yaw angle is 0. This is reached in less than one second in Fig. 5.34. The same simulation is now conducted by increasing the quadrotor mass until $2\,[kg]$. The system responses in this case are shown in Fig. 5.35 to Fig. 5.38. The responses become more oscillatory but remain acceptable.

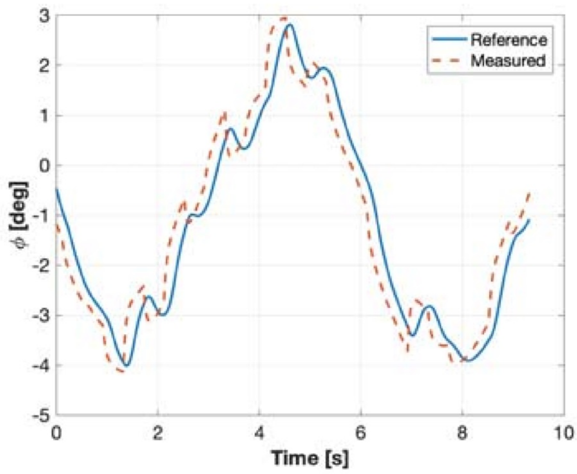


FIGURE 5.33: Measured and reference values for the roll angle

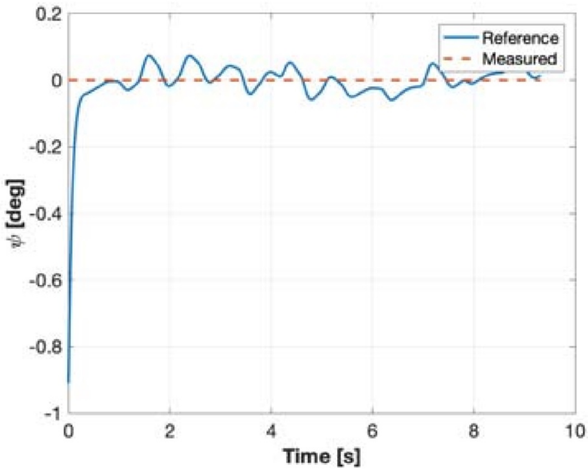


FIGURE 5.34: Measured and reference values for the yaw angle

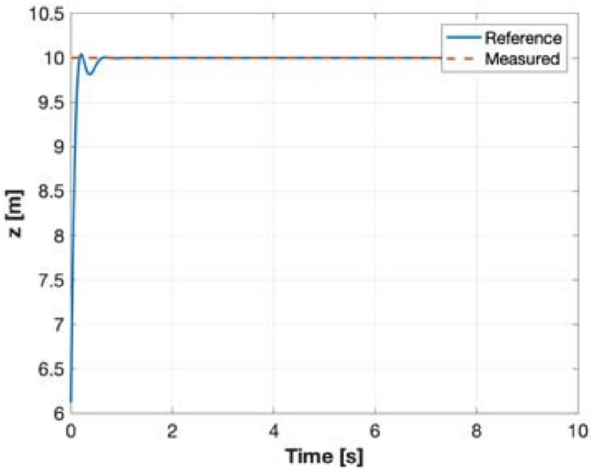


FIGURE 5.35: Measured and reference value for the altitude for added mass

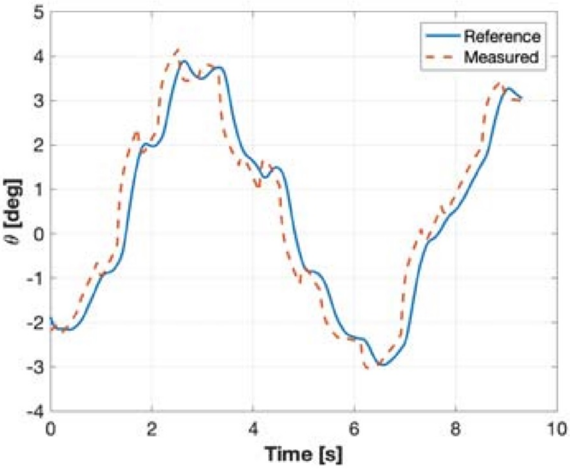


FIGURE 5.36: Measured and reference values for the pitch angle for added mass

5.7.5 Comments on the simulation results

This section addresses the problem of attitude/altitude control of an UAV. The focus is put on handling mass variation of the UAV according to the specific application of transporting different device types. Following an incremental model simplification, a simple LPV model is obtained for the selected motions. Afterwards, a 4 submodels TS model is derived. Thus the problem of reference tracking is formulated for static output feedback. It is solved using LMI conditions framework. The obtained controller is found to be able to follow the prescribed trajectory with high level of performance. Future works will both concern the exploration of reactive path planning realization and fault tolerant control.

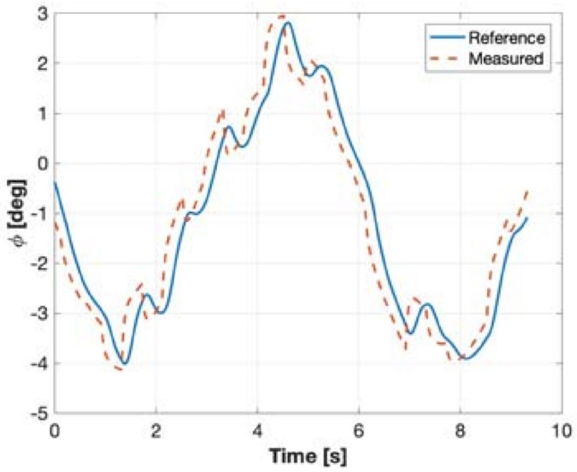


FIGURE 5.37: Measured and reference values for the roll angle for added mass

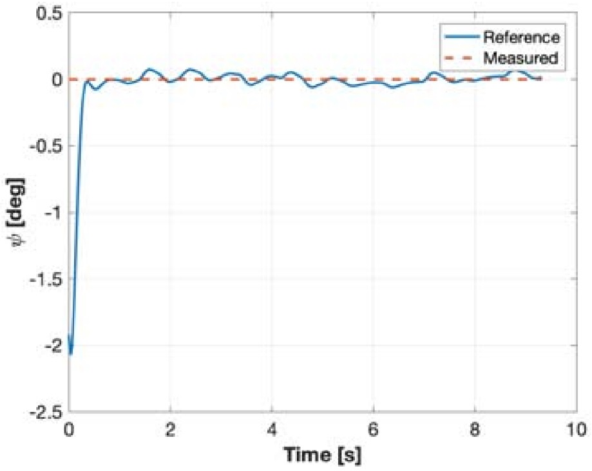


FIGURE 5.38: Measured and reference value for the yaw angle for added mass

5.8 Conclusion

In this chapter, we have examined the control problem for quadrotor applied in PA. In these applications, the mass of the quadrotor changes over time leads to changes in dynamic parameters of quadrotor such as the moments of inertia change. First, if the states of the system are available (measured or estimated), a H_∞ LPV controller was developed for a mass-varying quadrotors. Then, when states are not available, the LPV dynamic and static controllers have been designed. Then, this chapter also concern the Linear Parameter Varying (LPV) Unknown Input Observer (UIO) for the mass-varying quadrotor.

6

Simulation results

Chapter abstract

This chapter is dedicated for simulation results. In the first section 6.1, the Coverage Path Planning (CPP) for disease detection and crops spraying are generated. The section 6.2 is for simulation results of controlling the quadrotor. First, we show the stabilization simulation results for quadrotor using Linear Parameter Varying (LPV) H_∞ dynamic output feedback, LPV static output feedback, and LPV H_∞ state feedback controllers in subsection 6.2.1. The LPV Unknown Input Observer (UIO) for the altitude/attitude of the quadrotor will be shown in subsection 6.2.2. Then, in the subsection 6.2.3, the full control of quadrotor to follow the predefined points is given. The conclusion and some remarks are in section 6.3.

This Chapter contains:

6.1 Coverage Path Planning (CPP) simulations

6.1.1 Coverage Path Planning (CPP) for disease detection

Assume that we have a rectangular agricultural land as shown in figure 6.1. In this agricultural land, there are convex and concave obstacles (yellow polygons) and pest infestation plants (red dots). The positions of the obstacles are known in prior, while the locations of pests ridden areas are unknown.

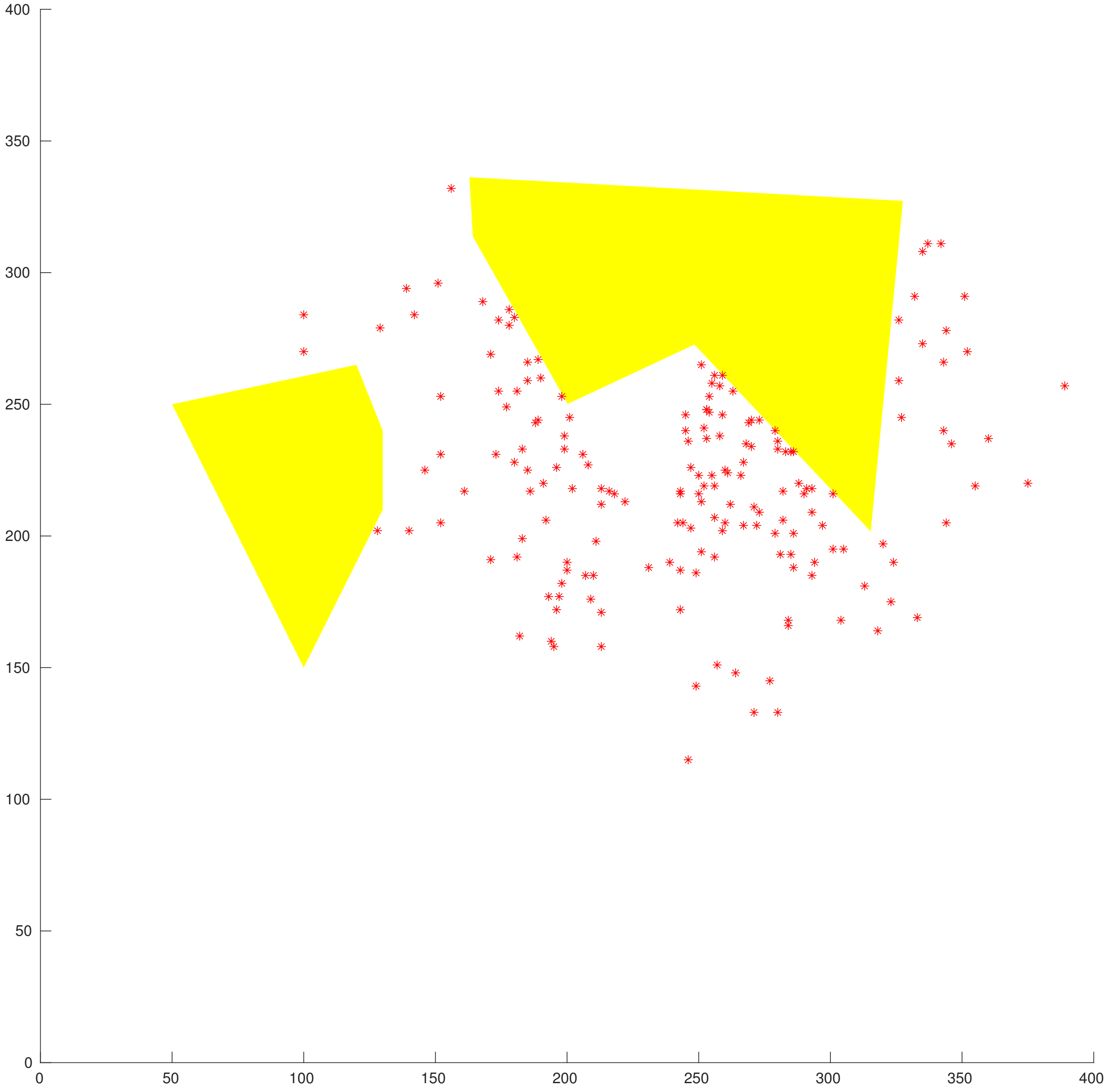


FIGURE 6.1: Agricultural area with obstacles and infected points

Scenario 1. *Our mission is to use UAVs to take pictures of the entire agricultural land so that the coordinates of the crop areas affected by pests can be determined.*

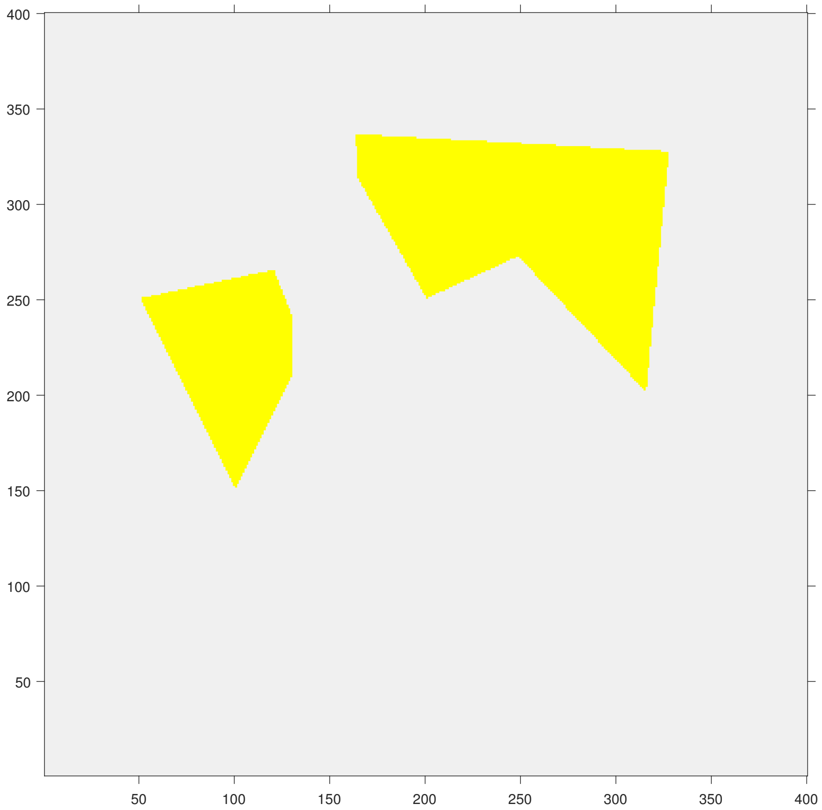


Figure 6.2: Scenario 1
Original images with obstacles used for infected area detection

Based on the proposed algorithm in Figure 6.3, first, it is necessary to define critical points (blue and red circles in Figure 6.3). Then from these critical points, the agricultural area is divided into cells as in Figure 6.4 by using the proposed algorithm on 4.3.2.1. In this case, the agricultural area is divided into four non-intersecting obstacle-free regions (as shown and numbered in Figure 6.4).

Using the path generation algorithm proposed in the 4.3.2.2 section, we will draw rectangles equal in size to the frame size that the camera-equipped on the UAV can capture. These rectangles should intersect in both the long and short sides (as shown in Figure 6.5) to ensure the completeness of coverage of the entire area.

In the rectangles, as shown in Figure 6.5, we see that there are a number of rectangles centered on the obstacles (Figure 6.6). These locations are the points that the UAV cannot reach. Therefore, only rectangles whose center is not on obstacles (Figure 6.7) are considered to create the motion trajectory for UAV.

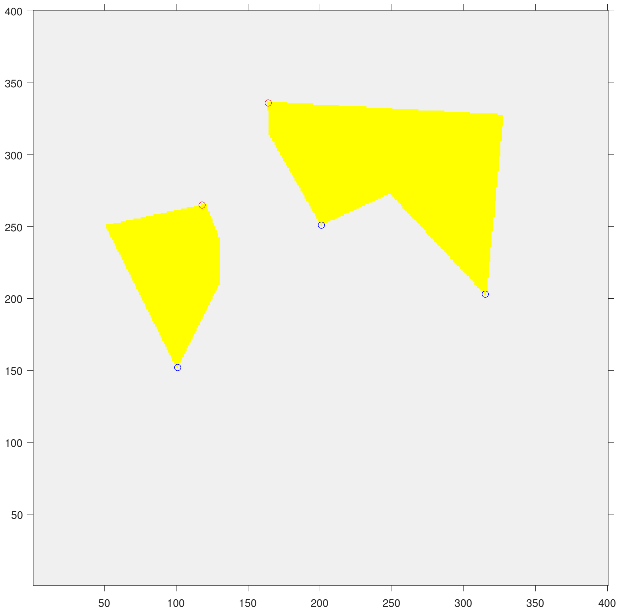


FIGURE 6.3: Scenario 1:
Infected area detection Critical points

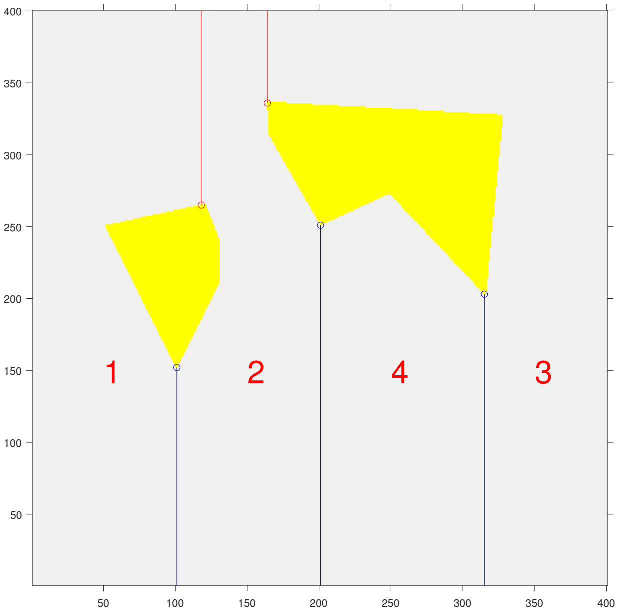


FIGURE 6.4: Scenario 1:
Infected area detection Cellular decomposition

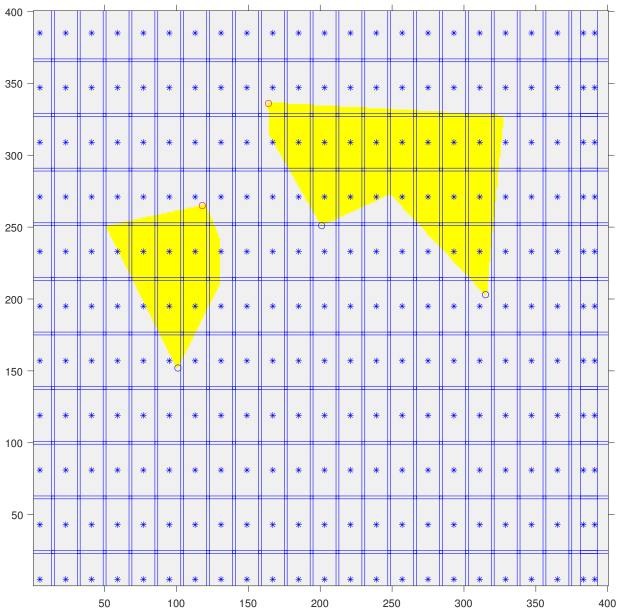


FIGURE 6.5: Scenario 1:
Infected area detection Image frames

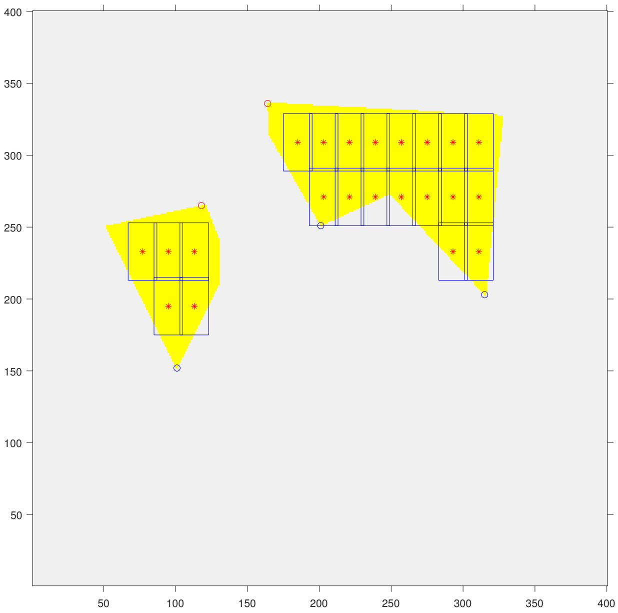


FIGURE 6.6: Scenario 1:
Infected area detection Image frames in obstacles

Rectangles whose center is not on obstacles are grouped into other groups depending on which cells the center coordinates are. As shown in Figure 6.7, rectangles whose center is not on obstacles are grouped into four groups with different colors.

From the center coordinates of the rectangles of these groups, the boustrophedon paths for the UAV in each cell (as shown in Figure 6.8a) are generated ensuring that UAV does not collide with the obstacles while traveling along with the generated path.

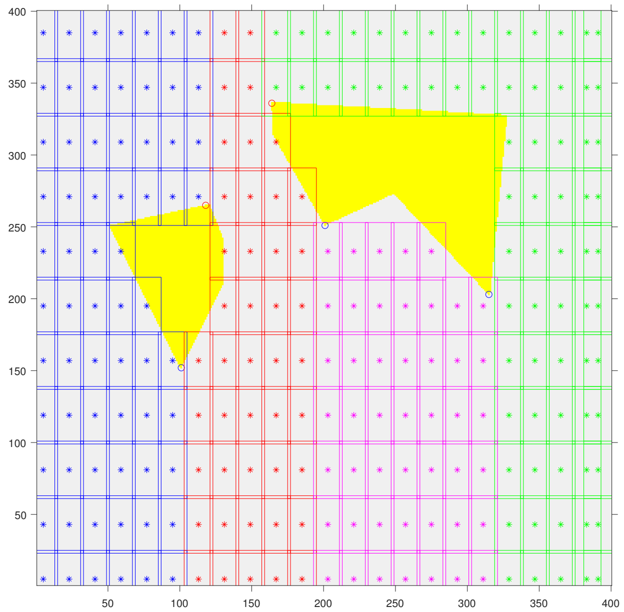
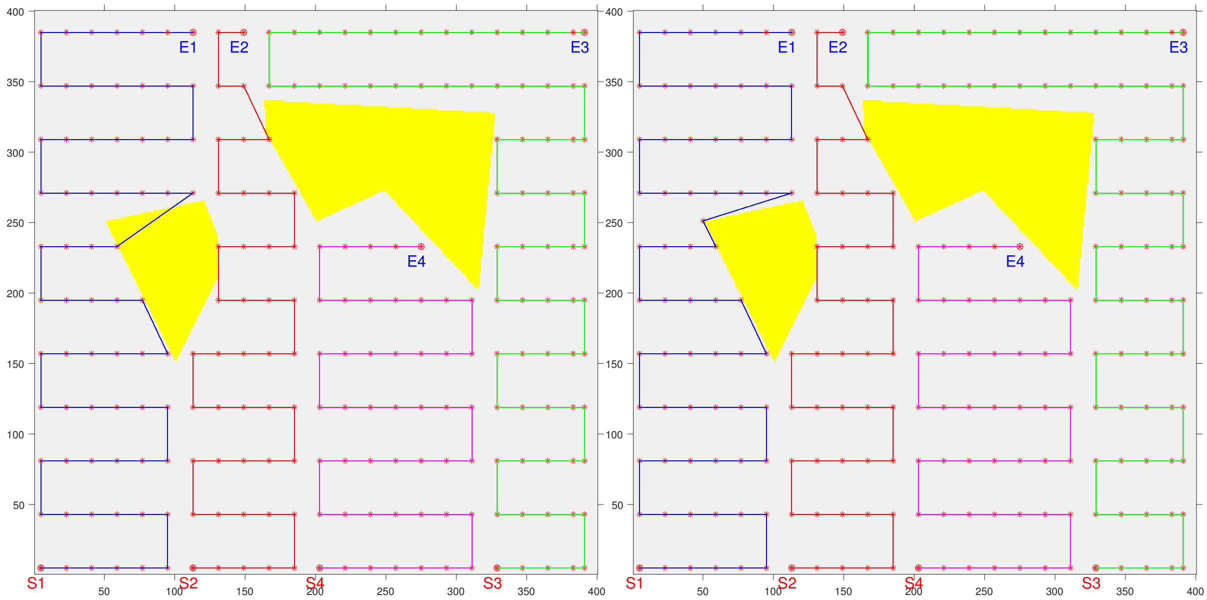


FIGURE 6.7: Scenario 1:
Infected area detection Image frames in obstacles free



(a) Boustrophedon path
contains collision with obstacles

(b) Boustrophedon path
without collision with obstacles

FIGURE 6.8: Scenario 1:
Infected area detection Boustrophedon path in cells

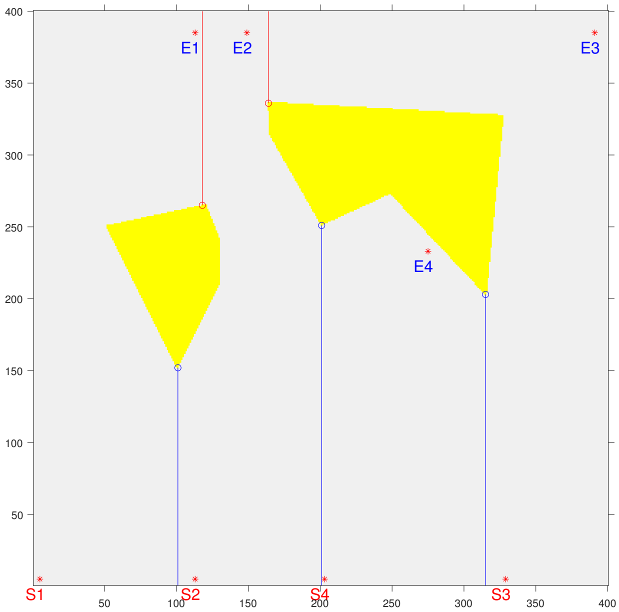


FIGURE 6.9: Scenario 1:
Infected area detection Start Stop Points in cells

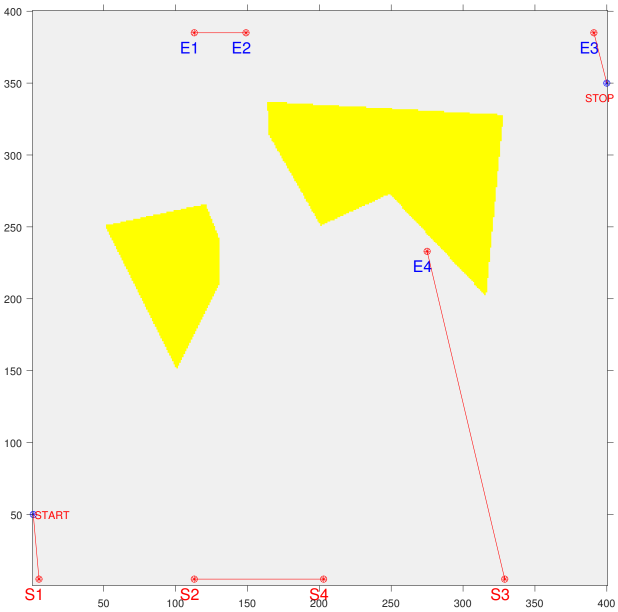


FIGURE 6.10: Scenario 1:
IA detection paths for changing cells using PRM TSP

Figure 6.8a shows that there exists one path that collides the obstacle. Consequently, the PRM should be applied for the two points that form the segment that collides obstacle to find the shortest obstacle-free path between them.

The final generated boustrophedon trajectory is shown in Figure 6.8b, we see that in the Boustrophedon trajectory generated on each cell, there are two points S_i *and* E_i *with* $i = 1, 2, 3, 4$. *These two points are the points where UAV enters or exits the cell. The entering and existing points* S_i *and* E_i *in cells are shown in Figure 6.9.*

Using the improved TSP algorithm detailed in the subsection 4.3.2.4, we can find the trajectory to change between cells so that the travel distance of the UAV is the smallest as shown in the Figure 6.10.

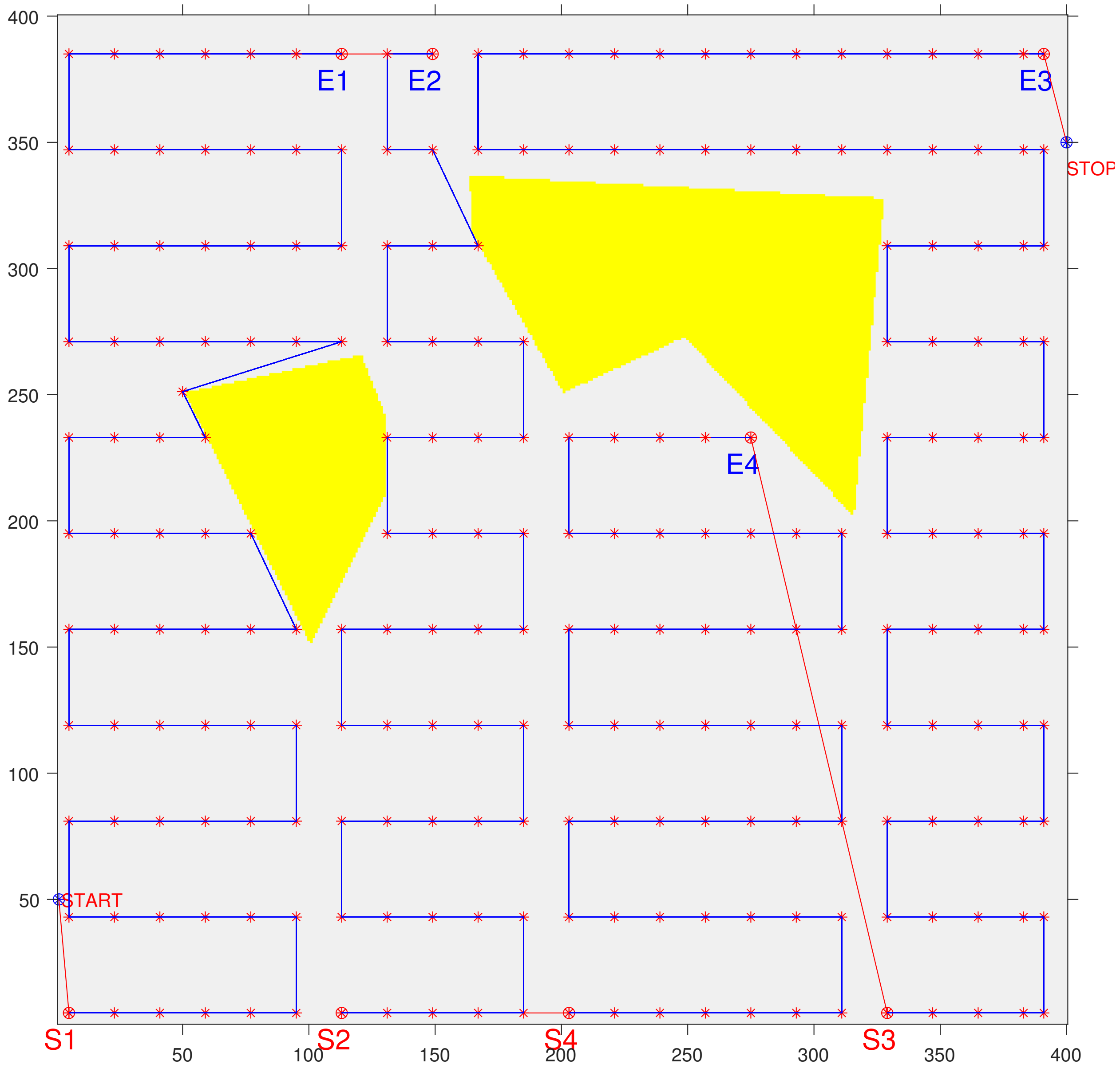


FIGURE 6.11: Scenario 1:
IA detection full path

Therefore, the moving trajectory of UAV is obtained as follows (as depicted on Figures 6.10 and 6.11): UAV starts at point $START$*, moves to point* S_1*, moves along the Boustrophedon trajectory in cell 1 to point* E_1*, moves to point* E_2*, moves along the Boustrophedon trajectory in cell 2 to point* S_2*, moves to point* S_4*, moves along the Boustrophedon trajectory in cell 4 to point* E_4*, moves to point* S_4*, moves along the Boustrophedon trajectory in cell 3 to point* E_E*, moves to point* $STOP$*.*

■

6.1.2 Coverage Path Planning (CPP) for crops spraying

Scenario 2. *In this section, we perform a simulation to apply pesticides to the entire pest-affected crop area identified in section 6.1.1.*

Assume agricultural area with fault and concave obstacles (yellow polygons) and pest-infested crop areas (red spots) as shown in the Figure 6.12. The coordinates of the obstacles and the crop areas affected by pests are known, thanks to the previous study.

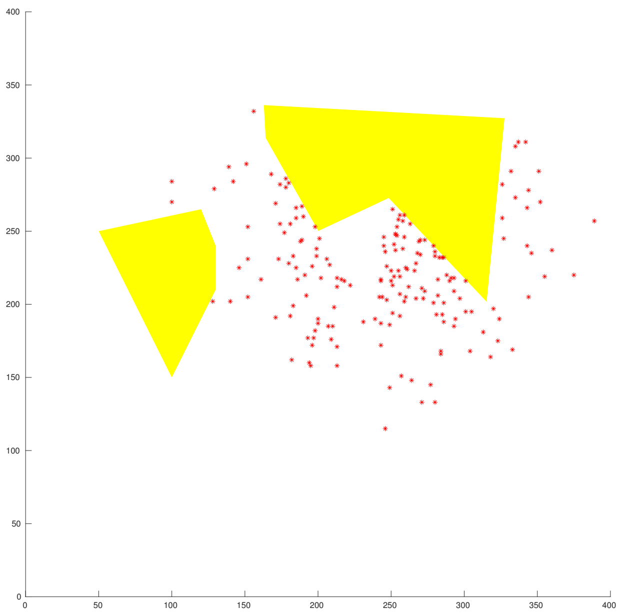


Figure 6.12: Scenario 2: Scenario 1 IA spraying obstacles and infected points

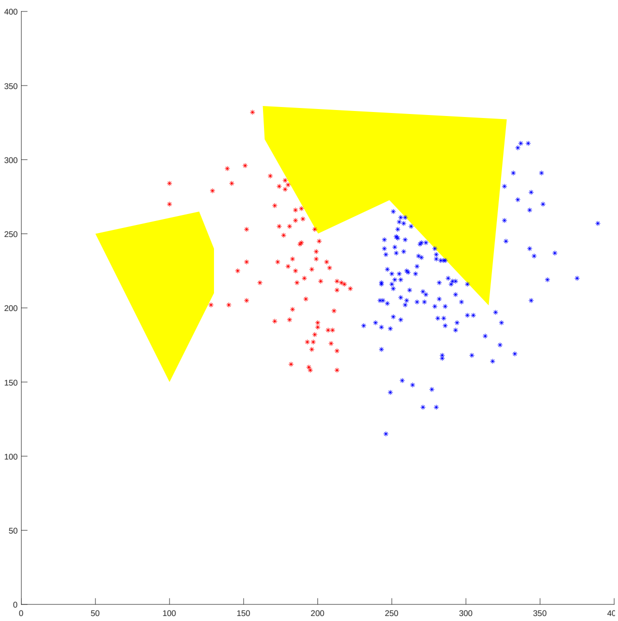


Figure 6.13: Scenario 2: IA spraying minimal Divide infected areas to classes

First, the agricultural areas affected by pests and diseases will be grouped into different classes using clustering as shown in Figure 6.13. In this simulation, the number of selected classes is 2. Therefore we see that the pests and diseases are divided into two groups (the dots have different colors in Figure 6.13).

As discussed on 4.4.2, there are two methods to create polygons that cover each class of infected crop points: minimal convex polygon (Figure 6.14) and boundary polygon (Figure 6.15). In the case boundary polygon, this is the smallest polygon which covers all given points. This polygon can be convex or concave. Meanwhile, in the case of minimal convex polygon, this is the smallest convex polygon containing all given points.

However, as shown in Figures 6.14 and 6.15, we see that these two polygons partially cover the obstacles. Therefore, in order to create an obstacle-free area, we can create new polygons that cover all the pests and diseases and obstacle-free as in Figures 6.16 and 6.17.

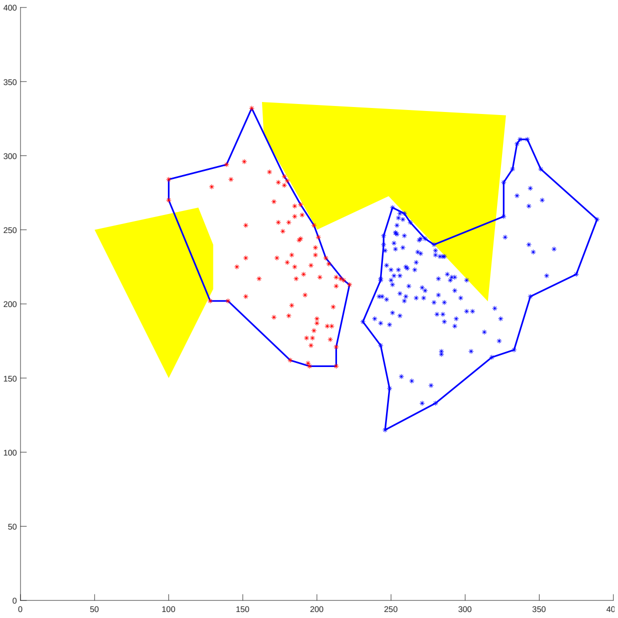


FIGURE 6.14: Scenario 2:
IA spraying minimal convexy polygons vs obstacles

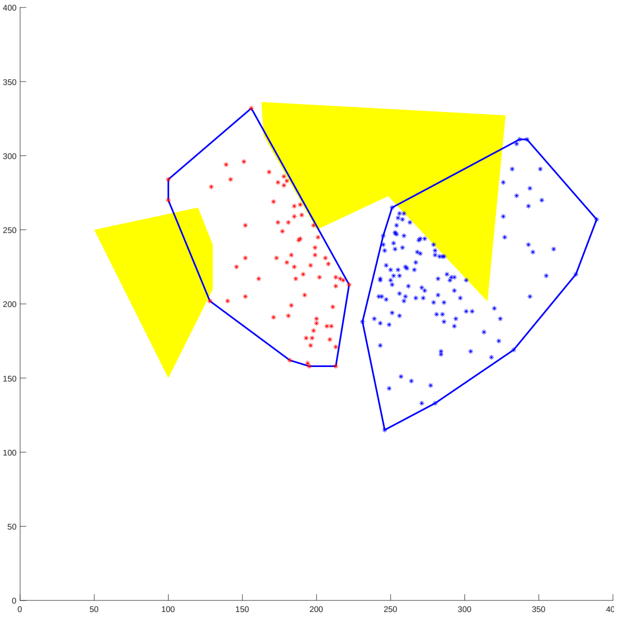


FIGURE 6.15: Scenario 2:
IA spraying boundary polygons vs obstacles

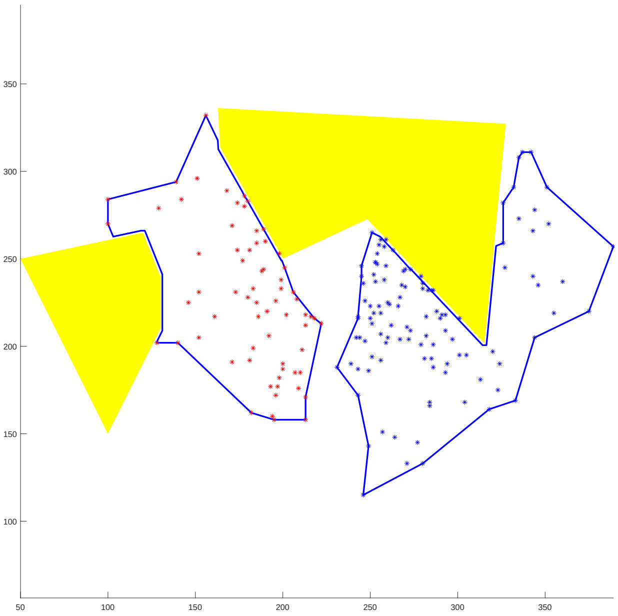


FIGURE 6.16: Scenario 2:
IA spraying obstacle free minimal convex polygons vs obstacles

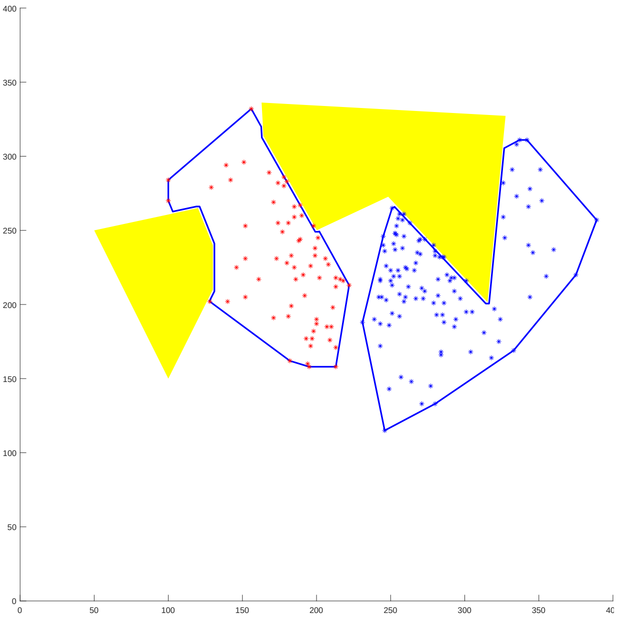


FIGURE 6.17: Scenario 2:
IA spraying obstacle free boundary polygons vs obstacles

Next, the resulting polygons in Figures 6.16 and 6.17 are divided into the minimal number of convex polygons as in Figures 6.18 and 6.19.

Remark 6.1. *For the purpose of clarity, the red and blue dots (infected areas) will not be shown in Figure 6.18 - Figure 6.24.*

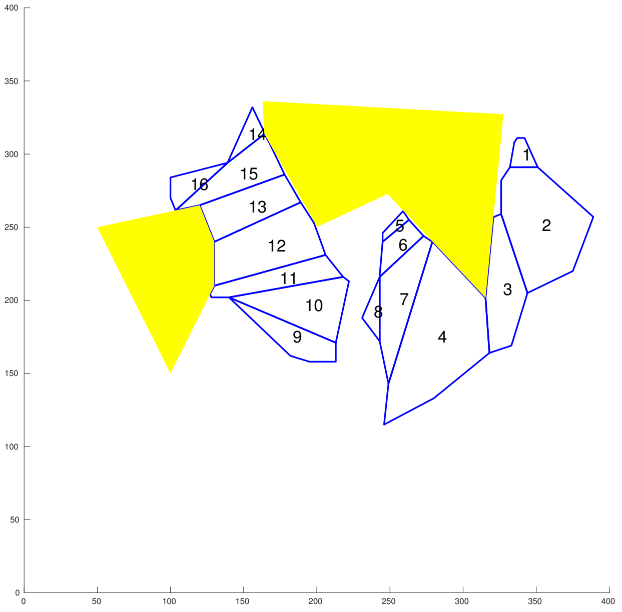


FIGURE 6.18: Scenario 2:
IA spraying minimal convex decomposition

FIGURE 6.19: Scenario 2:
IA spraying boundary polygon

Using the proposed algorithm for trajectory generation in convex polygon in 4.4.3, the boustrophedon trajectories are generated on each convex generated polygons as in Figures 6.20 and 6.21.

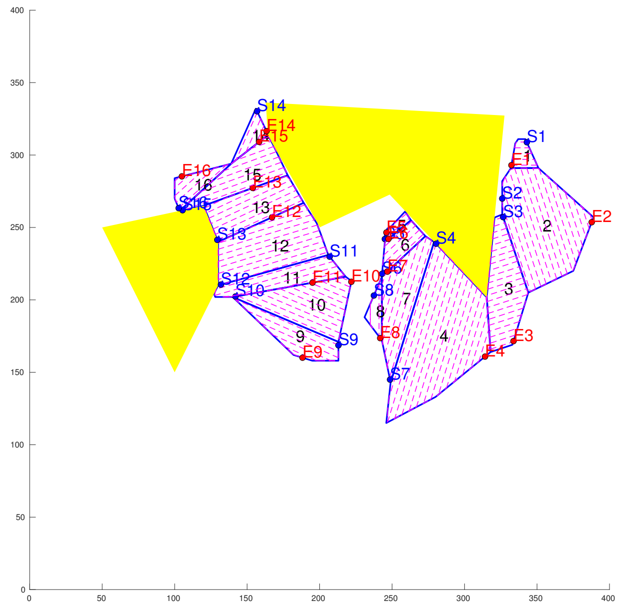


FIGURE 6.20: Scenario 2:
IA spraying minimal convex decomposition path

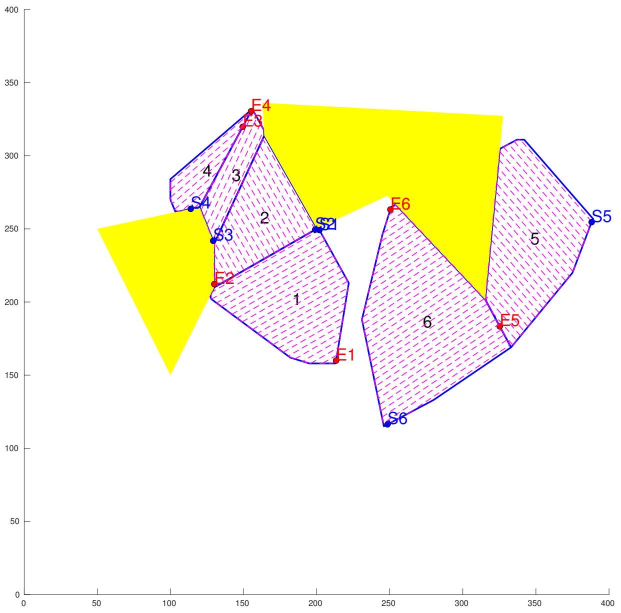


FIGURE 6.21: Scenario 2:
IA spraying boundary polygon path

Also in Figures 6.20 and 6.21, we see that in the Boustrophedon trajectory generated on each convex polygons, there are two points S_i and E_i with $i = 1, \ldots$. These two points are the points where UAV enters or exits the convex polygon. The entering and exiting points S_i and E_i in cells are shown in Figures 6.22 and 6.23.

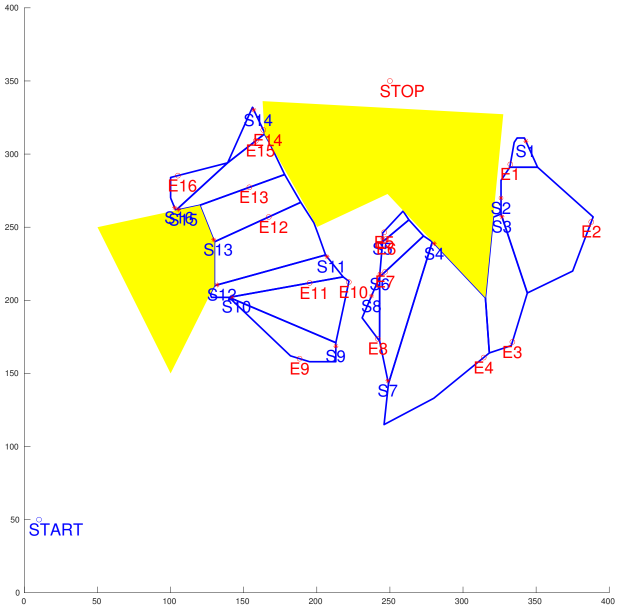


Figure 6.22: Scenario 2:
IA spraying minimal convex decomposition START STOP points

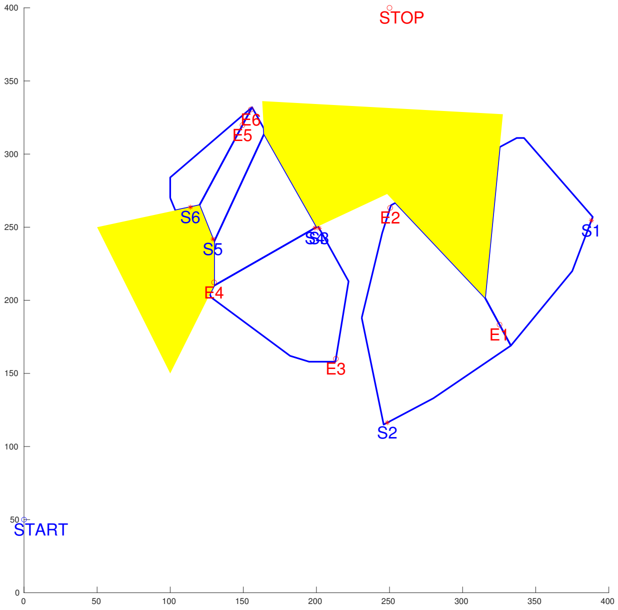


Figure 6.23: Scenario 2:
IA spraying boundary decomposition START STOP points

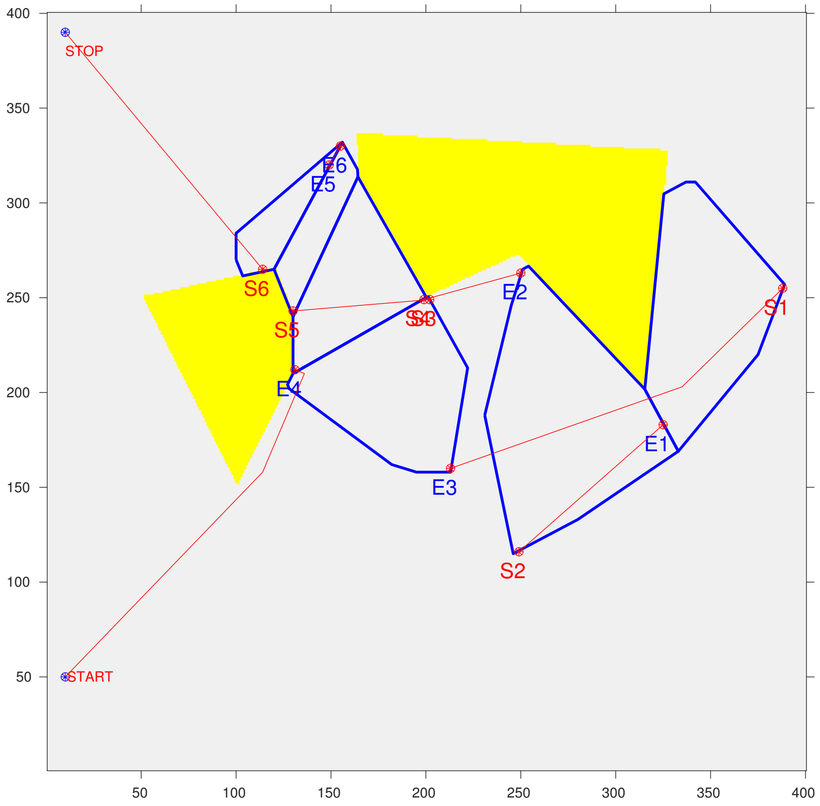


Figure 6.24: Scenario 2:
IA spraying paths for changing cells using prm tsp

Therefore, the moving trajectory of UAV is obtained as follows (as depicted on Figures 6.24: UAV starts at point START, moves to point E_4, moves along the Boustrophedon trajectory *in cell 4 to point* S_4*, moves to point* E_2*, moves along the Boustrophedon trajectory in cell 2 to point* S_2*, moves to point* E_1*, moves along the Boustrophedon trajectory in cell 1 to point* S_1*, moves to point* E_3*, moves along the Boustrophedon trajectory in cell 3 to point* S_3*, moves to point* S_5*, moves along the Boustrophedon trajectory in cell 5 to point* E_5*, moves to point* E_6*, moves along the Boustrophedon trajectory in cell 6 to point* S_6*, moves to point* $STOP$.

■

6.2 Quadrotor Control simulations

6.2.1 Quadrotor stabilization

6.2.1.1 Linear Parameter Varying (LPV) H_∞ state feedback controller

In this section, we simulate the reference tracking of the LPV H_∞ state feedback altitude/attitude controller for a mass-varying quadcopter under disturbances. Due to the variation of the mass, the moments of inertia with respect to the three axes x, y, and z are also changing. The mass can be changed abruptly or gradually.

- The disturbances d_φ, d_θ, d_ψ, and d_z to roll φ, pitch θ, yaw ψ, and altitude z of the quadcopter can be sinus, random, step, square wave, or constant.
- The reference signals for roll φ, pitch θ, yaw ψ, and altitude z can be of the form sinus, random, step, square wave, or constant.

The configuration of the LPV H_∞ Altitude/Attitude State Feedback Controller for mass-varying quadcopter is shown in Figure 6.25.

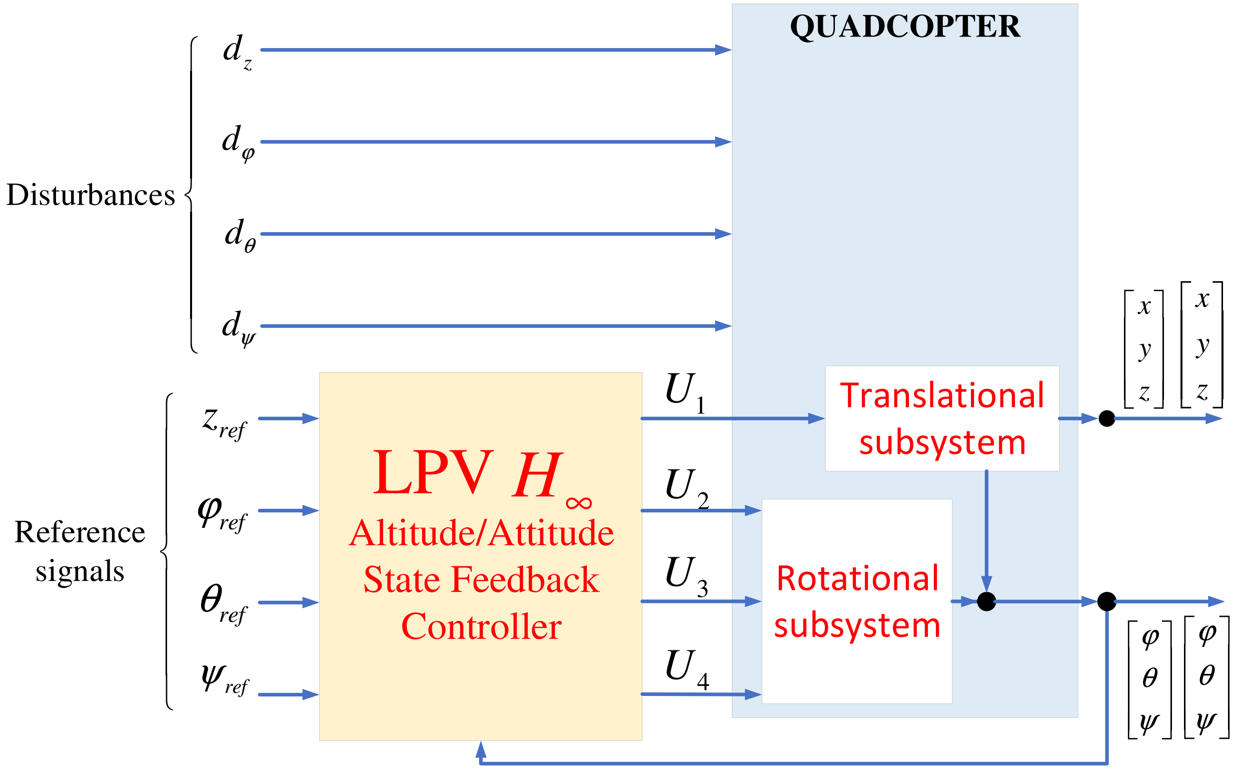


FIGURE 6.25: LPV H_∞ Altitude/Attitude State Feedback Controller for mass-varying quadcopter configuration

Scenario 3. *In this simulation:*

- *Simulation time is* $30\,(s)$
- *The mass is abruptly changes at* $5\,(s)$ *from* $2\,(kg)$ *to* $1.65\,(kg)$, *at* $10\,(s)$ *from* $1.65\,(kg)$ *to* $1.3\,(kg)$, *at* $15\,(s)$ *from* $1.3\,(kg)$ *to* $1.18\,(kg)$, *and at* $25\,(s)$ *from* $1.18\,(kg)$ *to* $1.12\,(kg)$. *Consequently, the moments of inertia also change abruptly at* $5\,(s)$, $10\,(s)$, $15\,(s)$, $25\,(s)$ *as shown in Figure 6.26.*
- *The reference signals* φ_{ref}, θ_{ref}, ψ_{ref}, *and* z_{ref} *(the first sub-figure of Figure 6.28) are:*
  - φ_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$, *max amplitude is* $30\,(deg)$, *min amplitude is* $0\,(deg)$
  - θ_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$, *max amplitude is* $29\,(deg)$, *min amplitude is* $0\,(deg)$
  - ψ_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$, *max amplitude is* $28\,(deg)$, *min amplitude is* $0\,(deg)$
  - z_{ref} *is a square wave signal of frequency* $1/5\,(Hz)$, *max amplitude is* $1.5\,(m)$, *min amplitude is* $1\,(m)$
- *The disturbances* d_φ, d_θ, d_ψ, *and* d_z *(the last four sub-figures of Figure 6.28 are:*
  - d_φ *is a random signal, max amplitude is* $2.1\,(Nm)$, *min amplitude is* $-2.1\,(Nm)$
  - d_θ *is a random signal, max amplitude is* $2.1\,(Nm)$, *min amplitude is* $-2.1\,(Nm)$
  - d_ψ *is a random signal, max amplitude is* $2.1\,(Nm)$, *min amplitude is* $-2.1\,(Nm)$
  - d_z *is a square wave signal of frequency* $1/4\,(Hz)$, *max amplitude is* $6\,(N)$, *min amplitude is* $-6\,(N)$

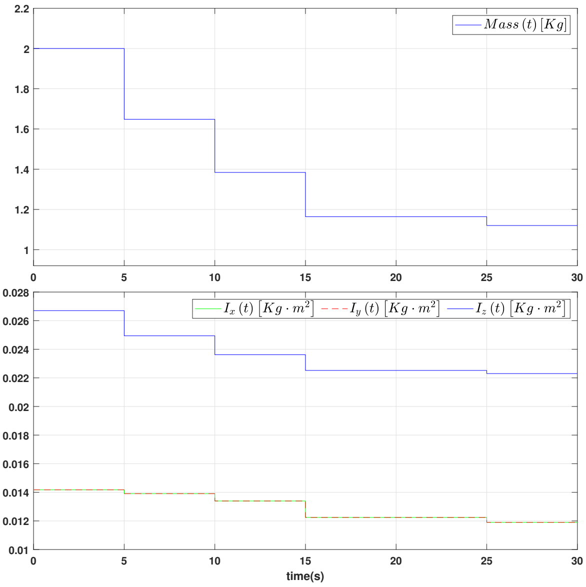


FIGURE 6.26: Scenario 3:
Mass, I_x, I_y, and I_z
LPV H_∞ Alt/Att State Feedback Controller

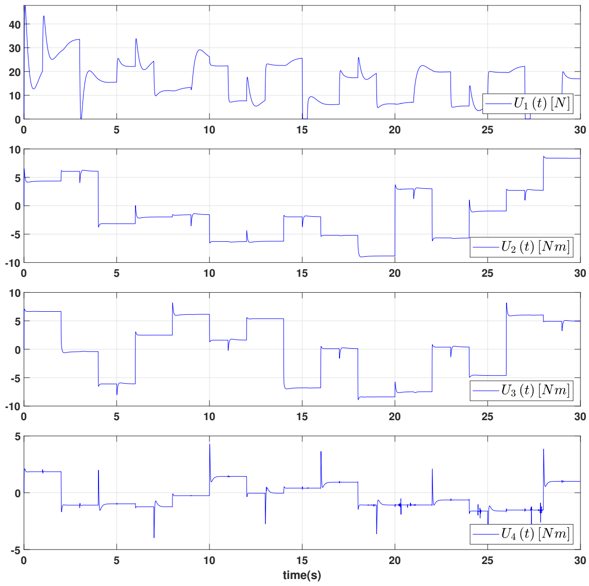


FIGURE 6.27: Scenario 3:
U_1, U_2, U_3, and U_4
LPV H_∞ Alt/Att State Feedback Controller

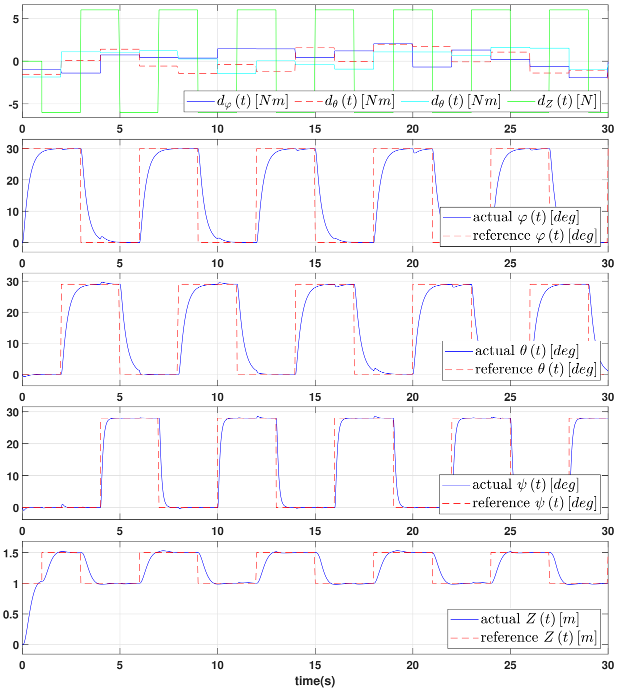


FIGURE 6.28: Scenario 3:
φ, θ, ψ, and z
LPV H_∞ Alt/Att State Feedback Controller

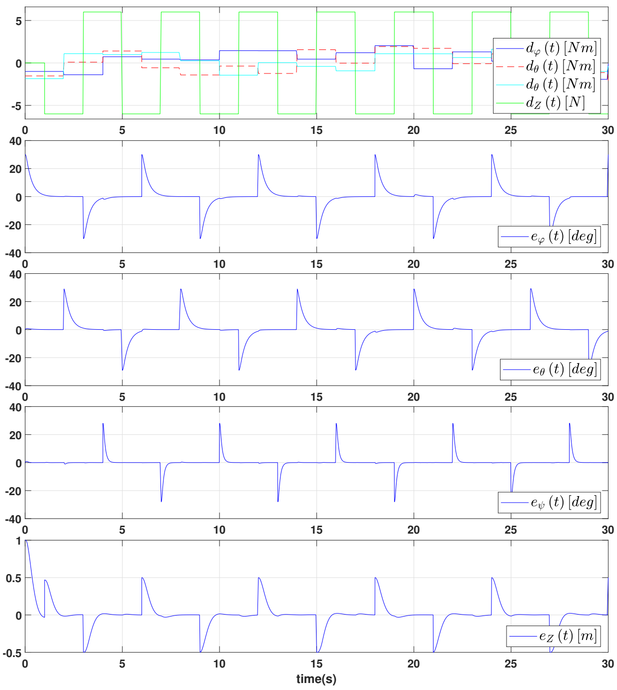


FIGURE 6.29: Scenario 3:
e_φ, e_θ, e_ψ, and e_z
LPV H_∞ Alt/Att State Feedback Controller

The last four sub-figures of the Figure 6.28 show the reference signals and responses of φ, θ, ψ, and z respectively.

We can see that φ, θ, ψ, and z track the reference signals well even under disturbances and the changes of mass and moments of inertia. The response time for φ, θ, ψ, and z are small.

The last four sub-figures of the Figure 6.29 show the differences between the actual signal and reference signals of φ, θ, ψ, and z. We can see that, these error signals tend to zero quickly.

The input signals U_1, U_2, U_3, and U_4 for the quadcopter are shown in Figure 6.27. ■

Scenario 4. *In this simulation:*

- *Simulation time is* $30\,(s)$
- *The mass gradually changes from* $2\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $5\,(s)$ *to* $25\,(s)$*. Consequently, the moments of inertia also gradually change over a period of time from* $5\,(s)$ *to* $25\,(s)$ *as shown in Figure 6.30.*
- *The reference signals* φ_{ref}*,* θ_{ref}*,* ψ_{ref}*, and* z_{ref} *(the first sub-figure of Figure 6.32) are:*
  - φ_{ref} *is a constant signal,* $\varphi_{ref} = 30\,(deg)$
  - θ_{ref} *is a step signal at* $4\,(s)$ *from* $0\,(deg)$ *to* $29\,(deg)$
  - ψ_{ref} *is a step signal at* $6\,(s)$ *from* $0\,(deg)$ *to* $28\,(deg)$
  - z_{ref} *is a constant signal,* $z_{ref} = 1\,(m)$
- *The disturbances* d_φ*,* d_θ*,* d_ψ*, and* d_z *(the last four sub-figures of Figure 6.32 are:*
  - d_φ *is a sine signal of frequency* $1/6\,(Hz)$*, maximum amplitude is* $2.1\,(Nm)$*, minimum amplitude is* $0\,(Nm)$
  - d_θ *is a sine signal of frequency* $1/6\,(Hz)$*, maximum amplitude is* $2.1\,(Nm)$*, minimum amplitude is* $0\,(Nm)$
  - d_ψ *is a sine signal of frequency* $1/6\,(Hz)$*, maximum amplitude is* $2.1\,(Nm)$*, minimum amplitude is* $0\,(Nm)$
  - d_z *is a random signal, maximum amplitude is* $3\,(N)$*, minimum amplitude is* $-6\,(N)$

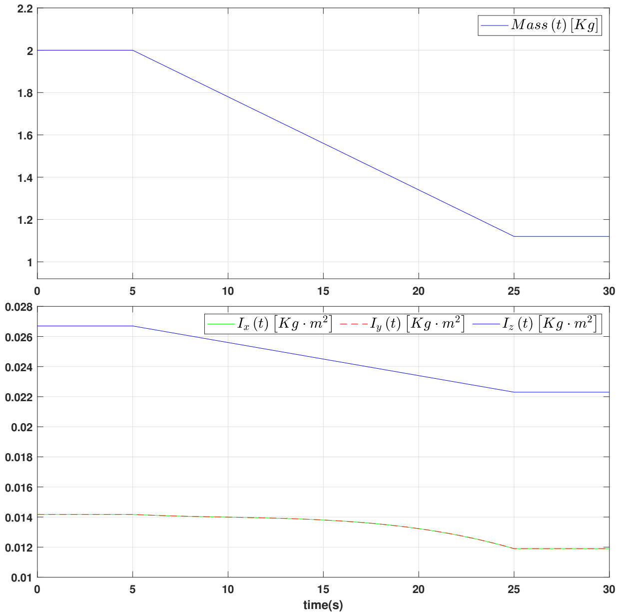


FIGURE 6.30: Scenario 4:
Mass, I_x, I_y, and I_z
LPV H_∞ Alt/Att State Feedback Controller

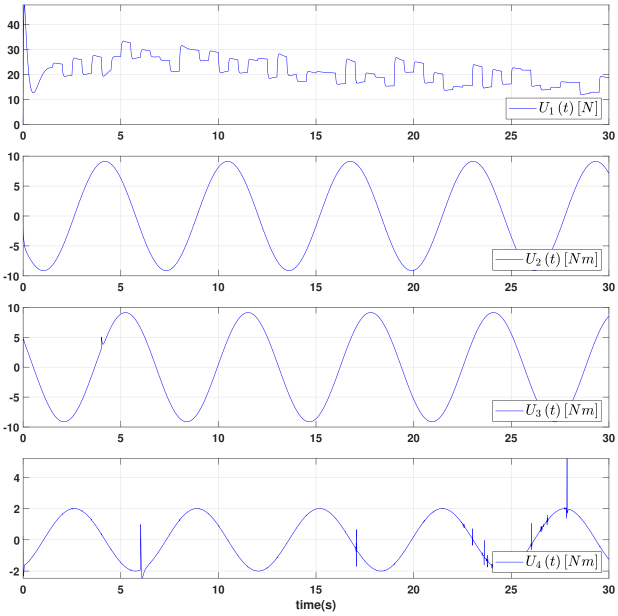


FIGURE 6.31: Scenario 4:
U_1, U_2, U_3, and U_4
LPV H_∞ Alt/Att State Feedback Controller

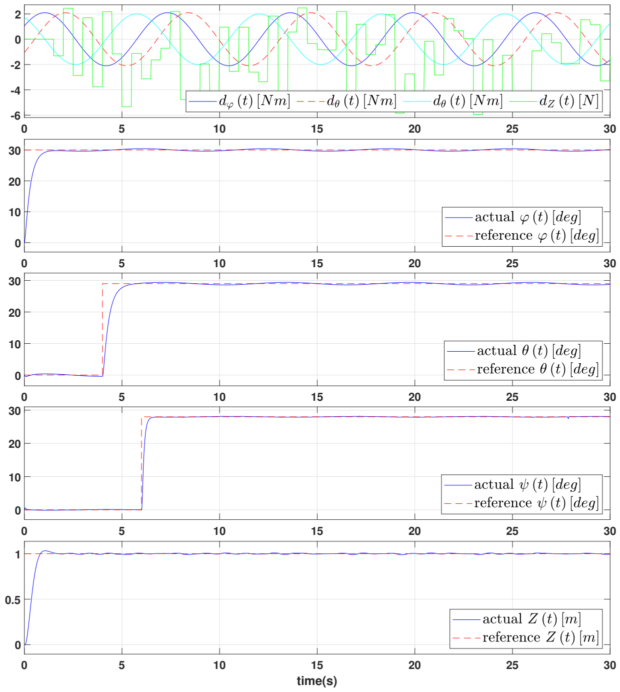


FIGURE 6.32: Scenario 4:
φ, θ, ψ, and z
LPV H_∞ Alt/Att State Feedback Controller

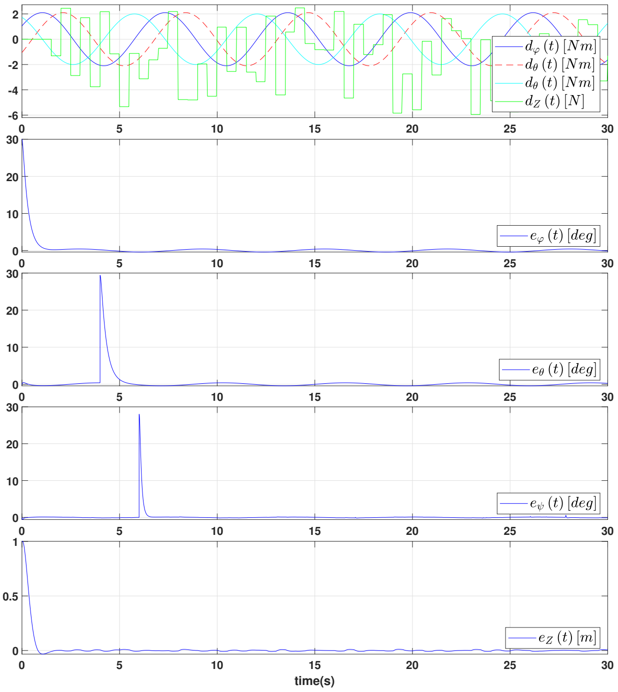


FIGURE 6.33: Scenario 4:
e_φ, e_θ, e_ψ, and e_z
LPV H_∞ Alt/Att State Feedback Controller

The last four sub-figures of the Figure 6.32 show the reference signals and responses of φ, θ, ψ, *and* z *respectively.*

We can see that φ, θ, ψ, *and* z *track the reference signals well even under disturbances and the changes of mass and moments of inertia. The response time for* φ, θ, ψ, *and* z *are small.*

The last four sub-figures of the Figure 6.33 show the differences between the actual signal and reference signals of φ, θ, ψ, *and* z. *We can see that, these error signals tend to zero quickly.*

The input signals U_1, U_2, U_3, *and* U_4 *for the quadcopter are shown in Figure 6.31.* ■

6.2.2 Quadrotor Unknown Input Observer (UIO)

To evaluate the effectiveness of the proposed Linear Parameter Varying (LPV) Unknown Input Observer (UIO), we conduct two types of simulation in this section. The first type of simulation concerns the UIO for the system without measurement noises in subsection 6.2.2.1. While the second one concerns the UIO for the system with measurement noises in subsection 6.2.2.2. The simulation results suggest that the proposed UIO works well for various reference signals (impulse, random, constant, and sine), several types of disturbances (impulse, random, constant, and sine), and with or without measurement noises.

6.2.2.1 Quadrotor Unknown Input Observer (UIO) without measurement noise

In this section, we simulate the Unknown Input Observer (UIO) for altitude/attitude of a mass-varying quadcopter under disturbances. Due to the variation of the mass, the moments of inertia with respect to the three axes x, y, and z are also changing. The mass can be changed abruptly or gradually.

- The disturbances d_φ, d_θ, d_ψ, and d_z to roll φ, pitch θ, yaw ψ, and altitude z of the quadcopter can be sinus, random, step, square wave, or constant.
- The reference signals for roll φ, pitch θ, yaw ψ, and altitude z can be of the form sinus, random, step, square wave, or constant.

The configuration of the LPV UIO for Altitude/Attitude of a mass-varying quadcopter is shown in Figure 6.34. For controlling the Altitude/Attitude of the quadrotor, the LPV H_∞ state feedback controller designed in 5.6 is used. In this simulation, we do not consider the measurement noises, consequently, the measured values are directly the inputs to the LPV UIO block as shown in Figure 6.34.

For simplify, In the simulation, the derivatives of the outputs are calculated by generic differentiator. However, in the real cases, the time derivatives of the outputs can be corrupted by measurement noises. In this case, in order to get the derivatives of the outputs which are less sensitive to the measurement noises, sliding mode or algebraic differentiators can be applied.

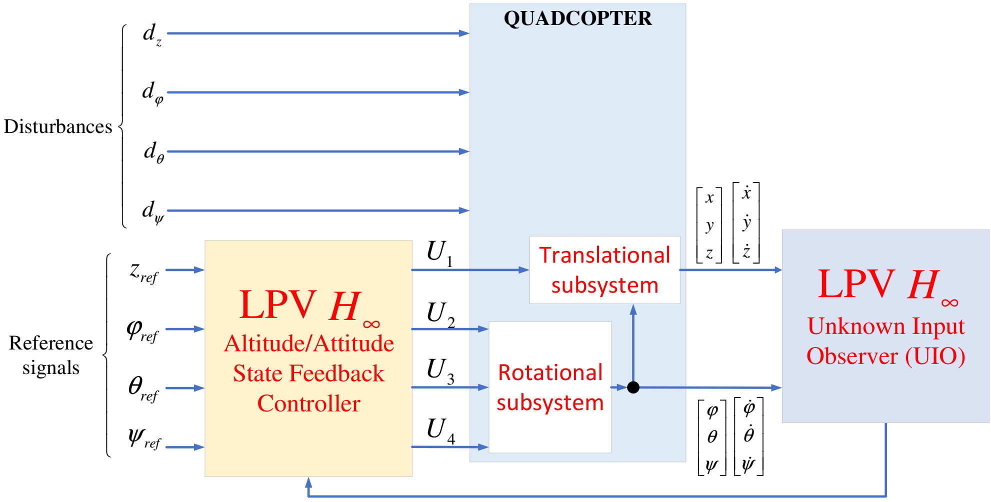


FIGURE 6.34: LPV H_∞ UIO for mass-varying quadcopter configuration

Scenario 5. *In this simulation:*

- *No measurement noises are considered*
- *Simulation time is* $30\,(s)$
- *The mass gradually changes from* $2\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $5\,(s)$ *to* $25\,(s)$*. Consequently, the moments of inertia also gradually change over a period of time from* $5\,(s)$ *to* $25\,(s)$ *as shown in Figure 6.35.*
- *The reference signals* φ_{ref}, θ_{ref}, ψ_{ref}, *and* z_{ref} *(Figure 6.38) are:*
  - φ_{ref} *is a step signal at* $2\,(s)$ *from* $0\,(deg)$ *to* $30\,(deg)$
  - θ_{ref} *is a step signal at* $4\,(s)$ *from* $0\,(deg)$ *to* $28\,(deg)$
  - ψ_{ref} *is a step signal at* $6\,(s)$ *from* $0\,(deg)$ *to* $26\,(deg)$
  - z_{ref} *is a constant signal,* $z_{ref} = 1\,(m)$
- *The disturbances* d_φ, d_θ, d_ψ, *and* d_z *(Figure 6.36) are:*
  - d_φ *is a sine wave signal of frequency* $1\,(rad/s)$*, amplitude is* $1.1\,(Nm)$
  - d_θ *is a sine wave signal of frequency* $1\,(rad/s)$*, amplitude is* $1.3\,(Nm)$
  - d_ψ *is a sine wave signal of frequency* $1\,(rad/s)$*, amplitude is* $1.5\,(Nm)$
  - d_z *is a random signal, maximum amplitude is* $6\,(N)$ *and , minimum amplitude is* $-30\,(N)$

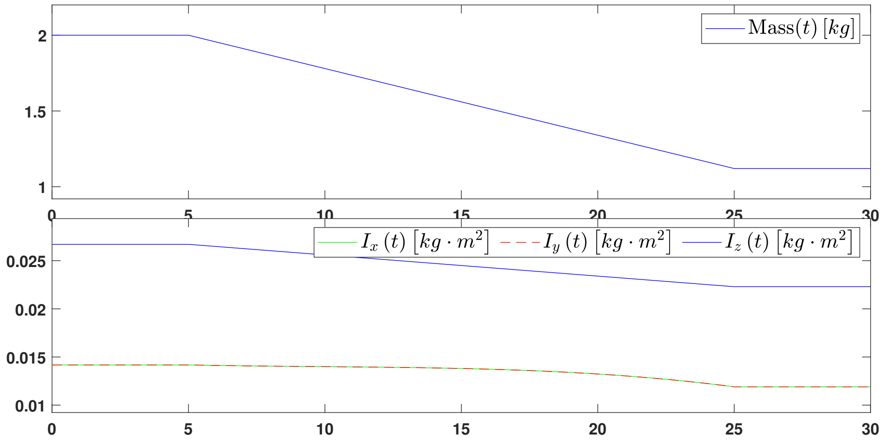


FIGURE 6.35: Scenario 5:
Mass, I_x, I_y, and I_z
LPV UIO for Alt/Att

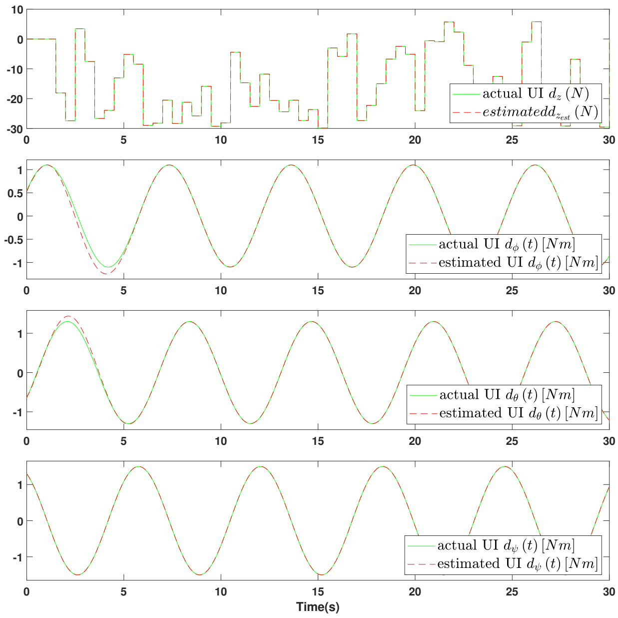


FIGURE 6.36: Scenario 5: UI and their estimation with LPV UIO for Alt/Att

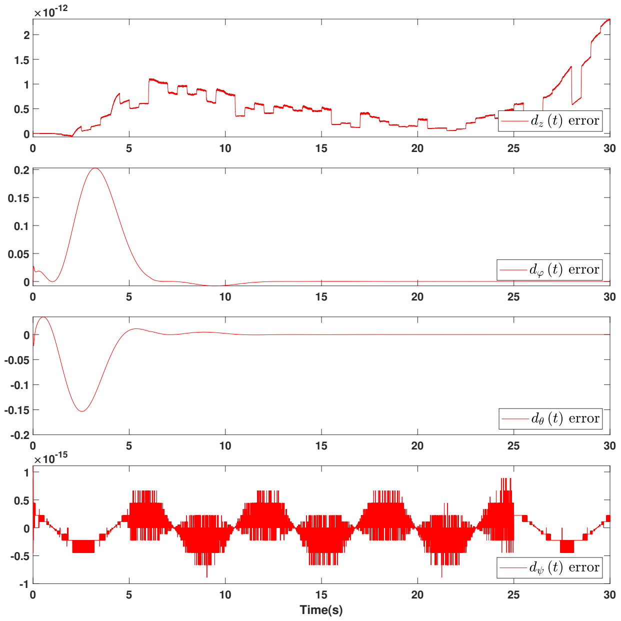


FIGURE 6.37: Scenario 5: UI estimations errors with LPV UIO for Alt/Att

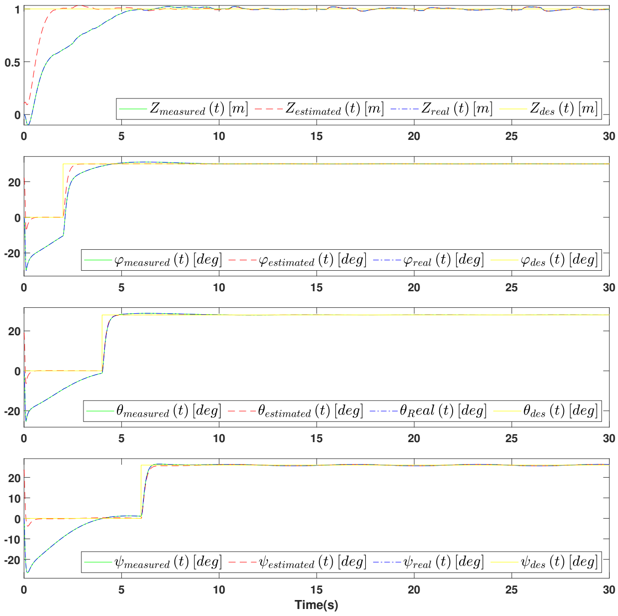


FIGURE 6.38: Scenario 5: Z, ϕ, θ, and ψ and estimations LPV UIO for Alt/Att

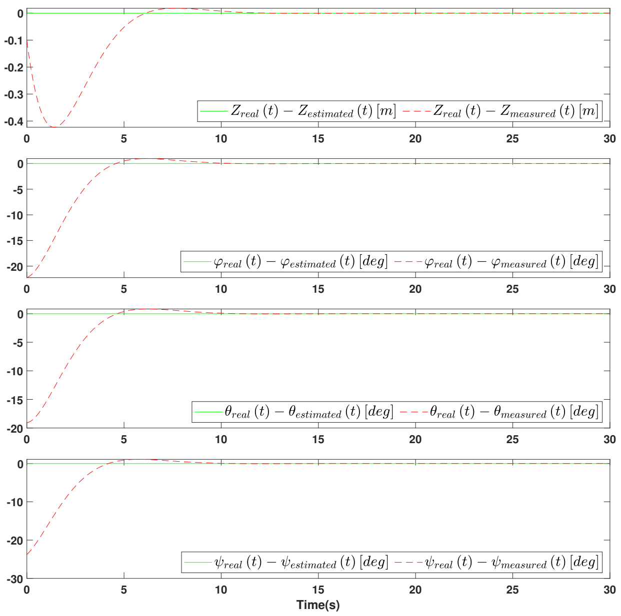


FIGURE 6.39: Scenario 5: Differences between (real and estimated) vs (real and measured) values LPV UIO for Alt/Att

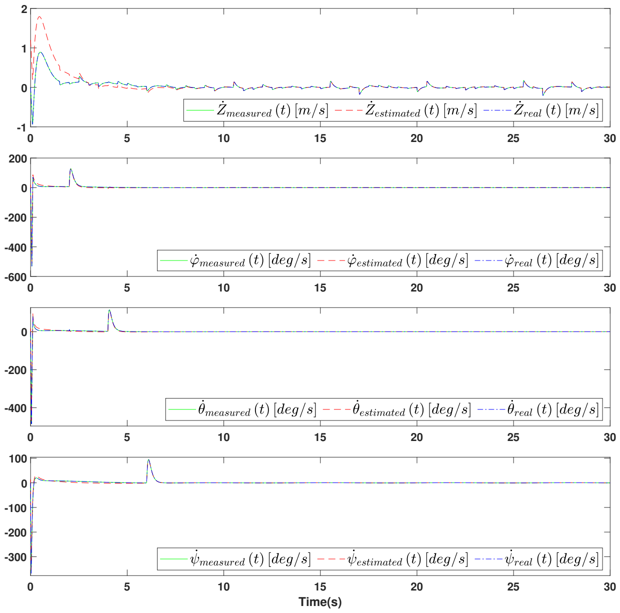


FIGURE 6.40: Scenario 5: $\dot{Z}$, $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ and estimations LPV UIO for Alt/Att

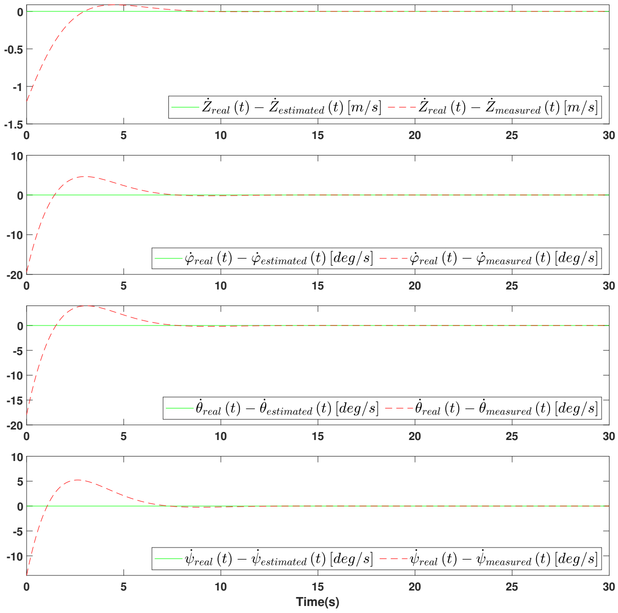


FIGURE 6.41: Scenario 5: Differences between (real and estimated) vs (real and measured) values LPV UIO for Alt/Att

The actual unknown inputs d_z, d_φ, d_θ, d_ψ *and their estimated values are shown in Figure 6.36 and Figure 6.37 shows the differences between actual unknown inputs and estimated unknown inputs. We can see that the estimated unknown inputs tend to actual unknown inputs in about* $5\,(s)$.

In Figure 6.38 are shown the desired, actual, and estimated value of φ, θ, ψ, *and* z. *We can see that* φ, θ, ψ, *and* z *track the reference signals well (as in Figure 6.38) even under disturbances and the changes of mass and moments of inertia. The response time for* φ, θ, ψ, *and* z *are small.*

The actual, and estimated value of $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, *and* $\dot{z}$ *are shown in Figure 6.40, while Figure 6.41 shows the difference between the actual and estimated values of* $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, *and* $\dot{z}$. ■

Scenario 6. *In this simulation:*

- *No measurement noise is considered*
- *Simulation time is* $30\,(s)$
- *The mass abruptly changes at* $5\,(s)$ *from* $2\,(kg)$ *to* $1.65\,(kg)$, *at* $10\,(s)$ *from* $1.65\,(kg)$ *to* $1.3\,(kg)$, *at* $15\,(s)$ *from* $1.3\,(kg)$ *to* $1.18\,(kg)$, *and at* $25\,(s)$ *from* $1.18\,(kg)$ *to* $1.12\,(kg)$. *Consequently, the moments of inertia also change abruptly at* $5\,(s)$, $10\,(s)$, $15\,(s)$, $25\,(s)$ *as shown in Figure 6.42.*
- *The reference signals* φ_{ref}, θ_{ref}, ψ_{ref}, *and* z_{ref} *(Figure 6.45) are:*
  - φ_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$, *maximum amplitude is* $30\,(deg)$, *minimum amplitude is* $0\,(deg)$
  - θ_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$, *maximum amplitude is* $28\,(deg)$, *minimum amplitude is* $0\,(deg)$
  - ψ_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$, *maximum amplitude is* $26\,(deg)$, *minimum amplitude is* $0\,(deg)$
  - z_{ref} *is a constant signal,* $z_{ref} = 1\,(m)$
- *The disturbances* d_φ, d_θ, d_ψ, *and* d_z *(Figure 6.43) are:*
  - d_φ *is a random signal, maximum amplitude is* $1.1\,(Nm)$ *and minimum amplitude is* $-1.1\,(Nm)$
  - d_θ *is a random signal, maximum amplitude is* $1.3\,(Nm)$ *and minimum amplitude is* $-1.3\,(Nm)$
  - d_ψ *is a random signal, maximum amplitude is* $1.5\,(Nm)$ *and minimum amplitude is* $-1.5\,(Nm)$
  - d_z *is a random signal, maximum amplitude is* $6\,(N)$ *and minimum amplitude is* $-30\,(N)$

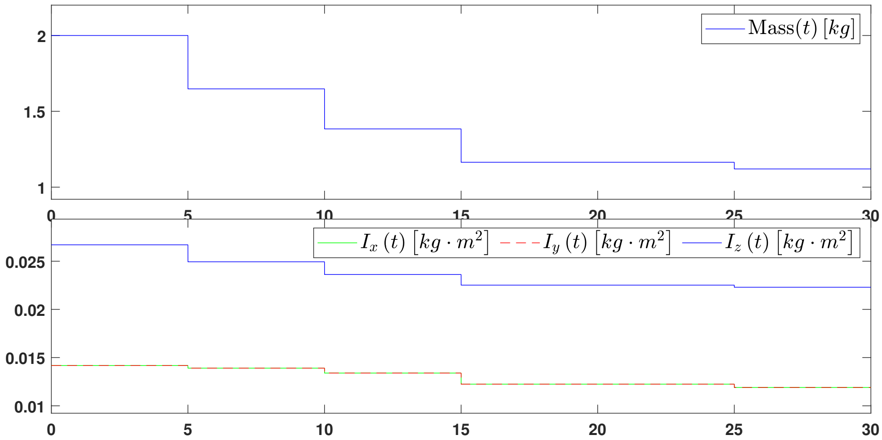


FIGURE 6.42: Scenario 6:
Mass, I_x, I_y, and I_z
LPV UIO for Alt/Att

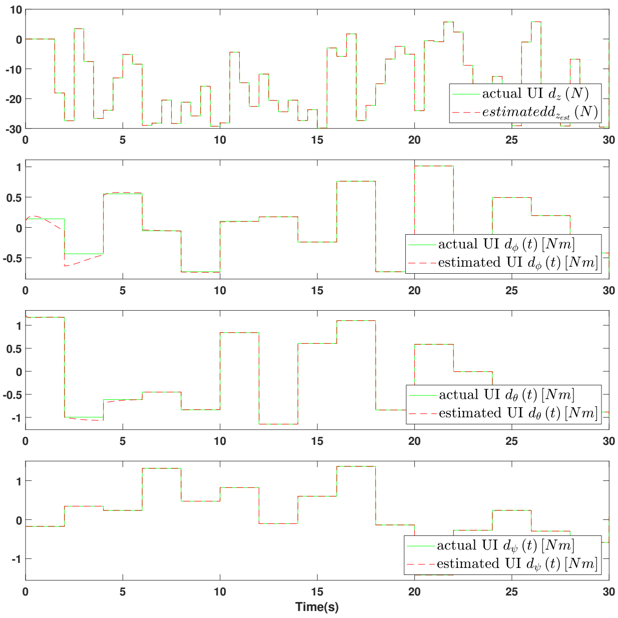


Figure 6.43: Scenario 6: UI and their estimation with LPV UIO for Alt/Att

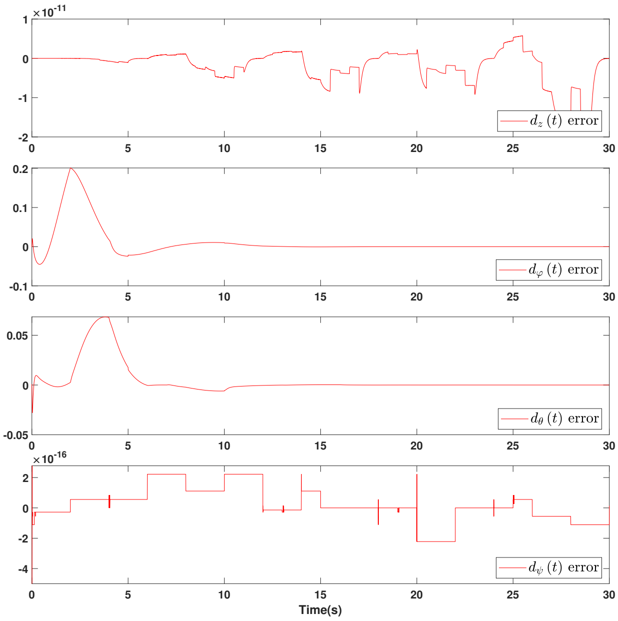


Figure 6.44: Scenario 6: UI estimations errors with LPV UIO for Alt/Att

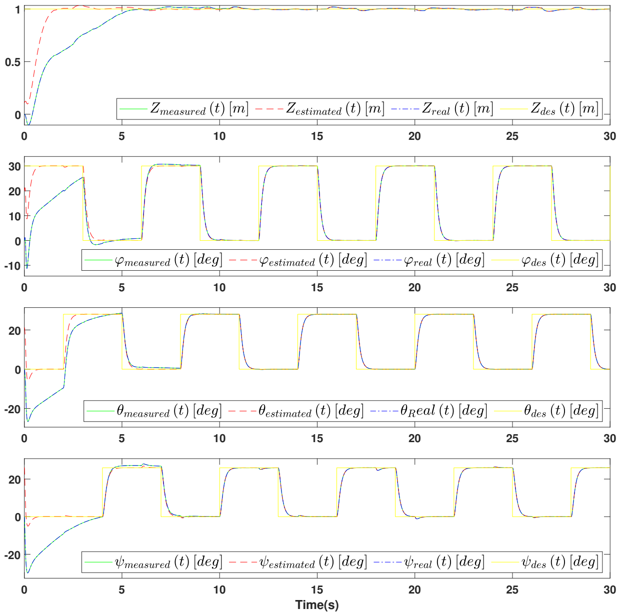


Figure 6.45: Scenario 6: Z, ϕ, θ, and ψ and estimations LPV UIO for Alt/Att

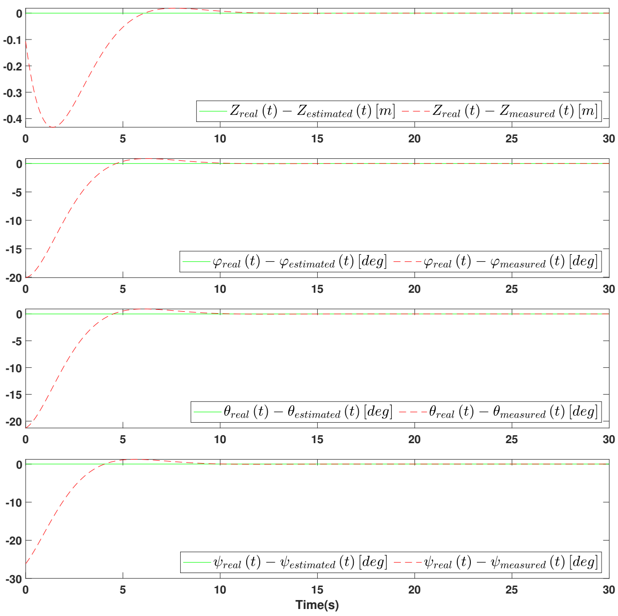


Figure 6.46: Scenario 6: Differences between (real and estimated) vs (real and measured) values LPV UIO for Alt/Att

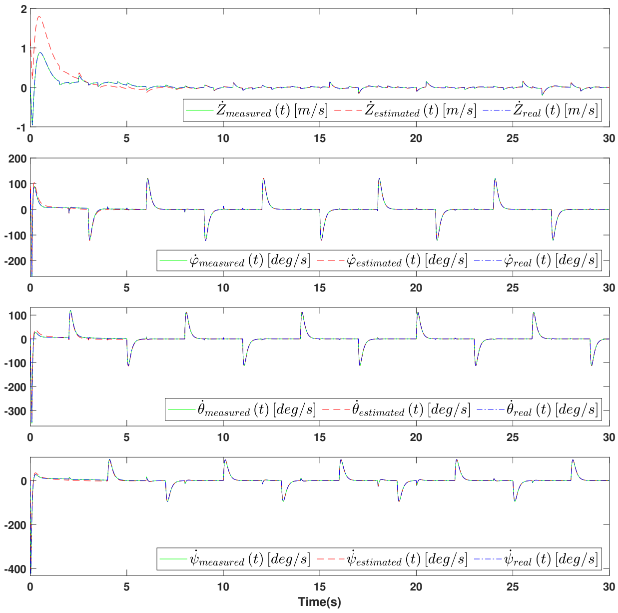


FIGURE 6.47: Scenario 6:
$\dot{Z}$, $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ and estimations
LPV UIO for Alt/Att

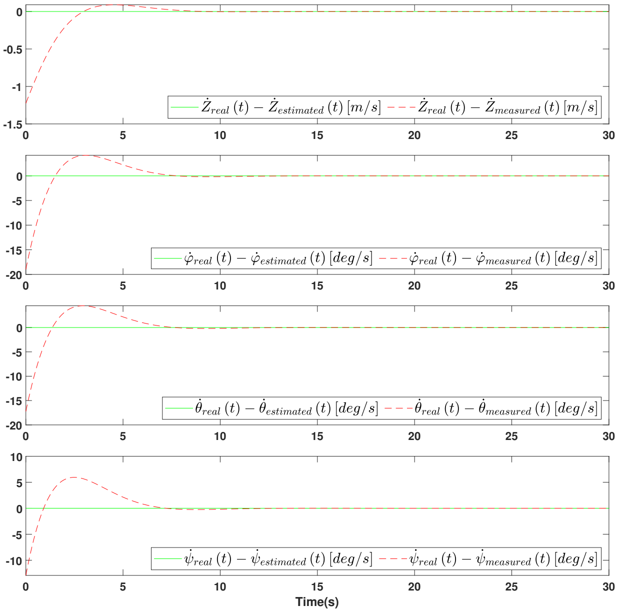


FIGURE 6.48: Scenario 6:
Differences between (real and estimated) vs (real and measured) values
LPV UIO for Alt/Att

The actual unknown inputs d_z, d_φ, d_θ, d_ψ and their estimated values are shown in Figure 6.43 and Figure 6.44 shows the differences between actual unknown inputs and estimated unknown inputs. We can see that the estimated unknown inputs tend to actual unknown inputs in about $5\,(s)$.

In Figure 6.45 are shown the desired, actual, and estimated value of φ, θ, ψ, and z. We can see that φ, θ, ψ, and z track the reference signals well (as in Figure 6.45) even under disturbances and the changes of mass and moments of inertia. The response time for φ, θ, ψ, and z are small.

The actual, and estimated value of $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, and $\dot{z}$ shown in Figure 6.47, while Figure 6.48 shows the difference between the actual and estimated values of $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, and $\dot{z}$. ■

6.2.2.2 Quadrotor Unknown Input Observer (UIO) with measurement noise

In this section, we simulate the Unknown Input Observer (UIO) for altitude/attitude of a mass-varying quadcopter under disturbances. Due to the variation of the mass, the moments of inertia with respect to the three axes x, y, and z are also changing. The mass can be changed abruptly or gradually.

- The disturbances d_φ, d_θ, d_ψ, and d_z to roll φ, pitch θ, yaw ψ, and altitude z of the quadcopter can be sinus, random, step, square wave, or constant.
- The reference signals for roll φ, pitch θ, yaw ψ, and altitude z can be of the form sinus, random, step, square wave, or constant.

The configuration of the LPV UIO for Altitude/Attitude of a mass-varying quadcopter is shown in Figure 6.49. For controlling the Altitude/Attitude of the quadrotor, the LPV H_∞ state feedback controller designed in 5.6 is used. In this simulation, we consider the measurement noises. The outputs of the quadcopter are measured by the Measurement block. The white noises are added to the outputs of the quadcopter here (in Figure 6.49), then the noisy outputs of the quadcopter (the outputs of the Measurement block) are the inputs to the LPV UIO block.

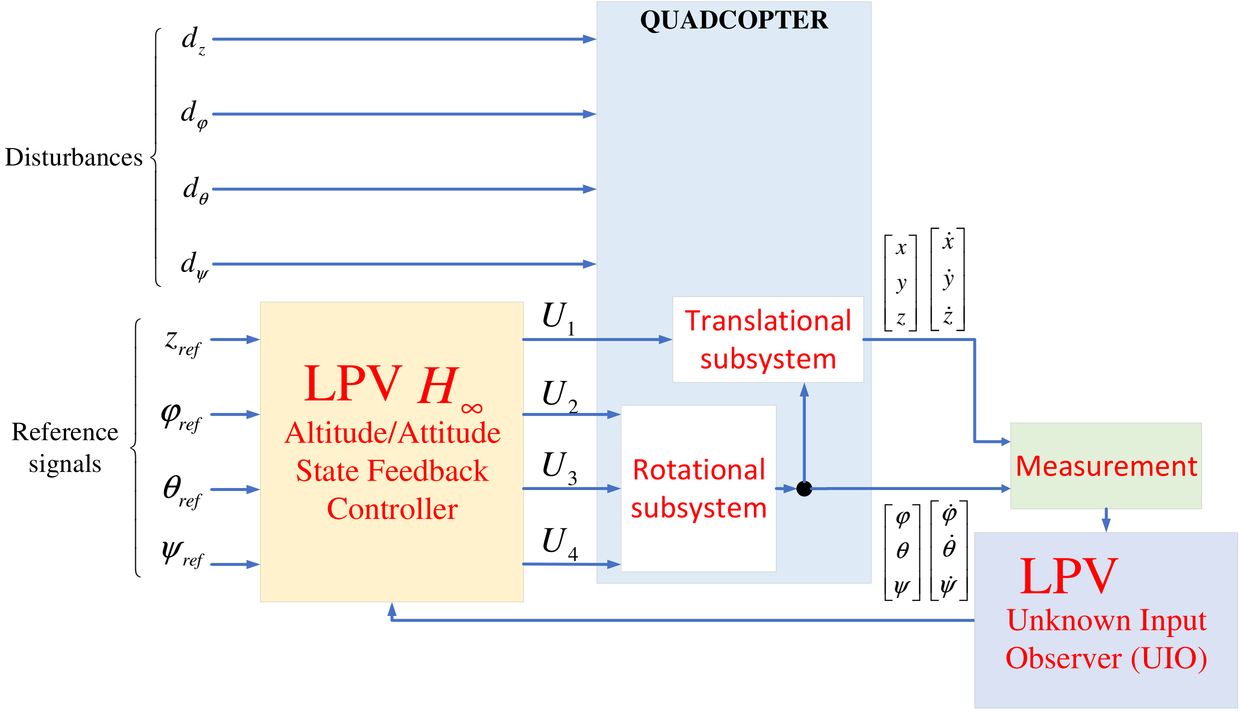


FIGURE 6.49: LPV H_∞ UIO for mass-varying quadcopter configuration

Scenario 7. *In this simulation:*

- *Measurement noises (random white noises) are considered*
- *Simulation time is* $30\,(s)$
- *The mass abruptly changes at* $5\,(s)$ *from* $2\,(kg)$ *to* $1.65\,(kg)$, *at* $10\,(s)$ *from* $1.65\,(kg)$ *to* $1.3\,(kg)$, *at* $15\,(s)$ *from* $1.3\,(kg)$ *to* $1.18\,(kg)$, *and at* $25\,(s)$ *from* $1.18\,(kg)$ *to* $1.12\,(kg)$. *Consequently, the moments of inertia also change abruptly at* $5\,(s)$, $10\,(s)$, $15\,(s)$, $25\,(s)$ *as shown in Figure 6.50.*
- *The reference signals* φ_{ref}, θ_{ref}, ψ_{ref}, *and* z_{ref} *(Figure 6.53) are:*
  - φ_{ref} *is a random signal, maximum amplitude is* $22\,(deg)$, *minimum amplitude is* $-40\,(deg)$
  - θ_{ref} *is a random signal, maximum amplitude is* $30\,(deg)$, *minimum amplitude is* $-22\,(deg)$
  - ψ_{ref} *is a random signal, maximum amplitude is* $48\,(deg)$, *minimum amplitude is* $-48\,(deg)$
  - z_{ref} *is a square wave signal of frequency* $\frac{1}{6}\,(Hz)$, *maximum amplitude is* $1.5\,(m)$, *minimum amplitude is* $1\,(m)$
- *The disturbances* d_φ, d_θ, d_ψ, *and* d_z *(Figure 6.51) are:*
  - d_φ *is a sinus signal of frequency* $\frac{1}{6}\,(Hz)$, *maximum amplitude is* $1.1\,(Nm)$ *and minimum amplitude is* $-1.1\,(Nm)$
  - d_θ *is a sinus signal, maximum amplitude is* $1.3\,(Nm)$ *and minimum amplitude is* $-1.3\,(Nm)$
  - d_ψ *is a sinus signal, maximum amplitude is* $1.5\,(Nm)$ *and minimum amplitude is* $-1.5\,(Nm)$
  - d_z *is a square wave signal of frequency* $\frac{1}{4}\,(Hz)$, *maximum amplitude is* $6\,(N)$ *and minimum amplitude is* $-40\,(N)$

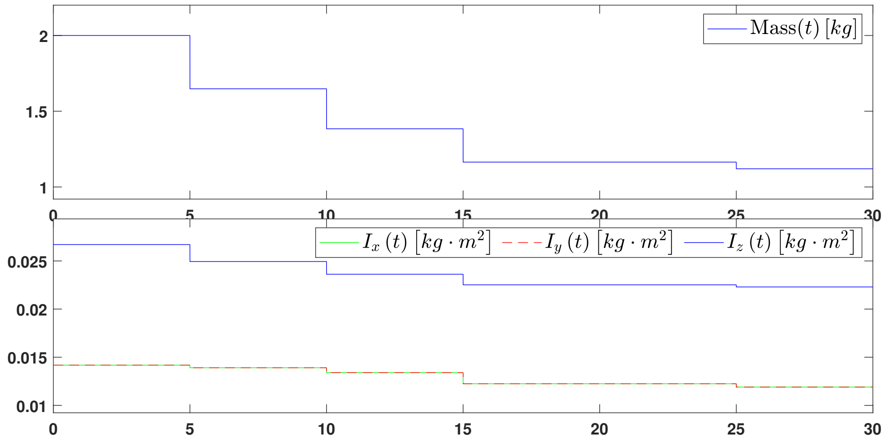


FIGURE 6.50: Scenario 7:
Mass, I_x, I_y, and I_z
LPV UIO for Alt/Att

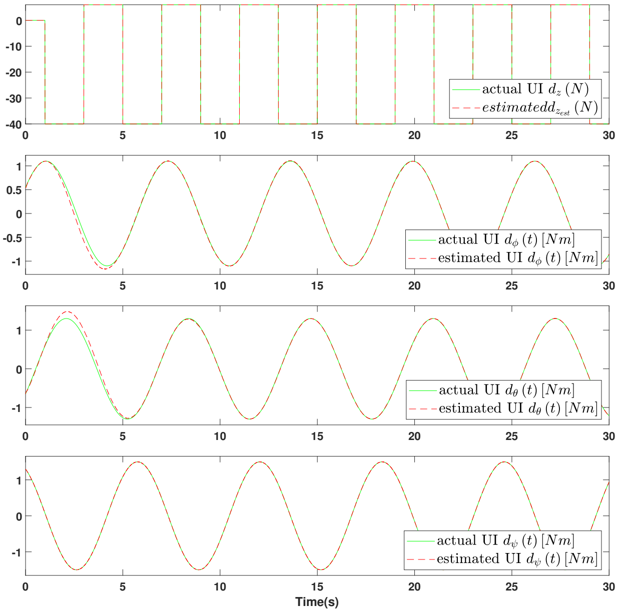


FIGURE 6.51: Scenario 7:
UI and their estimation with
LPV UIO for Alt/Att

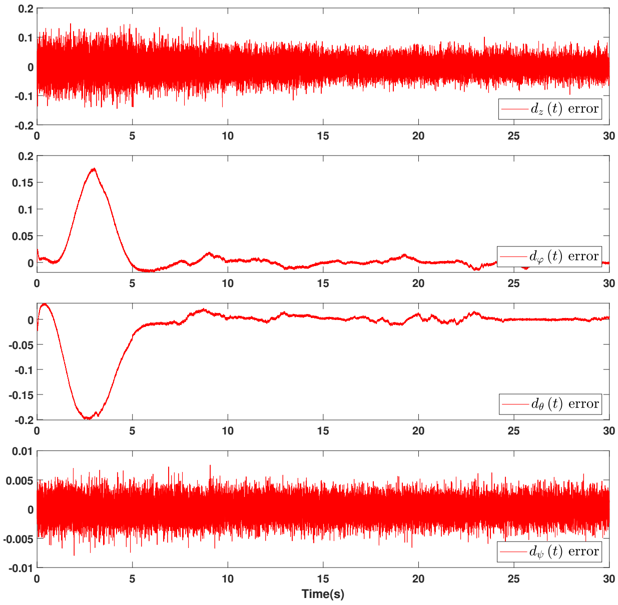


FIGURE 6.52: Scenario 7:
UI estimations errors with
LPV UIO for Alt/Att

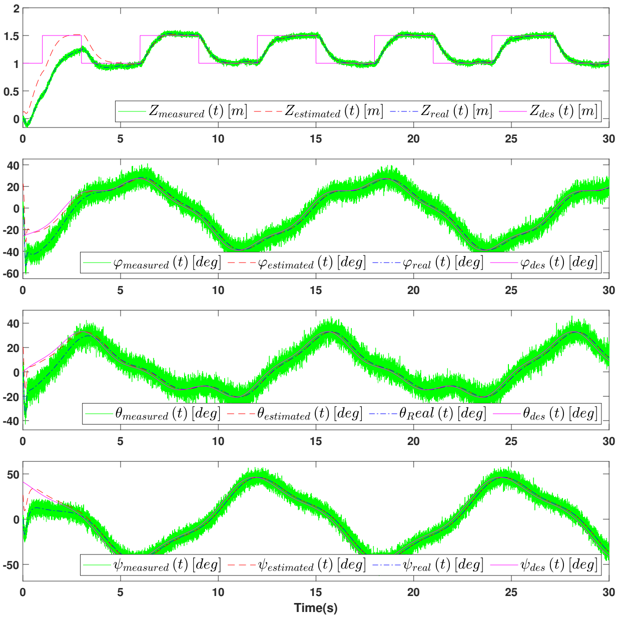


FIGURE 6.53: Scenario 7:
Z, ϕ, θ, and ψ and estimations
LPV UIO for Alt/Att

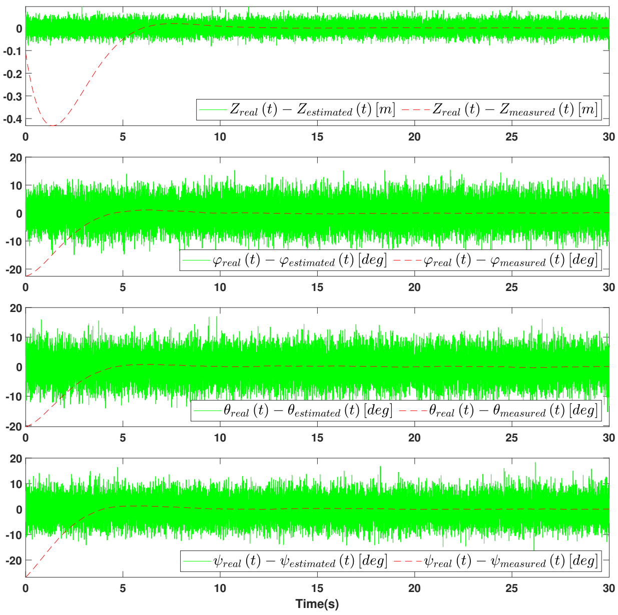


FIGURE 6.54: Scenario 7:
Differences between (real and estimated) vs
(real and measured) values
LPV UIO for Alt/Att

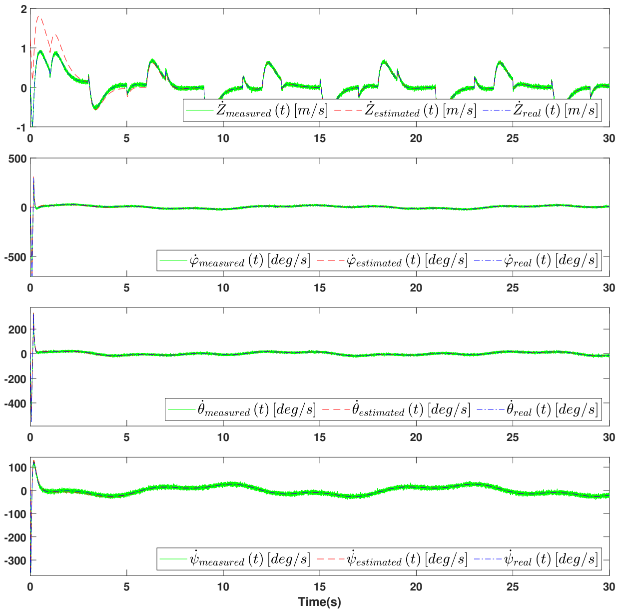


FIGURE 6.55: Scenario 7: $\dot{Z}$, $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ and estimations LPV UIO for Alt/Att

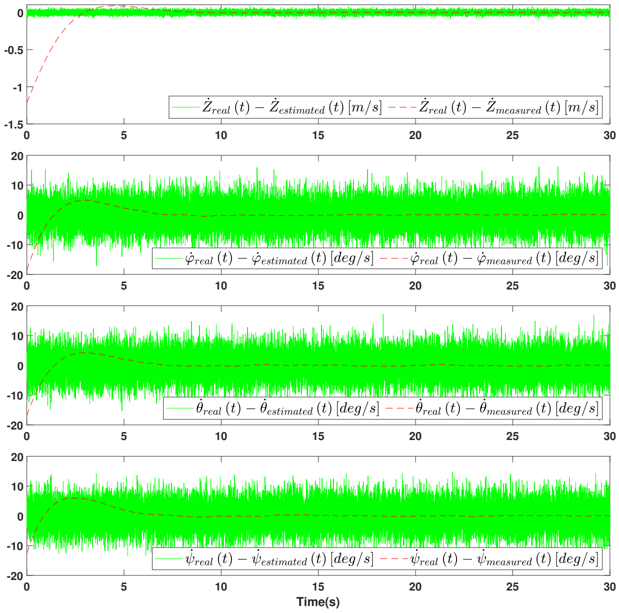


FIGURE 6.56: Scenario 7: Differences between (real and estimated) vs (real and measured) values LPV UIO for Alt/Att

The actual unknown inputs d_z, d_φ, d_θ, d_ψ *and their estimated values are shown in Figure 6.51 and Figure 6.52 shows the differences between actual unknown inputs and estimated unknown inputs. We can see that the estimated unknown inputs tend to actual unknown inputs in about* $5\,(s)$.

In Figure 6.53 are shown the desired, actual, measured, and estimated value of φ, θ, ψ, *and* z. *We can see that* φ, θ, ψ, *and* z *track the reference signals well (as in Figure 6.53) even under disturbances and the abrupt changes of mass and moments of inertia. The response time for* φ, θ, ψ, *and* z *are small.*

The Figure 6.54 shows the differences between the real and estimated values, and between real and measured values of φ, θ, ψ, *and* z. *These errors signals are really small compared to the actual values. We can see that, even under disturbances, the abrupt changes of mass and moments of inertia, and the measurement noises, the estimated values still track the actual values of* φ, θ, ψ, *and* z *well.*

The actual, and estimated value of $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, *and* $\dot{z}$ *are showed in Figure 6.55, while Figure 6.56 shows the difference between the actual and estimated, and between the actual and measured values of* $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, *and* $\dot{z}$. *We can see that, even under disturbances, the changes of mass and moments of inertia, and the measurement noises, the estimated values still track the actual values of* φ, θ, ψ, *and* z *well.*

Some simulations has been conducted when the outputs of the quadcopter are disturbed by the same random white noise signals as given in 6.53 and 6.54, if the UIO is not applied, the control is lost. ■

Scenario 8. *In this simulation:*

- *Measurement noises (random white noises) are considered*
- *Simulation time is* $30\,(s)$
- *The mass gradually changes from* $2\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $5\,(s)$ *to* $25\,(s)$*. Consequently, the moments of inertia also gradually change over a period of time from* $5\,(s)$ *to* $25\,(s)$ *as shown in Figure 6.57.*
- *The reference signals* φ_{ref}*,* θ_{ref}*,* ψ_{ref}*, and* z_{ref} *(Figure 6.60) are:*
  - φ_{ref} *is a step signal at* $2\,(s)$ *from* $0\,(deg)$ *to* $30\,(deg)$
  - θ_{ref} *is a step signal at* $4\,(s)$ *from* $0\,(deg)$ *to* $28\,(deg)$
  - ψ_{ref} *is a step signal at* $6\,(s)$ *from* $0\,(deg)$ *to* $26\,(deg)$
  - z_{ref} *is a square wave signal of frequency* $1/6\,(Hz)$*, maximum amplitude is* $1.5\,(m)$*, minimum amplitude is* $1\,(m)$
- *The disturbances* d_φ*,* d_θ*,* d_ψ*, and* d_z *(Figure 6.58) are:*
  - d_φ *is a random signal, maximum amplitude is* $1.1\,(Nm)$ *and minimum amplitude is* $-1.1\,(Nm)$
  - d_θ *is a random signal, maximum amplitude is* $1.3\,(Nm)$ *and minimum amplitude is* $-1.3\,(Nm)$
  - d_ψ *is a random signal, maximum amplitude is* $1.5\,(Nm)$ *and minimum amplitude is* $-1.5\,(Nm)$
  - d_z *is a random signal, maximum amplitude is* $6\,(N)$ *and minimum amplitude is* $-30\,(N)$

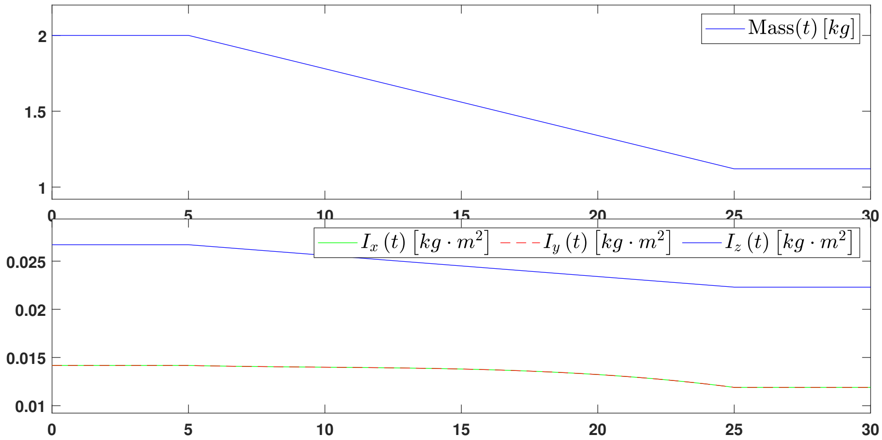


FIGURE 6.57: Scenario 8:
Mass, I_x, I_y, and I_z
LPV UIO for Alt/Att

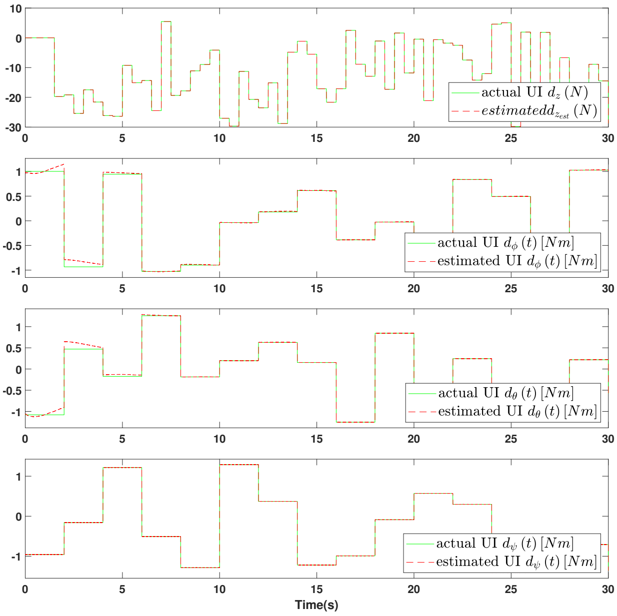


FIGURE 6.58: Scenario 8:
UIO Moments estimations
LPV UIO for Alt/Att

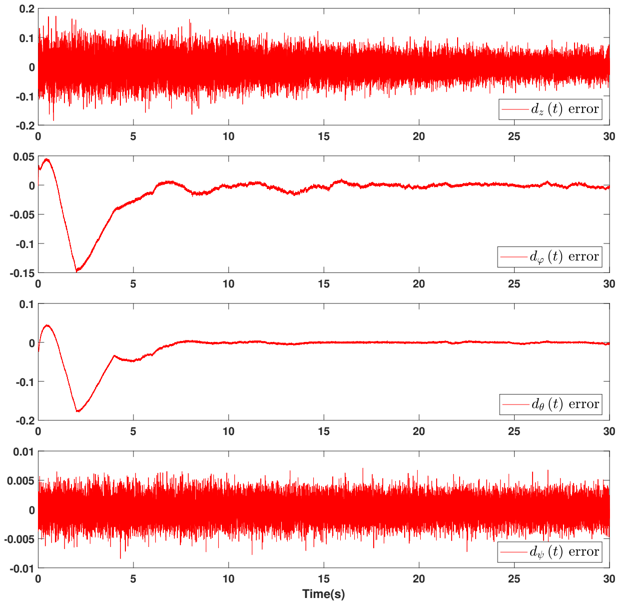


FIGURE 6.59: Scenario 8:
UIO Moments estimations errors
LPV UIO for Alt/Att

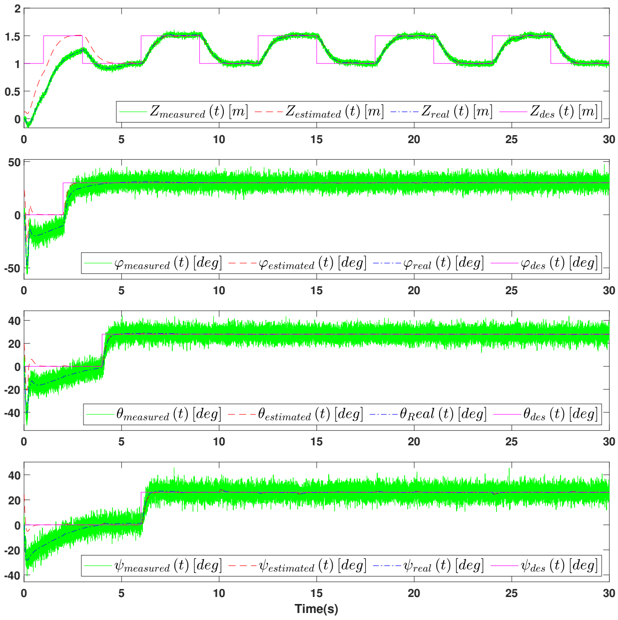


FIGURE 6.60: Scenario 8:
Z, ϕ, θ, and ψ and estimations
LPV UIO for Alt/Att

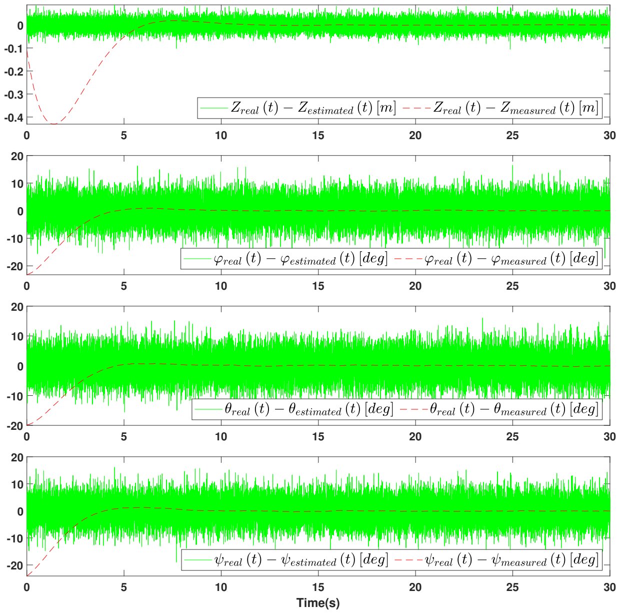


FIGURE 6.61: Scenario 8:
Differences between (real and estimated) vs
(real and measured) values
LPV UIO for Alt/Att

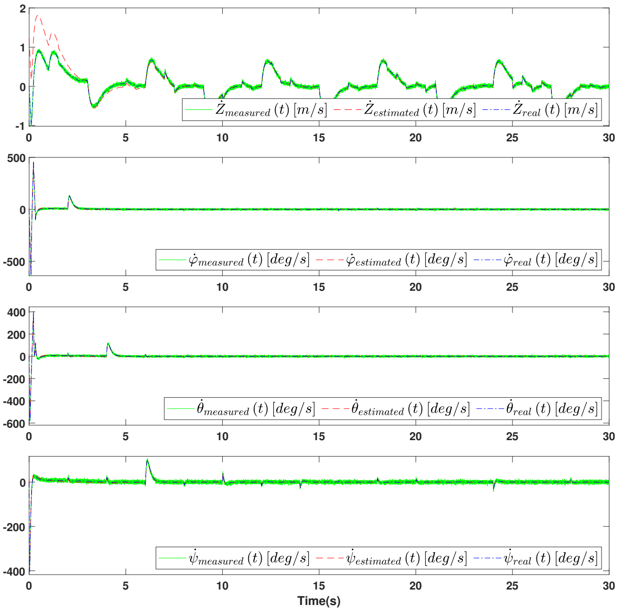


FIGURE 6.62: Scenario 8:
$\dot{Z}$, $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ and estimations
LPV UIO for Alt/Att

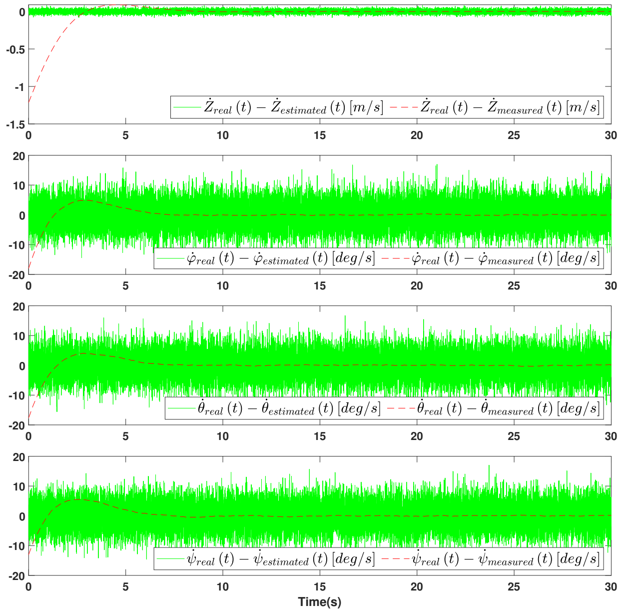


FIGURE 6.63: Scenario 8:
Differences between (real and estimated) vs
(real and measured) values
LPV UIO for Alt/Att

The actual unknown inputs d_z, d_φ, d_θ, d_ψ *and their estimated values are shown in Figure 6.58 and Figure 6.59 shows the differences between actual unknown inputs and estimated unknown inputs. We can see that the estimated unknown inputs tend to actual unknown inputs in about* $5\,(s)$.

In Figure 6.60 are shown the desired, actual, measured, and estimated value of φ, θ, ψ, *and* z. *We can see that* φ, θ, ψ, *and* z *track the reference signals well (as in Figure 6.60) even under disturbances and the gradual changes of mass and moments of inertia. The response time for* φ, θ, ψ, *and* z *are small.*

The Figure 6.61 shows the differences between the real and estimated values, and between real and measured values of φ, θ, ψ, *and* z. *We can see that, even under disturbances, the gradual changes of mass and moments of inertia, and the measurement noises, the estimated values still track the actual values of* φ, θ, ψ, *and* z *well.*

The actual, and estimated value of $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, *and* $\dot{z}$ *are shown in Figure 6.62, while Figure 6.63 show the difference between the actual and estimated, and between the actual and measured values of* $\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$, *and* $\dot{z}$. *We can see that, even under disturbances, the changes of mass and moments of inertia, and the measurement noises, the estimated values still track the actual values of* φ, θ, ψ, *and* z *well.*

Some simulations has been conducted when the outputs of the quadcopter are disturbed by the same random white noise signals as given in 6.60 and 6.61, if the UIO is not applied, the control is lost. ■

6.2.3 Quadrotor path following

In this section, we simulate the full control for a mass-varying quadcopter under disturbances to follow predefined trajectory. Due to the variation of the mass, the moments of inertia with respect to the three axes x, y, and z are also changing. The mass can be changed abruptly or gradually.

- The disturbances d_φ, d_θ, d_ψ, and d_z to roll φ, pitch θ, yaw ψ, and altitude z of the quadcopter can be of the form sinus, random, step, square wave, or constant.
- The reference signals for roll φ, pitch θ, yaw ψ, and altitude z can be of the form sinus, random, step, square wave, or constant.

The configuration of the the full control for a mass-varying quadcopter is shown in Figure 6.64.

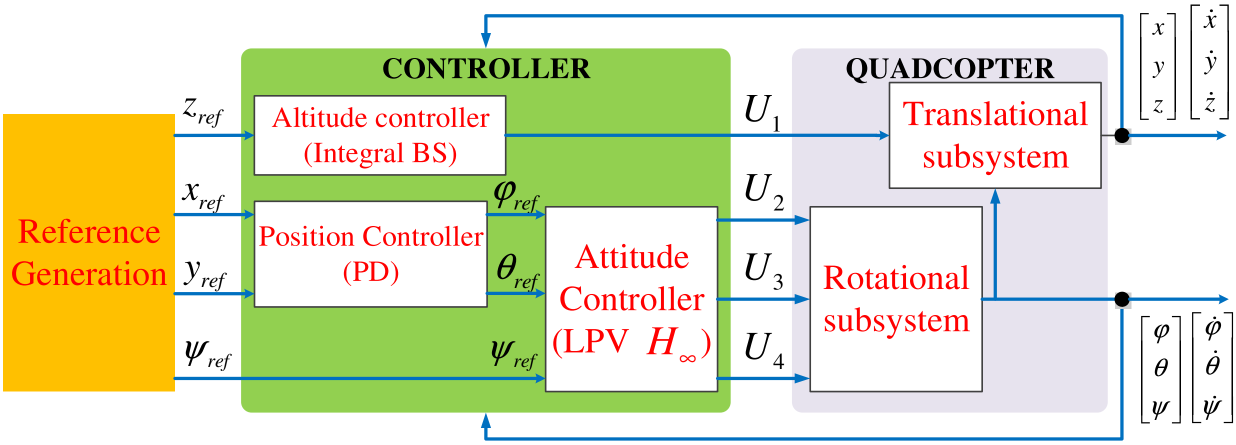


FIGURE 6.64: LPV H_∞ UIO for mass-varying quadcopter configuration

Scenario 9. *In this simulation:*

- *No measurement noise is considered*
- *Simulation time is* $90\,(s)$
- *The mass of the quadcopter changes in the two manners:*
    - *The mass gradually changes from* $2.0\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $10\,(s)$ *to* $85\,(s)$*. Consequently, the moments of inertia also gradually change over a period of time from* $10\,(s)$ *to* $85\,(s)$ *as shown in the first two sub-figures of Figure 6.65.*
    - *The mass abruptly changes at* $10\,(s)$ *from* $2.0\,(kg)$ *to* $1.824\,(kg)$*, at* $20\,(s)$ *from* $1.824\,(kg)$ *to* $1.736\,(kg)$*, at* $30\,(s)$ *from* $1.736\,(kg)$ *to* $1.56\,(kg)$*, at* $40\,(s)$ *from* $1.56\,(kg)$ *to* $1.472\,(kg)$*, at* $50\,(s)$ *from* $1.472\,(kg)$ *to* $1.384\,(kg)$*, at* $60\,(s)$ *from* $1.384\,(kg)$ *to* $1.296\,(kg)$*, at* $70\,(s)$ *from* $1.296\,(kg)$ *to* $1.208\,(kg)$*, and at* $85\,(s)$ *from* $1.208\,(kg)$ *to* $1.12\,(kg)$*. Consequently, the moments of inertia also change abruptly at* $10\,(s)$*,* $20\,(s)$*,* $30\,(s)$*,* $40\,(s)$*,* $50\,(s)$*,* $60\,(s)$*,* $70\,(s)$*,* $85\,(s)$ *as shown in the last two sub-figures of Figure 6.65.*
- *The quadrotor must follow a square trajectory as follows: Quadrotor moves from (0,0,0) to (0,0,2) over the interval of time* $[0\,(s)\,;5\,(s)]$*, moves from (0,0,2) to (10,0,2) over the interval of time* $[10\,(s)\,;25\,(s)]$*, moves from (10,0,2) to (10,10,2) over the interval of time* $[25\,(s)\,;45\,(s)]$*, moves from (10,10,2) to (0,10,2) over the interval of time* $[45\,(s)\,;65\,(s)]$*, moves from (0,10,2) to (0,0,2) over the interval of time* $[65\,(s)\,;85\,(s)]$*, moves from (0,0,2) to (0,0,0) over the interval of time* $[85\,(s)\,;90\,(s)]$ *as in Figure 6.70. Therefore, reference signals* x_{ref}*,* y_{ref}*,* z_{ref}*, and* ψ_{ref} *are:*
    - x_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s)\,;5\,(s)]$*, changes from* $0\,(m)$ *to* $10\,(m)$ *over the interval of time* $[5\,(s)\,;25\,(s)]$*, keeps at* $10\,(m)$ *over the interval of time* $[25\,(s)\,;45\,(s)]$*, changes from* $10\,(m)$ *to* $0\,(m)$ *over the interval of time* $[45\,(s)\,;65\,(s)]$*, keeps at* $0\,(m)$ *over the interval of time* $[65\,(s)\,;90\,(s)]$*.*
    - y_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s)\,;25\,(s)]$*, changes from* $0\,(m)$ *to* $10\,(m)$ *over the interval of time* $[25\,(s)\,;45\,(s)]$*, keeps at* $10\,(m)$ *over the interval of time* $[45\,(s)\,;65\,(s)]$*, changes from* $10\,(m)$ *to* $0\,(m)$ *over the interval of time* $[65\,(s)\,;85\,(s)]$*, keeps at* $0\,(m)$ *over the interval of time* $[85\,(s)\,;90\,(s)]$
    - z_{ref} *changes from* $0\,(m)$ *to* $2\,(m)$ *over the interval of time* $[0\,(s)\,;5\,(s)]$*, keeps at* $2\,(m)$ *over the interval of time* $[5\,(s)\,;85\,(s)]$*, and changes from* $2\,(m)$ *to* $0\,(m)$ *over the interval of time* $[85\,(s)\,;90\,(s)]$*.*
    - $\psi_{ref} = 0\,(deg)$
- *The disturbances* d_x*,* d_y*, and* d_z *are* $Wind_X$*,* $Wind_Y$*, and* $Wind_Z$ *in (the first sub-figure of Figure 6.66) are:*
    - $Wind_X$ *is a square wave signal of frequency* $1/9\,(Hz)$*, maximum amplitude is* $5.1\,(m/s)$*, minimum amplitude is* $0\,(m/s)$
    - $Wind_Y$ *is a step signal at* $5\,(s)$ *from* $0\,(m/s)$ *to* $4.8\,(m/s)$

- $Wind_Z$ *is a square wave signal of frequency* $1/10\,(Hz)$, *maximum amplitude is* $1.5\,(m/s)$, *minimum amplitude is* $0\,(m/s)$

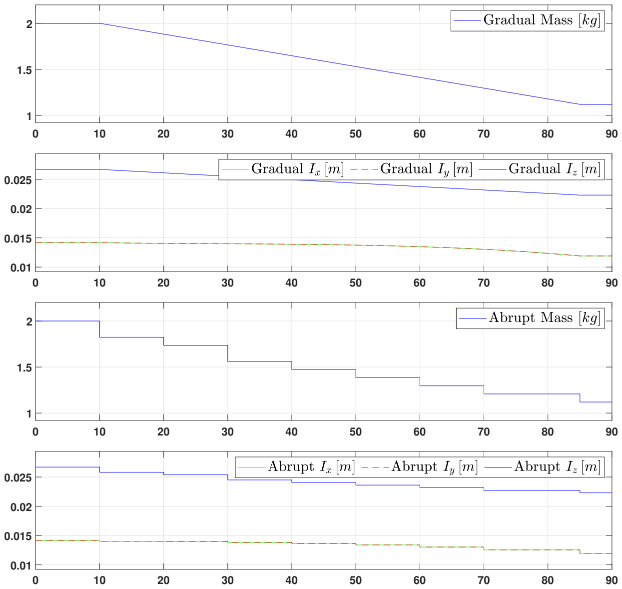


FIGURE 6.65: Scenario 9:
Mass, I_x, I_y, and I_z
Quadrotor path following

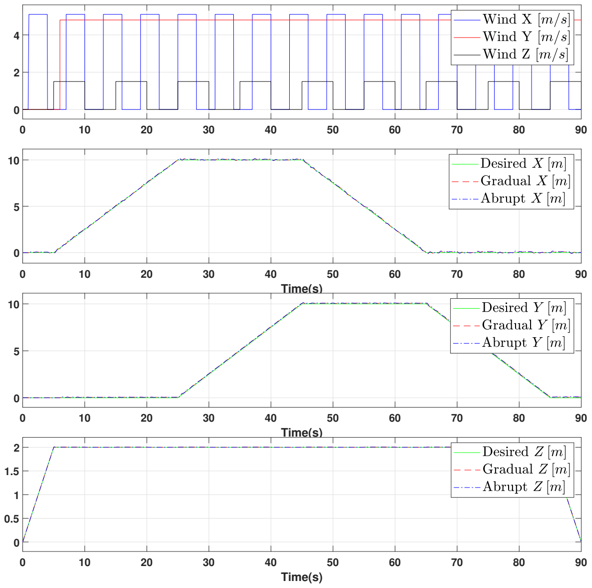


FIGURE 6.66: Scenario 9:
x, y, and z, their responses, and disturbances
Quadrotor path following

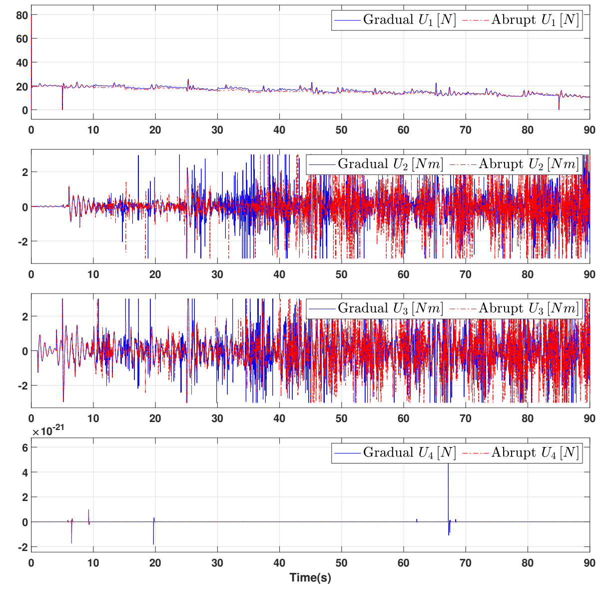


FIGURE 6.67: Scenario 9:
U_1, U_2, U_3 and U_4
Quadrotor path following

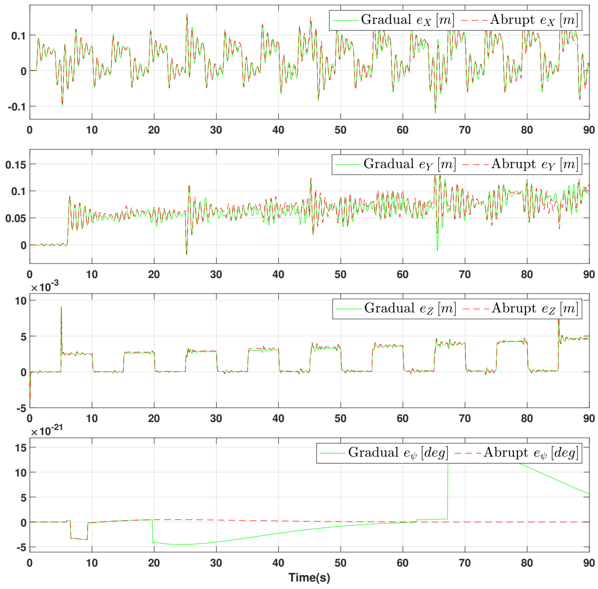


FIGURE 6.68: Scenario 9:
Errors on X, Y, Z and ψ
Quadrotor path following

The results in Figure 6.66, 6.69 and 6.70 show that the quadcopter follows the reference trajectory well even under the disturbances. The Figure 6.68 shows the errors of the outputs and reference signals. We can see that, for x *and* y*, the maximum values of the errors are less than* $0.15m$*. This errors happen when the disturbances are applied.*

Figure 6.67 show the inputs to the quadcopter U_1, U_2, U_3, and U_4.

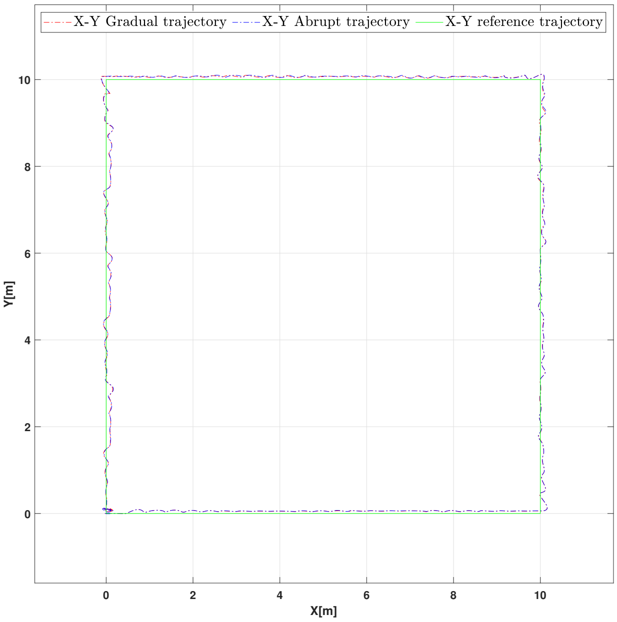


FIGURE 6.69: Scenario 9: X and Y in 2D Quadrotor path following

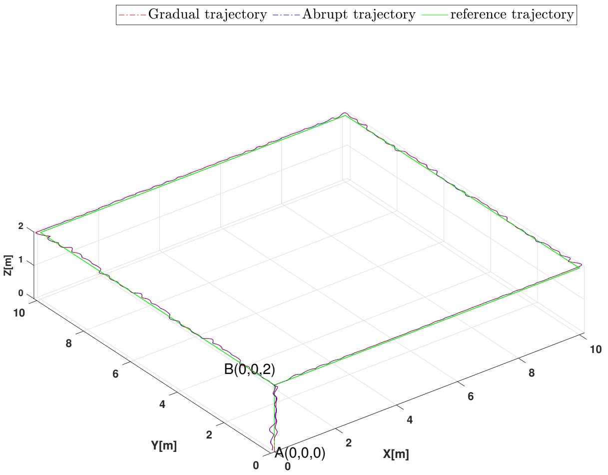


FIGURE 6.70: Scenario 9: X, Y, and Z in 3D Quadrotor path following

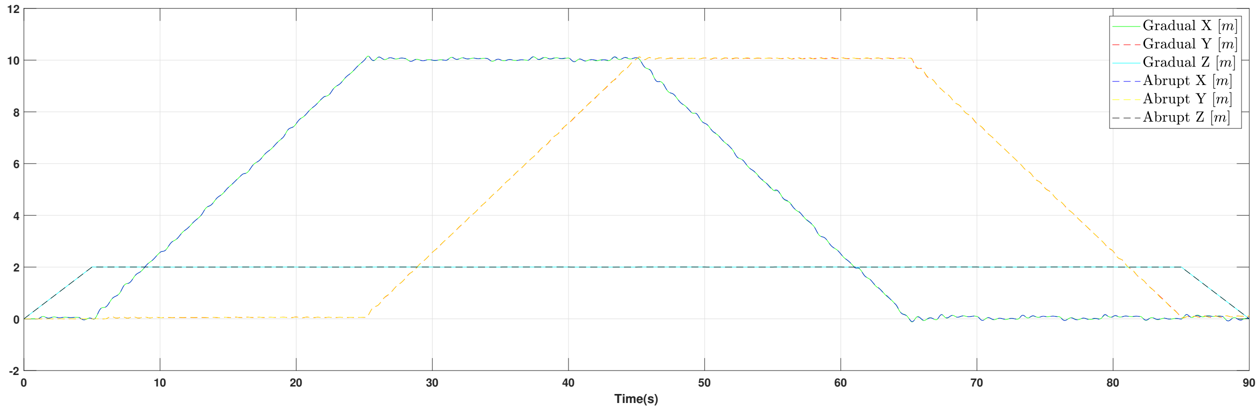


FIGURE 6.71: Scenario 9: X, Y, and Z vs time Quadrotor path following

The results in Figure 6.71 show that the mass variations (abrupt or gradual) do not effect to the ability of path following of the quadcopter. The quadcopter follows the reference trajectory even under the different types of disturbances and variations of the mass and moments of inertia. ■

Scenario 10. *In this simulation:*

- *No measurement noise is considered*
- *Simulation time is* $90\,(s)$
- *The mass of the quadcopter changes in the two manners:*
  - *The mass gradually changes from* $2.0\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $10\,(s)$ *to* $80\,(s)$. *Consequently, the moments of inertia also gradually change over a period of time from* $10\,(s)$ *to* $80\,(s)$ *as shown in the first two sub-figures of Figure 6.72.*
  - *The mass abruptly changes at* $10\,(s)$ *from* $2.0\,(kg)$ *to* $1.648\,(kg)$, *at* $30\,(s)$ *from* $1.648\,(kg)$ *to* $1.384\,(kg)$, *at* $50\,(s)$ *from* $1.384\,(kg)$ *to* $1.164\,(kg)$, *and at* $80\,(s)$ *from* $1.164\,(kg)$ *to* $1.12\,(kg)$. *Consequently, the moments of inertia also change abruptly at* $10\,(s)$, $30\,(s)$, $50\,(s)$, $80\,(s)$ *as shown in the last two sub-figures of Figure 6.72.*
- *The quadrotor must follow a square trajectory as follows: Quadrotor moves from (0,0,0) to (0,0,2) over the interval of time* $[0\,(s);5\,(s)]$, *moves from (0,0,2) to (10,0,2) over the interval of time* $[10\,(s);25\,(s)]$, *moves from (10,0,2) to (10,10,2) over the interval of time* $[25\,(s);40\,(s)]$, *moves from (10,10,2) to (0,10,2) over the interval of time* $[45\,(s);65\,(s)]$, *moves from (0,10,2) to (0,0,2) over the interval of time* $[65\,(s);85\,(s)]$, *moves from (0,0,2) to (0,0,0) over the interval of time* $[85\,(s);90\,(s)]$ *as in Figure 6.77. Therefore, reference signals* x_{ref}, y_{ref}, z_{ref}, *and* ψ_{ref} *are:*
  - x_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s);5\,(s)]$, *changes from* $0\,(m)$ *to* $10\,(m)$ *over the interval of time* $[5\,(s);25\,(s)]$, *keeps at* $10\,(m)$ *over the interval of time* $[25\,(s);45\,(s)]$, *changes from* $10\,(m)$ *to* $0\,(m)$ *over the interval of time* $[45\,(s);65\,(s)]$, *keeps at* $0\,(m)$ *over the interval of time* $[65\,(s);90\,(s)]$.
  - y_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s);25\,(s)]$, *changes from* $0\,(m)$ *to* $10\,(m)$ *over the interval of time* $[25\,(s);45\,(s)]$, *keeps at* $10\,(m)$ *over the interval of time* $[45\,(s);65\,(s)]$, *changes from* $10\,(m)$ *to* $0\,(m)$ *over the interval of time* $[65\,(s);85\,(s)]$, *keeps at* $0\,(m)$ *over the interval of time* $[85\,(s);90\,(s)]$
  - z_{ref} *changes from* $0\,(m)$ *to* $2\,(m)$ *over the interval of time* $[0\,(s);5\,(s)]$, *keeps at* $2\,(m)$ *over the interval of time* $[5\,(s);85\,(s)]$, *and changes from* $2\,(m)$ *to* $0\,(m)$ *over the interval of time* $[85\,(s);90\,(s)]$.
  - $\psi_{ref} = 0\,(deg)$
- *The disturbances* d_x, d_y, *and* d_z *are* $Wind_X$, $Wind_Y$, *and* $Wind_Z$ *in (the first sub-figure of Figure 6.73) are:*
  - $Wind_X$ *is a square wave signal of frequency* $1/9\,(Hz)$, *maximum amplitude is* $5.1\,(m/s)$, *minimum amplitude is* $0\,(m/s)$
  - $Wind_Y$ *is a sine signal of frequency* $1/40\,(Hz)$, *maximum amplitude is* $4.9\,(m/s)$, *minimum amplitude is* $-4.9\,(m/s)$
  - $Wind_Z$ *is a step signal at* $6\,(s)$ *from* $0\,(m/s)$ *to* $1.5\,(m/s)$

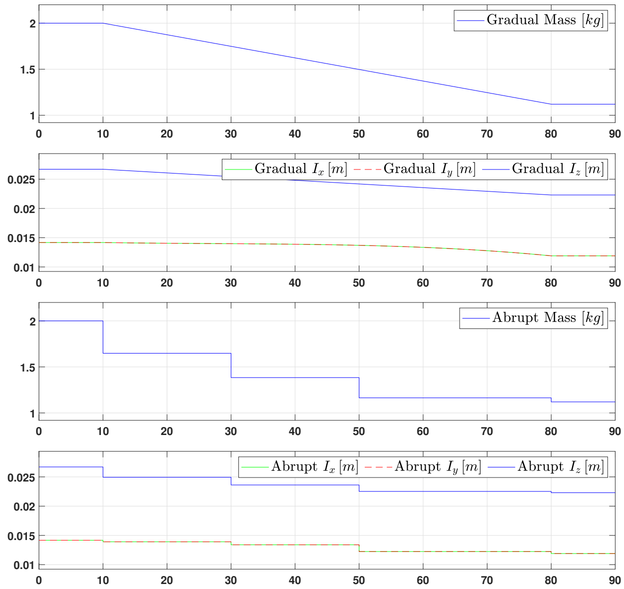


FIGURE 6.72: Scenario 10:
Mass, I_x, I_y, and I_z
Quadrotor path following

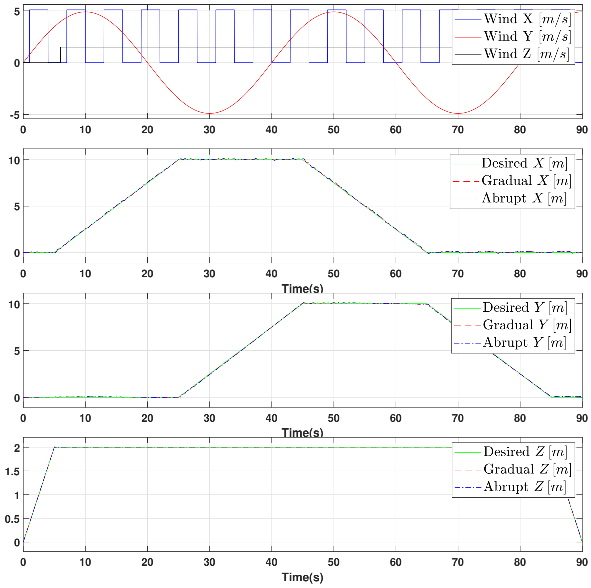


FIGURE 6.73: Scenario 10:
x, y, and z, their responses, and disturbances
Quadrotor path following

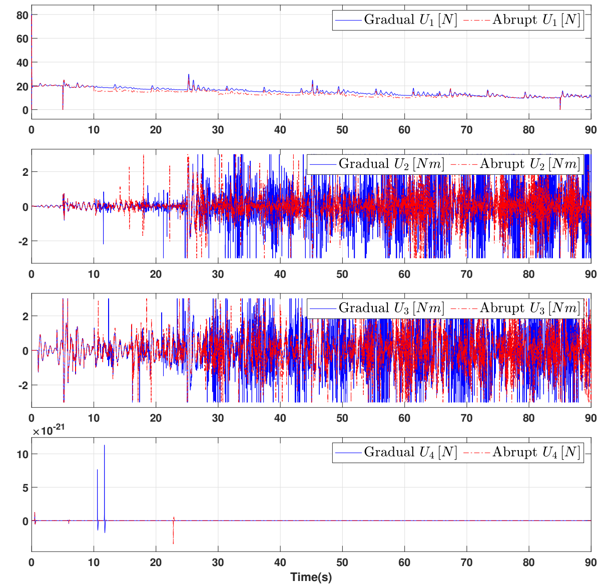


FIGURE 6.74: Scenario 10:
U_1, U_2, U_3 and U_4
Quadrotor path following

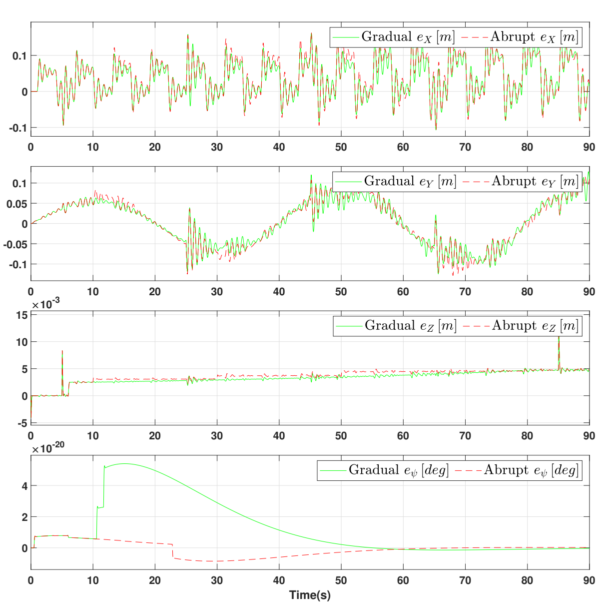


FIGURE 6.75: Scenario 10:
Errors on X, Y, Z and ψ
Quadrotor path following

The results in Figure 6.73, 6.76 and 6.77 show that the quadcopter follows the reference trajectory well even under the disturbances. The Figure 6.75 shows the errors of the outputs and reference signals. We can see that, for x and y, the maximum values of the errors are less than $0.15m$. This errors happen when the disturbances are applied.

Figure 6.74 show the inputs to the quadcopter U_1, U_2, U_3, and U_4.

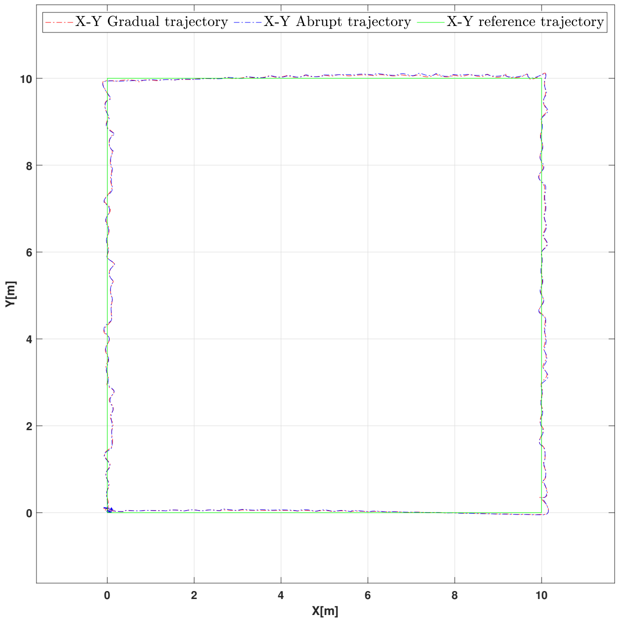


FIGURE 6.76: Scenario 10:
X and Y in 2D
Quadrotor path following

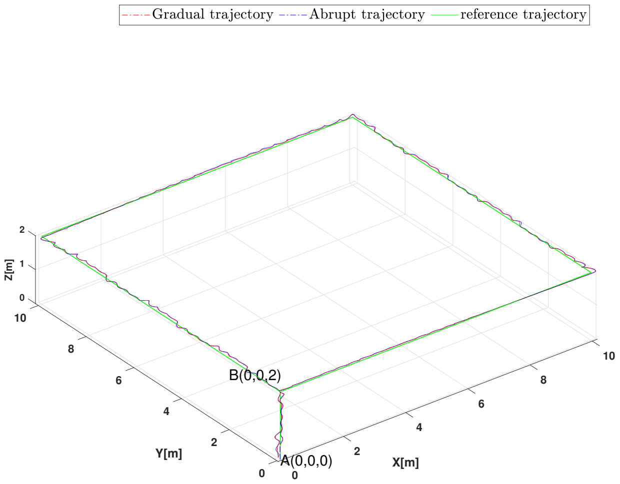


FIGURE 6.77: Scenario 10:
X, Y, and Z in 3D
Quadrotor path following

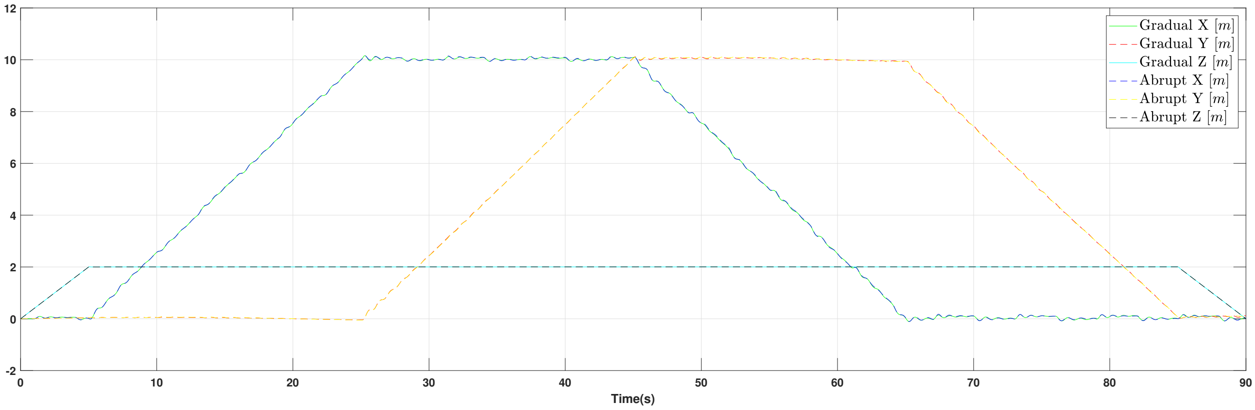


FIGURE 6.78: Scenario 10:
X, Y, and Z vs time
Quadrotor path following

The results in Figure 6.78 show that the methods of mass variation (abrupt or gradual) do not effect to the ability of path following of the quadcopter. The quadcopter follows the reference trajectory even under the different type of disturbances and variations of the mass and moments of inertia.

The quadcopter tracks the reference signal well even at the vertices of the rectangle as in Figures 6.76 and 6.77 ■

Scenario 11. *In this simulation:*

- *No measurement noise is considered*
- *Simulation time is* $90\,(s)$
- *The mass of the quadcopter changes in the two manners:*
  - *The mass gradually changes from* $2.0\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $5\,(s)$ *to* $80\,(s)$*. Consequently, the moments of inertia also gradually change over a period of time from* $5\,(s)$ *to* $80\,(s)$ *as shown in the first two sub-figures of Figure 6.79.*
  - *The mass abruptly changes at* $5\,(s)$ *from* $2.0\,(kg)$ *to* $1.824\,(kg)$*, at* $10\,(s)$ *from* $1.824\,(kg)$ *to* $1.648\,(kg)$*, at* $20\,(s)$ *from* $1.648\,(kg)$ *to* $1.560\,(kg)$*, at* $40\,(s)$ *from* $1.560\,(kg)$ *to* $1.472\,(kg)$*, at* $55\,(s)$ *from* $1.472\,(kg)$ *to* $1.164\,(kg)$*, and at* $80\,(s)$ *from* $1.164\,(kg)$ *to* $1.12\,(kg)$*. Consequently, the moments of inertia also change abruptly at* $5\,(s)$*,* $10\,(s)$*,* $20\,(s)$*,* $40\,(s)$*,* $55\,(s)$*,* $80\,(s)$ *as shown in the last two sub-figures of Figure 6.79.*
- *The quadrotor must follow a circle trajectory as follows: Quadrotor moves from (0,0,0) to (0,0,2) over the interval of time* $[0\,(s);5\,(s)]$*, moves along the circle with radius* $5m$ *and center (0,-5,2) over the interval of time* $[5\,(s);85\,(s)]$*, moves from (0,0,2) to (0,0,0) over the interval of time* $[85\,(s);90\,(s)]$ *as in Figure 6.84. Therefore, reference signals* x_{ref}*,* y_{ref}*,* z_{ref}*, and* ψ_{ref} *are:*
  - x_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s);5\,(s)]$*, changes along sine trajectory* $5\sin\left((t-5)\,\frac{\pi}{40}\right)$ *over the interval of time* $[5\,(s);85\,(s)]$*, keeps at* $0\,(m)$ *over the interval of time* $[85\,(s);90\,(s)]$*.*
  - y_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s);5\,(s)]$*, changes along sine trajectory* $5\cos\left((t-5)\,\frac{\pi}{40}\right)-5$ *over the interval of time* $[5\,(s);85\,(s)]$*, keeps at* $0\,(m)$ *over the interval of time* $[85\,(s);90\,(s)]$*.*
  - z_{ref} *changes from* $0\,(m)$ *to* $2\,(m)$ *over the interval of time* $[0\,(s);5\,(s)]$*, keeps at* $2\,(m)$ *over the interval of time* $[5\,(s);85\,(s)]$*, and changes from* $2\,(m)$ *to* $0\,(m)$ *over the interval of time* $[85\,(s);90\,(s)]$*.*
  - $\psi_{ref}=0\,(deg)$
- *The disturbances* d_x*,* d_y*, and* d_z *are* $Wind_X$*,* $Wind_Y$*, and* $Wind_Z$ *in (the first sub-figure of Figure 6.80) are:*
  - $Wind_X$ *is a sine signal of frequency* $1/40\,(Hz)$*, maximum amplitude is* $5.1\,(m/s)$*, minimum amplitude is* $-5.1\,(m/s)$
  - $Wind_Y$ *is a constant signal* $4.9\,(m/s)$
  - $Wind_Z$ *is a impulse signal of frequency* $1/10\,(Hz)$*, maximum amplitude is* $2.5\,(m/s)$*, minimum amplitude is* $0\,(m/s)$

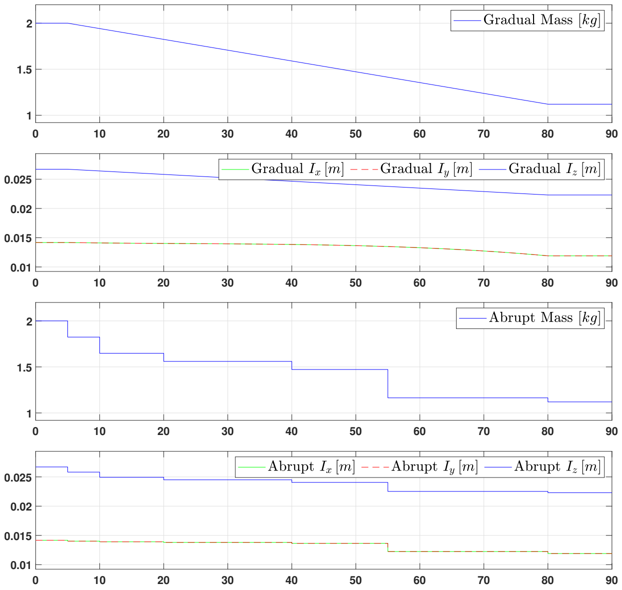


FIGURE 6.79: Scenario 11:
Mass, I_x, I_y, and I_z
Quadrotor path following

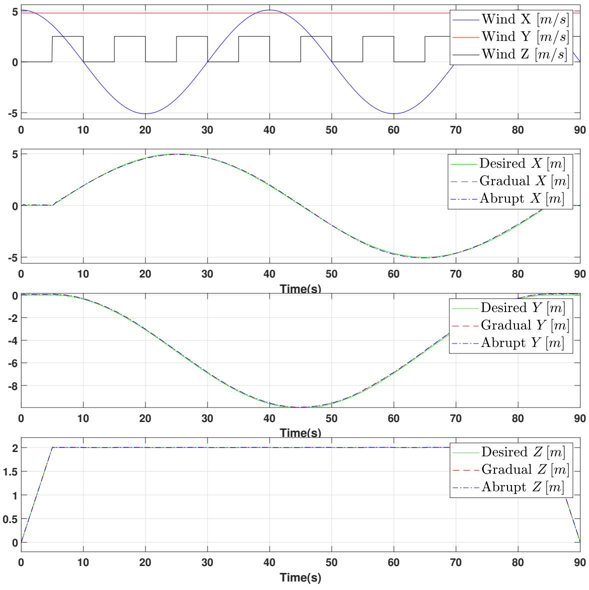


FIGURE 6.80: Scenario 11:
x, y, and z, their responses, and disturbances
Quadrotor path following

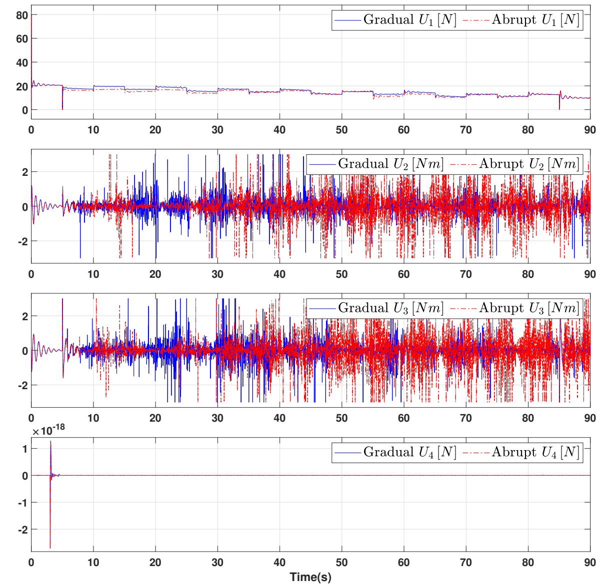


FIGURE 6.81: Scenario 11:
U_1, U_2, U_3 and U_4
Quadrotor path following

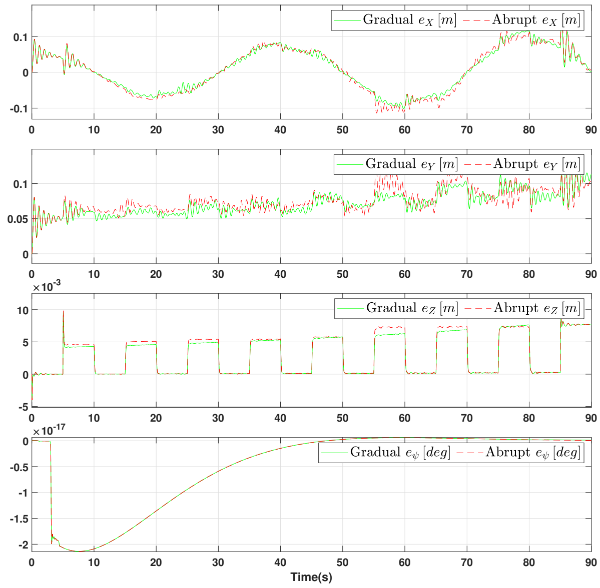


FIGURE 6.82: Scenario 11:
Errors on X, Y, Z and ψ
Quadrotor path following

The results in Figure 6.80, 6.83 and 6.84 show that the quadcopter follows the reference trajectory well even under the disturbances. The Figure 6.82 shows the errors of the outputs and reference signals. We can see that, for x and y, the maximum values of the errors are less than $0.15m$. This errors happen when the disturbances are applied.

Figure 6.81 shows the inputs to the quadcopter U_1, U_2, U_3, and U_4.

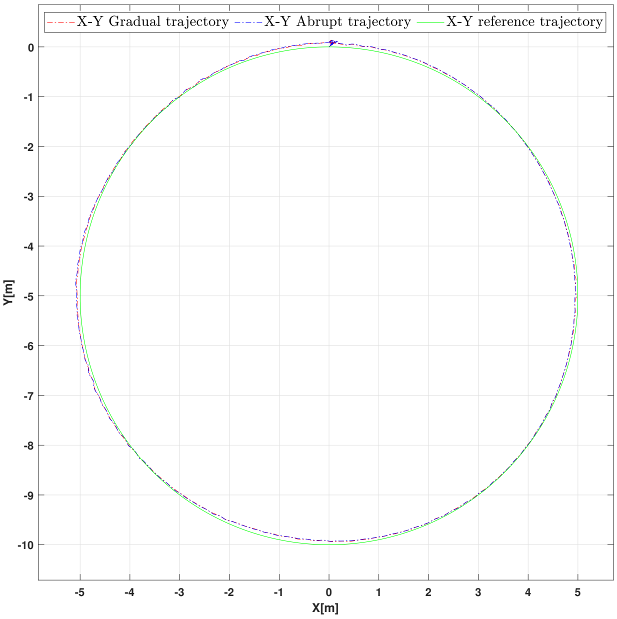


FIGURE 6.83: Scenario 11: X and Y in 2D Quadrotor path following

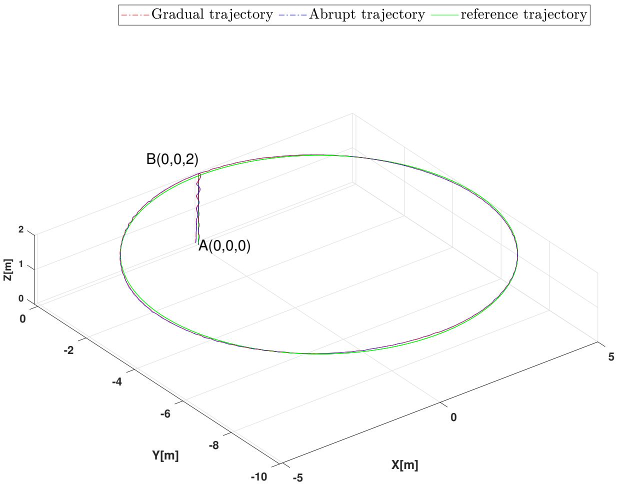


FIGURE 6.84: Scenario 11: X, Y, and Z in 3D Quadrotor path following

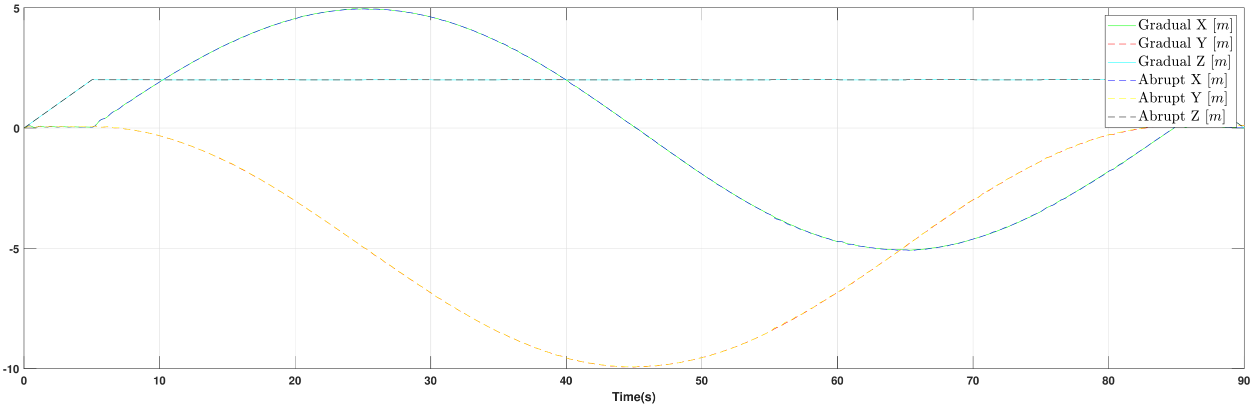


FIGURE 6.85: Scenario 11: X, Y, and Z vs time Quadrotor path following

The results in Figure 6.85 show that the mass variations (abrupt or gradual) do not effect to the ability of path following of the quadcopter. The quadcopter follows the reference trajectory even under the different type of disturbances and variations of the mass and moments of inertia.

■

Scenario 12. *In this simulation:*

- *No measurement noise is considered*
- *Simulation time is* $90\,(s)$
- *The mass of the quadcopter changes in the two manners:*
    - *The mass gradually changes from* $2.0\,(kg)$ *to* $1.12\,(kg)$ *over a period of time from* $15\,(s)$ *to* $80\,(s)$. *Consequently, the moments of inertia also gradually change over a period of time from* $15\,(s)$ *to* $80\,(s)$ *as shown in the first two sub-figures of Figure 6.86.*
    - *The mass abruptly changes at* $15\,(s)$ *from* $2.0\,(kg)$ *to* $1.736\,(kg)$, *at* $40\,(s)$ *from* $1.736\,(kg)$ *to* $1.472\,(kg)$, *and at* $80\,(s)$ *from* $1.472\,(kg)$ *to* $1.12\,(kg)$. *Consequently, the moments of inertia also change abruptly at* $15\,(s)$, $40\,(s)$, *and* $80\,(s)$ *as shown in the last two sub-figures of Figure 6.86.*
- *The quadrotor must follow a circle trajectory as follows: Quadrotor moves from (0,0,0) to (0,0,2) over the interval of time* $[0\,(s)\,;5\,(s)]$, *moves along the circle with radius* $5m$ *and center (0,-5,2) over the interval of time* $[5\,(s)\,;85\,(s)]$, *moves from (0,0,2) to (0,0,0) over the interval of time* $[85\,(s)\,;90\,(s)]$ *as in Figure 6.84. Therefore, reference signals* x_{ref}, y_{ref}, z_{ref}, *and* ψ_{ref} *are:*
    - x_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s)\,;5\,(s)]$, *changes along sine trajectory* $5\sin\left((t-5)\,\frac{\pi}{40}\right)$ *over the interval of time* $[5\,(s)\,;85\,(s)]$, *keeps at* $0\,(m)$ *over the interval of time* $[85\,(s)\,;90\,(s)]$.
    - y_{ref} *keeps at* $0\,(m)$ *over the interval of time* $[0\,(s)\,;5\,(s)]$, *changes along sine trajectory* $5\cos\left((t-5)\,\frac{\pi}{40}\right)-5$ *over the interval of time* $[5\,(s)\,;85\,(s)]$, *keeps at* $0\,(m)$ *over the interval of time* $[85\,(s)\,;90\,(s)]$.
    - z_{ref} *changes from* $0\,(m)$ *to* $2\,(m)$ *over the interval of time* $[0\,(s)\,;5\,(s)]$, *keeps at* $2\,(m)$ *over the interval of time* $[5\,(s)\,;85\,(s)]$, *and changes from* $2\,(m)$ *to* $0\,(m)$ *over the interval of time* $[85\,(s)\,;90\,(s)]$.
    - $\psi_{ref} = 0\,(deg)$
- *The disturbances* d_x, d_y, *and* d_z *are* $Wind_X$, $Wind_Y$, *and* $Wind_Z$ *in (the first sub-figure of Figure 6.87) are:*
    - $Wind_X$ *is a sine wave signal of frequency* $1/40\,(Hz)$, *maximum amplitude is* $5.1\,(m/s)$, *minimum amplitude is* $-5.1\,(m/s)$
    - $Wind_Y$ *is a sine wave signal of frequency* $\pi/40\,(Hz)$, *maximum amplitude is* $4.9\,(m/s)$, *minimum amplitude is* $-4.9\,(m/s)$
    - $Wind_Z$ *is a sine wave signal of frequency* $\pi/40\,(Hz)$, *maximum amplitude is* $2.5\,(m/s)$, *minimum amplitude is* $-2.5\,(m/s)$

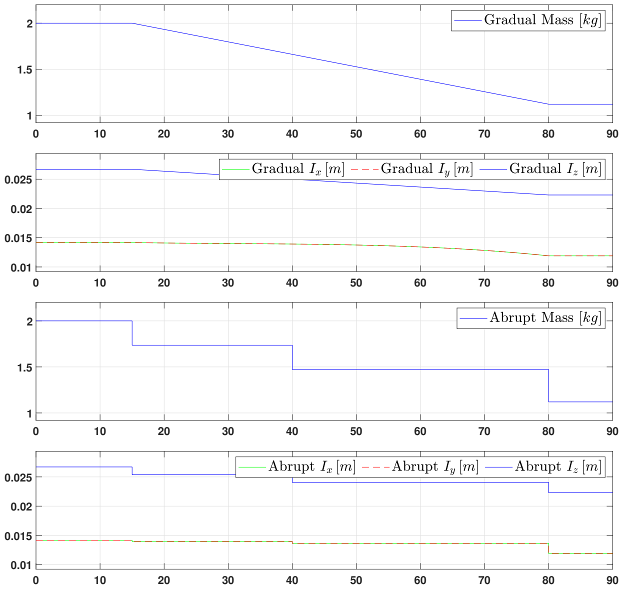


FIGURE 6.86: Scenario 12:
Mass, I_x, I_y, and I_z
Quadrotor path following

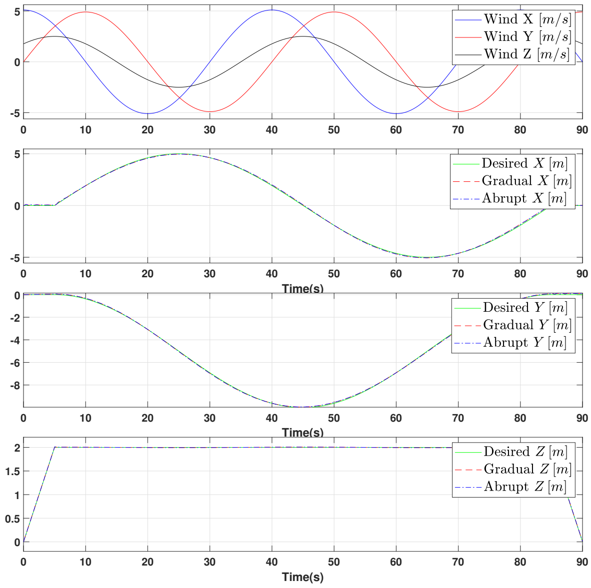


FIGURE 6.87: Scenario 12:
x, y, and z, their responses, and disturbances
Quadrotor path following

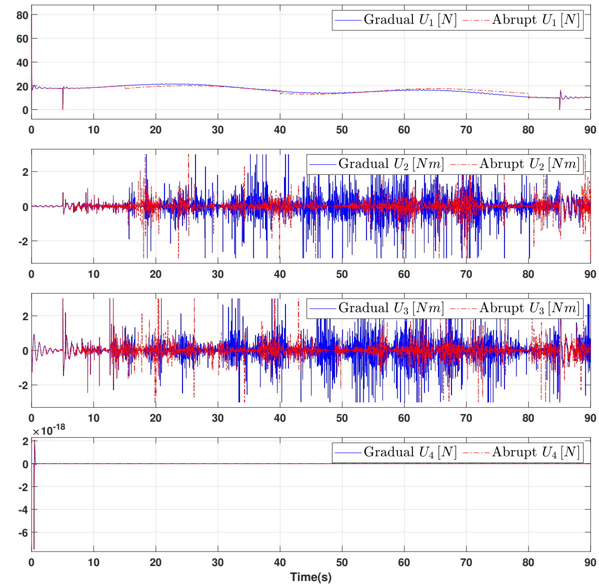


FIGURE 6.88: Scenario 12:
U_1, U_2, U_3 and U_4
Quadrotor path following

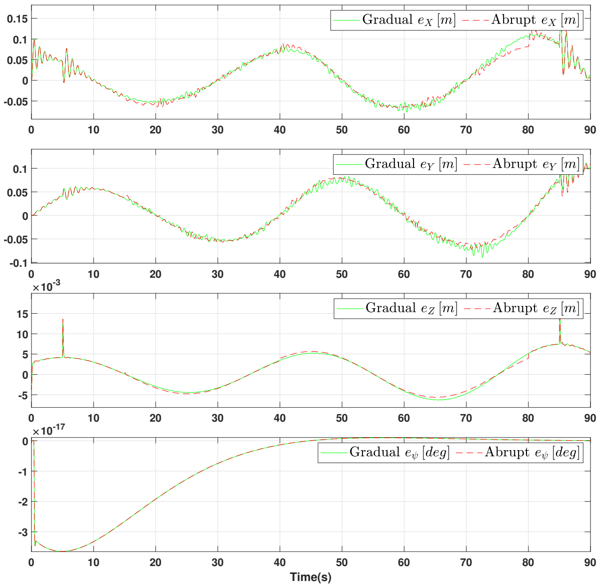


FIGURE 6.89: Scenario 12:
Errors on X, Y, Z and ψ
Quadrotor path following

The results in Figure 6.87, 6.90 and 6.91 show that the quadcopter follows the reference trajectory well even under the disturbances. The Figure 6.89 shows the errors of the outputs and reference signals. We can see that, for x and y, the maximum values of the errors are less than $0.1m$. This errors happen when the disturbances are applied.

Figure 6.88 show the inputs to the quadcopter U_1, U_2, U_3, and U_4.

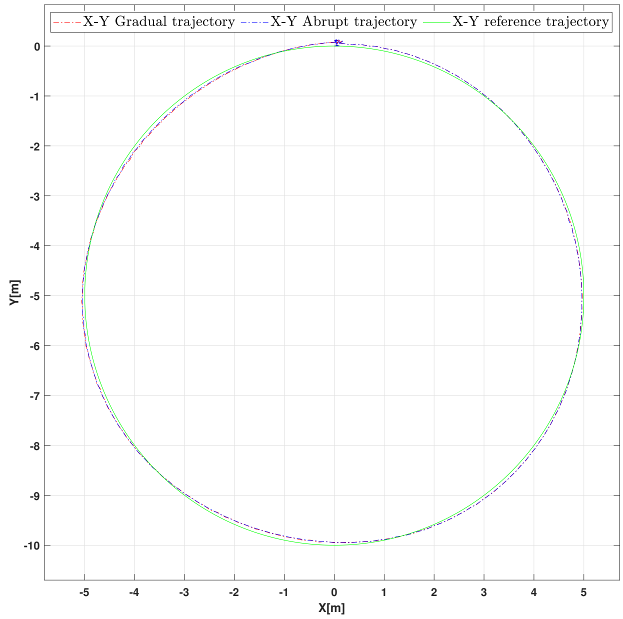


FIGURE 6.90: Scenario 12:
X and Y in 2D
Quadrotor path following

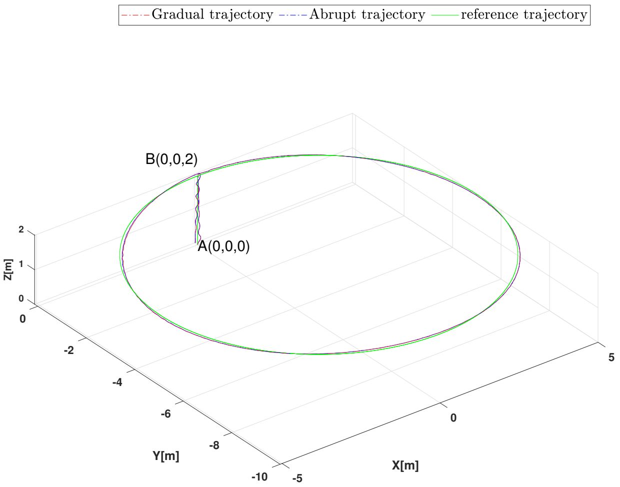


FIGURE 6.91: Scenario 12:
X, Y, and Z in 3D
Quadrotor path following

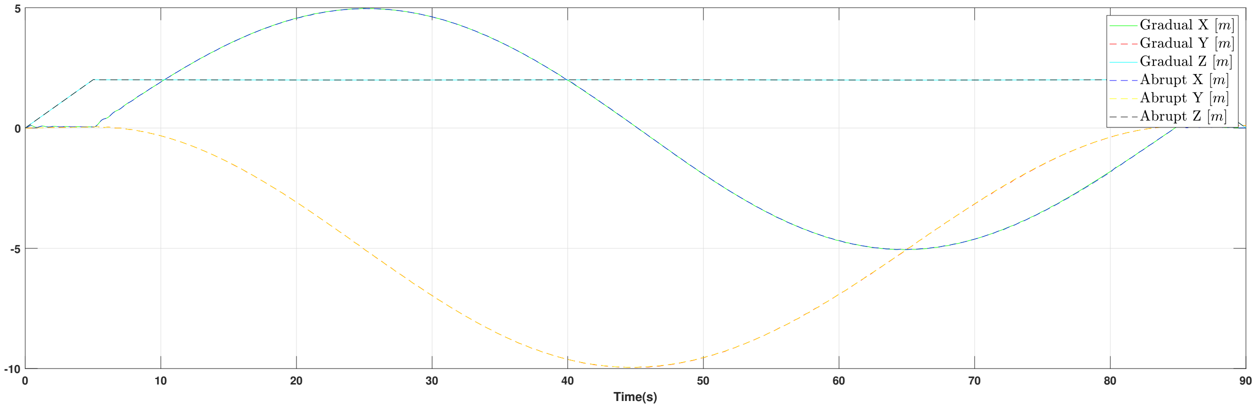


FIGURE 6.92: Scenario 12:
X, Y, and Z vs time
Quadrotor path following

The results in Figure 6.92 show that the methods of mass variation (abrupt or gradual) do not effect to the ability of path following of the quadcopter. The quadcopter follows the reference trajectory even under the different type of disturbances and variations of the mass and moments of inertia.

The quadcopter tracks the reference signal well even at the vertices of the rectangle as in Figures 6.90 and 6.91 ■

6.3 Conclusion

In this chapter, we have simulated different scenarios for both the complete coverage path planning and control of mass-varying quadrotor.

For the complete coverage path planning, the proposed algorithm generates the best trajectory for quadrotor to capture the pictures of the whole agriculture areas. Based on the gathered pictures, the infected crop areas are detected. The second task of complete coverage path planning is to generate the best trajectory for the quadrotor to spray pesticide to all the infected areas. The simulations show the ability of the proposed algorithm to generate the trajectory not only to cover the whole predefined areas but also with the minimum trajectory while avoiding obstacles.

For the control part, several control simulations for LPV H_∞ state feedback controller, static output feedback controller, dynamic output feedback controller have been conducted. the simulation results show the ability of good reference tracking (several types of references are considered such as sine, step, square wave, random) under the existence of disturbances (several types of disturbances are considered such as sine, step, square wave, random) and changes of dynamic parameters of quadrotor (mass and moments of inertia). We also conducted simulations for LPV UIO. The results show that the proposed LPV UIO can well estimate the unknown disturbances and the states of the system.

7

General conclusion and perspectives

7.1 Conclusions

In this research work, we have proposed some new algorithms for Coverage Path Planning (CPP) and control of quadcopter for Precision Agriculture (PA), such as pest-ridden area detection and spraying insecticides. We started the manuscript by an introduction to the importance and application of robotics in precision agriculture to increase crop yields. Then, we emphasized our goals and main contributions and we look at the drones, their classification, their applications. Based on the above analysis, the quadrotor has been selected as the research object in the content of this thesis. Next, we studied CPP algorithms and control laws that have been applied to the quadrotors.

In the first part, we present a new complete CPP algorithm by proposing a new Cellular Decomposition (CD) which is based on a generalization of a Boustrophedon variant, using Morse functions, with an extension of the representation of the critical points. This extension leads to a decrease in the number of cells after decomposition. Genetic Algorithm (GA) and Travelling Salesman Problem (TSP) algorithm are then applied to obtain the shortest path for complete coverage. Next, from the information on the map regarding the coordinates of the obstacles, non-infected areas, and infected areas, the infected areas are divided into several non-overlapping regions by using a clustering technique. Then an algorithm is proposed for generating the best path for a UAV to distribute medicine to all the infected areas of an agriculture environment which contains non-convex obstacles, pest-free areas, and pests-ridden areas.

In the second part, we study the design of a robust control law that allows the UAV to track the predefined trajectory for a dynamic model-changing helicopter due to the changes of dynamic coefficients such as the mass and moments of inertia. Therefore, robust observer and control law are required to adopt the changes in dynamic parameters as well as the impact of external forces. The proposed approach is to explore the modeling techniques, planning, and control by the Takagi-Sugeno type technique. To have easily implantable algorithms and adaptable to changes in parameters and conditions of use, we favor the synthesis of Linear Parameter Varying (LPV) Unknown Input Observer (UIO), LPV quadratic state feedback, robust state feedback, and static output feedback controllers. The observers and controllers are designed by solving a set of Linear Matrix Inequality (LMI) obtained from the Bounded Real Lemma and Linear Matrix Inequality (LMI) conditions.

In summary, besides the generic dynamic model that we have proposed for multirotors that may fly under wind gust, the performed work contains the following achievements:

- Coverage path planning
    1. Firstly, we propose a new approach for cellular decomposition. Based on the critical points as in [45] and [225], the extension of this approach is that the critical points have been categorized to several groups, and the cells have been created from these groups. This extension leads to a decrease of the number of cells after decomposition. The results show that this new cellular decomposition works well even with several concave obstacles inside the environment.
    2. Secondly, we have proposed a method for maximizing the percentage of coverage and we have given some comments for the trade-off between the percentage of coverage and the number of way-points for the UAV. The percentage of coverage is up to 97.9%.
    3. Finally, by using GA with some modification on the swap, flip and slide operations and the TSP with some additional constraints, the shortest path for changing the cells has been developed. This algorithm guarantees that the solution always exists and the time for calculation is decreasing.
    4. For the second task, we have proposed a method for generating a trajectory which allows the UAV to put the medicine to the entire pest-ridden area of an agricultural area. First, the pest-ridden areas have been divided into several smaller areas (clusters) by using clustering technique. After that, each cluster is divided again into several obstacle-free convex polygons. Then the shortest Boustrophedon trajectory is created in each obstacle-free convex polygon. Finally, the shortest trajectory for changing between obstacle-free convex polygons is generated for the final trajectory.
- Control
    1. If the states of UAV can be measured of estimated, we developed an LPV H_∞ state feedback controller for mass-varying quadrotor. This controller is robustly stable under the disturbances
    2. When the states might not be available, the dynamic and static LPV controllers have been proposed for mass-varying quadrotor.
    3. In order to overcome unmeasured states, we proposed also the Linear Parameter Varying (LPV) Unknown Input Observer (UIO) for mass-varying quadrotor.

7.2 Perspectives

In the following, we list some challenges that one may highlight in the future.

- Coverage path planning issues
    1. In the research content of this thesis, the issue of energy optimization has not been considered. Therefore, the energy optimization problem should be considered.

Furthermore, the UAVs must be aware of the remaining energy level and be able to return to the given locations before running out of energy.

2. For large agricultural areas, one UAV cannot completely perform the CPP task. Hence the expansion of the CPP problem for a team of UAVs is necessary. In addition, even if one quadrotor is totally damaged or in trouble, others may take over the coverage subtask.
3. The proposed coverage path-planning algorithm considers only off-line in $2D$ work-space with static obstacles. The $3D$ dynamic environment could be an interesting extension for the CPP.
4. The trajectory generated by the proposed CPP does not consider the dynamic constraints of the UAV. For example, the UAV is not able to make a sharp turn. We have to consider also constraints on the limit speed and acceleration of the UAV for the CPP problem.
5. Several extensions from this section are possible. One might consider recalculating the trajectory under the windy condition of the environment or trajectory generation for a team of UAV. Field tests are also subject of future work. The continuation of this work is also to add the UAV equations of motion as constraints on the path. Then the efficiency of the algorithm will be improved to be real-time usable.

- Control issues
    1. The mass and moments of inertia of the quadrotor in the research content of this thesis are calculated according to the amount of pesticide sprayed over time. An extension of this could be the on-line estimation of the quadrotor mass and moments of inertia according to parameters measured from sensors such as accelerometer, gyroscope, etc
    2. The control laws need to be simulated on Robot Operating System (ROS), developed on embedded computers, and applied to the real platform quadrotors.

A Preliminaries

This section presents some preliminaries about gain-scheduling, with particular emphasis on Linear Parameter Varying (LPV) systems. Then, we also review analysis and design of LPV control systems using the quadratic framework and discussing several possible specifications, like stability, $\mathcal{D}$-stability (pole clustering in a subset of the complex plane), H_∞/H_2 performance, and finite-time stability/boundedness.

A.0.1 Linear Parameter Varying (LPV) system

Definition A.1. *Linear Parameter Varying (LPV) systems [205][204][152] are linear dynamical systems whose mathematical description depends on parameters that change values over time. The system matrices are fixed functions of some varying parameters $\rho \in \mathbb{R}^{n_\rho}$, assumed to be unknown a priori but measured or estimated in real-time[203]. LPV systems are described by equations of the form as follows:*

$$\begin{array}{rcl} \sigma \cdot x(\tau) & = & A(\rho(\tau))\,x(\tau) + B(\rho(\tau)\,u(t)), \;\; \tau \geq 0 \\ y(\tau) & = & C(\rho(\tau))\,x(\tau) + D(\rho(\tau)\,u(\tau)) \\ x(0) & = & x_0 \end{array} \tag{A.1}$$

where

- *The notation $\sigma \cdot x(\tau)$ stands for $\dot{x}(\tau)$ for continuous systems and for $x(k+1)$ for discrete systems.*
- *$x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $y \in \mathbb{R}^{n_y}$ are the state, input, and output vectors respectively. The system matrices are $A(\rho(\tau)) \in \mathbb{R}^{n_x \times n_x}$, $B(\rho(\tau)) \in \mathbb{R}^{n_x \times n_u}$, $C(\rho(\tau)) \in \mathbb{R}^{n_y \times n_x}$, and $D(\rho(\tau)) \in \mathbb{R}^{n_y \times n_u}$.*

□

Remark A.1. *The equation (A.1) describes both Continuous Time Linear Parameter Varying (CT LPV) and Discrete Time Linear Parameter Varying (DT LPV).*

- *For CT LPV, $\tau = t$, and (A.1) becomes*

$$\begin{array}{rcl} \dot{x}(t) & = & A(\rho(t))\,x(t) + B(\rho(t)\,u(t)), \qquad t \geq 0 \\ y(t) & = & C(\rho(t))\,x(t) + D(\rho(t)\,u(t)) \\ x(0) & = & x_0 \end{array} \tag{A.2}$$

- *For DT LPV, $\tau = k$, and (A.1) becomes*

$$\begin{array}{rcl} x(k+1) & = & A(\rho(k))\,x(k)+B(\rho(k)\,u(k)), \qquad k \geq 0 \\ y(k) & = & C(\rho(k))\,x(k)+D(\rho(k)\,u(k)) \\ x(0) & = & x_0 \end{array} \tag{A.3}$$

☐

Definition A.2. *(Generic Parameter Dependent system [46]) The Generic Parameter Dependent system is the system in the form*

$$\begin{array}{rcl} \sigma\cdot x(\tau) & = & A(\rho(\tau))\,x(\tau)+B_u(\rho(\tau))\,u(t)+B_w(\rho(\tau))\,w(t) \\ z(\tau) & = & C_z(\rho(\tau))\,x(\tau)+D_{zu}(\rho(\tau))\,u(t)+D_{zw}(\rho(\tau))\,w(t) \\ y(\tau) & = & C_y(\rho(\tau))\,x(\tau)+D_{yw}(\rho(\tau))\,w(t) \\ x(0) & = & x_0 \end{array} \tag{A.4}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$, $w \in \mathbb{R}^{n_w}$, and , $z \in \mathbb{R}^{n_z}$ are the state, input, measured output, exogenous input, and controlled output vectors respectively. The system matrices are $A(\rho(\tau)) \in \mathbb{R}^{n_x \times n_x}$, $B_u(\rho(\tau)) \in \mathbb{R}^{n_x \times n_u}$, $B_w(\rho(\tau)) \in \mathbb{R}^{n_x \times n_w}$, $C_z(\rho(\tau)) \in \mathbb{R}^{n_z \times n_x}$, $D_{zu}(\rho(\tau)) \in \mathbb{R}^{n_z \times n_u}$, $D_{zw}(\rho(\tau)) \in \mathbb{R}^{n_z \times n_w}$, $C_y(\rho(\tau)) \in \mathbb{R}^{n_y \times n_x}$, $D_{yu}(\rho(\tau)) \in \mathbb{R}^{n_y \times n_u}$, $D_{yw}(\rho(\tau)) \in \mathbb{R}^{n_y \times n_w}$. ☐

Remark A.2. *The equation (A.4) describes both continuous time and discrete time generic parameter dependent system according to σ.*

- *For continuous time and generic parameter dependent system, $\tau = t$, and (A.4) becomes*

$$\begin{array}{rcl} \dot{x}(t) & = & A(\rho(t))\,x(t)+B_u(\rho(t))\,u(t)+B_w(\rho(t))\,w(t) \\ z(t) & = & C_z(\rho(t))\,x(t)+D_{zu}(\rho(t))\,u(t)+D_{zw}(\rho(t))\,w(t) \\ y(t) & = & C_y(\rho(t))\,x(t)+D_{yu}(\rho(t))\,u(t) \end{array} \tag{A.5}$$

- *For discrete time generic parameter dependent system, $\tau = k$, and (A.4) becomes*

$$\begin{array}{rcl} x(k+1) & = & A(\rho(k))\,x(k)+B_u(\rho(k))\,u(t)+B_w(\rho(k))\,w(t) \\ z(k) & = & C_z(\rho(k))\,x(k)+D_{zu}(\rho(k))\,u(t)+D_{zw}(\rho(k))\,w(t) \\ y(k) & = & C_y(\rho(k))\,x(k)+D_{yu}(\rho(k))\,u(t) \end{array} \tag{A.6}$$

☐

Definition A.3. *The N-unit simplex, denoted by Λ_N, is defined as the set*

$$\Lambda_N := \left\{ \chi \in \mathbb{R}^N_{\geq 0} : \sum_{i=1}^{N} \chi_i = 1 \right\} \tag{A.7}$$

☐

Definition A.4. *The LPV system in (A.1) is called polytopic if its state space system matrices can be represented as follows*

$$\begin{aligned} A(\rho(\tau)) &= \sum_{i=1}^{N} \mu_i(\tau) A_i \\ B(\rho(\tau)) &= \sum_{i=1}^{N} \mu_i(\tau) B_i \\ C(\rho(\tau)) &= \sum_{i=1}^{N} \mu_i(\tau) C_i \\ D(\rho(\tau)) &= \sum_{i=1}^{N} \mu_i(\tau) D_i \end{aligned} \tag{A.8}$$

where $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $C_i \in \mathbb{R}^{n_y \times n_x}$, *and* $D_i \in \mathbb{R}^{n_y \times n_u}$ *for* $i = 1,...,n$ *and* $\mu(t) \in \Lambda_N$. ☐

From definition A.4, the LPV polytopic system in (A.1) can be rewritten as

$$\begin{aligned} \sigma \cdot x(\tau) &= \sum_{i=1}^{N} \mu_i(\tau)\left[A_i x(\tau) + B_i u(\tau)\right] \\ y(\tau) &= \sum_{i=1}^{N} \mu_i(\tau)\left[C_i x(\tau) + D_i u(\tau)\right] \end{aligned} \tag{A.9}$$

where the quadruples (A_i, B_i, C_i, D_i) define the so-called vertex systems, and $\mu_i, i = 1,...,n$ are the coefficients of the polytopic decomposition such that

$$\sum_{i=1}^{N} \mu_i(\rho(\tau)) = 1, \ \ \mu_i(\rho(\tau)) \geq 0, \ \ \forall i = 1,...,N, \ \ \forall \rho \in \Theta \tag{A.10}$$

Definition A.5. *(Poles of an LPV system [82]) Given an autonomous LPV system*

$$\sigma \cdot x(\tau) = A(\rho(\tau))\, x(\tau) \tag{A.11}$$

where the state vector $x \in \mathbb{R}^{n_x}$, *the varying parameter vector* $\rho(\tau) \in \Theta \subset \mathbb{R}^{n_\rho}$, *and the system matrix* $A(\rho(\tau)) \in \mathbb{R}^{n_x \times n_x}$. *The poles of (A.11) are the set of all the Linear Time Invariant (LTI) system obtained by freezing* $\rho(\tau)$ *for all* $\rho \in \Theta$. ☐

Definition A.6. *($\mathcal{D}$-stability of an LPV system) Given a subset $\mathcal{D}$ of the complex plane. The autonomous LPV system (A.11) is called to be $\mathcal{D}$-stalbe if all its poles lie in $\mathcal{D}$.* ☐

Definition A.7. *(LMI regions [43]) A subset $\mathcal{D}$ of the complex plane is called a LMI region if there exist matrices* $\alpha = [\alpha_{kl}]_{k,l \in \{1,...,m\}} \in S^{m \times m}$ *and* $\beta = [\beta_{kl}]_{k,l \in \{1,...,m\}} \in S^{m \times m}$ *such that*

$$\mathcal{D} = \{z \in C : f_{\mathcal{D}}(z) \prec 0\} \tag{A.12}$$

where $f_{\mathcal{D}}$ *is the characteristic function*

$$f_D(z) = \alpha + \beta z + \beta^T z^* = \left[\alpha_{kl} + \beta_{kl} z + \beta_{kl} z^*\right]_{k,l \in \{1,...,m\}} \tag{A.13}$$

There are several typical LMI regions as follows:

- *Left-hand semiplanes* $\mathrm{Re}(z) < \lambda$

$$\alpha = -2\lambda \qquad \beta = 1$$

- *Right-hand semiplanes* $\mathrm{Re}(z) > \lambda$

$$\alpha = 2\lambda \qquad \beta = -1$$

- *Disks of radius* r *and center* $(-c, 0)$

$$\alpha = \begin{bmatrix} -r & c \\ c & -r \end{bmatrix} \qquad \beta = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

- *Horizontal strips* $-\omega \le \mathrm{Im}(z) \le \omega$

$$\alpha = \begin{bmatrix} -2\omega & 0 \\ 0 & -2\omega \end{bmatrix} \qquad \beta = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

□

Definition A.8. *(H_∞ norm [78]) The H_∞ norm of a stable real-rational transfer matrix $T(\sigma)$ of CT and DT systems are defined as*

- *for CT system:*

$$\|T(s)\|_\infty = \sup_{\omega \in R} \sigma_{\max}(T(j\omega)) \tag{A.14}$$

- *for DT system:*

$$\|T(z)\|_\infty = \sup_{\omega \in [-\pi,\pi]} \sigma_{\max}\left(T\left(e^{j\omega}\right)\right) \tag{A.15}$$

where $\sigma_{max}(*)$ is the largest singular value of the matrix $*$.

H_∞-norm is an important quantity for measuring robust stability, error in model order reduction. The H_∞-norm can also be used to measuring the system input-output gain of finite energy signals. □

Definition A.9. *(H_∞ performance of an LPV system [202]) Given a LPV system defined as follows:*

$$\begin{array}{rcl} \sigma \cdot x(t) & = & A(\rho(t))\,x(t) + B(\rho(t)\,\omega(t)) \\ z(t) & = & C(\rho(t))\,x(t) + D(\rho(t)\,\omega(t)) \end{array} \tag{A.16}$$

γ_∞ is called the $\mathcal{H}_\infty$ performance of the system (A.16) if $\|T_{zw}(\sigma,\rho)\|_\infty < \gamma_\infty$ for all $\rho \in \Theta$, where $T_{zw}(\sigma,\rho)$ is the closed-loop transfer function from $w(t)$ to $z(t)$. □

Definition A.10. *(H_2 norm [202]) Given a stable real-rational transfer matrix $T(\sigma)$. The $\mathcal{H}_2$ norm of CT and DT T is defined as:*

- *for CT system:*

$$\|T(s)\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} Tr\left[T(j\omega)^T T(j\omega)\right] d\omega} \tag{A.17}$$

- *for DT system:*

$$\|T(z)\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} Tr\left[T(e^{j\omega})^T T(e^{j\omega})\right] d\omega} \tag{A.18}$$

where $Tr\,(*)$ *is the trace of the matrix* $*$. ☐

Definition A.11. *(H_2 performance of a LPV system [202]) Given a LPV system defined as follows:*

$$\begin{array}{rcl} \sigma \cdot x\,(t) & = & A\,(\rho\,(t))\,x\,(t) + B\,(\rho\,(t)\,\omega\,(t)) \\ z\,(t) & = & C\,(\rho\,(t))\,x\,(t) \end{array} \tag{A.19}$$

γ_2 *is called the* $\mathcal{H}_2$ *performance of the system (A.19) if* $\|T_{zw}\,(\sigma,\rho)\|_2 < \gamma_2$ *for all* $\rho \in \Theta$, *where* $T_{zw}\,(\sigma,\rho)$ *is the closed-loop transfer function from* $w\,(t)$ *to* $z\,(t)$.

Note that, the H_2 performance is really effective for handling stochastic aspects such as measurement noise and random disturbances. ☐

Theorem A.1. *(Quadratic stability of CT LPV systems [23][83]) The autonomous CT LPV system in (A.11) is quadratically stable*

1. *if there exists a positive definite matrix P such that*

$$A(\rho)^T P + PA\,(\rho) \prec 0, \qquad \forall \rho \in \Theta \tag{A.20}$$

2. *if there exists a positive definite matrix Q such that*

$$QA(\rho)^T + A\,(\rho)\,Q \prec 0, \qquad \forall \rho \in \Theta \tag{A.21}$$

■

Theorem A.2. *(Quadratic stability of DT LPV systems [118][73]) The autonomous DT LPV system in (A.11) is quadratically stable*

1. *if there exists a positive definite matrix P such that*

$$A(\rho)^T PA - P\,(\rho) \prec 0, \qquad \forall \rho \in \Theta \tag{A.22}$$

2. *if there exists a positive definite matrix P such that*

$$\left[\begin{array}{cc} -P & PA\,(\rho) \\ A(\rho)^T P & -P \end{array}\right] \prec 0, \qquad \forall \rho \in \Theta \tag{A.23}$$

3. *if there exists a positive definite matrix P such that*

$$\left[\begin{array}{cc} -Q & A\,(\rho)\,Q \\ QA(\rho)^T & -Q \end{array}\right] \prec 0, \qquad \forall \rho \in \Theta \tag{A.24}$$

■

Theorem A.3. *(Quadratic D-stability of LPV systems [160]) Given a LMI region $\mathcal{D}$ defined as in (A.12), the autonomous LPV system:*

$$\sigma \cdot x\,(t) = A\,(\rho)\,x\,(t) \tag{A.25}$$

The system (A.25) is said to be quadratically $\mathcal{D}$-stable:

1. *if there exists* $P \succ 0$ *such that:*

$$\begin{aligned} &\alpha \otimes P + \beta \otimes PA + \beta^T \otimes A^T P \\ &= \left[\alpha_{kl} P + \beta_{kl} PA + \beta_{kl} A^T P\right]_{k,l \in \{1,\ldots,m\}} \prec 0, \quad \forall \rho \in \Theta \end{aligned} \tag{A.26}$$

2. *if there exists* $Q \succ 0$ *such that:*

$$\begin{aligned} &\alpha \otimes Q + \beta \otimes AQ + \beta^T \otimes QA^T \\ &= \left[\alpha_{kl} Q + \beta_{kl} AQ + \beta_{kl} QA^T\right]_{k,l \in \{1,\ldots,m\}} \prec 0, \quad \forall \rho \in \Theta \end{aligned} \tag{A.27}$$

■

Theorem A.4. *(Quadratic* H_∞ *performance of CT LPV systems [43]) The CT LPV system has quadratic* $\mathcal{H}_\infty$ *performance* γ_∞

1. *if there exists* $P \succ 0$ *such that:*

$$\begin{bmatrix} A(\rho)^T P + PA(\rho) & PB(\rho) & C(\rho)^T \\ B(\rho)^T P & -I & D(\rho)^T \\ C(\rho) & D(\rho) & -\gamma_\infty^2 I \end{bmatrix} \prec 0, \quad \forall \rho \in \Theta \tag{A.28}$$

or equivalent to [15]

$$\begin{bmatrix} A(\rho)^T P + PA(\rho) & PB(\rho) & C(\rho)^T \\ B(\rho)^T P & -\gamma_\infty I & D(\rho)^T \\ C(\rho) & D(\rho) & -\gamma_\infty I \end{bmatrix} \prec 0, \quad \forall \rho \in \Theta \tag{A.29}$$

2. *if there exists* $Q \succ 0$ *such that:*

$$\begin{bmatrix} A(\rho) Q + QA(\rho)^T & B(\rho) & QC(\rho)^T \\ B(\rho)^T & -I & D(\rho)^T \\ C(\rho) Q & D(\rho) & -\gamma_\infty^2 I \end{bmatrix} \prec 0, \quad \forall \rho \in \Theta \tag{A.30}$$

or equivalent to [15]

$$\begin{bmatrix} A(\rho) Q + QA(\rho)^T & B(\rho) & QC(\rho)^T \\ B(\rho)^T & -I & D(\rho)^T \\ C(\rho) Q & D(\rho) & -\gamma_\infty I \end{bmatrix} \prec 0, \quad \forall \rho \in \Theta \tag{A.31}$$

■

Theorem A.5. *(Quadratic* H_∞ *performance of DT LPV systems) The DT LPV system has quadratic* $\mathcal{H}_\infty$ *performance* γ_∞

1. *if there exists* $P \succ 0$ *such that:*

$$\begin{bmatrix} P & PA(\rho) & PB(\rho) & 0 \\ A(\rho)^T P & P & 0 & C(\rho)^T \\ B(\rho)^T P & 0 & \gamma_\infty^2 I & D(\rho)^T \\ 0 & C(\rho) & D(\rho) & I \end{bmatrix} \succ 0, \quad \forall \rho \in \Theta \tag{A.32}$$

or equivalent to [15]

$$\begin{bmatrix} P & PA(\rho) & PB(\rho) & 0 \\ A(\rho)^T P & P & 0 & C(\rho)^T \\ B(\rho)^T P & 0 & \gamma_\infty I & D(\rho)^T \\ 0 & C(\rho) & D(\rho) & \gamma_\infty I \end{bmatrix} \succ 0, \quad \forall \rho \in \Theta \tag{A.33}$$

2. *if there exists $Q \succ 0$ such that:*

$$\begin{bmatrix} Q & A(\rho)Q & B(\rho) & 0 \\ QA(\rho)^T & Q & 0 & QC(\rho)^T \\ B(\rho)^T & 0 & \gamma_\infty^2 I & D(\rho)^T \\ 0 & C(\rho)Q & D(\rho) & I \end{bmatrix} \succ 0, \qquad \forall \rho \in \Theta \tag{A.34}$$

or equivalent to [15]

$$\begin{bmatrix} Q & A(\rho)Q & B(\rho) & 0 \\ QA(\rho)^T & Q & 0 & QC(\rho)^T \\ B(\rho)^T & 0 & \gamma_\infty & D(\rho)^T \\ 0 & C(\rho)Q & D(\rho) & \gamma_\infty I \end{bmatrix} \succ 0, \quad \forall \rho \in \Theta \tag{A.35}$$

■

The problem with the conditions provided in Theorems A.1-A.5 is that they rely on the satisfaction of infinite number of constraints. This is really hard or even impossible for satisfaction. This difficulty can be overcome by considering the polytopic form [46] which allow to extract tractable conditions as in the following results.

Theorem A.6. *(Quadratic stability of polytopic CT LPV systems[191]) Given an autonomous polytopic CT LPV system*

$$\dot{x}(t) = \sum_{i=1}^{N} \mu_i(\rho(t)) A_i x(t) \tag{A.36}$$

where $\mu_i, i = 1, ..., N$ satisfies (A.10). The system (A.36) is said to be quadratically stable:

1. *if there exists $P \succ 0$ such that:*

$$A_i^T P + PA_i \prec 0, \qquad \forall i = 1, ..., N \tag{A.37}$$

2. *if there exists $Q \succ 0$ such that:*

$$QA_i^T + A_i Q \prec 0, \qquad \forall i = 1, ..., N \tag{A.38}$$

■

Theorem A.7. *(Quadratic stability of polytopic DT LPV systems[191]) Given an autonomous polytopic DT LPV system*

$$x(k+1) = \sum_{i=1}^{N} \mu_i(\rho(k)) A_i x(k) \tag{A.39}$$

where $\mu_i, i = 1, ..., N$ satisfies (A.10). The system (A.39) is said to be quadratically stable:

1. *if there exists $P \succ 0$ such that:*

$$\begin{bmatrix} -P & PA_i \\ -A_i^T P & -P \end{bmatrix} \prec 0, \quad \forall i = 1, ..., N \tag{A.40}$$

2. *if there exists $Q \succ 0$ such that:*

$$\begin{bmatrix} -Q & A_i Q \\ -Q A_i^T & -Q \end{bmatrix} \prec 0, \quad \forall i = 1, ..., N \tag{A.41}$$

■

Theorem A.8. *(Quadratic D-stability of polytopic LPV systems[191]) Given an LMI region $\mathcal{D}$ defined as in (A.12), the autonomous polytopic LPV system:*

$$\sigma \cdot x(t) = \sum_{i=1}^{N} \mu_i(\rho(t)) A_i x(t) \tag{A.42}$$

where $\mu_i, i = 1, ..., N$ satisfies (A.10). The system (A.42) is said to be quadratically $\mathcal{D}$-stable:

1. *if there exists $P \succ 0$ such that:*

$$\begin{aligned} &\alpha \otimes P + \beta \otimes PA_i + \beta^T \otimes A_i^T P \\ &= \left[\alpha_{kl} P + \beta_{kl} PA_i + \beta_{kl} A_i^T P\right]_{k,l \in \{1,...,m\}} \prec 0, \quad \forall i = 1, ..., N \end{aligned} \tag{A.43}$$

2. *if there exists $Q \succ 0$ such that:*

$$\begin{aligned} &\alpha \otimes Q + \beta \otimes A_i Q + \beta^T \otimes Q A_i^T \\ &= \left[\alpha_{kl} Q + \beta_{kl} A_i Q + \beta_{kl} Q A_i^T\right]_{k,l \in \{1,...,m\}} \prec 0, \quad \forall i = 1, ..., N \end{aligned} \tag{A.44}$$

■

Theorem A.9. *(Quadratic H_∞ performance of polytopic CT LPV systems[191]) Given an CT LPV system defined as follows:*

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{N} \mu_i(\rho(t)) \left[A_i x(t) + B_i \omega(t)\right] \\ z(t) &= \sum_{i=1}^{N} \mu_i(\rho(t)) \left[C_i x(t) + D_i \omega(t)\right] \end{aligned} \tag{A.45}$$

where $\mu_i, i = 1, ..., N$ satisfies (A.10). The system (A.45) has quadratic H_∞ performance γ_∞:

1. *if there exists $P \succ 0$ such that:*

$$\begin{bmatrix} A_i^T P + PA_i & PB_i & C_i^T \\ B_i^T P & -I & D_i^T \\ C_i & D_i & -\gamma_\infty^2 I \end{bmatrix} \prec 0, \ \forall i = 1, ..., N \tag{A.46}$$

2. *if there exists* $Q \succ 0$ *such that:*

$$\begin{bmatrix} A_iQ+QA_i^T & B_i & QC_i^T \\ B_i^TP & -I & D_i^T \\ C_iQ & D_i & -\gamma_\infty^2 I \end{bmatrix} \prec 0, \quad \forall i=1,...,N \tag{A.47}$$

■

Theorem A.10. *(Quadratic* H_∞ *performance of polytopic DT LPV systems[191]) Given an DT LPV system defined as follows:*

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{N} \mu_i(\rho(k))[A_ix(k)+B_iw(k)] \\ z(k) &= \sum_{i=1}^{N} \mu_i(\rho(k))[C_ix(k)+D_iw(k)] \end{aligned} \tag{A.48}$$

where $\mu_i, i=1,...,N$ *satisfies (A.10). The system (A.48) has quadratic* H_∞ *performance* γ_∞*:*

1. *if there exists* $P \succ 0$ *such that:*

$$\begin{bmatrix} P & PA_i & PB_i & 0 \\ A_i^TP & P & 0 & C_i^T \\ B_i^TP & 0 & I & D_i^T \\ 0 & C_i & D_i & \gamma_\infty^2 I \end{bmatrix} \succ 0, \quad \forall \rho \in \Theta \tag{A.49}$$

2. *if there exists* $Q \succ 0$ *such that:*

$$\begin{bmatrix} Q & A_iQ & B_i & 0 \\ QA_i^T & Q & 0 & QC_i^T \\ B_i^T & 0 & I & D_i^T \\ 0 & C_iQ & D_i & \gamma_\infty^2 I \end{bmatrix} \succ 0, \quad \forall \rho \in \Theta \tag{A.50}$$

■

Theorem A.11. *(Quadratic Stabilization of CT LPV systems - State feedback[46]) Consider CT LPV system given in the first equation of (A.1). This CT LPV system is quadratically stable by a state feedback control law given by*

$$u(t) = K(\rho(t))x(t) \tag{A.51}$$

if there exists $Q \succ 0$ *and* $K(\rho(t)) \in R^{n_u \times n_x}$ *such that*

$$He\{A(\rho)Q + B(\rho)K(\rho)Q\} \prec 0, \qquad \forall \rho \in \Theta \tag{A.52}$$

■

Theorem A.12. *(Quadratic Stabilization of DT LPV systems - State feedback[191]) Consider DT LPV system given in the first equation of (A.1). This DT LPV system is quadratically stable by a state feedback control law given by*

$$u(t) = K(\rho(t))x(t) \tag{A.53}$$

if there exist $Q \succ 0$ *and* $K(\rho(t)) \in R^{n_u \times n_x}$ *such that*

$$\begin{bmatrix} -Q & A(\rho)Q + B(\rho)K(\rho)Q \\ (A(\rho)Q + B(\rho)K(\rho)Q)^T & -Q \end{bmatrix} \prec 0, \qquad \forall \rho \in \Theta \tag{A.54}$$

■

Theorem A.13. *(Quadratic D-Stabilizability of LPV systems - State feedback[202]) Consider an LMI region given in (A.12). This LPV given in the first equation of (A.1) is quadratically stable by a state feedback control law given by*

$$u(t) = K(\rho(t))x(t) \tag{A.55}$$

if there exist $Q \succ 0$ *and* $K(\rho(t)) \in R^{n_u \times n_x}$ *such that*

$$\alpha \otimes Q + He\{\beta \otimes [A(\rho)Q + B(\rho)K(\rho)Q]\} \prec 0, \qquad \forall \rho \in \Theta \tag{A.56}$$

■

Theorem A.14. *(Quadratic H_∞ State feedback for CT LPV systems[202]) Given the CT LPV system*

$$\begin{array}{rcl} \dot{x}(t) & = & A(\rho(t))x(t) + B_1(\rho(t))w(t) + B_2(\rho(t)u(t)) \\ z(t) & = & C_1(\rho(t))x(t) + D_{11}(\rho(t)w(t)) + D_{12}(\rho(t)u(t)) \end{array} \tag{A.57}$$

The CT LPV system (A.57) with state feedback control law has quadratic H_∞ performance γ_∞ if there exist $Q \succ 0$ and $K(\rho(t)) \in R^{n_u \times n_x}$ such that

$$\begin{bmatrix} He[A(\rho)Q + B(\rho)K(\rho)Q] & * & * \\ B_1(\rho)^T & -I & * \\ C_1(\rho)Q + D_{12}(\rho)K(\rho)Q & D_{11}(\rho) & -\gamma_\infty^2 I \end{bmatrix} \prec 0, \qquad \forall \rho \in \Theta \tag{A.58}$$

■

Remark A.3. *It is possible to consider γ_∞ as a variable, then $\bar{\gamma}_\infty = \gamma_\infty^2$. Consequently, the LMI in Theorem A.14 can be rewritten as follows:*

$$\begin{bmatrix} He[A(\rho)Q + B(\rho)K(\rho)Q] & * & * \\ B_1(\rho)^T & -I & * \\ C_1(\rho)Q + D_{12}(\rho)K(\rho)Q & D_{11}(\rho) & -\bar{\gamma}_\infty I \end{bmatrix} \prec 0, \qquad \forall \rho \in \Theta \tag{A.59}$$

A.0.2 Observability and Detectability of LPV systems

Theorem A.15. *(Obsevability) LPV system (A.1) is quadratically detectable, if there exists a matrix $P \succ 0$ and a matrix function $L(\rho)$ such that:*

$$(A(\rho(t)) + L(\rho(t))C)^T P + P(A(\rho(t)) + L(\rho(t))C) \prec 0, \ \forall \rho(t) \tag{A.60}$$

■

A.0.3 Filtering the input

In many cases, the inputs matrices of polytopic LPV systems can be varying and leads to the difficulty in controller designing. However, by using appropriate low-pass filter [14] and augmenting the system with the state of the filter, we can change the system to new system with constant inputs matrices.

Given an LPV system

$$\sigma\cdot x(\tau) = A(\rho(\tau))x(\tau) + B_1(\rho(\tau))w(\tau) + B_2(\rho(\tau))u(\tau) \tag{A.61a}$$

$$z_\infty(\tau) = C_{1_\infty}(\rho(\tau))x(\tau) + D_{11_\infty}(\rho(\tau))w(\tau) + D_{12_\infty}(\rho(\tau))u(\tau) \tag{A.61b}$$

$$z_2(\tau) = C_{1_2}(\rho(\tau))x(\tau) + D_{11_2}(\rho(\tau))w(\tau) + D_{12_2}(\rho(\tau))u(\tau) \tag{A.61c}$$

Define a new control input $\tilde{u}(\tau)$ such that

$$\sigma\cdot x_u(\tau) = A_u(\rho(\tau))x(\tau) + B_u\tilde{u}(\tau) \tag{A.62a}$$

$$u(\tau) = C_u x_u(\tau) \tag{A.62b}$$

where $A_u(\rho(\tau))$ is parameter varying stable matrix, while B_u and C_u are constant matrices.

From (A.62), we obtain new LPV system

$$\begin{bmatrix} \sigma\cdot x(\tau) \\ \sigma\cdot x_u(\tau) \end{bmatrix} = \begin{bmatrix} A(\rho(k)) & B(\rho(\tau))C_u \\ 0 & A_u(\rho(\tau)) \end{bmatrix} \begin{bmatrix} x(\tau) \\ x_u(\tau) \end{bmatrix} + \begin{bmatrix} B_1(\rho(\tau)) \\ 0 \end{bmatrix} w(\tau) + \begin{bmatrix} 0 \\ B_u \end{bmatrix} \tilde{u}(\tau) \tag{A.63a}$$

$$z_\infty(\tau) = \begin{bmatrix} C_{1_\infty}(\rho(\tau)) & D_{12_\infty}(\rho(\tau)) \end{bmatrix} \begin{bmatrix} x(\tau) \\ x_u(\tau) \end{bmatrix} + D_{11_\infty}(\rho(\tau))w(\tau) \tag{A.63b}$$

$$z_2(\tau) = \begin{bmatrix} C_{1_2}(\rho(\tau)) & D_{12_2} \end{bmatrix} \begin{bmatrix} x(\tau) \\ x_u(\tau) \end{bmatrix} \tag{A.63c}$$

We can rewrite (A.61) as

$$\begin{array}{rcl} \sigma\cdot\tilde{x}(\tau) & = & \tilde{A}(\rho(\tau))\tilde{x}(\tau) + \tilde{B}_1(\rho(\tau))w(\tau) + \tilde{B}_2\tilde{u}(\tau) \\ z_\infty(\tau) & = & \tilde{C}_{1_\infty}(\rho(\tau))\tilde{x}(\tau) + \tilde{D}_{11_\infty}(\rho(\tau))w(\tau) \\ z_2(\tau) & = & \tilde{C}_{1_2}(\rho(\tau))\tilde{x}(\tau) \end{array} \tag{A.64}$$

where

$$\tilde{A}(\rho(\tau)) = \begin{bmatrix} A(\rho(k)) & B(\rho(\tau))C_u \\ 0 & A_u(\rho(\tau)) \end{bmatrix}; \tilde{B}_1(\rho(\tau)) = \begin{bmatrix} B_1(\rho(\tau)) \\ 0 \end{bmatrix}; \tilde{B}_2 = \begin{bmatrix} 0 \\ B_u \end{bmatrix}$$

$$\tilde{C}_{1_\infty}(\rho(\tau)) = \begin{bmatrix} C_{1_\infty}(\rho(\tau)) & D_{12_\infty}(\rho(\tau)) \end{bmatrix}; \tilde{D}_{11_\infty}(\rho(\tau)) = D_{11_\infty}(\rho(\tau))$$

$$\tilde{C}_{1_2}(\rho(\tau)) = \begin{bmatrix} C_{1_2}(\rho(\tau)) & D_{12_2} \end{bmatrix}$$

A.0.4 Controller design for polytopic LPV systems

In this section, we review some results on designing the state feedback controller of the form

$$u(t) = \sum_{i=1}^{N} \mu_i(\rho(t))K_i x(t) \tag{A.65}$$

for polytopic LPV system.

In this case, the matrices $A(\rho(t))$, $B_1(\rho(t))$, $C_1(\rho(t))$, $D_{11}(\rho(t))$, $C_2(\rho(t))$, $W(\rho(t))$ are assumed to be in the polytopic form as follows

$$\begin{bmatrix} A(\rho(t)) \\ B_1(\rho(t)) \\ C_1(\rho(t)) \\ D_{11}(\rho(t)) \\ C_2(\rho(t)) \\ W(\rho(t)) \end{bmatrix} = \sum_{i=1}^{N} \mu_i(\rho(t)) \begin{bmatrix} A_i \\ B_{1,i} \\ C_{1,i} \\ D_{11,i} \\ C_{2,i} \\ W_i \end{bmatrix} \tag{A.66}$$

Theorem A.16. *(Quadratic H_∞ State feedback for DT LPV systems[191]) Given the DT LPV system*

$$\begin{array}{rcl} x(k+1) & = & A(\rho(k))\,x(k) + B_1(\rho(k))\,w(k) + B_2(\rho(k)\,u(k)) \\ z(k) & = & C_1(\rho(t))\,x(k) + D_{11}(\rho(k)\,w(k)) + D_{12}(\rho(k)\,u(k)) \end{array} \tag{A.67}$$

The CT LPV system (A.67) with state feedback control law has quadratic H_∞ performance γ_∞ if there exist $Q \succ 0$ and $K(\rho(t)) \in R^{n_u \times n_x}$ such that

$$\begin{bmatrix} Q & A(\rho)Q + B(\rho)K(\rho)Q & B_1(\rho) & 0 \\ * & Q & 0 & QC_1(\rho)^T + QK(\rho)^T D_{12}(\rho)^T \\ * & * & I & D_{11}(\rho) \\ * & * & * & \gamma_\infty^2 I \end{bmatrix} \succ 0, \qquad \forall \rho \in \Theta \tag{A.68}$$

■

Theorem A.17. *(Quadratically stabilizing polytopic state-feedback controller for CT LPV systems[191]) The CT LPV system is quadratically stable by a state feedback of the form (A.65) if there exist $Q \succ 0$ and $\Gamma_i \in R^{n_u \times n_x}, i = 1, ..., N$ such that*

$$\begin{bmatrix} -Q & A_iQ + B\Gamma_i \\ (A_iQ + B\Gamma_i)^T & -Q \end{bmatrix} \prec 0, \ \forall i = 1, ..., N \tag{A.69}$$

Then the polytopic state feedback control gain can be calculated as $K_i = \Gamma_i Q^{-1}, \ \forall i = 1, ..., N$.

■

Theorem A.18. *(Quadratically stabilizing polytopic state-feedback controller for DT LPV systems[191]) The DT LPV system is quadratically stabilizable by a state feedback of the form (A.65) if there exist $Q \succ 0$ and $\Gamma_i \in R^{n_u \times n_x}, i = 1, ..., N$ such that*

$$He\{A_iQ + B\Gamma_i\} \prec 0, \ \forall i = 1, ..., N \tag{A.70}$$

Then the polytopic state feedback control gain can be calculated as $K_i = \Gamma_i Q^{-1}, \ \forall i = 1, ..., N$.

■

Theorem A.19. *(Quadratically D-stabilizing polytopic state-feedback controller for LPV systems[191]) Given an LMI region $\mathcal{D}$ defined as in (A.12). Let $Q \succ 0$ and $\Gamma_i \in R^{n_u \times n_x}, i = 1, ..., N$ such that*

$$\alpha \otimes Q + He\{\beta \otimes (A_iQ + B\Gamma_i)\} \prec 0, \ \forall i = 1, ..., N \tag{A.71}$$

Then, the closed-loop system made up by the LPV system (A.1) for the case $B(\rho(t)) = B$ *and polytopic system matrices in (A.66) and the polytopic state-feedback control law given by (A.65) with the gains calculated as* $K_i = \Gamma_i Q^{-1}, i = 1,...,N$*, is quadratically* $\mathcal{D}$ *stable.* ■

Theorem A.20. *(Quadratic* H_∞ *polytopic State feedback control for CT LPV systems[191]) The polytopic CT LPV is quadratically stabilizable by a state feedback of the form (A.65) if and only if there exist a matrix* $Q \succ 0$ *and* $\Gamma_i \in R^{n_u \times n_x}, i = 1,...,N$ *such that*

$$\begin{bmatrix} He\{A_iQ + B\Gamma_i\} & * & * \\ B_{1,i}^T & -I & * \\ C_{1_\infty i}Q + D_{1_\infty 2}\Gamma_i & D_{1_\infty 2} & -\gamma_\infty^2 I \end{bmatrix} \prec 0, \ \forall i = 1,...,N \tag{A.72}$$

Moreover, the state feedback control law given by (A.65) with matrices $K_i = \Gamma_i Q^{-1}$, $\forall i = 1,...,N$ *has the quadratic* H_∞ *performance* γ_∞*.* ■

Theorem A.21. *(Quadratic* H_∞ *polytopic State feedback control for DT LPV systems[191]) The polytopic DT LPV is quadratically stabilizable by a state feedback of the form (A.65) if and only if there exist a matrix* $Q \succ 0$ *and* $\Gamma_i \in R^{n_u \times n_x}, i = 1,...,N$ *such that*

$$\begin{bmatrix} Q & A_iQ + B\Gamma_i & B_{1i} & 0 \\ * & Q & 0 & QC_{1_\infty i}^T \\ * & * & I & D_{1_\infty 1,i}^T \\ * & * & * & \gamma_\infty^2 I \end{bmatrix} \prec 0, \ \forall i = 1,...,N \tag{A.73}$$

Moreover, the state feedback control law given by (A.65) with matrices $K_i = \Gamma_i Q^{-1}$, $\forall i = 1,...,N$ *has the quadratic* H_∞ *performance* γ_∞*.* ■

B

Clustering method

Clustering or cluster analysis is the task of dividing the data points or objects into groups (called clusters) in such a way data points in the same cluster are more similar (in some sense) to other data points in the same group than the other data points in other clusters.

- **Density-Based Methods**: Density-based Clustering methods are the clustering methods that are based on detecting areas where points are concentrated and where they are separated by areas that are empty or sparse. The data points, that do not belong to any cluster, are considered as noise. Density-based Clustering methods have good accuracy and ability to merge two clusters. Some representatives of Density-based Clustering methods are DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [67], OPTICS (Ordering Points to Identify Clustering Structure) [251], etc.
- **Hierarchical Based Methods[163]**: The clusters formed in these methods form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one. Hierarchical based clustering methods can be divided into two main types:
  - Agglomerative clustering (bottom-up approach)[4]: An object is initially considered as a single-element cluster (leaf). The two most similar clusters are combined into a new bigger cluster (nodes) at each step of the algorithm. Then, this procedure is repeated till all the data points belong to just one single big cluster (called root). Result of agglomerative clustering method is the tree which can be plotted as a dendrogram.
  - Divisive hierarchical clustering (top-down approach)[242]: This algorithm is in an inverse order of Agglomerative clustering. The algorithm starts with the root, in which all objects are included in a single cluster. The most heterogeneous cluster is divided into two clusters at each iteration step. The process is repeated until all objects are in their own cluster.

  Some examples of Hierarchical clustering Based Methods are CURE (Clustering Using Representatives) [90], BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies) [248], etc.
- **Partitioning clustering Methods**: are the clustering methods that partition the objects into groups based on their similarity and each partition forms one cluster. This

method is commonly implemented by optimizing an objective criterion similarity function especially when the distance is a major parameter. Some examples of Partitioning clustering Methods are K-means [146][173], CLARANS (Clustering Large Applications based upon Randomized Search) [159], K-medoids clustering or PAM [119].

- **Grid-based clustering Methods**: are the clustering methods where the data space is divided into a finite number of cells that form a grid-like structure. All the clustering operation performed on these grids are fast and independent of the number of data objects. Some examples of Grid-based clustering Methods are STING (Statistical Information Grid) [234], wave cluster [207], CLIQUE (CLustering In Quest) [165], etc.

C

Minimal convex polygon decomposition

A partition of a polygon P is a set of primitive units (or polygons, these primitive units can be triangles, rectangles, squares, convex or concave polygons) such that these polygons do not intersect each other and the union of these polygons is exactly equal to the interior of the original polygon P. A polygon partition problem is a problem of finding a partition which is optimal in some sense. The optimal criterion for this polygon partition problem can be partition with a smallest number of polygons, with polygons of smallest total side-length, or minimal number of convex polygons.

Polygon partitioning is a really important class of problems in computational geometry because most geometric problems are simpler and faster on convex objects than on non-convex ones. There exists a number of different polygon partitioning problems. They are depending on the type of polygon that is being partitioned and also on the types of units (polygons) allowed for partitioning.

The term polygon decomposition is often used as a general term that includes both covering and partitioning[143].

Polygon decomposition has been widely applied in several areas:

- Polygon decomposition has been widely used in computational geometry. Generally, problems that restrict the types of polygons such as convex or star-shaped are always simpler than the ones with general polygons. Therefore, for computational geometry problems, polygons are first decomposed into simpler component parts. Then, the algorithm is applied to the gathered simpler polygons. Finally, partial solutions are combined to get the full solution. Examples can be seen in [164] [8].
- Very large-scale integration (VLSI) is the procedure for generating an integrated circuit (IC) by combining millions of MOS transistors onto a single chip. Layouts are illustrated as polygons [137] in VLSI data processing, and one method for preparing the electron-beam lithography is to decompose these polygon regions into fundamental simple polygons. Polygon decomposition is also applied for dividing the routing region into channels.
- Pattern recognition [25] is the process of recognizing patterns by using machine learning algorithm. These techniques extract information from an object for identifying, classifying, or describing it. General polygons are usually recognized by decomposing

it into simpler components, then identify the components and their interrelationships. Finally, the gathered information is used to determine the shape of the object.

- Some others applications of Polygon decomposition include image processing [153], computer graphics [222], database systems [132], and data compression [139].

Several methods for polygon decomposition are:

- Partitioning a polygon to triangles [1][117]
- Partitioning a polygon to pseudo-triangles [10][96]
- Partitioning a rectilinear polygon into rectangles [52]
- Partitioning a polygon to trapezoids [39][16]
- Partitioning a polygon to convex quadrilaterals [187][145]
- Partitioning a polygon to m-gons [131][125]

D

CGAL library

The Computational Geometry Algorithms Library (CGAL) is an open source software library of computational geometry algorithms that provides easy access to efficient and reliable geometric algorithms in the form of a C++ library. It was initially released in 1996 and primarily written in C++. Nowadays, CGAL libraries are also available for Python and Java.

CGAL is used and applied in a number of areas that require geometric computation such as computer vision, medical imaging, computer-aided design, geographic information system, molecular biology, and robotics.

The CGAL library covers the following topics:

- Geometry kernels - basic geometric operations on geometric primitives
- Arithmetic and algebra
- Convex hull algorithms
- Polygons and polyhedra
- Polygon and polyhedron operations
- Arrangements
- Point set triangulations
- Delaunay triangulations
- Voronoi diagrams
- Mesh generation
- Geometry processing
- Search structures
- Shape analysis, fitting, and distances
- Interpolation
- Kinetic data structures

The main CGAL package was used in this thesis is "2D Polygon Partitioning". This package provides functions for partitioning polygons in monotone or convex polygons. The algorithms can produce results with the minimal number of polygons, as well as approximations which have no more than four times the optimal number of convex pieces but they differ in their runtime complexities.

Bibliography

[1] Fournier A. and Montuno D. Y. "Triangulating Simple Polygons and Equivalent Problems". In: *ACM Transactions on Graphics* 3 (2 Apr. 1984). DOI: `10.1145/357337.357341`.

[2] M. Abdolhosseini, Youmin Zhang, and C. Rabbath. "An Efficient Model Predictive Control Scheme for an Unmanned Quadrotor Helicopter". In: *Journal of Intelligent & Robotic Systems* 70 (Apr. 2013). DOI: `10.1007/s10846-012-9724-3`.

[3] Ercan Acar et al. "Path Planning for Robotic Demining: Robust Sensor-Based Coverage of Unstructured Environments and Probabilistic Methods". In: *The International Journal of Robotics Research* 22.8 (2003), pp. 441–466. DOI: `10.1177/027836403128965213`.

[4] Marcel R. Ackermann et al. "Analysis of Agglomerative Clustering". In: *Algorithmica* 69 (1 May 2014). DOI: `10.1007/s00453-012-9717-4`.

[5] O. Adiyatov and H. A. Varol. "Rapidly-exploring random tree based memory efficient motion planning". In: *2013 IEEE International Conference on Mechatronics and Automation* (2013), pp. 354–359. DOI: `10.1109/ICMA.2013.6617944`.

[6] D. Adjiashvili and D. Peleg. "Equal-area locus-based convex polygon decomposition". In: *Theoretical Computer Science* 411 (2010), pp. 1648–1667. DOI: `https://doi.org/10.1016/j.tcs.2010.01.012`.

[7] Pankaj K. Agarwal, Eyal Flato, and Dan Halperin. "Polygon decomposition for efficient construction of Minkowski sums". In: *Computational Geometry* 21 (2002), pp. 39–61. DOI: `https://doi.org/10.1016/S0925-7721(01)00041-4`.

[8] Alok Aggarwal et al. "A linear-time algorithm for computing the voronoi diagram of a convex polygon". In: *Discrete & Computational Geometry* 4 (1 1989). DOI: `10.1007/bf02187749`.

[9] Nigar Ahmed and Mou Chen. "Sliding mode control for quadrotor with disturbance observer". In: *Advances in Mechanical Engineering* 10.7 (2018), pp. 1–16. DOI: `10.1177/1687814018782330`.

[10] Oswin Aichholzer et al. "On the number of pseudo-triangulations of certain point sets". In: *Journal of Combinatorial Theory, Series A* 115 (2 2008). DOI: `10.1016/j.jcta.2007.06.002`.

[11] Isidori Alberto. *Nonlinear Control Systems*. Springer. ISBN: 978-3-540-17069-3.

[12] C. Albornoz and L. F. Giraldo. "Trajectory design for efficient crop irrigation with a UAV". In: *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*. 2017, pp. 1–6. DOI: `10.1109/CCAC.2017.8276401`.

[13] Barry Allred et al. "Effective and efficient agricultural drainage pipe mapping with UAS thermal infrared imagery: A case study". In: *Agricultural Water Management* 197 (Jan. 2018). DOI: `10.1016/j.agwat.2017.11.011`.

[14] P. Apkarian and P. Gahinet. “A convex characterization of gain-scheduled H_∞ controllers”. In: *IEEE Transactions on Automatic Control* 40.5 (1995), pp. 853–864. DOI: `10.1109/9.384219`.

[15] Pierre Apkarian, Pascal Gahinet, and Greg Becker. “Self-Scheduled H_∞ Control of Linear Parameter-Varying Systems: A Design Example”. In: *Automatica* 31 (Sept. 1995), pp. 1251–1261. DOI: `10.1016/0005-1098(95)00038-X`.

[16] Takao Asano, Tetsuo Asano, and H. Imai. “Partitioning a polygonal region into trapezoids”. In: *Journal of the Association for Computing Machinery* 33 (2 1986). DOI: `10.1145/5383.5387`.

[17] Prasad N. Atkar et al. “Uniform Coverage of Automotive Surface Patches”. In: *The International Journal of Robotics Research* 24.11 (2005), pp. 883–898. ISSN: 0278-3649. DOI: `10.1177/0278364905059058`.

[18] Reg Austin. *Unmanned Aircraft Systems: UAVS Design, Development and Deployment.* Wiley, 2010. ISBN: 978-0-470-05819-0.

[19] M. D. Bah, A. Hafiane, and R. Canals. “Weeds detection in UAV imagery using SLIC and the hough transform”. In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA).* 2017, pp. 1–6. DOI: `10.1109/IPTA.2017.8310102`.

[20] Moses Bangura and Robert Mahony. “Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor”. In: *Proceedings of Australasian Conference on Robotics and Automation, 3-5 Dec 2012, Victoria University of Wellington, New Zealand.* 2012. ISBN: 978-0-9807404-3-1. URL: `https://openresearch-repository.anu.edu.au/bitstream/1885/65230/2/01_Bangura_Nonlinear_Dynamic_Modeling_for_2012.pdf`.

[21] Hichem Barki, Florence Denis, and Florent Dupont. “Contributing vertices-based Minkowski sum computation of convex polyhedra”. In: *Computer-Aided Design* (2009), pp. 525–538. DOI: `10.1016/j.cad.2009.03.008`.

[22] Marvin E. Bauer and Jan E. Cipra. “Identification of Agricultural Crops by Computer Processing of ERTS MSS Data”. In: (1973), p. 20.

[23] G. Becker et al. “Control of Parametrically-Dependent Linear Systems: A Single Quadratic Lyapunov Approach”. In: *1993 American Control Conference.* 1993, pp. 2795–2799. DOI: `10.23919/ACC.1993.4793406`.

[24] Lenaick Besnard, Yuri Shtessel, and David Landrum. “Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer”. In: *Journal of The Franklin Institute-engineering and Applied Mathematics - J FRANKLIN INST-ENG APPL MATH* 349 (Mar. 2012), pp. 1–27. DOI: `10.1016/j.jfranklin.2011.06.031`.

[25] Christopher M Bishop. *Pattern Recognition and Machine Learning.* Heidelberg, Germany: Springer, 2006, p. 738. ISBN: 0-387-31073-8.

[26] B. S. Blackmore and Griepentrog. “A future view of precision farming”. In: *Precision Agriculture Tage, Bonn, 13–15th March 2002. (KTBL, Darmstadt, Germany).* 2002.

[27] R. Bohlin and L.E. Kavraki. “Path planning using lazy PRM”. In: *IEEE Proceedings 2000 ICRA. IEEE International Conference on Robotics and Automation - San Francisco, CA, USA.* 2000. ISBN: 0-7803-5886-4. DOI: `10.1109/ROBOT.2000.844107`.

[28] M. Bosse, N. Nourani-Vatani, and J. Roberts. “Coverage Algorithms for an Underactuated Car-Like Vehicle in an Uncertain Environment”. In: *Proceedings 2007 IEEE*

International Conference on Robotics and Automation (Apr. 2007), pp. 698–703. DOI: `10.1109/ROBOT.2007.363068`.

[29] S. Bouabdallah, A. Noth, and R. Siegwart. "PID vs LQ control techniques applied to an indoor micro quadrotor". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. 2004, 2451–2456 vol.3. DOI: `10.1109/IROS.2004.1389776`.

[30] Samir Bouabdallah. "Design and Control of quadrotors with application to autonomous flying". PhD thesis. Jan. 2007. DOI: `10.5075/epfl-thesis-3727`.

[31] Samir Bouabdallah and Roland Siegwart. "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor". In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference* (2007), pp. 153–158. DOI: `10.1109/ROBOT.2005.1570447`.

[32] Yasser BOUZID. "Guidance and Control system for autonomous aerial vehicles navigation". PhD thesis. Université Paris-Saclay - Université d'Evry Val d'Essonne, 2018. URL: `http://www.theses.fr/2018SACLE014`.

[33] Yuandong Yang; Oliver Brock. "Elastic roadmaps—motion generation for autonomous mobile manipulation". In: *Autonomous Robots* 28 (1 Jan. 2010). DOI: `10.1007/s10514-009-9151-x`.

[34] Siciliano Bruno and Khatib Oussama. "Springer Handbook of Robotics || Robotics in Agriculture and Forestry". In: vol. 10.1007/978-3-540-30301-5. 2008. ISBN: 978-3-540-23957-4,978-3-540-30301-5. DOI: `10.1007/978-3-540-30301-5_47`.

[35] D. Cabecinhas, R. Cunha, and C. Silvestre. "A Globally Stabilizing Path Following Controller for Rotorcraft With Wind Disturbance Rejection". In: *IEEE Transactions on Control Systems Technology* 23.2 (2015), pp. 708–714. DOI: `10.1109/TCST.2014.2326820`.

[36] Zuo Llang Cao, Yuyu Huang, and Ernest L. Hall. "Region filling operations with random obstacle avoidance for mobile robots". en. In: *Journal of Robotic Systems* 2.5 (1988), pp. 87–102. DOI: `10.1002/rob.4620050202`.

[37] X. Chang and G. Yang. "New Results on Output Feedback H_∞ Control for Linear Discrete-Time Systems". In: *IEEE Transactions on Automatic Control* 59.5 (2014), pp. 1355–1359. DOI: `10.1109/TAC.2013.2289706`.

[38] Konstantinos Chartzoulakis and Maria Bertaki. "Sustainable Water Management in Agriculture under Climate Change". In: *Agriculture and Agricultural Science Procedia* 4 (2015). DOI: `10.1016/j.aaspro.2015.03.011`.

[39] Bernard Chazelle. "Triangulating a simple polygon in linear time". In: *Discrete & Computational Geometry* 6 (1 1991). DOI: `10.1007/bf02574703`.

[40] Auat Cheein, Fernando Alfredo, and Ricardo Carelli. "Agricultural Robotics: Unmanned Robotic Service Units in Agricultural Tasks". In: *IEEE Industrial Electronics Magazine* 7 (3 Sept. 2013). DOI: `10.1109/mie.2013.2252957`.

[41] Jie Chen and Patton R J. *Robust Model-Based Fault Diagnosis for Dynamic Systems.* Springer, 1999. ISBN: 978-1-4615-5149-2.

[42] P. Chen and J. Luo. "Modeling and H_∞ of quadrotor and design of loop shaping controller". In: *Journal of Nanjing University of Science and Technology* 33 (Aug. 2009), pp. 81–86.

[43] Mahmoud Chilali and Pascal Gahinet. "H_∞ Design with Pole Placement Constraints: an LMI Approach". In: *Automatic Control, IEEE Transactions on* 41 (Mar. 1996), pp. 358 –367. DOI: `10.1109/9.486637`.

[44] Howie Choset. “Coverage for robotics – A survey of recent results”. In: *Annals of Mathematics and Artificial Intelligence* 31 (1-4 Oct. 2001). DOI: `https://10.1023/a:1016639210559`.

[45] Howie Choset and Philippe Pignon. “Coverage Path Planning: The Boustrophedon Cellular Decomposition”. en. In: *SpringerLink* (1998), pp. 203–209. DOI: `10.1007/978-1-4471-1273-0_32`.

[46] Briat Corentin. *Linear Parameter-Varying and Time-Delay Systems. Analysis, Observation, Filtering & Control.* Springer, 2014. ISBN: 978-3-662-44050-6.

[47] National Research Council. *Precision Agriculture in the 21st Century: Geospatial and Information Technologies in Crop Management.* Washington, DC: The National Academies Press, 1997. ISBN: 978-0-309-46616-5. DOI: `10.17226/5491`. URL: `https://www.nap.edu/catalog/5491/precision-agriculture-in-the-21st-century-geospatial-and-information-technologies`.

[48] I. D. Cowling et al. “A prototype of an autonomous controller for a quadrotor UAV”. In: *2007 European Control Conference (ECC).* 2007, pp. 4001–4008. DOI: `10.23919/ECC.2007.7068316`.

[49] B. Dai et al. “A vision-based autonomous aerial spray system for precision agriculture”. In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO).* 2017, pp. 507–513.

[50] M. Darouach, M. Zasadzinski, and S. J. Xu. “Full-order observers for linear systems with unknown inputs”. In: *IEEE Transactions on Automatic Control* 39.3 (1994), pp. 606–609. DOI: `10.1109/9.280770`.

[51] Jonathan P. Dash et al. “Assessing very high resolution UAV imagery for monitoring forest health during a simulated disease outbreak”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 131 (Sept. 2017). DOI: `10.1016/j.isprsjprs.2017.07.007`.

[52] Eppstein David. “Graph-Theoretic Solutions to Computational Geometry Problems”. In: vol. 5911. Aug. 2009, pp. 1–16. DOI: `10.1007/978-3-642-11409-0_1`.

[53] Laloui Derafa, Abdelaziz Benallegue, and Leonid Fridman. “Super-twisting control algorithm for the attitude tracking of a four rotors UAV”. In: *Journal of Fracklin Institute* 349 (Mar. 2012), pp. 685–699. DOI: `10.1016/j.jfranklin.2011.10.011`.

[54] C. Diao et al. “A nonlinear adaptive control approach for quadrotor UAVs”. In: *2011 8th Asian Control Conference (ASCC).* 2011, pp. 223–228.

[55] Edsger Dijkstra. “A Note on Two Problems in Connexion With Graphs”. In: *Numerische Mathematik* 1 (Dec. 1959), pp. 269–271. DOI: `10.1007/BF01386390`.

[56] J. Dong and G. Yang. “Static Output Feedback Control Synthesis for Linear Systems With Time-Invariant Parametric Uncertainties”. In: *IEEE Transactions on Automatic Control* 52.10 (2007), pp. 1930–1936. DOI: `10.1109/TAC.2007.906227`.

[57] Jiuxiang Dong and Guang-Hong Yang. “Static output feedback H_∞ control of a class of nonlinear discrete-time systems”. In: *Fuzzy Sets and Systems* 160.19 (2009), pp. 2844–2859.

[58] Wei Dong et al. “Development of a Quadrotor Test Bed — Modelling, Parameter Identification, Controller Design and Trajectory Generation”. In: *International Journal of Advanced Robotic Systems* 12 (Feb. 2015), p. 1. DOI: `10.5772/59618`.

[59] Wei Dong et al. “High-Performance Trajectory Tracking Control of a Quadrotor with Disturbance Observer”. In: *Sensors and Actuators A: Physical* 211 (May 2014). DOI: `10.1016/j.sna.2014.03.011`.

[60] Ibrahima N Doye et al. "Design of unknown input fractional-order observers for fractional-order systems". In: *International Journal of Applied Mathematics and Computer Science* 23.23 (2013), pp. 491–500. DOI: `10.2478/amcs-2013-0037`.

[61] Altug E., Ostrowski J.P., and Taylor C.J. "Quadrotor control using dual camera visual feedback". In: *IEEE IEEE International Conference on Robotics and Automation. IEEE ICRA 2003 - Taipei, Taiwan (14-19 Sept. 2003).* Vol. 3. 2003. ISBN: 0-7803-7736-2. DOI: `10.1109/ROBOT.2003.1242264`.

[62] Altug E., Ostrowski J.P., and Mahony R. "Control of a quadrotor helicopter using visual feedback". In: *IEEE 2002 IEEE International Conference on Robotics and Automation - Washington, DC, USA (11-15 May 2002).* Vol. 1. 2002. ISBN: 0-7803-7272-7. DOI: `10.1109/ROBOT.2002.1013341`.

[63] Frazzoli E., Dahleh M.A., and Feron E. "Real-time motion planning for agile autonomous vehicles". In: *IEEE Proceedings of American Control Conference - Arlington, VA, USA (2001.6.25-2001.6.27).* 2001. ISBN: 0-7803-6495-3. DOI: `10.1109/ACC.2001.945511`.

[64] Jeff El-Osery Aly; Prevost. "[Studies in Systems, Decision and Control] Control and Systems Engineering Volume 27 || Proportional-Integral Observer in Robust Control, Fault Detection, and Decentralized Control of Dynamic Systems". In: 2015, pp. 13–43. ISBN: 978-3-319-14635-5,978-3-319-14636-2. DOI: `10.1007/978-3-319-14636-2_2`.

[65] Frazzoli Emilio, Dahleh Munther A., and Feron Eric. "Real-Time Motion Planning for Agile Autonomous Vehicles". In: *Journal of Guidance Control and Dynamics* 25 (1 Jan. 2002). DOI: `10.2514/2.4856`.

[66] Bara J Emran et al. "A Cascaded Approach for Quadrotor's Attitude Estimation". In: *Procedia Technology* 15 (2014). DOI: `10.1016/j.protcy.2014.09.080`.

[67] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: vol. 96. Jan. 1996, pp. 226–231.

[68] Critical Exploratiions. *Food Science and Nutrition: Breakthroughs in Research and Practice.* 1st ed. Vol. 4. 10. An optional note. The address: IGI Global, Feb. 2018. ISBN: 1522552073.

[69] Bruno S Faical et al. "An adaptive approach for UAV-based pesticide spraying in dynamic environments". In: *Computers and Electronics in Agriculture* 138 (June 2017). DOI: `10.1016/j.compag.2017.04.011`.

[70] Ole Falkenberg et al. "Model Identification and H_∞ Attitude Control for Quadrotor MAV's". In: vol. 7507. Oct. 2012, pp. 460–471. DOI: `10.1007/978-3-642-33515-0_46`.

[71] Z. Fang and W. Gao. "Adaptive integral backstepping control of a Micro-Quadrotor". In: *2nd International Conference on Intelligent Control and Information Processing.* Vol. 2. 2011, pp. 910–915. DOI: `10.1109/ICICIP.2011.6008382`.

[72] Gary Fay. "Derivation of the Aerodynamic Forces for the Mesicopter Simulation". In: *Stanford University, USA.* 2001. URL: `https://www.scienceopen.com/document?vid=b3638dea-34c3-4033-b05b-ec69db015c1d`.

[73] G. Feng. "A Survey on Analysis and Design of Model-Based Fuzzy Control Systems". In: *IEEE Transactions on Fuzzy Systems* 14.5 (2006), pp. 676–697. DOI: `10.1109/TFUZZ.2006.883415`.

[74] Dave Ferguson and Anthony Stentz. "Using interpolation to improve path planning: The Field D* algorithm". In: *Journal of Field Robotics* 23 (2 2006). DOI: `10.1002/rob.20109`.

[75] J. Fernandez, L. Canovas, and B. Pelegrin. “Algorithms for the decomposition of a polygon into convex polygons”. In: *European Journal of Operational Research* 121 (2000), pp. 330–342. DOI: `https://doi.org/10.1016/S0377-2217(99)00033-8`.

[76] Foresight. *The Future of Food and Farming: Challenges and choices for global sustainability.* Tech. rep. The Government Office for Science, London, 2011.

[77] F. Forte et al. “Impedance control of an aerial manipulator”. In: *2012 American Control Conference (ACC).* 2012, pp. 3839–3844. DOI: `10.1109/ACC.2012.6315568`.

[78] Bruce A Francis. *A Course in H_∞ Control Theory.* Vol. 1. Springer US, 1995. ISBN: 978-1-84628-615-5.

[79] Hoffmann Gabriel et al. “Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment”. In: *American Institute of Aeronautics and Astronautics AIAA Guidance, Navigation and Control Conference and Exhibit - Hilton Head, South Carolina.* 2007. ISBN: 978-1-62410-015-4. DOI: `10.2514/6.2007-6461`.

[80] Marc Galceran Enric; Carreras. “A survey on coverage path planning for robotics”. In: *Robotics and Autonomous Systems* 61 (12 Dec. 2013). DOI: `10.1016/j.robot.2013.09.004`.

[81] Parthasarathy Garre and Alladi Harish. “Autonomous Agricultural Pesticide Spraying UAV”. In: *IOP Conference Series: Materials Science and Engineering* 455 (Dec. 2018), p. 012030. DOI: `10.1088/1757-899X/455/1/012030`.

[82] A. S. Ghersin and R. S. Sanchez Pena. “LPV control of a 6-DOF vehicle”. In: *IEEE Transactions on Control Systems Technology* 10.6 (2002), pp. 883–887. DOI: `10.1109/TCST.2002.804123`.

[83] Rafal Goebel, Tingshu Hu, and Andrew Teel. “Dual Matrix Inequalities in Stability and Performance Analysis of Linear Differential/Difference Inclusions”. In: Jan. 2006, pp. 103–122. DOI: `10.1007/0-8176-4470-9_6`.

[84] C. Goerzen, Z. Kong, and B. Mettler. “A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance”. In: *Journal of Intelligent & Robotic Systems* 57 (1-4 Jan. 2010). DOI: `10.1007/s10846-009-9383-1`.

[85] Herbert Goldstein, Charles P. Poole, and John Safko. *Classical Mechanics.* 3rd ed. Pearson, 2001. ISBN: 9788131758915.

[86] S. Gonzalez-Vazquez and J. Moreno-Valenzuela. “A New Nonlinear PI/PID Controller for Quadrotor Posture Regulation”. In: *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference.* 2010, pp. 642–647. DOI: `10.1109/CERMA.2010.78`.

[87] Anne Goodchild and Jordan Toy. “Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO_2 emissions in the delivery service industry”. In: *Transportation Research Part D: Transport and Environment* (Mar. 2017). DOI: `10.1016/j.trd.2017.02.017`.

[88] Ole Green et al. “Commercial autonomous agricultural platform: Kongskilde Robotti”. In: 2014, pp. 351–356.

[89] Daniel H. Greene. “The decomposition of polygons into convex parts”. In: *Advances in Computational Mathematics* 1 (1983), pp. 235–259.

[90] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. “Cure: an efficient clustering algorithm for large databases”. In: *Information Systems* 26 (1 2001). DOI: `10.1016/s0306-4379(01)00008-4`.

[91] Kil-Su Han et al. “Strawberry Harvesting Robot for Bench-type Cultivation”. In: *Journal of Biosystems Engineering* 37 (Feb. 2012). DOI: `10.5307/JBE.2012.37.1.065`.

[92] Liang Han et al. “Clustering Field-Based Maize Phenotyping of Plant-Height Growth and Canopy Spectral Dynamics Using a UAV Remote-Sensing Approach”. In: *Frontiers in Plant Science* 9 (Nov. 2018). DOI: `10.3389/fpls.2018.01638`.

[93] John A. Hartigan. “Clustering Algorithms”. In: *John Wiley & Sons New York , London , Sydney , Toronto* (1975), p. 351.

[94] J. Hemming et al. “A robot for harvesting sweet-pepper in greenhouses”. In: (2014).

[95] E.J. Van Henten et al. “An Autonomous Robot for De-leafing Cucumber Plants grown in a High-wire Cultivation System”. In: *Biosystems Engineering* 94 (3 2006). DOI: `10.1016/j.biosystemseng.2006.03.005`.

[96] Bronnimann Herve et al. “Counting and Enumerating Pointed Pseudotriangulations with the Greedy Flip Algorithm”. In: *SIAM Journal on Computing* 36 (3 Jan. 2006). DOI: `10.1137/050631008`.

[97] David Hsu et al. “On Finding Narrow Passages with Probabilistic Roadmap Planners”. In: *Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics: The Algorithmic Perspective.* 1998, pp. 141–153. DOI: `https://doi.org/10.1145/276675.276685`.

[98] Huasheng Huang et al. “A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery”. In: *PLoS ONE* 13 (4 Apr. 2018). DOI: `10.1371/journal.pone.0196302`.

[99] Joseph E. Hunter et al. “Integration of Remote-Weed Mapping and an Autonomous Spraying Unmanned Aerial Vehicle for Site-Specific Weed Management”. In: *Pest Management Science* (Oct. 2019). DOI: `10.1002/ps.5651`.

[100] Dalil Ichalal and Said Mammar. “On Unknown Input Observers for LPV Systems”. In: *IEEE Transactions on Industrial Electronics* 62 (9 Sept. 2015), pp. 5870 –5880. DOI: `10.1109/TIE.2015.2448055`.

[101] Weikuan Jia et al. “Preprocessing method of night vision image application in apple harvesting robot”. In: *International Journal of Agricultural and Biological Engineering* 11 (Jan. 2018), pp. 158–163. DOI: `10.25165/j.ijabe.20181102.2822`.

[102] Meng Jianliang, Shang Haikun, and Bian Ling. “The Application On Intrusion Detection Based On K-Means Cluster Algorithm”. In: *IEEE International Conference On Information Technology And Applications* (2009), pp. 150 –152. DOI: `10.1109/IFITA.2009.34`.

[103] Kuffner J.J. and LaValle S.M. “RRT-connect: An efficient approach to single-query path planning”. In: *IEEE 2000 ICRA. IEEE International Conference on Robotics and Automation - San Francisco, CA, USA (24-28 April 2000).* Vol. 2. 2000. ISBN: 0-7803-5886-4. DOI: `10.1109/robot.2000.844730`.

[104] S. John. “Artificial intelligent-based feedforward optimized PID wheel slip controller”. In: *2013 Africon.* 2013, pp. 1–6. DOI: `10.1109/AFRCON.2013.6757683`.

[105] Rasmus Jorgensen et al. “HortiBot: A System Design of a Robotic Tool Carrier for High-tech Plant Nursing”. In: *CIGR J. Sci. Res. Dev.* IX (Nov. 2006).

[106] Jinha Jung et al. “Unmanned aerial system assisted framework for the selection of high yielding cotton genotypes”. In: *Computers and Electronics in Agriculture* 152 (Sept. 2018). DOI: `10.1016/j.compag.2018.06.051`.

[107] Dalamagkidis K. “Classification of UAVs”. In: Valavanis K. Vachtsevanos G. (eds) Handbook of Unmanned Aerial Vehicles, Springer Dordrecht, 2014, pp. 83–91.

[108] Xiao K. et al. “Improved Theta*: Improved any-angle path planning on girds”. In: *Journal of Computational Information Systems* 10 (Oct. 2014), pp. 8881–8890. DOI: `10.12733/jcis12043`.

[109] Von T Kaczorec. “Proportional-integral observers for linear multivariable time-varying systems”. In: *Regelungstechnik* 27 (1979), pp. 359–362. DOI: `https://doi.org/10.1524/auto.1979.27.112.359`.

[110] P. Katsigiannis et al. “An autonomous multi-sensor UAV system for reduced-input precision agriculture applications”. In: *2016 24th Mediterranean Conference on Control and Automation (MED)*. 2016, pp. 60–64.

[111] L. Kavraki and M. Kolountzakis; J. Latombe. “Analysis of probabilistic roadmaps for path planning”. In: (1998). DOI: `10.1109/70.660866`.

[112] L.E. Kavraki et al. “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. In: *IEEE Transactions on Robotics and Automation* 12 (4 1996). DOI: `10.1109/70.508439`.

[113] Jonathan Kelly and Gaurav Sukhatme. “Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration”. In: *I. J. Robotic Res.* 30 (Jan. 2011), pp. 56–79. DOI: `10.1177/0278364910382802`.

[114] Farid Kendoul. “Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems”. In: *Journal of Field Robotics* 29 (Mar. 2012), pp. 315 –378. DOI: `10.1002/rob.20414`.

[115] Daniel Kenny et al. “Theta*: Any-Angle Path Planning on Grids”. In: *J. Artif. Intell. Res. (JAIR)* 39 (Jan. 2014). DOI: `10.1613/jair.2994`.

[116] Mohamed Kerkecha, Adel Hafianea, and Raphael Canals. “Deep leaning approach with colorimetric spaces and vegetation indices forvine diseases detection in UAV images”. In: *Computers and Electronics in Agriculture* 155 (2018), pp. 237–243. DOI: `https://doi.org/10.1016/j.compag.2018.10.006`.

[117] Capp Kerry and Julstrom Bryant A. “[ACM Press the 1998 ACM symposium - Atlanta, Georgia, United States (1998.02.27-1998.03.01)] Proceedings of the 1998 ACM symposium on Applied Computing - SAC ’98 - A weight-coded genetic algorithm for the minimum weight triangulation problem”. In: 1998. ISBN: 0897919696. DOI: `10.1145/330560.330833`.

[118] Won Kim, Sang Ahn, and Wook-Hyun Kwon. “Stability analysis and stabilization of Fuzzy state space models”. In: *Fuzzy Sets and Systems* 71 (Apr. 1995), 131–142. DOI: `10.1016/0165-0114(94)00190-I`.

[119] Mark Van der Laan, Katherine Pollard, and Jennifer Bryan. “A new partitioning around medoids algorithm”. In: *Journal of Statistical Computation and Simulation* 73 (8 Aug. 2003). DOI: `10.1080/0094965031000136012`.

[120] Jean-Claude Latombe. *Robot Motion Planning*. 1st ed. Vol. 124. 1. Robotics: Vision, Manipulation and Sensors. The address: Springer US, Dec. 1991. ISBN: 978-0-7923-9206-4.

[121] S. M. LaValle. *Planning Algorithms*. Available at http://planning.cs.uiuc.edu/. Cambridge, U.K.: Cambridge University Press, 2006.

[122] S. M. LaValle. “Randomized Kinodynamic Planning”. In: *Internatinal Journal of Robotics Research* 20 (5 May 2001). DOI: `10.1177/02783640122067453`.

[123] Steven M. Lavalle. *Rapidly-Exploring Random Trees: A New Tool for Path Planning.* Tech. rep. 1998.

[124] Junghwan Lee et al. "SR-RRT: Selective retraction-based RRT planner". In: *IEEE 2012 IEEE International Conference on Robotics and Automation (ICRA) - St Paul, MN, USA (2012.05.14-2012.05.18).* 2012. ISBN: 978-1-4673-1405-3,978-1-4673-1403-9,978-1-4673-1578-4,978-1-4673-1404-6, DOI: `10.1109/icra.2012.6224928`.

[125] Christos Levcopoulos, Andrzej Lingas, and Jorg-R. Sack. "Heuristics for optimum binary search trees and minimum weight triangulation problems". In: *Theoretical Computer Science* 66 (2 1989). DOI: `10.1016/0304-3975(89)90134-5`.

[126] Danjun Li et al. "Autonomous landing of quadrotor based on ground effect modelling". In: *IEEE 2015 34th Chinese Control Conference (CCC) - Hangzhou, China (2015.7.28-2015.7.30).* 2015. ISBN: 978-9-8815-6389-7. DOI: `10.1109/ChiCC.2015.7260521`.

[127] J. Li and Y. Li. "Dynamic analysis and PID control for a quadrotor". In: *2011 IEEE International Conference on Mechatronics and Automation.* 2011, pp. 573–578. DOI: `10.1109/ICMA.2011.5985724`.

[128] L Li and F Liao. "Parameter-dependent preview control with robust tracking performance". In: *IET Control Theory Applications* 11.1 (2017), pp. 38–46. DOI: `10.1049/iet-cta.2016.0361`.

[129] M. Li et al. "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation". In: *2014 IEEE International Conference on Robotics and Automation (ICRA).* 2014, pp. 409–416. DOI: `10.1109/ICRA.2014.6906889`.

[130] Yang Liang et al. "Survey of Robot 3D Path Planning Algorithms". In: *Journal of Control Science and Engineering* 2016 (2016). DOI: `10.1155/2016/7426913`.

[131] Andrzej Lingas, Christos Levcopoulos, and Jorg Sack. "Algorithms for minimum length partitions of polygons". In: *BIT Numerical Mathematics* 27 (4 1987). DOI: `10.1007/bf01937272`.

[132] Witold Lipski et al. "On two-dimensional data organization". In: *Annales Societatis Mathematicae Polonae. Series IV* 2 (Jan. 1979).

[133] H. Liu et al. "Brief Paper - Robust attitude control of uncertain quadrotors". In: *IET Control Theory Applications* 7.11 (2013), pp. 1583–1589. DOI: `10.1049/iet-cta.2012.0964`.

[134] Hao Liu, Jianxiang Xi, and Yisheng Zhong. "Robust motion control of quadrotors". In: *Journal of the Franklin Institute* 351 (Oct. 2014), pp. 5494–5510. DOI: `10.1016/j.jfranklin.2014.10.003`.

[135] G. Loianno and V. Kumar. "Cooperative Transportation Using Small Quadrotors Using Monocular Vision and Inertial Sensing". In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 680–687. DOI: `10.1109/LRA.2017.2778018`.

[136] G. Loianno, J. Thomas, and V. Kumar. "Cooperative localization and mapping of MAVs using RGB-D sensors". In: *2015 IEEE International Conference on Robotics and Automation (ICRA).* 2015, pp. 4021–4028. DOI: `10.1109/ICRA.2015.7139761`.

[137] Mario A. Lopez and Dinesh P. Mehta. "Efficient decomposition of polygons into L-shapes with application to VLSI layouts". In: *ACM Transactions on Design Automation of Electronic Systems* 1 (3 July 1996). DOI: `10.1145/234860.234865`.

[138] Garcia Carrillo L.R. et al. *Quad Rotorcraft Control. Vision-Based Hovering and Navigation.* Springer-Verlag London, 2013. ISBN: 1430-9491.

[139] Anna Lubiw. "The Boolean Basis Problem and How to Cover Some Polygons by Rectangles". In: *SIAM Journal on Discrete Mathematics* 3 (1 1990). DOI: `10.1137/0403010`.

[140] D. Luenberger. "An introduction to observers". In: *IEEE Transactions on Automatic Control* 16.6 (1971), pp. 596 –602. DOI: `10.1109/TAC.1971.1099826`.

[141] L. Luong Tuan and S. Won. "PID based sliding mode controller design for the micro quadrotor". In: *13th International Conference on Control, Automation and Systems (ICCAS 2013)*. 2013, pp. 1860–1865. DOI: `10.1109/ICCAS.2013.6704244`.

[142] T. Madani and A. Benallegue. "Backstepping Control for a Quadrotor Helicopter". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006, pp. 3255–3260. DOI: `10.1109/IROS.2006.282433`.

[143] J. Mark Keil. "Handbook of Computational Geometry || Polygon Decomposition". In: 2000. ISBN: 9780444825377. DOI: `10.1016/b978-044482537-7/50012-7`.

[144] Alessandro Matese et al. "Intercomparison of UAV, Aircraft and Satellite Remote Sensing Platforms for Precision Viticulture". In: *Remote Sensing* 7 (Mar. 2015), pp. 2971–2990. DOI: `10.3390/rs70302971`.

[145] Muller Hannemann Matthias and Karsten Weihe. "[ACM Press the thirteenth annual symposium - Nice, France (1997.06.04-1997.06.06)] Proceedings of the thirteenth annual symposium on Computational geometry - SCG '97 - Minimum strictly convex quadrangulations of convex polygons". In: 1997. ISBN: 0897918789. DOI: `10.1145/262839.262960`.

[146] J. McQueen. "Some methods for classification and analysis of multivariate observations". In: *Computer and Chemistry* 4 (Jan. 1967), pp. 257–272.

[147] S.S. Mehta and T.F. Burks. "Vision-based control of robotic manipulator for citrus harvesting". In: *Computers and Electronics in Agriculture* 102 (Mar. 2014). DOI: `10.1016/j.compag.2014.01.003`.

[148] Berenice Mettler, Mark Tischler, and Takeo Kanade. "System Identification of Small-Size Unmanned Helicopter Dynamics". In: *Annual Forum Proceedings - American Helicopter Society* 2 (Feb. 1999), 1706–1717.

[149] Nathan Michael et al. "Collaborative mapping of an earthquake-damaged building via ground and aerial robots". In: *Journal of Field Robotics* 29 (5 2012). DOI: `10.1002/rob.21436`.

[150] Ly Dat Minh and Cheolkeun Ha. "Modeling and control of quadrotor MAV using vision-based measurement". In: *International Forum on Strategic Technology 2010*. 2010, pp. 70–75. DOI: `10.1109/IFOST.2010.5668079`.

[151] Mostafa Mohammadi and Alireza Mohammad Shahri. "Adaptive Nonlinear Stabilization Control for a Quadrotor UAV: Theory, Simulation and Experimentation". In: *Journal of Intelligent and Robotic Systems* 72 (Oct. 2013), pp. 105–122. DOI: `10.1007/s10846-013-9813-y`.

[152] J. Mohammadpour and C.W. Scherer. *Control of Linear Parameter Varying Systems with Applications*. Springer, New York, 2012. ISBN: 978-1-4614-1833-7.

[153] Dipen Moitra. "Finding a minimal cover for binary images: An optimal parallel algorithm". In: *Algorithmica* 6 (1-6 June 1991). DOI: `10.1007/bf01759065`.

[154] A. Mokhtari, Abdelaziz Benallegue, and Boubaker Daachi. "Robust feedback linearization and GH_∞ controller for a quadrotor unmanned aerial vehicle". In: vol. 57. Sept. 2005, pp. 1198 –1203. DOI: `10.1109/IROS.2005.1545112`.

[155] Daniel Montero and Claudia Rueda. "Detection of palm oil bud rot employing artificial vision". In: *IOP Conference Series: Materials Science and Engineering* 437 (Oct. 2018), p. 012004. DOI: `10.1088/1757-899X/437/1/012004`.

[156] Andre Mora et al. “Land Cover Classification from Multispectral Data Using Computational Intelligence Tools: A Comparative Study”. In: *Information* 8 (2017), pp. 1–15. DOI: `10.3390/info8040147`.

[157] M. Morales, S. Rodriguez, and N.M. Amato. “2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422) - Improving the connectivity of PRM roadmaps”. In: *Improving the connectivity of PRM roadmaps*. Vol. 3. 2003. ISBN: 0-7803-7736-2. DOI: `10.1109/robot.2003.1242286`.

[158] Hassanalian Mostafa and Abdelkefi Abdessattar. “Classifications, applications, and design challenges of drones: A review”. In: *Progress in Aerospace Sciences* (May 2017). DOI: `10.1016/j.paerosci.2017.04.003`.

[159] R.T. Ng and Jiawei Han. “CLARANS: a method for clustering objects for spatial data mining”. In: *IEEE Transactions on Knowledge and Data Engineering* 14 (5 Sept. 2002). DOI: `10.1109/TKDE.2002.1033770`.

[160] Sing Nguang and Peng Shi. “Robust H_∞ output feedback control design for fuzzy dynamic systems with quadratic D stability constraints: An LMI approach”. In: *Information Sciences* 176 (Aug. 2006), pp. 2161–2191. DOI: `10.1016/j.ins.2005.02.005`.

[161] Tien Thanh Nguyen et al. “Detection of red and bicoloured apples on tree with an RGB-D camera”. In: *Biosystems Engineering* (Feb. 2016). DOI: `10.1016/j.biosystemseng.2016.01.007`.

[162] C.L. Nielsen and L.E. Kavraki. “A two level fuzzy PRM for manipulation planning”. In: *IEEE Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) - Takamatsu, Japan (2000.10.31-2000.11.5)*. 2000. ISBN: 0-7803-6348-5. DOI: `10.1109/iros.2000.895219`.

[163] Frank Nielsen. *Introduction to HPC with MPI for Data Science*. Sept. 2016. ISBN: 978-3-319-21902-8.

[164] Joseph O’Rourke et al. “A new linear algorithm for intersecting convex polygons”. In: *Computer Graphics and Image Processing* 19 (4 1982). DOI: `10.1016/0146-664x(82)90023-5`.

[165] Guang Ouyang, Dipak K. Dey, and Panpan Zhang. “Clique-Based Method for Social Network Clustering”. In: *Journal of Classification* (Apr. 2019). DOI: `10.1007/s00357-019-9310-5`.

[166] T. Ozaslan et al. “Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs”. In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1740–1747. DOI: `10.1109/LRA.2017.2699790`.

[167] Castillo P., Dzul A., and R. Lozano. “Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft”. In: *IEEE Transactions on Control Systems Technology* 12 (4 July 2004). DOI: `10.1109/TCST.2004.825052`.

[168] Luis Padua et al. “UAS, sensors, and data processing in agroforestry: a review towards practical applications”. In: *International Journal of Remote Sensing* (Mar. 2017). DOI: `10.1080/01431161.2017.1297548`.

[169] M L Parry et al. “Effects of climate change on global food production under SRES emissions and socio-economic scenarios”. In: *Global Environmental Change* 14.1 (Apr. 2004). An optional note, pp. 53 –67. DOI: `https://doi.org/10.1016/j.gloenvcha.2003.10.008`.

[170] Ron Patton, Robert Clark, and Paul M. Frank. *Fault Diagnosis in Dynamic Systems: Theory and Application*. Prentice-Hall, 1989. ISBN: 978-0133082630.

[171] Liam Paull, Sajad Saeedi, and Howard Li. “Path Planning for Autonomous Underwater Vehicles”. In: Springer, New York, NY, 2013, pp. 177–223. ISBN: 978-1-4614-5658-2 978-1-4614-5659-9. DOI: `10.1007/978-1-4614-5659-9\_4`.

[172] Hart P.E., Nilsson N.J., and Raphael B. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4 (2 1968). DOI: `10.1109/tssc.1968.300136`.

[173] Dan Pelleg and Andrew Moore. “Accelerating Exact k-means Algorithms with Geometric Reasoning”. In: (July 1999). DOI: `10.1145/312129.312248`.

[174] Jose M Pena-Barragan et al. “Weed Mapping in Early-Season Maize Fields Using Object-Based Analysis of Unmanned Aerial Vehicle (UAV) Images”. In: *PloS one* 8 (Oct. 2013), e77151. DOI: `10.1371/journal.pone.0077151`.

[175] The Hung Pham, Yasmina Bestaoui, and Said Mammar. “Aerial robot coverage path planning approach with concave obstacles in precision agriculture”. In: *Proceedings 2107 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)* (), pp. 43 –48. DOI: `10.1109/RED-UAS.2017.8101641`.

[176] Sammons Phil, Furukawa Tomonari, and Bulgin Andrew. “Autonomous Pesticide Spraying Robot for use in a Greenhouse”. In: *Australasian Conference on Robotics & Automation (ACRA2005), December 5 - 7, 2005, Canberra, Australia.* Sept. 2005.

[177] Bristeau Pierre-Jean et al. “The Role of Propeller Aerodynamics in the Model of a Quadrotor UAV”. In: *2009 European Control Conference, ECC 2009* (Aug. 2009).

[178] David Pimentel. “Global warming, population growth, and natural resources for food production”. In: *Society & Natural Resources* 4.4 (1991). PMID: 12344889, pp. 347–363. DOI: `10.1080/08941929109380766`.

[179] Pauline Pounds et al. “Design of a four-rotor aerial robot”. In: (2002), pp. 145–150.

[180] Jacopo Primicerio et al. “A flexible unmanned aerial vehicle for precision agriculture”. In: *Precision Agriculture* 13 (4 Aug. 2012). DOI: `10.1007/s11119-012-9257-6`.

[181] Feng Qingchun et al. “Design and test of robotic harvesting system for cherry tomato”. In: *International Journal of Agricultural and Biological Engineering* 11 (Jan. 2018), pp. 96–100. DOI: `10.25165/j.ijabe.20181101.2853`.

[182] Feng Qingchun et al. “Design of structured-light vision system for tomato harvesting robot”. In: *International Journal of Agricultural and Biological Engineering* 7 (Apr. 2014), pp. 19–26. DOI: `10.3965/j.ijabe.20140702.003`.

[183] L. Quebrajo et al. “Linking thermal imaging and soil remote sensing to enhance irrigation management of sugar beet”. In: *Biosystems Engineering* (Sept. 2017). DOI: `10.1016/j.biosystemseng.2017.08.013`.

[184] Shamshiri R R et al. “Robotic harvesting of fruiting vegetables: A simulation approach in V-REP, ROS and MATLAB”. In: *Proceedings in Automation in Agriculture-Securing Food Supplies for Future Generations, 2018, InTech.* 2018, pp. 81–105. DOI: `10.5772/intechopen.73861`.

[185] Paulina Lyubenova Raeva, Jaroslav Sedina, and Adam Dlesk. “Monitoring of crop fields using multispectral and thermal imagery from UAV”. In: *European Journal of Remote Sensing* (Oct. 2018). DOI: `10.1080/22797254.2018.1527661`.

[186] G.V. Raffo, Manuel Ortega, and Francisco Rubio. “An integral predictive nonlinear H_∞ control structure for a quadrotor helicopter”. In: *Automatica* 46 (Jan. 2010), pp. 29–39. DOI: `10.1016/j.automatica.2009.10.018`.

[187] Suneeta Ramaswami, Pedro Ramos, and Godfried Toussaint. "Converting triangulations to quadrangulations". In: *Computational Geometry* 9 (4 1998). DOI: `10.1016/s0925-7721(97)00019-9`.

[188] M. Rich, N. Elia, and P. Jones. "Design and implementation of an H_∞ controller for a quadrotor helicopter". In: *21st Mediterranean Conference on Control and Automation.* 2013, pp. 1189–1198. DOI: `10.1109/MED.2013.6608871`.

[189] Mahony Robert, Kumar Vijay, and Peter Corke. "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor". In: *IEEE Robotics & Automation Magazine* 19 (3 Sept. 2012). DOI: `10.1109/mra.2012.2206474`.

[190] Giulia Ronchetti et al. "Crop Row Detection through UAV Surveys to Optimize On-farm Irrigation Management". In: *Remote Sensing* 12 (12 June 2020). DOI: `10.3390/rs12121967`.

[191] Damiano Rotondo. "Advances in Gain-Scheduling and Fault Tolerant Control Techniques". PhD thesis. 2018. ISBN: 978-3-319-62901-8,978-3-319-62902-5. DOI: `10.1007/978-3-319-62902-5`.

[192] Francisco Rovira-Mas, Qin Zhang, and John F. Reid. "Stereo vision three-dimensional terrain maps for precision agriculture". In: *Computers and Electronics in Agriculture* 60 (2 2008). DOI: `10.1016/j.compag.2007.07.007`.

[193] A. Ruangwiset. "The application of unmanned aerial vehicle to precision agriculture: Verification experiments of the power consumption". In: *2014 International Conference on Information Science, Electronics and Electrical Engineering.* Vol. 2. 2014, pp. 968–971.

[194] K. Runcharoon and V. Srichatrapimuk. "Sliding Mode Control of quadrotor". In: *The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE).* 2013, pp. 552–557. DOI: `10.1109/TAEECE.2013.6557334`.

[195] Bouabdallah S., Murrieri P., and Siegwart R. "Design and control of an indoor micro quadrotor". In: *IEEE IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 - New Orleans, LA, USA (2004.04.26-2004.05.1).* 2004. ISBN: 0-7803-8232-3. DOI: `10.1109/robot.2004.1302409`.

[196] Pierpaolo Saccon. "Water for agriculture, irrigation management". In: *Applied Soil Ecology* (Nov. 2017). DOI: `10.1016/j.apsoil.2017.10.037`.

[197] Atheer Salih et al. "Flight PID Controller Design for a UAV Quadrotor". In: *Scientific research and essays* 5 (Dec. 2010), pp. 3660–3667.

[198] L.G Santesteban et al. "High-resolution UAV-based thermal imaging to estimate the instantaneous and seasonal variability of plant water status within a vineyard". In: *Agricultural Water Management* (Sept. 2016). DOI: `10.1016/j.agwat.2016.08.026`.

[199] Pablo Gonzalez-de Santos et al. "Fleets of robots for environmentally-safe pest control in agriculture". In: *Precision Agriculture* (2016). DOI: `10.1007/s11119-016-9476-3`.

[200] Alessandro dos Santos Ferreira et al. "Weed detection in soybean crops using ConvNets". In: *Computers and Electronics in Agriculture* 143 (Dec. 2017). DOI: `10.1016/j.compag.2017.10.027`.

[201] A. C. Satici, H. Poonawala, and M. W. Spong. "Robust Optimal Control of Quadrotor UAVs". In: *IEEE Access* 1 (2013), pp. 79–93. DOI: `10.1109/ACCESS.2013.2260794`.

[202] Carsten Scherer, Pascal Gahinet, and Mahmoud Chilali. "Multiobjective Output-feedback Control Via LMI Optimization". In: *IEEE Trans. Automat. Contr.* 42 (July 1997), pp. 896–911. DOI: `10.1109/9.599969`.

[203] Jeff Shamma and James Cloutiert. “Gain-Scheduled Missile Autopilot Design using Linear Parameter Varying Transformation”. In: *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM* 16 (Mar. 1993), pp. 256–263. DOI: `10.2514/3.20997`.

[204] J.S. Shamma. “An Overview of LPV Systems”. In: *Control of Linear Parameter Varying Systems with Applications*. Ed. by Mohammadpour Javad and Scherer Carsten W. Springer, 2012, pp. 3–26.

[205] J.S. Shamma. “Analysis and design of gain-scheduled control systems”. PhD thesis. Massachusetts Institute of Technology: Department Massachusetts Institute of Technology. Dept. of Mechanical Engineering, 1988.

[206] Redmond Shamshiri, Wan Ishak, and Wan Ishak Wan Ismail. “Nonlinear tracking control of a two link oil palm harvesting robot manipulator”. In: *International Journal of Agricultural and Biological Engineering* 5 (June 2012). DOI: `10.3965/j.ijabe.20120502.002`.

[207] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. “WaveCluster: a wavelet-based clustering approach for spatial data in very large databases”. In: *The VLDB Journal* 8 (3-4 Feb. 2000). DOI: `10.1007/s007780050009`.

[208] Guang Song, Miller S., and Amato N.M. “Customizing PRM roadmaps at query time”. In: *IEEE 2001 ICRA. IEEE International Conference on Robotics and Automation - Seoul, South Korea (21-26 May 2001)*. Vol. 2. 2001. ISBN: 0-7803-6576-3. DOI: `10.1109/robot.2001.932823`.

[209] Madeleine Stein, Suchet Bargoti, and James Underwood. “Image Based Mango Fruit Detection, Localisation and Yield Estimation Using Multiple View Geometry”. In: *Sensors* 16 (Nov. 2016), p. 1915. DOI: `10.3390/s16111915`.

[210] A. Stentz. “Optimal and efficient path planning for partially-known environments”. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. 1994, 3310–3317 vol.4. DOI: `10.1109/ROBOT.1994.351061`.

[211] J. Su, P. Fan, and K. Cai. “Attitude control of quadrotor aircraft via nonlinear PID”. In: 37 (Sept. 2011), pp. 1054–1058.

[212] Gerard Sylvester. *E-AGRICULTURE IN ACTION: DRONES FOR AGRICULTURE*. Tech. rep. Food, Agriculture Organization of the United Nations, and International Telecommunication Union, 2018.

[213] Hamel T., Mahony R., and Chriette A. “Visual servo trajectory tracking for a four rotor VTOL aerial vehicle”. In: *IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*. Vol. 3. 2002. ISBN: 0-7803-7272-7. DOI: `10.1109/ROBOT.2002.1013653`.

[214] Kazuo Tanaka and Hua O. Wang. *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. Wiley Inc, New York, 2001. DOI: `10.1002/0471224596`.

[215] Kanae Tanigaki et al. “Cherry-harvesting robot”. In: *Computers and Electronics in Agriculture* 63 (1 2008). DOI: `10.1016/j.compag.2008.01.018`.

[216] Hamel Tarek et al. “DYNAMIC MODELLING AND CONFIGURATION STABILIZATION FOR AN X4-FLYER”. In: *IFAC Proceedings Volumes* 35 (1 2002). DOI: `10.3182/20020721-6-es-1901.00848`.

[217] Alberto Tellaeche et al. “A new vision-based approach to differential spraying in precision agriculture”. In: *Computers and Electronics in Agriculture* 60 (2 2008). DOI: `10.1016/j.compag.2007.07.008`.

[218] J. Thomas et al. “Visual Servoing of Quadrotors for Perching by Hanging From Cylindrical Objects”. In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 57–64. DOI: `10.1109/LRA.2015.2506001`.

[219] Goran Tmusic et al. “Current Practices in UAS-based Environmental Monitoring”. In: *Remote Sensing* 12 (Mar. 2020), p. 1001. DOI: `10.3390/rs12061001`.

[220] T. Tomic et al. “Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue”. In: *IEEE Robotics Automation Magazine* 19.3 (2012), pp. 46–56.

[221] C. T. Ton and W. MacKunis. “Robust attitude tracking control of a quadrotor helicopter in the presence of uncertainty”. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC).* 2012, pp. 937–942. DOI: `10.1109/CDC.2012.6426266`.

[222] S. B. Tor and A. E. Middleditch. “Convex Decomposition of Simple Polygons”. In: *ACM Transactions on Graphics* 3 (4 Oct. 1984). DOI: `10.1145/357346.357348`.

[223] V. K. Tripathi, L. Behera, and N. Verma. “Disturbance observer based backstepping controller for a quadcopter”. In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society.* 2016, pp. 108–113. DOI: `10.1109/IECON.2016.7794007`.

[224] Asma Troudi et al. “Sizing of the Drone Delivery Fleet Considering Energy Autonomy”. In: *Sustainability* 10 (Sept. 2018), p. 3344. DOI: `10.3390/su10093344`.

[225] Acar Ercan U et al. “Morse Decompositions for Coverage Tasks”. en. In: *The International Journal of Robotics Research* (2002), pp. 331–344. DOI: `https://doi.org/10.1177/027836402320556359`.

[226] Fayyad U.M et al. *Advances In Knowledge Discovery And Data Mining.* American Association for Artificial Intelligence, 445 Burgess Drive Menlo Park, CAUnited States, 1996. ISBN: 978-0-262-56097-9.

[227] James P. Underwood et al. “Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors”. In: *Computers and Electronics in Agriculture* 130 (Nov. 2016). DOI: `10.1016/j.compag.2016.09.014`.

[228] Joao Valente et al. “Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach”. In: *Computers and Electronics in Agriculture* 99 (Nov. 2013). DOI: `10.1016/j.compag.2013.09.008`.

[229] E.J. Van Henten et al. “An Autonomous Robot for Harvesting Cucumbers in Greenhouses”. In: *Autonomous Robots* 13 (Nov. 2002), pp. 241–258. DOI: `10.1023/A:1020568125418`.

[230] Ibrahim Wahab, Ola Hall, and Magnus Jirström. “Remote Sensing of Yields: Application of UAV Imagery-Derived NDVI for Estimating Maize Vigor and Yields in Complex Farming Systems in Sub-Saharan Africa”. In: *Drones* 2 (3 Aug. 2018). DOI: `10.3390/drones2030028`.

[231] S. Waharte and N. Trigoni. “Supporting Search and Rescue Operations with UAVs”. In: *2010 International Conference on Emerging Security Technologies.* 2010, pp. 142–147. DOI: `DOI:10.1109/EST.2010.31`.

[232] Jing Wang. “Quadrotor analysis and model free control with comparisons”. PhD thesis. Université Paris Sud - Paris XI, 2013. URL: `https://tel.archives-ouvertes.fr/tel-00952401/document`.

[233] L.L. Wang et al. “Development of a tomato harvesting robot used in greenhouse”. In: *International Journal of Agricultural and Biological Engineering* 10 (Jan. 2017), pp. 140–149. DOI: `10.25165/j.ijabe.20171004.3204`.

[234] W. Wang, J. Yang, and Richard Muntz. "STING: A statistical information grid approach to spatial data mining". In: *VLDB* 97 (Jan. 1997), pp. 186–195.

[235] Sheng Wen et al. "Design of Plant Protection UAV Variable Spray System Based on Neural Networks". In: *Sensors* 19 (5 Mar. 2019). DOI: `10.3390/s19051112`.

[236] Jan Wendel et al. "An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter". In: *Aerospace Science and Technology* 10 (6 2006). DOI: `10.1016/j.ast.2006.04.002`.

[237] S.A. Wilmarth, N.M. Amato, and P.F. Stiller. "MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the free space". In: *IEEE International Conference on Robotics and Automation - Detroit, MI, USA (10-15 May 1999)*. Vol. 2. 1999. ISBN: 0-7803-5180-0. DOI: `10.1109/robot.1999.772448`.

[238] B. Wojciechowski. "Analysis and synthesis of proportional-integral observers for single-input-single-output time-invariant continuous systems". PhD thesis. July 1978.

[239] V. Wuest, V. Kumar, and G. Loianno. "Online Estimation of Geometric and Inertia Parameters for Multirotor Aerial Vehicles". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 1884–1890.

[240] www.fao.org. *Healthy people depend on healthy food systems*. Tech. rep. Food and Agriculture Organization of the United Nations (FAO) (2013), 2013.

[241] Jing-Jing Xiong and En-Hui Zheng. "Optimal Kalman Filter for state estimation of a quadrotor UAV". In: *Optik - International Journal for Light and Electron Optics* (July 2015). DOI: `10.1016/j.ijleo.2015.07.032`.

[242] Tengke Xiong et al. "DHCC: Divisive hierarchical clustering of categorical data". In: *Data Mining and Knowledge Discovery* 24 (1 Jan. 2012). DOI: `10.1007/s10618-011-0221-2`.

[243] R. Xu and U. Ozguner. "Sliding Mode Control of a Quadrotor Helicopter". In: *Proceedings of the 45th IEEE Conference on Decision and Control*. 2006, pp. 4957–4962. DOI: `10.1109/CDC.2006.377588`.

[244] Xinyu Xue et al. "Develop an unmanned aerial vehicle based automatic aerial spraying system". In: *Computers and Electronics in Agriculture* 128 (Oct. 2016). DOI: `10.1016/j.compag.2016.07.022`.

[245] Zhai Yuyi et al. "Control System Design for a Surface Cleaning Robot". In: *International Journal of Advanced Robotic Systems* 10.5 (2013), p. 220. ISSN: 1729-8814. DOI: `10.5772/56200`.

[246] Guy Zaidner and Amir Shapiro. "A novel data fusion algorithm for low-cost localisation and navigation of autonomous vineyard sprayer robots". In: *Biosystems Engineering* (May 2016). DOI: `10.1016/j.biosystemseng.2016.05.002`.

[247] Ji Zhang and Sanjiv Singh. "Laser-visual-inertial Odometry and Mapping with High Robustness and Low Drift". In: *Journal of Field Robotics* (Aug. 2018), pp. 1–23. DOI: `10.1002/rob.21809`.

[248] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. "BIRCH: An Efficient Data Clustering Method for Very Large Databases". In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 25 (Sept. 1999). DOI: `10.1145/233269.233324`.

[249] H. Zheng et al. "Evaluation of a UAV-based hyperspectral frame camera for monitoring the leaf nitrogen concentration in rice". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016, pp. 7350–7353.

[250] X. Zhou et al. "Predicting grain yield in rice using multi-temporal vegetation indices from UAV-based multispectral and digital imagery". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 130 (Aug. 2017). DOI: `10.1016/j.isprsjprs.2017.05.003`.

[251] Jan M. Zytkow and Jan Rauch. "[Lecture Notes in Computer Science] Principles of Data Mining and Knowledge Discovery Volume 1704 || OPTICS-OF: Identifying Local Outliers". In: vol. 10.1007/b72280. 1999. ISBN: 978-3-540-66490-1,978-3-540-48247-5. DOI: `10.1007/978-3-540-48247-5_28`.

[252] Z.Zhang and M. Cong. "Controlling Quadrotors Based on Linear Quadratic Regulator". en. In: *Journal of Robotic Systems* 5.38 (2011), pp. 38–42.

ÉCOLE DOCTORALE
Sciences et technologies
de l'information et de
la communication (STIC)

université
PARIS-SACLAY

Titre: Planification et Commande robustes de véhicules aériens sans pilote

Mots clés: Systèmes multi-modèle Takagi-Sugeno, Commande robuste, Planification et suivi de trajectoire, Mission coordonnée, Optimisation énergétique, Commande déportée

Résumé: L'objectif de cette thèse est de réaliser la modélisation, la planification de trajectoire et le contrôle d'un robot hélicoptère sans pilote pour la surveillance de grandes surfaces, en particulier dans des applications d'agriculture de précision. Plusieurs tâches de l'agriculture de précision sont abordées. Dans les missions de surveillance des ravageurs, les drones seront équipés de caméras spécialisées. Une trajectoire sera recherchée et créée pour permettre aux aéronefs sans pilote de capturer des images de zones de cultures entières et d'éviter les obstacles pendant le vol. Les zones infectées seront ensuite identifiées en analysant les images prises. Lors de la pulvérisation d'insecticides, l'aéronef doit être contrôlé pour voler selon une trajectoire préprogrammée et pulvériser l'insecticide sur toutes les zones de culture infectées.

Dans la première partie, nous présentons un nouvel algorithme de planification de chemin de couverture complet en proposant une nouvelle décomposition cellulaire qui repose sur une généralisation de la variante Boustrophédon, à l'aide de fonctions Morse, avec une extension de la représentation des points critiques. Cette extension conduit à un nombre réduit de cellules après décomposition. L'algorithme génétique (GA) et l'algorithme de problème du voyageur de commerce (TSP) sont ensuite appliqués pour obtenir le chemin le plus court pour une couverture complète. Ensuite, à partir des informations sur la carte concernant les coordonnées des obstacles, des zones infectées et non infectées, les zones infectées sont divisées en plusieurs régions non chevauchantes en utilisant une technique de regroupement. Ensuite, un algorithme est proposé pour générer le meilleur chemin pour qu'un véhicule aérien sans pilote (UAV) distribue des médicaments à toutes les zones infectées d'un environnement agricole qui contient des obstacles non convexes, des zones exemptes de parasites et des zones infestées de parasites.

Dans la deuxième partie, nous étudions la conception d'un système de contrôle robuste qui permet au véhicule de suivre la trajectoire prédéfinie d'un hélicoptère à modèle dynamique variable en raison des changements de coefficients dynamiques tels que la masse et les moments d'inertie. Par conséquent, les lois robustes d'observation et de contrôle sont nécessaires pour adopter les changements des paramètres dynamiques ainsi que l'impact des forces externes. La méthode proposée consiste à explorer les techniques de modélisation, de planification et de contrôle par l'approche LPV. Pour avoir des algorithmes facilement implantables et adaptables aux changements de paramètres et de conditions d'utilisation, nous privilégions la synthèse de l'Observateur d'Entrées Inconnues (UIO) à Paramètre Linéaire Variable (LPV), et des controlleurs retour d'état quadratique LPV, retour d'état robuste et retour de sortie statique. L'observateur et les contrôleurs sont conçus en résolvant un ensemble d'inégalités matricielles linéaires (LMI) obtenues à partir du lemme réel borné et de la caractérisation des régions LMI.

Enfin, pour mettre en évidence les performances des algorithmes de planification de trajectoire et des lois de contrôle générées, nous effectuons une série de simulations à l'aide de MATLAB Simulink. Les résultats de la simulation sont assez prometteurs. L'algorithme de planification de trajectoire de couverture suggère que la trajectoire générée raccourcit la distance de vol de l'aéronef mais évite toujours les obstacles et couvre toute la zone d'intérêt. Les simulations pour l'observateur LPV UIO et les contrôleurs LPV sont effectuées avec les cas où la masse et les moments d'inertie changent brusquement et lentement. Le LPV UIO est capable d'estimer les variables d'état et les perturbations inconnues et les valeurs estimées convergent vers les vraies valeurs des variables d'état et les perturbations inconnues de manière asymptotique. Les contrôleurs LPV fonctionnent bien pour divers signaux de référence (impulsion, aléatoire, constant et sinusoïdale) et plusieurs types de perturbations (impulsionnelle, aléatoire, constante et sinusoïdale).

Title: Robust planning and control of unmanned aerial vehicles

Keywords: Takagi-Sugeno Multi-model systems, robust control, trajectory planning and tracking, coordinated mission, optimization in energy

Abstract: The objective of this thesis is to realize the modeling, trajectory planning, and control of an unmanned helicopter robot for monitoring large areas, especially in precision agriculture applications. Several tasks in precision agriculture are addressed. In pest surveillance missions, drones will be equipped with specialized cameras. A trajectory will be researched and created to enable unmanned aircraft to capture images of entire crop areas and avoid obstacles during flight. Infected areas will be then identified by analyzing taken images. In insecticides spraying, the aircraft must be controlled to fly in a pre-programmed trajectory and spray the insecticide over all the infected crop areas.

In the first part, we present a new complete coverage path planning algorithm by proposing a new cellular decomposition which is based on a generalization of the Boustrophedon variant, using Morse functions, with an extension of the representation of the critical points. This extension leads to a reduced number of cells after decomposition. Genetic Algorithm (GA) and Travelling Salesman Problem (TSP) algorithm are then applied to obtain the shortest path for complete coverage. Next, from the information on the map regarding the coordinates of the obstacles, non-infected areas, and infected areas, the infected areas are divided into several non-overlapping regions by using a clustering technique. Then an algorithm is proposed for generating the best path for a Unmanned Aerial Vehicle (UAV) to distribute medicine to all the infected areas of an agriculture environment which contains non-convex obstacles, pest-free areas, and pests-ridden areas.

In the second part, we study the design of a robust control system that allows the vehicle to track the predefined trajectory for a dynamic model-changing helicopter due to the changes of dynamic coefficients such as the mass and moments of inertia. Therefore, the robust observer and control laws are required to adopt the changes in dynamic parameters as well as the impact of external forces. The proposed approach is to explore the modeling techniques, planning, and control by the Linear Parameter Varying (LPV) type technique. To have easily implantable algorithms and adaptable to changes in parameters and conditions of use, we favor the synthesis of Linear Parameter Varying (LPV) Unknown Input Observer (UIO), LPV quadratic state feedback, robust state feedback, and static output feedback controllers. The observer and controllers are designed by solving a set of Linear Matrix Inequality (LMI) obtained from the Bounded Real Lemma and LMI regions characterization.

Finally, to highlight the performances of the path planning algorithms and generated control laws, we perform a series of simulations in MATLAB Simulink environment. Simulation results are quite promising. The coverage path planning algorithm suggests that the generated trajectory shortens the flight distance of the aircraft but still avoids obstacles and covers the entire area of interest. Simulations for the LPV UIO and LPV controllers are conducted with the cases that the mass and moments of inertias change abruptly and slowly. The LPV UIO is able to estimate state variables and the unknown disturbances and the estimated values converge to the true values of the state variables and the unknown disturbances in asymptotically. The LPV controllers work well for various reference signals (impulse, random, constant, and sine) and several types of disturbances (impulse, random, constant, and sine).