FUSED FILAMENT FABRICATION VIA MULTI QUADCOPTER COLLABORATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MATIN GHAZIANI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

JULY 2023

Approval of the thesis:

FUSED FILAMENT FABRICATION VIA MULTI QUADCOPTER COLLABORATION

submitted by **MATIN GHAZIANI** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. M. A. Sahir Arıkan
Head of Department, **Mechanical Engineering** _____

Assoc. Prof. Dr. Ulaş Yaman
Supervisor, **Mechanical Engineering, METU** _____

Assoc. Prof. Dr. Ali Emre Turgut
Co-supervisor, **Mechanical Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. A. Buğra Koku
Mechanical Engineering, METU _____

Assoc. Prof. Dr. Ulaş Yaman
Mechanical Engineering, METU _____

Assoc. Prof. Dr. Kıvanç Azgın
Mechanical Engineering, METU _____

Assoc. Prof. Dr. Hakkı Özgür Ünver
Mechanical Engineering, TOBB ETU _____

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering, TED University _____

Date:12.07.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: MATIN GHAZIANI

Signature :

ABSTRACT

FUSED FILAMENT FABRICATION VIA MULTI QUADCOPTER COLLABORATION

GHAZIANI, MATIN
Ph.D., Department of Mechanical Engineering
Supervisor: Assoc. Prof. Dr. Ulaş Yaman
Co-Supervisor: Assoc. Prof. Dr. Ali Emre Turgut

July 2023, 134 pages

This thesis explores the integration of multirotor unmanned aerial vehicles (UAVs) with three Dimensional (3D) extruders to overcome the limitations of traditional 3D printers regarding size and production volume. The motivation behind this research is to enable the production of large and complex structures without the need for costly expansions of fixed-size 3D printers. The limitations of existing 3D printers and industrial robots in size and working area are discussed, highlighting the potential of UAVs, particularly multirotor UAVs, in overcoming these limitations. Multirotor UAVs equipped with 3D extrusion systems offer the advantage of producing or repairing structures without being constrained by size or positional conditions, making them especially useful in hazardous or inaccessible environments.

The thesis addresses various aspects related to 3D printing with multirotor UAVs. It emphasizes the design considerations for lightweight manipulator systems, such as gimbals that can be attached to UAVs to perform the 3D printing process. The need for highly accurate flight controllers to compensate for unwanted movements of the UAVs during printing operations is also highlighted. Furthermore, the concept

of swarm UAVs is explored, where multiple UAVs work together autonomously to perform a specific task, offering advantages such as time savings, reduced labor, and increased operational efficiency.

Based on the conclusions drawn from each chapter, it is evident that integrating multirotor UAVs with 3D extruders holds significant potential for expanding the capabilities of 3D printing. The research findings demonstrate the feasibility and advantages of using UAVs for 3D printing, including precise and autonomous operations. The thesis contributes to the field of additive manufacturing by showcasing the potential of multirotor UAVs in overcoming the limitations of traditional 3D printers.

Future research in this area could focus on further improving the accuracy and reliability of the 3D printing process with UAVs. This could involve advancements in lightweight manipulator designs, more robust flight control algorithms, and enhanced navigation and obstacle detection systems. Additionally, exploring the scalability and coordination of swarm UAVs for additive manufacturing could be an exciting avenue for future investigation.

Keywords: Additive Manufacturing, Multirotor UAVs, Gimbal, Path Planning

ÖZ

ÇOKLU DRON İŞBİRLİĞİNDE ERGİYİK FİLAMENT İLE İMALAT

GHAZIANI, MATIN
Doktora, Makina Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Ulaş Yaman
Ortak Tez Yöneticisi: Doç. Dr. Ali Emre Turgut

Temmuz 2023 , 134 sayfa

Bu tez çalışması, çok rotorlu insansız hava araçlarının (İHA'lar) 3B ekstrüzyon sistemleriyle entegrasyonunu araştırarak geleneksel 3B yazıcıların boyut ve üretim hacmiyle ilgili kısıtlamalarını aşmayı amaçlamaktadır. Bu araştırmanın motivasyonu, sabit boyutlu 3B yazıcıların maliyetli genişlemelere ihtiyaç duymadan büyük ve karmaşık yapıların üretimini mümkün kılmaktır. Mevcut 3B yazıcılar ve endüstriyel robotlarının boyut ve çalışma alanı kısıtlamaları tartışılarak, İHA'ların, özellikle çok rotorlu İHA'ların, bu kısıtlamaları aşma potansiyeli vurgulanmaktadır. 3B ekstrüzyon sistemleriyle donatılmış çok rotorlu İHA'lar, boyut veya konumsal koşullarla sınırlanmadan yapı üretme veya tamir etme avantajı sunarak, özellikle tehlikeli veya erişilemez ortamlarda faydalı olmaktadır.

Tez çalışması, çok rotorlu İHA'larla 3B baskı ile ilgili çeşitli yönleri ele almaktadır. İHA'ya bağlanabilen gimbal gibi hafif manipülatör sistemlerinin tasarım hususlarına vurgulanmaktadır. Baskı işlemleri sırasında İHA'ların istenmeyen hareketlerini telafi etmek için yüksek hassasiyetli uçuş kontrol cihazlarının gerekliliği de vurgulanmıştır. Ayrıca, bir görevi gerçekleştirmek için birden fazla İHA'nın otonom olarak birlikte

çalıştığı sürü İHA'larının kavramı da araştırılmış olup sunduğu zaman tasarrufu, azaltılmış işgücü ve artırılmış operasyonel verimlilik gibi avantajlar açıklanmıştır.
Gerçekleştirilen çalışmaların sonuçları değerlendirildiğinde çok rotorlu İHA'ların 3B ekstrüzyon cihazlarıyla entegrasyonunun 3B baskı yeteneklerini genişletme konusunda önemli bir potansiyele sahip olduğu açıktır. Araştırma bulguları, İHA'ların hassas ve otonom işlemler de dahil olmak üzere 3B baskı için kullanımının olası avantajlarını göstermektedir. Tez, çok rotorlu İHA'ların geleneksel 3B yazıcıların kısıtlamalarını aşma potansiyelini göstererek, katı üretim alanına katkıda bulunmaktadır.
Bu alanda gelecekteki araştırmalar, İHA'larla 3B baskı sürecinin doğruluğunu ve güvenilirliğini daha da iyileştirmeye odaklanabilir. Örneğin, hafif manipülatör tasarımlarında, daha sağlam uçuş kontrol algoritmaları ve gelişmiş navigasyon ve engel algılama sistemlerinde araştırmalar gerçekleştirilebilir. Ayrıca, katı üretim için sürü İHA'ların ölçeklenebilirliği ve koordinasyonunu keşfetmek, gelecekteki araştırmalar için heyecan verici bir alan olabilir.

Anahtar Kelimeler: Eklemeli imalat , Çok Rotorlu İHA'lar, Gimbal, Yörünge Planlama

I dedicate my dissertation to my daughter Venüs

ACKNOWLEDGMENTS

First and foremost, I express my sincere gratitude to my supervisors, Associate Professor Dr. Ulaş YAMAN, and Associate Professor Dr. Ali Emre TURGUT for their exceptional guidance, unwavering support, valuable insights, and constructive criticism throughout this study. Their expertise and dedication have been instrumental in shaping the direction and quality of this research, and I am truly grateful for their mentorship.

I am deeply indebted to my loving family for their constant support, unconditional love, and encouragement. Their belief in my abilities and unwavering encouragement has motivated and inspired me. I want to express my profound gratitude to my beloved wife, Ezgi ÇAĞLAR GHAZİANİ, for her understanding, patience, and unwavering support. Her love, encouragement, and sacrifices have been instrumental in enabling me to focus on this work and complete it successfully. Her presence has been a constant source of strength and happiness.

I want to thank my dear friends and colleagues, Hamit TEKİN and Masoud LATIFINAVID, for their friendship, encouragement, and technical assistance throughout the thesis. Their collaboration and shared insights have greatly enriched the research process, and I am fortunate to have had their support.

I express my profound appreciation to all those who have contributed to this research and supported me along this journey. Your support and encouragement have been invaluable, and I am genuinely grateful for the privilege of working with such exceptional individuals.

Finally, I sincerely thank Turkey's Scientific and Technological Research Council (TÜBİTAK) for supporting this study financially with 122E678 project number of TÜBİTAK 1002B. Their support has played a crucial role in facilitating the research process and has made it possible to carry out this study effectively.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xix

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation and Problem Definition 1

1.2 Contributions and Novelties 6

1.3 Proposed Methods and Models 7

1.4 The Outline of the Thesis 10

2 DYNAMIC MODEL OF THE SYSTEM 11

2.1 Introduction 11

2.2 Quadcopter Model 13

2.2.1 Kinematic of a Quadcopter 14

2.2.2 Dynamics Model of a Quadcopter 15

2.3 Design and Manufacturing of the Gimbal 24
2.3.1 System Design 25
2.3.2 Gimbal Model 27
2.3.2.1 Kinematics of the Gimbal Mechanism 29
Forward Kinematics of the Gimbal Mechanism: 31
2.3.2.2 Dynamic Model of the Gimbal Mechanism 34
2.3.2.3 Kinetic Energy of the Gimbal Mechanism 35
2.4 Controlling of the Gimbal 38
2.4.1 Tracking a Target Object 40
2.4.2 Object State Estimation by Using The Kalman Filter 41
2.4.3 Target Object Tracking Algorithm 45
2.5 Conclusion 50
3 CAMERA BASED MULTI QUADCOPTER SIMULATION 53
3.1 Introduction 53
3.2 Controlling of the Gimbal Using Camera 54
3.3 Image Processing 58
3.3.1 Moment method 58
3.3.2 Contour Line Detection 59
3.3.3 Hough Detection 59
3.4 Simulation Environment 60
3.5 Results of the Simulation 66
3.6 Conclusion 71
4 MULTI QUADCOPTER PATH PLANNING FOR AREA COVERAGE . . 73

4.1 Introduction . . . 73
4.2 Problem Definition . . . 75
4.3 Methodology . . . 78
4.4 Simulations . . . 81
4.4.1 CPP Algorithm with Constraints . . . 82
4.4.2 Simulations in Gazebo . . . 87
4.4.3 Real-World Tests . . . 89
4.5 Conclusion . . . 93
5 TEST PLATFORMS . . . 95
5.1 Introduction . . . 95
5.2 Design of the Test Platform . . . 96
5.3 System Model . . . 99
5.3.1 Possible Errors in the System . . . 100
5.3.1.1 Possible Errors in the XY Plane . . . 101
5.3.1.2 Possible Errors in the Z-Direction . . . 104
5.3.1.3 Resistance during 3D Printing with Quadcopters . . . 106
5.4 Experimental Results . . . 108
5.4.1 Line Following Test . . . 110
5.4.2 Gimbal Performance Test . . . 111
5.4.2.1 Gimbal Performance in XY Plane . . . 112
5.4.2.2 Gimbal Performance in Z-direction . . . 113
5.4.2.3 Gimbal Performance during 3D Printing . . . 116
5.5 Conclusion . . . 119

6 CONCLUSIONS 121

REFERENCES 123

LIST OF TABLES

TABLES

Table 3.1 Open-Source Simulator 60

Table 3.2 Result of the Multi Drone Simulation using Contour Method 70

Table 3.3 Result of the Multi Drone Simulation using Hough Method 70

Table 3.4 Result of the Multi Drone Simulation using Moment Method 70

LIST OF FIGURES

FIGURES

Figure 1.1 Flowchart of the proposed system 8

Figure 2.1 Quadcopter Forces and Moments 18

Figure 2.2 Different types of the gimbals 25

Figure 2.3 CAD model of the gimbal 26

Figure 2.4 Dimension of the gimbal 27

Figure 2.5 Coordinate system of the gimbal 28

Figure 2.6 Simulink of the control system 39

Figure 2.7 The output of the X_{cmd} with the output of Kalman filter 43

Figure 2.8 The output of the Y_{cmd} with the output of Kalman filter 44

Figure 2.9 The performance of the quadcopter in tracking the target in X direction 45

Figure 2.10 The performance of the quadcopter in tracking the target in Y direction 46

Figure 2.11 The performance of the gimbal in tracking the target in X direction 47

Figure 2.12 The performance of the gimbal in tracking the target in Y direction 47

Figure 2.13 θ_1 of the gimbal mechanism 49

Figure 2.14 θ_2 of the gimbal mechanism 49

Figure 2.15 θ_3 of the gimbal mechanism 50

Figure 3.1 The flow diagram for the algorithm 55

Figure 3.2 Coordinate system of the camera image, Cartesian and NED . . . 57

Figure 3.3 Heading of the quadcopter 61

Figure 3.4 Road position in the image of gazebo 63

Figure 3.5 ROS Graph for multi-drone simulation 64

Figure 3.6 Gazebo world for testing the multi-drone simulation 65

Figure 3.7 Contour method 67

Figure 3.8 Hough method 68

Figure 3.9 Moment method 69

Figure 4.1 Infill Pattern Types 76

Figure 4.2 Infill patterns for the rectilinear method 77

Figure 4.3 Path of the multi-drone 78

Figure 4.4 Sweep direction 79

Figure 4.5 Path in the square shape 82

Figure 4.6 The result of the CPP with constraints 86

Figure 4.7 Operation plan. Each UAV takes off, operates at mission altitude, and RTLs at the allotted altitude. 87

Figure 4.8 Position of the three drones in Gazebo simulation 88

Figure 4.9 The result of the CPP in Gazebo simulation 88

Figure 4.10 Real-world test for CPP 90

Figure 4.11 The result of the real-world CPP 91

Figure 4.12 Performance of drones 92

Figure 5.1 Test platform with mounted gimbal 97

Figure 5.2 Dimension of the test platform 98

Figure 5.3 The quadcopter's instantaneous variations 99

Figure 5.4 Basic principles of quadcopter maneuvering 100

Figure 5.5 Test Platform 109

Figure 5.6 Test Platform - Right view 110

Figure 5.7 The result of the circle and square shape single layer 3D printing 111

Figure 5.8 Position of the IMUs on the platform 112

Figure 5.9 The effect of the gimbal 113

Figure 5.10 Z axis disturbance generator 114

Figure 5.11 Position of the Z axis, vibration damping, and gimbal 115

Figure 5.12 Result of the vibration damping part test 115

Figure 5.13 Circle method 117

Figure 5.14 Square method 118

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
CAD	Computer-Aided Design
CPP	Coverage Path Planning
DoF	Degree of Freedom
EKF	Extended Kalman Filter
FFF	Fused Filament Fabrication
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MAVLink	Micro Air Vehicle Link
MAVROS	MAVLink on ROS
NED	North, East, Down
PID	Proportional, Integral, Derivative
RPM	Revolutions Per Minute
RTk	Real Time Kinematic
RTL	Return to Launch
ROS	Robot Operating System
SINS	Strap-down Inertial Navigation System
SITL	Software in the Loop
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VRP	Vehicle Routing Problem

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Nowadays, 3D printers have revolutionized production methods in various applications by enabling the quick and precise production of geometrically complex parts. However, the size of the parts that can be produced is limited by the size of the 3D printer itself. The printer must be expanded to manufacture larger parts, which incurs additional costs [1]. Similarly, industrial robots, while facilitating the fabrication of non-standard structures and saving time and costs, have limitations in their working areas. The maximum workpiece size that a robot can handle is constrained by its maximum working area [2]. To overcome these limitations, mobile robot arms have been employed to expand production volumes [3]. However, a single mobile robot arm may not be sufficient for large-scale objects or construction applications, necessitating multiple mobile robot arms.

Recently, UAVs, also known as drones, have emerged as a solution to overcome limitations in production volume. Multirotor UAVs, in particular, have gained popularity and are being utilized in various applications such as remote sensing [4], aerial photography [5], infrastructure supervision, distribution, intelligent agriculture [6, 7, 8], surveillance [9], and delivery [10]. One recent application involves using multirotor UAVs equipped with 3D extrusion systems for additive manufacturing [11]. This type of UAV allows for the production or repair of structures without being constrained by size and positional conditions, making it especially useful in hazardous or inaccessible environments.

Different types of equipment can be employed to enable 3D printing with multirotor UAVs. In the literature, special manipulators have been designed to hold the 3D

printer extruders [12, 13, 14, 15]. Due to the payload limitations of multirotor UAVs, the equipment designed for these UAVs must be lightweight. One common type of manipulator used is a gripper, which is a simple form of the manipulator. However, since these grippers are directly installed on the UAVs, they cannot compensate for the unwanted movements of the UAV during operation.

To address this challenge, highly accurate flight controllers are required when using grippers with UAVs. Achieving precise flight control is difficult for UAVs due to the aerodynamic conditions. Different types of manipulator systems have been proposed to overcome this sensitivity problem, such as integrating the kinematic model of the manipulator with the UAV [16, 17, 18, 19].

The application of UAVs is generally limited by payload capacity, flight time, and the maximum distance from the ground control station. As a result, flights are often interrupted, requiring the UAV to return to the central ground station for battery recharge or to add/remove payloads. In specific tasks, the use of multiple UAVs becomes inevitable. In such cases, UAVs must be capable of autonomous flight, making decisions using sensors, and communicating with each other. Swarm UAVs are a group of UAVs that fly autonomously or are controlled through a ground control station simultaneously to perform a specific task. Communication and adherence to the assigned flight paths are crucial to prevent collisions between UAVs.

Recently, there has been an emphasis on the applicability of UAVs in operations involving lightweight structures, as evidenced by several studies. For example, at ETH Zurich, a project utilized four UAVs to construct a 6-meter indoor tower made of foam bricks [20]. Another project at ETH Zurich used UAVs to connect cable structures [21]. Lindsey et al. utilized UAVs in a swarm configuration to connect small-scale columns and beam structures using magnets [22]. A separate literature review related to 3D production (additive manufacturing) using UAVs, production materials, navigation, and applications of UAVs in swarm configurations has been conducted and presented below.

1. The construction industry has shown great interest in initially utilizing 3D printers to produce components (e.g., joints, structural elements) and eventually moving towards constructing entire buildings. However, the size limitations of existing 3D printers have led to envision using UAVs capable of 3D production

to construct future buildings. There are several studies in the literature on additive manufacturing with UAVs [23, 24]. While using UAVs offers numerous advantages in this context, some limitations need to be addressed, such as minimizing weight, improving positional accuracy, battery charging, and stopping at central stations for loading production materials.

2. One of the challenges in 3D production with UAVs is their low load capacity [25], which makes it impossible to place 3D printers and materials onto the UAVs directly. Therefore, appropriate designs need to be developed for UAVs to support 3D production [26]. In this study, a special extruder and auxiliary equipment will be used. The equipment used for 3D production with UAVs includes various types of manipulators, such as simple graspers, high-degree-of-freedom mechanisms, extruders, and electromechanical parts. Previous studies in the literature have explored different gripper designs for UAVs. One common approach is using single Degree of Freedom (DoF) graspers [12, 25, 27, 28], but the position of the UAV limits these gripper systems. To achieve precise control and manipulation, lightweight and multi-degree-of-freedom mechanisms are required. Keemink et al. elaborated on the mechanical design and modeling of a manipulator system for UAVs in their study [29]. Other studies have focused on the modeling and control of miniature UAVs [30, 31], as well as the use of manipulators with multiple degrees of freedom and vehicles with double-arm systems [32, 33, 34, 35, 36, 37, 38, 39, 40]. Supervisory control has also been employed in some research [41, 42], where a Cartesian-based impedance controller was simulated for 2 DoF manipulators connected to a UAV. Additionally, researchers have explored controlling the balance of UAVs based on variations in the center of mass and moment of inertia [16, 43, 44, 45]. Previous studies have utilized simple 2 DoF manipulators for 3D production with UAVs [23, 26], but these mechanisms have not achieved precise 3D outputs. To address this challenge, the delta mechanism has been tested [24], where 3D production was achieved by integrating delta manipulators [46, 47] with UAVs to perform high-precision tasks.

3. The navigation system and supporting subsystems are crucial for enabling the autonomous flight of UAVs. To ensure proper operation of the navigation sys-

tem, the position (x, y, z) and orientations (ϕ, θ, ψ) of the UAV must be estimated using information from various sensors. Additionally, other supporting systems are required to detect and avoid obstacles (static or dynamic) and update motion trajectories. The navigation systems typically consist of three main subsystems:

(a) Location Estimation: Location estimation involves estimating the position and direction of the UAV during movement. This process relies on information from one or more sensors, such as Inertial Measurement Unit (IMU), Global Positioning System (GPS), camera, laser, and ultrasonic sensors. The position information can be obtained from each sensor individually or by combining data from different sensors. Location estimation is considered a fundamental step in any navigation or mapping process. GPS is a well-known method for location estimation. In some systems, GPS and Inertial Navigation System (INS) are used together to compensate for GPS errors or inaccessibility to satellites in bad weather conditions. Various approaches have been employed to improve position estimation, such as combining data from GPS and INS using a linear Kalman filter [48] or an Extended Kalman Filter (EKF) [49], a Kalman-Complementary filter [50], or GPS-combined SINS (Strapdown Inertial Navigation System) [51, 52]. Other studies have focused on combining GPS data with camera information and IMU data to improve position estimation using Unscented Kalman Filter (UKF) [53]. Some studies have also incorporated altitude sensors to enhance the measurement accuracy of GPS/INS [54, 55], although it has been noted that this approach may not work well at low altitudes due to the ground effect on barometric sensors [56].

(b) Obstacle Detection and Avoidance: Obstacle detection and avoidance play a critical role in the autonomy of any UAV system. These systems enable the detection of obstacles and the planning of routes to avoid collisions. Image processing-based navigation systems employ various strategies for obstacle detection and avoidance. Techniques such as 3D modeling of obstacles [57, 58, 59] and estimating obstacle distances [58, 59, 60] are

commonly used in the literature.

(c) Object Tracking: Data captured from cameras are used as feedback for the UAV's control unit. The control unit employs different algorithms to maintain the UAV's balance. Common techniques include PID (Proportional-Integral-Derivative) control [61, 62, 63], optimal control [64], sliding mode control [65], fuzzy logic control [66], cascade control structure [67], and waypoint range or path following [68, 69].

4. Swarm UAV technologies emerged in the early 1990s, primarily for military applications [70]. Recently, Intel developed UAV swarms utilized in the 2018 Winter Olympics [27]. The literature contains various approaches to control, route planning, and task allocation for swarm UAVs [71, 72, 73]. Swarm UAVs offer advantages such as time savings, reduced working hours, reduced labor, and lower operational costs. While typical remote piloting with a single UAV would be inefficient for package delivery, swarm UAVs are more suitable. Autonomy is a key aspect of swarm UAVs, and the algorithm responsible for decision-making in place of human operators is critical. The autonomy level of a UAV is defined based on the number of missions it can perform without human intervention. Full autonomy refers to a UAV's capability to autonomously complete a mission from takeoff to landing without human interference. Fully autonomous UAVs require advanced algorithms to plan and optimize trajectories, collaborate with other UAVs, and make decisions based on environmental conditions and mission objectives. Some studies have proposed algorithms that use machine learning techniques for swarm UAV control. Other studies have focused on optimizing flight routes for swarm UAVs to reduce mission completion time or energy consumption [74, 75].

5. One key factor in the success of UAVs in 3D production is the ability to load new production materials without having to return to the central station frequently. One approach is establishing a ground station network where UAVs can land, recharge batteries, and load new materials [23]. Alternatively, researchers have proposed docking mechanisms for UAVs to exchange payloads. These docking mechanisms can load new production materials or exchange batteries. Swarms of UAVs can be employed, with some UAVs performing

tasks while others return to the central station for recharging and reloading. The transfer of materials between UAVs can be accomplished through various mechanisms such as docking, grappling, or handover [76, 77, 78].

Based on the above literature review, the key challenges in the proposed system include the followings.

- Designing lightweight and precise manipulator systems that can be mounted on UAVs to enable 3D printing or additive manufacturing tasks.
- Developing accurate and robust navigation systems to ensure precise positioning and orientation control of the UAVs during 3D printing operations.
- Implementing obstacle detection and avoidance systems to prevent collisions and ensure safe operation in complex environments.
- Enabling autonomous flight and decision-making capabilities for swarm UAVs to collaborate and coordinate tasks efficiently.

By addressing these challenges, the proposed system aims to enable the use of multi-rotor UAVs for large-scale 3D printing or additive manufacturing applications in various industries. In the proposed system within the scope of this study, the UAVs are planned to return to the base station after completing their assigned tasks to recharge the battery and receive new payloads, such as filaments. At the same time, another UAV continues with the assigned task, ensuring uninterrupted workflow.

1.2 Contributions and Novelties

Traditional 3D printers face limitations when printing on non-flat or irregular surfaces, as they are typically designed to operate on fixed platforms. Researchers and innovators have recognized this constraint and explored alternative approaches to overcome this challenge. In this thesis, it is proposed to integrate an extruder mounted on a gimbal system coupled with a camera-based path-tracking technique for 3D printing. This approach allows the printer to adapt to the surface geometry and maintain precise control throughout printing.

The printer can adjust its orientation dynamically by mounting the extruder on a gimbal system, which provides rotational freedom. This flexibility enables the printer to navigate complex surfaces, such as curved or uneven objects while maintaining accurate deposition of the printing material. The camera-based path-following technique further enhances the printer's capabilities by tracking and following a predefined path, ensuring precise alignment with the desired printing trajectory.

The main contributions of this thesis are twofold. A methodology for efficient Coverage Path Planning (CPP) using UAVs is proposed. This approach optimizes the mission time for multi-UAV coverage tasks while utilizing the fewest possible UAVs. By intelligently planning the coverage path, the thesis aims to minimize redundancy and maximize efficiency in completing the area coverage task. Additionally, the proposed approach optimizes the sweep direction, reducing the number of rotations required during the operation and improving overall performance.

Secondly, the thesis encompasses the design, construction, and application of a specialized test bench specifically tailored to evaluate the performance of gimbals attached to quadcopters. This dedicated test platform allows researchers and practitioners to assess and demonstrate the effectiveness of various flight algorithms in conjunction with the gimbal's performance. A test bench is a valuable tool for refining and optimizing the overall system by thoroughly evaluating the gimbal's capabilities under different flight conditions and scenarios.

Through these contributions, the thesis aims to push the boundaries of 3D printing technology by addressing the limitations of traditional printers and paving the way for enhanced capabilities in printing on non-flat or irregular surfaces. By leveraging the potential of gimbal-mounted extruders and UAVs for coverage path planning, this research offers promising advancements in additive manufacturing, opening new opportunities for fabricating complex objects with precision and efficiency.

1.3 Proposed Methods and Models

This thesis aimed to simulate a system of swarm UAVs to be used in additive manufacturing in hard-to-reach places and desired dimensions and evaluate the performance of a quadcopter installed with a gimbal and an extruder system. The swarm movement

can complete the production process quickly and without interruption. The flowchart of the proposed system is shown in Figure 1.1.

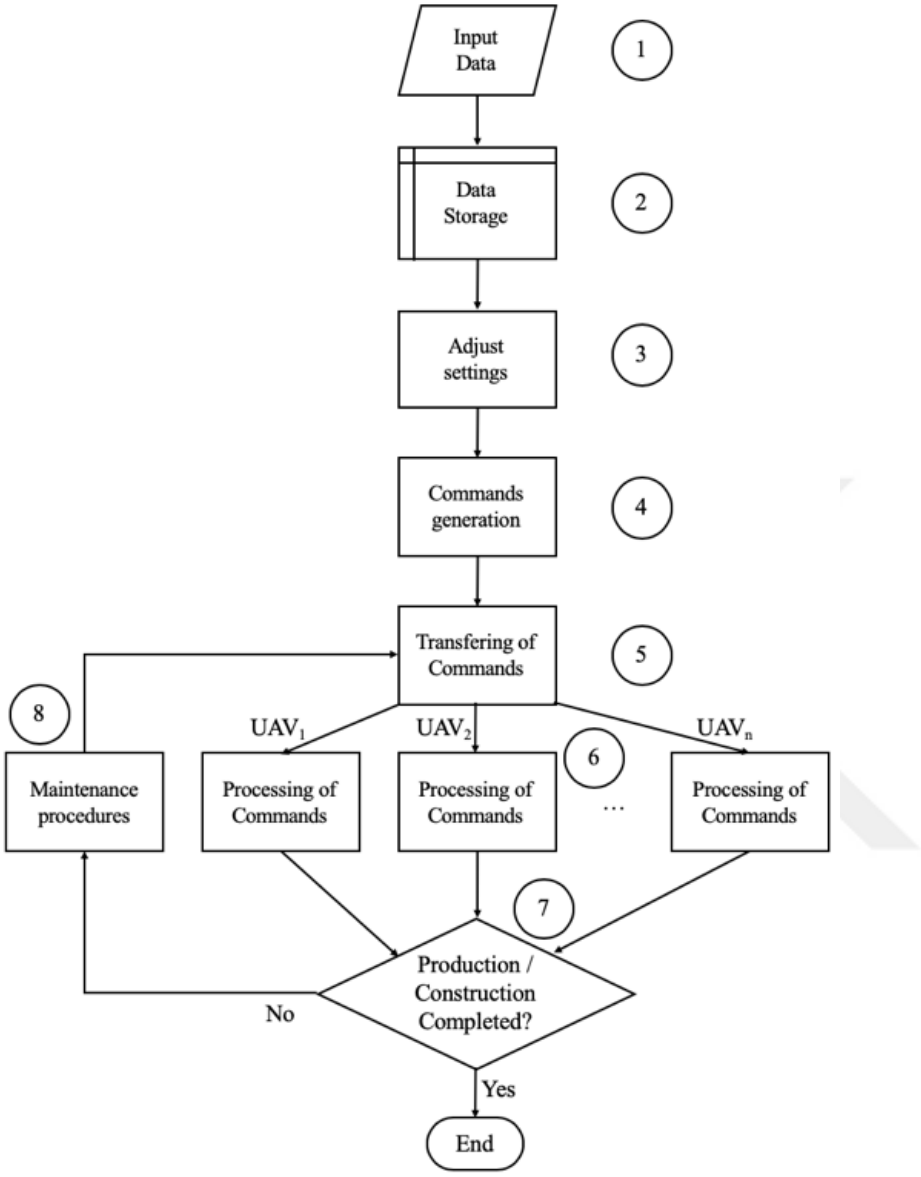


Figure 1.1: Flowchart of the proposed system

Chapter 2 of this study delves into the mathematical model of the gimbal, which is mounted on the quadcopter. The calculation assumes that the selected quadcopter possesses a distinctive $(+)$ shape model. Through this mathematical model, the study aims to accurately represent and understand the dynamics and behavior of the gimbal system, contributing to the overall comprehension of its functionality and perfor-

mance.

Moving on to Chapter 3, the focus shifts towards exploring and testing camera-based image processing methods. Within this chapter, three different quadcopters are selected as hypothetical models for the experiments. The objective is to evaluate the effectiveness and reliability of various image processing techniques when applied to these quadcopter models. The study aims to identify the most suitable image processing methods to enhance quadcopter systems' overall performance and capabilities by conducting extensive tests and analyses.

Chapter 4 is dedicated to calculating optimal paths for multi-UAV systems. The methodology developed in this chapter assumes a scenario with three UAVs operated by a single operator. The study also considers the setup time required for each UAV, which is assumed to be six minutes and the maximum flight time of ten minutes per UAV. By utilizing sophisticated path planning algorithms and considering these assumptions, the study aims to optimize the paths of multiple UAVs, enabling efficient coordination and collaboration between them.

Finally, Chapter 5 sheds light on the test platform employed in the study. The selected extruder dimension for the 3D printing process is set at one millimeter. This chapter provides a detailed explanation of the test platform's configuration, design, and components. It serves as a crucial foundation for conducting experiments and gathering data to validate the effectiveness and performance of the gimbal system, as well as other aspects related to multi-UAV coordination, image processing, and path planning.

Collectively, these chapters contribute to the comprehensive understanding and evaluation of various aspects related to the gimbal system, image processing methods, multi-UAV coordination, and the overall test platform. The study aims to provide valuable insights into the gimbal system's performance, capabilities, and potential applications and integration within the larger UAV ecosystem by combining theoretical models, experimental results, and assumptions.

1.4 The Outline of the Thesis

This thesis comprises six chapters, each addressing key aspects of the research. Chapter 1 serves as an introduction, focusing on the literature survey that underpins the thesis and providing a detailed account of the motivations behind the research. In Chapter 2, the design of the gimbal is presented, along with an explanation of the mathematical model that governs its operation. By integrating the gimbal mechanism into the quadcopter's design, precise control and manipulation of the end effector's orientation are possible.

Chapter 3, titled "Camera-Based Multi Quadcopter Simulation," delves into the simulation of a multi-UAV system for tracking a designated line under the UAVs. This line, generated as a path for the quadcopters to perform 3D printing, is followed using image processing techniques, ensuring the UAVs remain on the designated path.

Moving on to Chapter 4, "Path Planning for Area Coverage," the thesis explores the control strategy for coordinating multiple quadcopters during area coverage operations. The proposed method is thoroughly discussed, showcasing the results through simulations and real-world experiments.

Chapter 5 focuses on the test platform developed to assess the extruder's performance on the gimbal under various conditions. The experimental results of the 3D printing process are presented in detail in this chapter.

Finally, in Chapter 6, the thesis concludes by summarizing the essential findings and contributions of the research. The successful completion of this project holds great potential for overcoming many of the limitations faced by current 3D printers. Precisely, it enables rapidly fabricating products with higher complexity and larger sizes, particularly within hazardous or otherwise inaccessible environments. This breakthrough promises significant advancements in additive manufacturing, paving the way for transformative applications.

CHAPTER 2

DYNAMIC MODEL OF THE SYSTEM

2.1 Introduction

UAV has made significant strides across many applications in recent years. Quadcopters have become one of the most often used UAV platforms due to their versatility, stability, and ease of maneuvering. Outstanding capabilities for aerial surveillance, photography, mapping, and even delivery services are provided by quadcopters. Quadcopters frequently come with extra payloads like cameras, sensors, or other gadgets to expand their capability and enable precise control during complicated operations. The gimbal system is an essential component that adds to the versatility of quadcopters. A gimbal is a mechanical mechanism that stabilizes and controls the orientation of the payload, ensuring that it stays level and steady independent of the motion of the quadcopter or outside disturbances. A gimbal enables smooth and high-quality image or data acquisition by dissociating the payload's motion from the quadcopter's motion, which is crucial for applications requiring precise and stable observations. It is essential to have a thorough grasp of the dynamic behavior of quadcopters and their gimbal systems to create effective control algorithms and maximize the performance of these devices. This knowledge makes it possible to create sophisticated control schemes that account for outside disturbances, reduce vibrations, and enhance flight stability.

This chapter is dedicated to determining the system's parameters required for effectively controlling the endpoint on the gimbal and aligning it with the inertial coordinate system. The chapter is divided into three parts, each addressing a specific aspect of the system's modeling and parameter calculation.

The first part focuses on the kinematic and dynamic models of the quadcopter. This

involves formulating mathematical representations that describe the quadcopter's motion and behavior. The kinematic model encompasses the quadcopter's position, velocity, and orientation, while the dynamic model considers the forces, torques, and accelerations acting upon the quadcopter. By developing a comprehensive understanding of the quadcopter's kinematics and dynamics, it becomes possible to derive the necessary parameters for adequate control.

The second part delves into the kinematic and dynamic model of the gimbal. Like the quadcopter, the gimbal system's motion and behavior are represented through mathematical models. The kinematic model describes the gimbal's position, orientation, and motion relative to the quadcopter, while the dynamic model considers the forces, torques, and accelerations acting on the gimbal. By analyzing and modeling the gimbal's kinematics and dynamics, crucial parameters can be determined to ensure accurate control of the gimbal system.

The final part revolves around calculating the coordinates of the endpoint on the gimbal concerning the inertial coordinate system. This step involves transforming the measured or estimated positions of the gimbal endpoint, which are typically relative to the quadcopter or gimbal reference frame, into the absolute coordinates of the inertial reference frame. By accurately determining the endpoint's coordinates in the inertial coordinate system, precise control and alignment of the gimbal system can be achieved.

By addressing these three key components—the quadcopter's kinematic and dynamic model, the gimbal's kinematic and dynamic model, and the calculation of the endpoint's coordinates—, the chapter aims to provide a comprehensive understanding of the system's parameters necessary for effective control and coordination between the quadcopter and gimbal systems. Section 2.2 provides a detailed explanation of the kinematic analysis of the quadcopter. This section delves into the intricate mechanics and movements of the quadcopter. Moving on to Section 2.3, this portion of the thesis focuses on the gimbal's design and the mathematical model. Section 2.4 depicts the mathematical simulation of the gimbal mounted on the quadcopter, providing an in-depth understanding of how it operates and contributes to its overall stability and functionality. Lastly, in Section 2.5, the chapter concludes with a comprehensive discussion. It combines the essential findings and insights presented throughout the chapter, summarizing the main points and highlighting their significance.

2.2 Quadcopter Model

The first part of this section focuses on the kinematic analysis of the quadcopter, aiming to establish an understanding of the relationships between various parameters involved in its motion. This analysis determines how the quadcopter moves and behaves in different scenarios.

The kinematic analysis examines the quadcopter's position, velocity, and orientation. It investigates how these quantities change over time and how they relate to one another. Critical parameters such as displacement, velocity vectors, and angular rates can be derived by studying the quadcopter's kinematics. These parameters provide valuable insights into the quadcopter's spatial movement, speed, and rotational characteristics. Furthermore, the kinematic analysis explores the relationship between the quadcopter's position, velocity, and orientation. By understanding how these parameters interact, it becomes possible to determine the quadcopter's trajectory and predict its future motion based on its current state. This analysis also assists in studying the quadcopter's stability, maneuverability, and control.

Moving on to the next part, the dynamic model of the quadcopter is derived. The dynamic model considers the forces, torques, and accelerations acting on the quadcopter. It provides a mathematical representation of how these physical quantities affect the quadcopter's motion.The derivation of the dynamic model involves analyzing the forces and torques generated by the quadcopter's propellers and the forces of gravity, air resistance, and other external factors. By considering these forces and torques, the dynamic model captures the quadcopter's response to different inputs and disturbances, enabling a deeper understanding of its behavior.

The dynamic model also facilitates the development of control strategies for the quadcopter. By accurately modeling the quadcopter's dynamics, it becomes possible to design control algorithms that can effectively stabilize and maneuver the quadcopter, ensuring precise and reliable performance in various flight conditions.

This section comprehensively explains the quadcopter's motion and behavior by incorporating both the kinematic and dynamic analyses. It establishes the relationship between different parameters and provides a solid foundation for further analysis and control design, ultimately contributing to quadcopter systems' overall understanding and advancement.

2.2.1 Kinematic of a Quadcopter

The quadcopter's position is represented by the inertial coordinate system, which is fixed to the Earth, and the body coordinate system, which is mobile and can rotate. Euler's Angles model two coordinate systems' rotation.

The ZYX rotation sequence can break down the body frame's rotation concerning the inertial frame into three simple sequential rotations. The body rotates around the Z axis (yaw angle (ψ)) and then about the Y axis (pitch angle (θ)) in the ZYX rotation sequence. Finally, it is rotated through a rolling angle (ϕ) about the X-axis.

The quadcopter rotates from intermediate frame i to 1 around $Z_i = Z_1$.

$$C_i^1(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

The quadcopter rotates from intermediate frame 1 to 2 around $Y_1 = Y_2$.

$$C_1^2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{2.2}$$

The quadcopter rotates from intermediate frame 2 to b around $X_2 = X_b$.

$$C_2^b(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \tag{2.3}$$

Rotating a vector from the inertial frame to the body frame requires a transformation matrix that multiplies the three transformations.

$$C_i^b = C_2^b \times C_1^2 \times C_i^1 \tag{2.4}$$

$$C_i^b = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix} \tag{2.5}$$

Rotate a vector in the body frame back to the inertial frame using the transformation matrix C_b^i equation. Therefore,

$$C_b^i = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{2.6}$$

To establish the relationship between Euler angle changes and body frame angular velocity components, a transformation is required to express the angular velocity as components. This can be achieved using the following transformation:

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.7}$$

In this equation, ω represents the angular velocity vector, and (p, q, r) are the body frame angular velocity components.

2.2.2 Dynamics Model of a Quadcopter

Quadcopters are highly maneuverable aerial vehicles capable of both rotation and translation. Roll, pitch, and yaw can be precisely controlled in their motion, making them fully actuated, while movements along the x, y, and altitude axes are underactuated, requiring careful consideration. This dynamics model is based on the work in [76]. To accurately describe the quadcopter's motion, accounting for the external forces and torques acting upon it and applying Newton's law of motion is necessary. Incorporating the quadcopter's linear and rotational accelerations into the forces and moments, the body forces and torques can be expressed as follows.

Rotational Equations of Motion:

The general equation governing the rotational motion of the quadcopter's body frame, derived using the Newton-Euler method, is given by:

$$J\dot{\omega} + \omega \times J\omega + M_G + M_a = M_B \tag{2.8}$$

Here, J represents the inertia matrix of the quadcopter, ω denotes the angular body rates, M_G corresponds to the gyroscopic moments generated by the rotor's inertia, M_a represents the aerodynamic moments acting on the quadcopter's body, and M_B signifies the moment acting on the body frame of the quadcopter. The terms $J\dot{\omega}$ and $\omega \times J\omega$ account for the rate of change of angular momentum in the body frame. The inertia matrix of the quadcopter is a diagonal matrix with zero off-diagonal elements,

which are the products of inertia. It can be expressed as:

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.9}$$

In this matrix, I_{xx}, I_{yy}, and I_{zz} represent the area moments of inertia about the principal axes in the body frame of the quadcopter.

By comprehensively considering these dynamics and utilizing the derived equations of motion, valuable insights into the behavior and control of quadcopters can be gained. Gyroscopic moments of motors link the spin axis with the inertial z-axis [77]. Formulating the gyroscopic moment yields:

$$M_G = \overline{\omega}_b^i \times \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} \tag{2.10}$$

M_G is the gyroscopic moment, J_r is the motor moment, and Ω_r is the motor's total rotational speed in rpm. Equation of Ω_r is

$$\Omega_r = \overline{\omega}_1 - \overline{\omega}_2 + \overline{\omega}_3 - \overline{\omega}_4 \tag{2.11}$$

Moments Acting on the Quadrotor (M_B)

The moments exerted on the quadrotor play a crucial role in its dynamics and control. Understanding these moments is essential for accurate modeling and effective maneuvering of the quadcopter.

Thrust of the Propeller: The propulsion force exerted by its propellers is the primary force responsible for generating lift in the quadcopter. As the propellers spin through the air, they experience aerodynamic forces comprising lift and drag components. The lift force acts perpendicular to the propellers' direction and contributes to the quadcopter's upward motion. According to the findings in [78, 79], the lift force acting on the propellers can be expressed as:

$$F_n = \frac{1}{2} C_T \rho A_r r^2 \overline{\omega}_n^2 \tag{2.12}$$

In this equation, C_T represents the thrust coefficient of the propeller, ρ denotes the air density, A_r corresponds to the propeller's rotating area, r is the distance from the center to the tip of the propeller, and $\overline{\omega}_n$ denotes the angular velocity of the n^{th} motor

in terms of rpm (revolutions per minute). Combining all the constant parameters into a single variable simplifies the equation. Assuming the air density remains constant, the following equation is defined:

$$K_f = C_T \rho A_r r^2 \tag{2.13}$$

Subsequently, the equation for the lift force can be expressed as:

$$F_n = K_f \overline{\omega}_n^2 \tag{2.14}$$

Torque of the Propeller: In addition to generating lift, the rotation of the propellers also creates moments or torques that act on the quadcopter's body frame along the Z axis. As mentioned earlier, the aerodynamic forces resulting from the propeller rotation consist of lift and drag components. While the lift force operates in the direction perpendicular to the propellers, the drag force acts opposite to the propeller's motion. The drag component of the aerodynamic force induces a torque on the quadcopter's body. According to [78], the moment produced by the propellers due to drag force can be expressed as:

$$M_n = (-1)^n \frac{1}{2} \rho A C_D r^2 \overline{\omega}_n^2 \tag{2.15}$$

Here, M_n represents the moment generated by the n^{th} motor, ρ is the air density, A represents the frontal area of the propeller (obtained by multiplying the chord length by the maximum thickness), C_D denotes the drag coefficient, r is the distance from the center to the tip of the propeller, and $\overline{\omega}_n$ represents the angular velocity of the n^{th} motor in rpm. Additionally, the equation includes a factor of $(-1)^n$, which accounts for Newton's third law of motion. This component ensures that the propeller experiences a clockwise moment when rotating in a clockwise direction and vice versa. For instance, if motor 1 rotates counterclockwise, the body will experience a clockwise moment if the air density remains constant. To simplify the equation, all the constant parameters into a single variable can be combined:

$$K_m = \frac{1}{2} \rho A C_D r^2 \tag{2.16}$$

Finally, the equation for the propeller moment due to drag force becomes:

$$M_n = (-1)^n K_m \overline{\omega}_n^2 \tag{2.17}$$

Total Moments in the x, y, and z Directions: Figure 2.1 visually represents the moments and forces acting on the quadcopter. In the figure, each quadcopter rotor

generates a thrust force, denoted as F_i, and a moment, represented by M_i. These forces and moments are exerted in the direction opposite to the rotor's rotation.

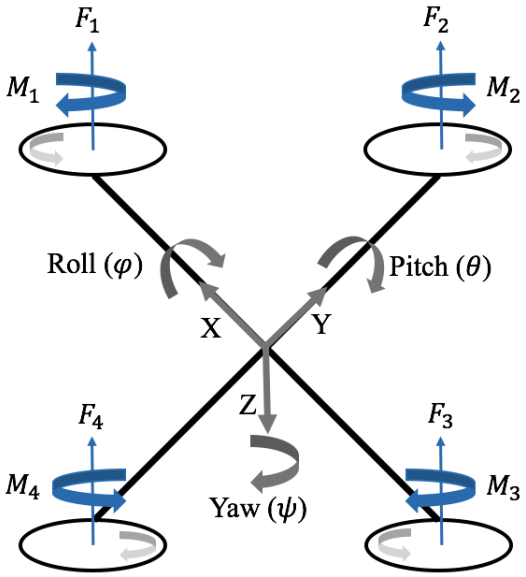


Figure 2.1: Quadcopter Forces and Moments

The total moments acting on the quadrotor can be expressed in the x, y, and z directions. These moments are crucial for understanding the rotational dynamics of the quadcopter.

$$M_x = F_2 d - F_4 d = dK_f \overline{\omega}_2^2 - dK_f \overline{\omega}_4^2 \tag{2.18}$$

Equation 2.18 represents the moment in the x direction, where d is the distance from the center of the hub to the motor.

$$M_y = -F_1 d + F_3 d = -dK_f \overline{\omega}_1^2 + dK_f \overline{\omega}_3^2 \tag{2.19}$$

Equation 2.19 represents the moment in the y direction.

$$\begin{aligned} M_z &= -M_1 + M_2 - M_3 + M_4 \\ &= -K_m \overline{\omega}_1^2 + K_m \overline{\omega}_2^2 - K_m \overline{\omega}_3^2 + K_m \overline{\omega}_4^2 \end{aligned} \tag{2.20}$$

Equation 2.20 represents the moment in the z direction, which is the sum of the moments produced by all four motors. Here, M_i represents the moment produced by the i^{th} motor.

By combining equations (2.18), (2.19), and (2.20) in vector form, the total moments acting on the quadcopter's body frame can be expressed as follows:

$$M_B = \begin{bmatrix} dK_f\overline{\omega}_2^2 - dK_f\overline{\omega}_4^2 \\ -dK_f\overline{\omega}_1^2 + dK_f\overline{\omega}_3^2 \\ -K_m\overline{\omega}_1^2 + K_m\overline{\omega}_2^2 - K_m\overline{\omega}_3^2 + K_m\overline{\omega}_4^2 \end{bmatrix} \tag{2.21}$$

In Equation (2.21), d represents the distance between the quadcopter's rotor and the center of the quadrotor. K_f and K_m are the constants that govern the aerodynamic force and moment, respectively. $\overline{\omega}_i$ represents the angular velocity of rotor i. Moving on to the translational equations of motion, they use Newton's second law to describe the quadrotor's motion in the Earth's inertial frame. The equation is given by:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + C_i^b F_B \tag{2.22}$$

In Equation (2.22), r represents the distance of the quadcopter from the inertial frame. The mass of the quadcopter is denoted by m, and g represents the acceleration due to gravity ($g = 9.81\text{m/s}^2$). The term F_B represents the non-gravitational forces acting on the quadrotor in the body frame, and C_i^b denotes the rotation matrix that transforms forces from the body frame to the inertial frame.

Equation (2.22) encapsulates the quadcopter's translational motion, considering the gravitational force and the non-gravitational forces acting on the quadrotor. By considering both the rotational and translational equations of motion, the dynamics of the quadcopter in both its rotational and translational aspects can be analyzed. This understanding is crucial for developing control strategies and optimizing the quadcopter's flight performance.

$$\begin{aligned} F_{Bz} &= F_1 + F_2 + F_3 + F_4 \\ &= K_f\overline{\omega}_1^2 + K_f\overline{\omega}_2^2 + K_f\overline{\omega}_3^2 + K_f\overline{\omega}_4^2 \\ &= \begin{bmatrix} 0 \\ 0 \\ K_f\overline{\omega}_1^2 + K_f\overline{\omega}_2^2 + K_f\overline{\omega}_3^2 + K_f\overline{\omega}_4^2 \end{bmatrix} \end{aligned} \tag{2.23}$$

Aerodynamic Effects: The aerodynamic impacts on the quadrotor body were disregarded in the formulation of the prior dynamic. However, aerodynamic factors should

be considered to create a precise and realistic model for simulation usage. Drag forces and drag moments are the two main categories of aerodynamic effects.

Drag Forces: Drag forces act on the quadrotor body, opposing its motion due to the friction between the moving body and the air. As the quadrotor's travel speed increases, the drag forces also increase. These drag forces, denoted as F_a, can be roughly calculated using the equation:

$$F_a = K_t \dot{r} \tag{2.24}$$

Here, K_t represents the aerodynamic translation coefficient matrix, and $\dot{r}$ is the time derivative of the position vector r. This equation shows an additional force acting on the quadrotor body. Therefore, Equation (2.22), which represents the translational equation of motion, should be modified to:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + C_i^b F_B - F_a \tag{2.25}$$

Drag Moments: There is a drag moment M_a operating on the quadrotor body that is equivalent to the drag force caused by air friction and can be approximated by,

$$M_a = K_r \dot{\eta} \tag{2.26}$$

Here, $\dot{\eta}$ represents the Euler rates, and K_r is a constant matrix known as the aerodynamic rotation coefficient matrix. Consequently, Equation (2.8), which represents the rotational equation of motion, can be rewritten as:

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \quad 0 \quad J_r\Omega_r]^T = M_B - M_a \tag{2.27}$$

This modified equation accounts for the drag moments caused by air friction, providing a more accurate representation of the quadrotor's rotational dynamics. Considering both the drag forces and drag moments, the aerodynamic effects are appropriately integrated into the quadcopter's dynamic model, allowing for a more realistic simulation and analysis.

Control Input U of the Quadcopter: The complete dynamic model of the quadcopter can be described by four control inputs and 12 state vectors. The quadcopter consists of four inputs as follows:

$$U = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix} \tag{2.28}$$

where U_1, U_2, U_3 and U_4 are the desired altitude, roll angle, pitch angle, and heading. They are defined as,

$$\begin{aligned}\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} &= \begin{bmatrix} K_f(\bar{\omega}_1^2 + \bar{\omega}_2^2 + \bar{\omega}_3^2 + \bar{\omega}_4^2) \\ K_f(-\bar{\omega}_2^2 + \bar{\omega}_4^2) \\ K_f(\bar{\omega}_1^2 - \bar{\omega}_3^2) \\ K_m(\bar{\omega}_1^2 - \bar{\omega}_2^2 + \bar{\omega}_3^2 - \bar{\omega}_4^2) \end{bmatrix} \\ &= \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_m & -K_m & K_m & -K_m \end{bmatrix} \begin{bmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{bmatrix}\end{aligned} \tag{2.29}$$

The matrix in Equation 2.29 can be inverted to obtain the inverse relationship between the control inputs and the rotor velocities required if the rotor velocities must be derived from the control inputs.

$$\begin{bmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & 0 & \frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & -\frac{1}{4K_m} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & -\frac{1}{4K_m} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{2.30}$$

The rotors' velocities can be computed from the control inputs as follows by taking the square root of that:

$$\begin{cases} \bar{\omega}_1 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 + \frac{1}{4K_m}U_4} \\ \bar{\omega}_2 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 - \frac{1}{4K_m}U_4} \\ \bar{\omega}_3 = \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 + \frac{1}{4K_m}U_4} \\ \bar{\omega}_4 = \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 - \frac{1}{4K_m}U_4} \end{cases} \tag{2.31}$$

The mathematical model for the rotational equation of motion of the quadcopter can be derived by substituting Equation (2.29) into Equation (2.21). This substitution yields:

$$M_B = \begin{bmatrix} dU_2 \\ dU_3 \\ U_4 \end{bmatrix} \tag{2.32}$$

By further substituting Equation (2.32) into Equation (2.8), the rotational equation of motion can be obtained as follows:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r \bar{\omega}_r \end{bmatrix} = \begin{bmatrix} dU_2 \\ dU_3 \\ U_4 \end{bmatrix} \tag{2.33}$$

By solving Equation (2.33), the following equations can be obtained:

$$\begin{cases} \ddot{\phi} = \frac{1}{I_{xx}}(dU_2 - J_r \dot{\theta} \Omega_r + I_{yy} \dot{\psi} \dot{\theta} - I_{zz} \dot{\theta} \dot{\psi}) \\ \ddot{\theta} = \frac{1}{I_{yy}}(U_3 - J_r \dot{\phi} \Omega_r + I_{zz} \dot{\phi} \dot{\psi} - I_{xx} \dot{\psi} \dot{\phi}) \\ \ddot{\psi} = \frac{1}{I_{zz}}(dU_4 + I_{xx} \dot{\theta} \dot{\phi} - I_{yy} \dot{\phi} \dot{\theta}) \end{cases} \tag{2.34}$$

These equations describe the rotational dynamics of the quadcopter and are dependent on the control input vector U. The rotational subsystem is fully actuated and exclusively depends on the rotational state variables x_1 to x_6, which correspond to $\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}$, respectively. By controlling these rotational variables, the dynamics of the quadcopter's rotation can be effectively managed.

The mathematical model of the translational equation of the quadcopter is derived by substituting equation (2.29) into (2.23) to obtain the expression for the total body forces acting on the quadcopter:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \tag{2.35}$$

Using this expression, the translational equation of motion can be written as: Then,

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} c\psi c\theta & c\psi s\phi s\theta & s\phi s\psi + c\phi c\psi s\theta \\ c\theta s\psi & c\theta c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\theta \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \tag{2.36}$$

Simplifying further, the equations of motion can be expressed as follows:

$$\begin{cases} \ddot{x} = \frac{-U_1}{m}(\sin\phi \sin\psi + \cos\phi \cos\psi \sin\theta) \\ \ddot{y} = \frac{-U_1}{m}(\cos\phi \sin\psi \sin\theta - \cos\psi \sin\phi) \\ \ddot{z} = g - \frac{U_1}{m}(\cos\phi \cos\theta) \end{cases} \tag{2.37}$$

These equations describe the acceleration of the quadcopter along the x, y, and z axes as functions of the control input U_1. The Euler angles ϕ, θ, ψ, and the gravitational constant g.

State Space Model of the Quadcopter

To simplify the control problem, a state space model for the quadcopter can be formulated. The state vector X of the quadcopter is defined as follows:

$$X = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}]^T \tag{2.38}$$

where x_1 to x_6 correspond to $\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}$, and x_7 to x_{12} correspond to $z, \dot{z}, x, \dot{x}, y, \dot{y}$ respectively. The state vector represents the quadcopter's position, angular, and linear velocity. The state vector of the quadrotor represents its current state and consists of the following variables:

$$X = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad z \quad \dot{z} \quad x \quad \dot{x} \quad y \quad \dot{y}]^T \tag{2.39}$$

Each state vector element corresponds to a specific parameter of the quadrotor's motion. The state vector includes the roll angle ϕ and its derivative $\dot{\phi}$, pitch angle θ and its derivative $\dot{\theta}$, yaw angle ψ and its derivative $\dot{\psi}$, as well as the vertical position z and its derivative $\dot{z}$, horizontal position x and its derivative $\dot{x}$, and horizontal position y and its derivative $\dot{y}$.

The state vector of the quadrotor provides a comprehensive representation of its kinematic and dynamic state. It encapsulates information about the quadrotor's orientation, angular velocities, and linear velocities in three-dimensional space.

The complete mathematical model of the quadrotor can be obtained by combining the rotational and translational equations, resulting in the following set of differential equations given in 2.40.

These equations describe the time derivatives of the state variables, where $\dot{x}_i$ represents the derivative of the i^{th} state variable. The right-hand side of each equation involves various combinations of state variables, control inputs (U_1, U_2, U_3, and U_4), the quadrotor's moments of inertia (I_{xx}, I_{yy}, I_{zz}) and other parameters.

The complete mathematical model of the quadrotor's dynamics can be represented using these equations, allowing for analysis, control design, and simulation of its behavior. The model considers the complex interplay between the rotational and

translational motions of the quadrotor, capturing its intricate dynamics compactly and expressively.

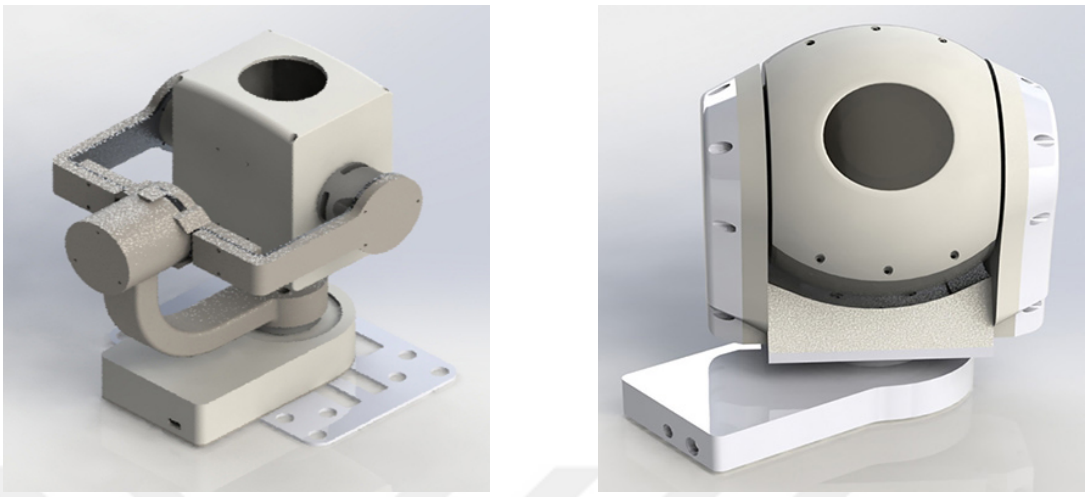
$$\begin{cases} \dot{x}_1 = \dot{\phi} = x_2 \\ \dot{x}_2 = \ddot{\phi} = x_4 x_6 (\frac{I_{yy} - I_{zz}}{I_{xx}}) - x_4 \Omega_r (\frac{J_r}{I_{xx}}) + (\frac{d}{I_{xx}}) U_2 \\ \dot{x}_3 = \dot{\theta} = x_4 \\ \dot{x}_4 = \ddot{\theta} = x_2 x_6 (\frac{I_{zz} - I_{xx}}{I_{yy}}) + x_2 \Omega_r (\frac{J_r}{I_{yy}}) + (\frac{d}{I_{yy}}) U_3 \\ \dot{x}_5 = \dot{\psi} = x_6 \\ \dot{x}_6 = \ddot{\psi} = x_2 x_4 (\frac{I_{xx} - I_{yy}}{I_{zz}}) + (\frac{d}{I_{zz}}) U_4 \\ \dot{x}_7 = \dot{z} = x_8 \\ \dot{x}_8 = \ddot{z} = g - \frac{U_1}{m} (\cos x_1 \cos x_3) \\ \dot{x}_9 = \dot{x} = x_{10} \\ \dot{x}_{10} = \ddot{x} = \frac{-U_1}{m} (\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \\ \dot{x}_{11} = \dot{y} = x_{12} \\ \dot{x}_{12} = \ddot{y} = \frac{U_1}{m} (\sin x_1 \cos x_5 - \cos x_1 \sin x_3 \sin x_5) \end{cases} \tag{2.40}$$

2.3 Design and Manufacturing of the Gimbal

Generally, gimbals predominantly come in two main designs: two-axis and three-axis gimbals. The two-axis design stabilizes the pitch and roll axes, compensating for angular movements and vibrations induced by the UAV's flight dynamics. The three-axis design enables precise control and stabilization of the camera or payload, regardless of the UAV's orientation and motion. This gimbal type stabilizes all three axes: pitch, roll, and yaw. Figure 2.2 illustrates these two gimbals commonly used in various applications. The first type, depicted in Figure 2.2a, is specifically designed for multi-rotor UAVs. On the other hand, the second type, illustrated in Figure 2.2b, is commonly utilized for fixing cameras in fixed-wing UAVs.

The proposed approach aims to incorporate the Fused Filament Fabrication (FFF) method by positioning the extruder at the end of the gimbal system. By integrating the extruder at the end of the gimbal system, the proposed approach offers the advantage of leveraging the existing stabilization capabilities of gimbals. This allows

for enhanced control and compensation of the extruder's movements during the FFF process, improving print quality and accuracy.



(a) Gimbals used in multi-rotor UAVs

(b) Gimbals used in fixed-wing UAVs

Figure 2.2: Different types of the gimbals

The proposed approach ensures compatibility and optimal performance by selecting the appropriate gimbal type based on the specific application requirements, whether it is for multi-rotor UAVs or fixed-wing UAVs. The utilization of gimbals enables efficient and reliable FFF printing in aerial platforms, offering the potential for a wide range of applications such as aerial prototyping, remote sensing, and rapid response manufacturing.

2.3.1 System Design

The gimbal is designed using Computer-Aided Design (CAD) software, and a preliminary design is illustrated in Figure 2.3. The 3-axis gimbals consist of four parts connected by three rotating joints, each controlled by a dedicated motor. Once the design is finalized, the gimbal will be manufactured using 3D printers. The final design must satisfy the following requirements.

- The total weight of the gimbal, including the control card, motors, camera, and material extruder, must be less than 2 kg. This weight limitation is crucial

because the payload can be carried with quadcopters.

- The dimensions of the gimbal should be smaller than the landing gear of the chosen UAV. This criterion ensures the gimbal does not collide with the ground during take-off and landing. Figure 2.4 demonstrates the importance of adhering to these size constraints.

- The gimbal's coverage area on the yaw axis should span $360°$, and a slip ring will be employed to achieve this. Incorporating a slip ring makes it unnecessary to rotate the UAV when performing 3D printing in different directions.

- The gimbal should have a minimum coverage of $\pm 15°$ in pitch. Since the extruder's direction is downward, the maximum angle required for 3D printing will be $30°$. Software connected to the gimbal control card will enforce this restriction.

- The gimbal's movement range on the roll axis will be at least $\pm 15°$. The attached camera used for trajectory tracking must remain flat relative to the gimbal.

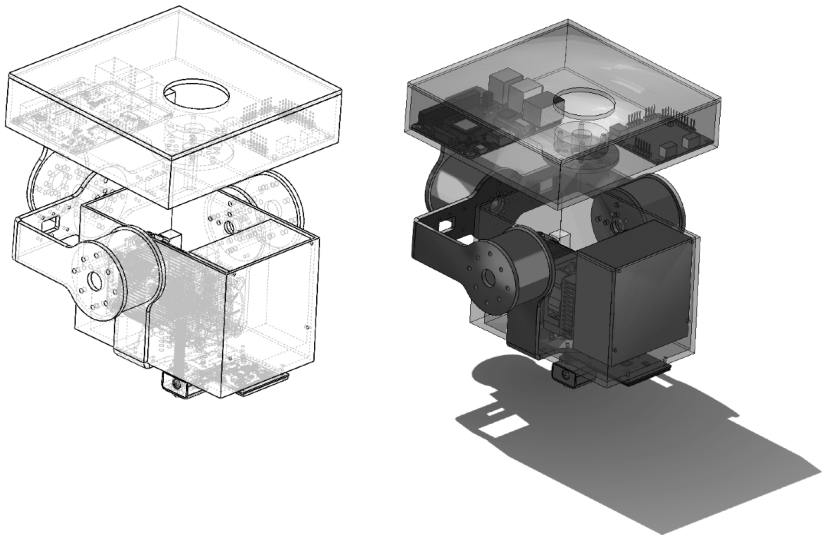


Figure 2.3: CAD model of the gimbal

This constraint will also be enforced through software connected to the gimbal control card. By fulfilling these requirements, the designed gimbal system will provide the

necessary stability, control, and flexibility to perform FFF printing tasks accurately and precisely on aerial platforms.

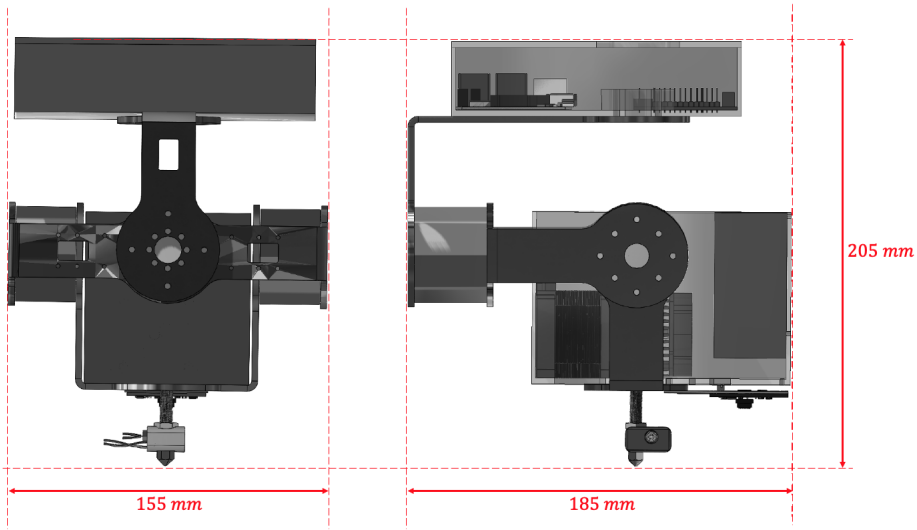


Figure 2.4: Dimension of the gimbal

2.3.2 Gimbal Model

The gimbal is a mechanical device that comprises interconnected links and revolute joints. Its primary function is to support and stabilize an attached component, such as a camera or laser, at the end effector. Gimbal is connected to the platform and is equipped with a damping joint to mitigate or reduce the effects of various noise sources, such as those from the UAV, wind, and gimbal motors. The end effector of the gimbal contains the camera, and the extruder is positioned in the direction of θ_1. An IMU comprising gyroscopes and accelerometers is near the gimbal-camera-extruder connection point. This IMU facilitates the measurement of altitude, roll (θ_2), pitch (θ_3), and yaw axis changes (θ_1) relative to the ground. Based on these measurements, the camera and extruder are always oriented downward. The design of the gimbal mechanism involves three essential components: kinematics, kinetics, and control.

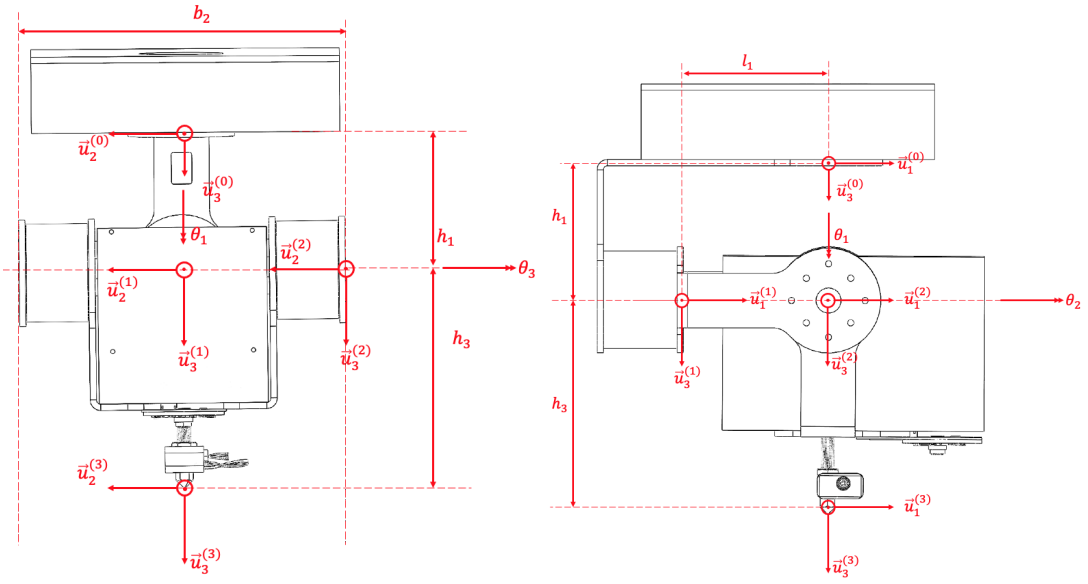
The kinematic model of the gimbal describes how it moves and how the joints and links influence its motion. It analyzes the relationships between the various components, such as the angles and positions of the joints, to determine the overall motion

of the gimbal system. This model allows for predicting and calculating the gimbal's orientation and position based on the inputs received.

The dynamic model of the gimbal explains how forces and torques acting on the system affect its movement. It considers factors such as the weight of the attached component, external forces, and the inertial properties of the gimbal itself. By understanding the system's dynamic behavior, engineers can optimize its design for stability and responsiveness.

To effectively control the gimbal and stabilize the end effector, the desired motion of the gimbal must be translated into the appropriate torque applied to its motors. This requires a control algorithm that considers both the gimbal and the target's current and desired positions and orientations. By calculating the necessary torque inputs to the motors, the control algorithm ensures that the gimbal accurately tracks the target and maintains stability throughout the motion.

The gimbal system can achieve smooth and accurate end effector motion by considering and integrating these three components: kinematics, kinetics, and control. This enables precise positioning and stabilization of the attached component, resulting in high-quality output and improved performance in 3D printing.



(a) Gimbal front side

(b) Gimbal right side

Figure 2.5: Coordinate system of the gimbal

2.3.2.1 Kinematics of the Gimbal Mechanism

The gimbal system has four links and three joints that enable its movement (Figure 2.5). Link (0) is attached to the quadcopter's body frame through the gimbal hole and as the base. Joint (1) connects to link (0) and rotates around the quadcopter's z-axis. Joint (2) is connected to link (1) and revolves around the x-axis. Finally, joint (3) is connected to link (2) and rotates around the y-axis. The camera and 3D printer extruder are attached to the link (4) and fixed to the joint (3).

Representing the transformations in the gimbal system involves accounting for the movement of the body frame relative to the inertial fixed frame. Since the body frame can experience various movements, it is essential to consider these changes and their impact on its relationship with the inertial frame. To accurately represent these movements, three types of transformations are typically used:

1. **Rotation Transformations:** When the body frame rotates around an axis relative to the inertial frame, a pure rotation occurs. There are three types of pure rotations: rotations about the body frame's x, y, and z axes. These rotations can be described using rotation matrices or Euler angles.

$$R_x(\theta_2) = R_2^1(\theta_2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_2 & -\sin\theta_2 \\ 0 & \sin\theta_2 & \cos\theta_2 \end{bmatrix}, \tag{2.41}$$

$$R_y(\theta_3) = R_3^2(\theta_3) = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 \\ 0 & 1 & 0 \\ -\sin\theta_3 & 0 & \cos\theta_3 \end{bmatrix}, \tag{2.42}$$

$$R_z(\theta_1) = R_1^0(\theta_1) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.43}$$

2. **Translation Transformations:** Pure translation involves the movement of the body frame in the direction of the axes of the inertial frame. This type of transformation describes the displacement of the body frame without any rotation.

It can be represented using translational vectors.

$$d_1^0 = \begin{bmatrix} -l_1 \cos\theta_1 \\ -l_1 \sin\theta_1 \\ h_1 \end{bmatrix} \tag{2.44}$$

$$d_2^1 = \begin{bmatrix} l_2 \\ -\frac{1}{2} b_2 \cos\theta_2 \\ -\frac{1}{2} b_2 \sin\theta_2 \end{bmatrix} \tag{2.45}$$

$$d_3^2 = \begin{bmatrix} h_3 \sin\theta_3 \\ \frac{1}{2} b_2 \\ h_3 \cos\theta_3 \end{bmatrix} \tag{2.46}$$

3. **Rotation and Translation Combinations:** In some cases, the movement of the body frame can be a combination of rotations and translations. A sequence of rotation matrices is multiplied to represent such complex movements to produce the overall transformation matrix. This matrix combines the rotational and translational components, creating a homogeneous transformation matrix.

$$T_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & -l_1 \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & -l_1 \sin\theta_1 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.47}$$

$$T_2^1 = \begin{bmatrix} R_2^1 & d_2^1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & \cos\theta_2 & -\sin\theta_2 & -\frac{1}{2} b_2 \cos\theta_2 \\ 0 & \sin\theta_2 & \cos\theta_2 & -\frac{1}{2} b_2 \sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.48}$$

$$T_3^2 = \begin{bmatrix} R_3^2 & d_3^2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 & h_3 \sin\theta_3 \\ 0 & 1 & 0 & \frac{1}{2} b_2 \\ -\sin\theta_3 & 0 & \cos\theta_3 & h_3 \cos\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.49}$$

By utilizing these representation techniques, the gimbal system can accurately describe and track the movements of the body frame relative to the inertial frame. This allows for precise control and coordination of the gimbal's motion, ensuring effective stabilization and positioning of the attached camera and 3D printer extruder.

Forward Kinematics of the Gimbal Mechanism: The gimbal mechanism's forward kinematics involves determining the end-effector's position and orientation, such as a camera or a 3D printer extruder, based on the joint angles and the transformation matrix between the inertial coordinate system and the quadcopter body frame. In this context, a known transformation matrix exists between the inertial coordinate system and the quadcopter body frame. The equation can represent this transformation:

$$T_3^0 = T_1^0 T_2^1 T_3^2 = \begin{bmatrix} R_3^0 & d_3^0 \\ 0 & 1 \end{bmatrix} \tag{2.50}$$

Here, T_3^0 denotes the transformation matrix from the quadcopter body frame (3) to the inertial reference frame (0). It can be obtained by multiplying the transformation matrices T_1^0, T_2^1, and T_3^2 corresponding to the individual segments of the gimbal mechanism.

The rotational matrix R_3^0 within the transformation matrix represents the rotation of the quadcopter body frame relative to the inertial reference frame. It defines the orientation of the quadcopter and can be expressed as:

$$R_3^0 = \begin{bmatrix} c\theta_1 c\theta_3 - s\theta_1 s\theta_2 s\theta_3 & -c\theta_2 s\theta_1 & c\theta_1 s\theta_3 + c\theta_3 s\theta_2 s\theta_1 \\ c\theta_3 s\theta_1 + c\theta_1 s\theta_2 s\theta_3 & c\theta_2 c\theta_1 & s\theta_3 s\theta_1 - c\theta_3 c\theta_1 s\theta_2 \\ -c\theta_2 s\theta_3 & s\theta_2 & c\theta_3 c\theta_2 \end{bmatrix} \tag{2.51}$$

In this matrix, $c\theta$ represents the $cosine$ of the angle θ and $s\theta$ represents the $sine$ of the angle θ. The elements of the matrix R_3^0 describe the relationships between the different coordinate axes of the quadcopter body frame and the inertial frame.

Furthermore, d_3^0 represents the position of the quadcopter relative to the inertial frame. It is a vector that defines the translation of the quadcopter's origin concerning the inertial coordinate system. The vector can be expressed as:

$$d_3^0 = \begin{bmatrix} l_2 c\theta_1 - l_1 c\theta_1 + h_3 c\theta_1 s\theta_3 + h_3 s\theta_1 s\theta_2 c\theta_3 \\ l_2 s\theta_1 - l_1 s\theta_1 + h_3 s\theta_1 s\theta_3 - h_3 c\theta_1 s\theta_2 c\theta_3 \\ h_1 + h_3 c\theta_2 c\theta_3 \end{bmatrix} \tag{2.52}$$

Rotating between the body frame (3) and the ground frame (g) involves a transformation represented by the rotation matrix R_3^g. The matrix is given by:

$$R_3^g = \begin{bmatrix} c\alpha_1 c\alpha_3 - s\alpha_1 s\alpha_2 s\alpha_3 & -c\alpha_2 s\alpha_1 & c\alpha_1 s\alpha_3 + c\alpha_3 s\alpha_2 s\alpha_1 \\ c\alpha_3 s\alpha_1 + c\alpha_1 s\alpha_2 s\alpha_3 & c\alpha_2 c\alpha_1 & s\alpha_3 s\alpha_1 - c\alpha_3 c\alpha_1 s\alpha_2 \\ -c\alpha_2 s\alpha_3 & s\alpha_2 & c\alpha_3 c\alpha_2 \end{bmatrix} \tag{2.53}$$

In this matrix, α_1 represents the yaw angle, α_2 represents the roll angle, and α_3 represents the pitch angle. These angles are derived using information from the IMU. The rotation matrix R_3^g describes the transformation between the body and ground frames, allowing for the quadcopter's orientation relative to the ground.

To quantify the error in the desired orientation, a new rotation matrix R_e^3 is introduced. This matrix is defined as:

$$R_e^3 = \begin{bmatrix} c\epsilon_1 c\epsilon_3 - s\epsilon_1 s\epsilon_2 s\epsilon_3 & -c\epsilon_2 s\epsilon_1 & c\epsilon_1 s\epsilon_3 + c\epsilon_3 s\epsilon_2 s\epsilon_1 \\ c\epsilon_3 s\epsilon_1 + c\epsilon_1 s\epsilon_2 s\epsilon_3 & c\epsilon_2 c\epsilon_1 & s\epsilon_3 s\epsilon_1 - c\epsilon_3 c\epsilon_1 s\epsilon_2 \\ -c\epsilon_2 s\epsilon_3 & s\epsilon_2 & c\epsilon_3 c\epsilon_2 \end{bmatrix} \tag{2.54}$$

Here, ϵ represents the error in the orientation, calculated as the difference between the desired orientation (α_{desire}) and the actual orientation (α). The matrix R_e^3 characterizes the rotation necessary to align the desired orientation with the current orientation. Finally, the total rotation matrix between the error frame (e) and the gimbal base frame can be obtained by combining the rotation matrices:

$$R_e^g(\Theta, \epsilon) = R_e^3(\Theta) R_3^g(\epsilon) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.55}$$

$$\begin{cases} \theta_{1new} = \arctan 2(-r_{12}, r_{22}) \\ \theta_{2new} = \arcsin(r_{32}) \\ \theta_{3new} = \arctan 2(-r_{31}, r_{33}) \end{cases} \tag{2.56}$$

The resulting matrix R_e^g represents the transformation between the error frame and the gimbal base frame, enabling the adjustment of the gimbal mechanism to compensate for the orientation error.

The input for the gimbal mechanism can be a set of desired angular velocities Φ of

the camera in its coordinate frame. The vector represents these angular velocities:

$$\Phi_3^3 = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = \begin{bmatrix} \phi_{roll} \\ \phi_{pitch} \\ \phi_{yaw} \end{bmatrix} \tag{2.57}$$

The relation between the angular velocities in the camera's coordinate frame and the local angular velocities in the gimbal joints can be expressed as:

$$\Phi_3^3 = J\Omega \tag{2.58}$$

Here, Ω represents the local angular velocities in the gimbal joints, and J is the Jacobian matrix. The local angular velocities Ω are defined as:

$$\Omega = (\omega_1, \omega_2, \omega_3)^T = (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)^T \tag{2.59}$$

where $\dot{\theta}_1$, $\dot{\theta}_2$, and $\dot{\theta}_3$ are the time derivatives of the joint angles.
The Jacobian matrix J is given by:

$$J = \begin{bmatrix} \hat{\theta}_1^3 & \hat{\theta}_2^3 & \hat{\theta}_3^3 \end{bmatrix} \tag{2.60}$$

Here, $\hat{\theta}_1^3$, $\hat{\theta}_2^3$, and $\hat{\theta}_3^3$ represent the angular velocity directions of the joints in the camera's coordinate frame.
The Jacobian matrix J can also be expressed as:

$$J = \begin{bmatrix} R_1^3\hat{\theta}_1^1 & R_2^3\hat{\theta}_2^3 & \hat{\theta}_3^3 \end{bmatrix} \tag{2.61}$$

where R_1^3 and R_2^3 are the rotational matrices from joint 1 to joint 3 and from joint 2 to joint 3, respectively.The angular velocity directions of the joints in the camera's coordinate frame are given by:

$$\hat{\theta}_1^1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \hat{\theta}_2^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \hat{\theta}_3^3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{2.62}$$

Combining these expressions, the Jacobian matrix J becomes:

$$J = \begin{bmatrix} (R_1^2R_2^3)^T\hat{\theta}_1^1 & (R_2^3)^T\hat{\theta}_2^2 & \hat{\theta}_3^3 \end{bmatrix} \tag{2.63}$$

Furthermore, the desired angular velocities Φ can be obtained as:

$$\begin{bmatrix} \phi_{roll} \\ \phi_{pitch} \\ \phi_{yaw} \end{bmatrix} = \begin{bmatrix} -c\theta_2 s\theta_3 & c\theta_3 & 0 \\ s\theta_2 & 0 & 1 \\ c\theta_2 c\theta_3 & s\theta_3 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{2.64}$$

This equation relates the desired roll, pitch, and yaw angular velocities to the joint angular velocities. After simplification, the following result is obtained:

$$\Rightarrow \begin{bmatrix} \phi_{yaw} \\ \phi_{roll} \\ \phi_{pitch} \end{bmatrix} = \begin{bmatrix} c\theta_2 c\theta_3 & s\theta_3 & 0 \\ -c\theta_2 s\theta_3 & c\theta_3 & 0 \\ s\theta_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{2.65}$$

These equations provide the relationship between the desired camera angular velocities and the joint angular velocities. To determine the desired joint velocities Ω, the Jacobian matrix J can be inverted.

$$\Omega = J^{-1} \begin{bmatrix} \phi_{yaw} \\ \phi_{roll} \\ \phi_{pitch} \end{bmatrix} \tag{2.66}$$

This equation calculates the joint velocities required to achieve the desired camera angular velocities. Finally, the current translational velocity $\dot{X}$ of the end-effector can be determined using the Jacobian matrix J_D:

$$\dot{X} = J_D \Omega = \begin{bmatrix} \hat{\theta}_0^3 \times d_1^3 & \hat{\theta}_1^3 \times d_2^3 & \hat{\theta}_2^3 \times d_3^3 \end{bmatrix} \Omega \tag{2.67}$$

In this equation, $\hat{\theta}_0^3$, $\hat{\theta}_1^3$, and $\hat{\theta}_2^3$ represent the directions of the end-effector's translation relative to the gimbal joints, and d_1^3, d_2^3, and d_3^3 are the distances from the joints to the end-effector. By multiplying the Jacobian matrix J_D with the joint velocities Ω, the current translational velocity $\dot{X}$ can be determined.

2.3.2.2 Dynamic Model of the Gimbal Mechanism

In the dynamics of a gimbal mechanism, the focus is on establishing the relationship between the motion of the gimbal mechanism's links and joints and the forces that cause this motion. For analysis, the gimbal is assumed to be a rigid body. When deriving the dynamic model of a rigid body, either Lagrangian equations or Newton's equations can be employed. However, when dealing with complex systems, Newton's equations of motion can become quite challenging due to the consideration of constraint forces and the geometric nature of the system. Consequently, the Lagrangian approach offers a more straightforward way to derive the equations of motion by considering the system's total kinetic and potential energies. Hence, the dynamics of the

gimbal mechanism are described using the Lagrangian equation. The gimbal mechanism can be regarded as a robot with n joints. To describe such a robot, the following Lagrange equation is utilized:

$$Q_i = \frac{d}{dt}(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}) - \frac{\partial \mathcal{L}}{\partial \theta_i} \qquad i = 1, 2, ..., n \tag{2.68}$$

In this equation, Q_i represents the generalized force acting on the joint i. The Lagrange equation relates this force to the time derivative of the partial derivative of the Lagrangian $\mathcal{L}$ concerning the joint velocity $\dot{\theta}_i$, and the partial derivative of $\mathcal{L}$ concerning the joint angle θ_i. It should be noted that this equation applies to each joint of the gimbal mechanism. The Lagrangian $\mathcal{L}$ is defined as:

$$\mathcal{L} = K - V \tag{2.69}$$

Here, V represents the potential energy of the system. In contrast, K represents its kinetic energy. The Lagrangian captures the difference between these two energy components. Using the Lagrange and Lagrangian equations, the dynamic model of the gimbal mechanism can be established, allowing for the analysis of its motion and the forces involved.

2.3.2.3 Kinetic Energy of the Gimbal Mechanism

To determine the kinetic energy of the gimbal mechanism, it is necessary to calculate the velocity of each link relative to the base frame. Initially, the center of gravity's position is determined to find each link's velocity. The position of a point P on link i can be transformed into the base frame using the transformation matrix ${}^0T_i$ as follows:

$${}^0r_P = {}^0\,T_i\;{}^ir_P \tag{2.70}$$

Next, the velocity of point P relative to the base frame is calculated as:

$${}^0\dot{r}_P = \sum_{j=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_i}\,\dot{\theta}_j\;{}^ir_P \tag{2.71}$$

The square of the velocity in matrix form can be expressed as:

$$\begin{aligned} {}^0\dot{r}_P^2 &= {}^0\dot{r}_P \,.\, {}^0\dot{r}_P = tr({}^0\dot{r}_P \, {}^0\dot{r}_P^T) \\ &= tr\left(\sum_{j=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_i} \dot{\theta}_i \, {}^i r_P \sum_{k=1}^{i} \left[\frac{\partial {}^0T_i}{\partial \theta_i} \dot{\theta}_i \, {}^i r_P \right]^T \right) \\ &= tr\left(\sum_{j=1}^{i} \sum_{k=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_j} \, {}^i r_P \, {}^i r_P^T \left[\frac{\partial {}^0T_i}{\partial \theta_k} \right]^T \dot{\theta}_j \dot{\theta}_k \right) \end{aligned} \tag{2.72}$$

Since kinetic energy is proportional to velocity and mass, the kinetic energy of a point P with a small mass dm can be given as:

$$\begin{aligned} dK_P &= \frac{1}{2} tr\left(\sum_{j=1}^{i} \sum_{k=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_j} \, {}^i r_P \, {}^i r_P^T \left[\frac{\partial {}^0T_i}{\partial \theta_k} \right]^T \dot{\theta}_j \dot{\theta}_k \right) dm \\ &= \frac{1}{2} tr\left(\sum_{j=1}^{i} \sum_{k=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_j} \, ({}^i r_P \, dm \, {}^i r_P^T) \left[\frac{\partial {}^0T_i}{\partial \theta_k} \right]^T \dot{\theta}_j \dot{\theta}_k \right) \end{aligned} \tag{2.73}$$

For link i, the kinetic energy is obtained by integrating over the volume B_i as follows:

$$K_i = \int_{B_i} dK_P = \frac{1}{2} tr\left(\sum_{j=1}^{i} \sum_{k=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_j} \, {}^i\bar{I}_i \, \frac{\partial {}^0T_i}{\partial \theta_k}^T \dot{\theta}_j \dot{\theta}_k \right) \tag{2.74}$$

Here, ${}^i\bar{I}_i$ represents the inertia matrix for link i and is defined as:

$${}^i\bar{I}_i = \int_{B_i} {}^i r_P \, {}^i r_P^T \, dm \tag{2.75}$$

Finally, the kinetic energy of the gimbal, which consists of n links, can be expressed as:

$$K = \sum_{i=1}^{n} K_i = \frac{1}{2} tr \sum_{i=1}^{n} \left(\sum_{j=1}^{i} \sum_{k=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_j} \, {}^i\bar{I}_i \, \frac{\partial {}^0T_i}{\partial \theta_k}^T \dot{\theta}_j \dot{\theta}_k \right) \tag{2.76}$$

In this equation, the trace function $tr(\cdot)$ calculates the sum of the diagonal elements of a matrix, and $\dot{\theta}_i$ represents the angular velocity of joint i. The kinetic energy is obtained by summing the contributions from each link, considering their respective inertia matrices and the angular velocities.

The only energy source considered for the gimbal's potential energy is gravity. The potential energy of each link i can be calculated using the below equation.

$$V_i = -m_i \, {}^0g \, {}^0r_i = -m_i \, {}^0g^T \, {}^0T_i \, {}^i r_i \tag{2.77}$$

Here, ${}^0g = [g_x \quad g_y \quad g_z \quad 0]^T$ represents the gravitational acceleration, typically in the direction of $-z_g$, and ${}^0r_i$ represents the coordinates of the center of mass of each

link expressed in the base frame. The potential energy of the entire gimbal is obtained by summing the potential energies of all the links:

$$V = \sum_{i=1}^{n} V_i = -\sum_{i=1}^{n} m_i \, {}^0g^T \, {}^0T_i \, {}^i r_i \tag{2.78}$$

By substituting equations (2.76) and (2.78) into equation (2.69), the Lagrangian of the gimbal can be expressed as:

$$\begin{aligned} \mathcal{L} &= K - V \\ &= \frac{1}{2} tr \sum_{i=1}^{n} \left(\sum_{j=1}^{i} \sum_{k=1}^{i} \frac{\partial^0 T_i}{\partial \theta_j} \, {}^i\bar{I}_i \, \frac{\partial^0 T_i}{\partial \theta_k}^T \right) \dot{\theta}_j \dot{\theta}_k + \sum_{j=1}^{n} m_i \, {}^0g^T \, {}^0T_i \, {}^i r_i \end{aligned} \tag{2.79}$$

To solve the Lagrange equation, it is necessary to compute $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}$ and $\frac{\partial \mathcal{L}}{\partial \theta_i}$ and substitute them into equation (2.68). The expressions for these derivatives are as follows:

$$\begin{aligned} \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} &= \sum_{i=r}^{n} \sum_{j=1}^{i} tr \left(\frac{\partial \, {}^0T_i}{\partial \theta_j} \, {}^i\bar{I}_i \, \frac{\partial \, {}^0T_i}{\partial \theta_r}^T \right) \ddot{\theta}_j \\ &\quad + \sum_{i=r}^{n} \sum_{j=1}^{i} \sum_{k=1}^{i} tr \left(\frac{\partial^2 \, {}^0T_i}{\partial \theta_j \partial \theta_k} \, {}^i\bar{I}_i \, \frac{\partial \, {}^0T_i}{\partial \theta_r}^T \right) \dot{\theta}_j \dot{\theta}_k \\ &\quad + \sum_{i=r}^{n} \sum_{j=1}^{i} \sum_{k=1}^{i} tr \left(\frac{\partial^2 \, {}^0T_i}{\partial \theta_r \partial \theta_k} \, {}^i\bar{I}_i \, \frac{\partial \, {}^0T_i}{\partial \theta_j}^T \right) \dot{\theta}_j \dot{\theta}_k \end{aligned} \tag{2.80}$$

and

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_i} &= \sum_{i=r}^{n} \sum_{j=1}^{i} \sum_{k=1}^{i} tr \left(\frac{\partial^2 \, {}^0T_i}{\partial \theta_r \partial \theta_j} \, {}^i\bar{I}_i \, \frac{\partial \, {}^0T_i}{\partial \theta_k}^T \right) \dot{\theta}_j \dot{\theta}_k \\ &\quad + \sum_{i=r}^{n} m_i g^T \frac{\partial \, {}^0T_r}{\partial \theta_i} \, {}^i r_i \end{aligned} \tag{2.81}$$

Therefore, the Lagrange equation takes the form:

$$\begin{aligned} Q_i &= \sum_{i=i}^{n} \sum_{k=1}^{j} tr \left(\frac{\partial \, {}^0T_j}{\partial \theta_k} \, {}^j\bar{I}_j \, \frac{\partial \, {}^0T_j}{\partial \theta_i}^T \right) \ddot{\theta}_k \\ &\quad + \sum_{j=i}^{n} \sum_{k=1}^{j} \sum_{m=1}^{j} tr \left(\frac{\partial^2 \, {}^0T_j}{\partial \theta_k \partial \theta_m} \, {}^j\bar{I}_j \, \frac{\partial \, {}^0T_j}{\partial \theta_i}^T \right) \dot{\theta}_k \dot{\theta}_m \\ &\quad - \sum_{j=i}^{n} m_j g^T \frac{\partial \, {}^0T_j}{\partial \theta_i} \, {}^j r_j \end{aligned} \tag{2.82}$$

To simplify this equation, the matrices $D_{ij}(\theta)$, H_{ikm}, and vector G_i are defined as follows:

$$Q_i = \sum_{j=1}^{n} D_{ij}(\theta) \ddot{\theta}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} H_{ikm} \dot{\theta}_k \dot{\theta}_m + G_i \tag{2.83}$$

$$D_{ij} = \sum_{r=max(i,j)}^{n} tr\left(\frac{\partial^0 T_r}{\partial \theta_i}\ {}^r\bar{I}_r\left(\frac{\partial^0 T_r}{\partial \theta_j}\right)^T\right) \tag{2.84}$$

$$H_{ijk} = \sum_{r=max(i,j,k)}^{n} tr\left(\frac{\partial^2\ {}^0T_r}{\partial \theta_j \partial \theta_k}\ {}^r\bar{I}_r\left(\frac{\partial^0 T_r}{\partial \theta_i}\right)^T\right) \tag{2.85}$$

$$G_i = -\sum_{r=i}^{n} m_r g^T \frac{\partial\ {}^0T_r}{\partial \theta_i}\ {}^r r_r \tag{2.86}$$

where ${}^r\bar{I}_r$ is the inertia matrix of body r expressed in frame r. It takes the following form:

$$\bar{I} = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & I_{xy} & I_{xz} & mx_{cm} \\ I_{yx} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & I_{yz} & my_{cm} \\ I_{zx} & I_{zy} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & mz_{cm} \\ mx_{cm} & my_{cm} & mz_{cm} & m \end{bmatrix}, \tag{2.87}$$

where I_{ii} are the elements of the mass moment of inertia I_r of body r, and (x_{cm}, y_{cm}, z_{cm}) are the coordinates of the body's center of mass.

In summary, the Lagrange equation for the gimbal system is given by:

$$Q_i = \sum_{j=1}^{n} D_{ij}(\theta)\ddot{\theta}j + \sum j = 1^n \sum_{k=1}^{n} H_{ikm}\dot{\theta}_k\dot{\theta}_m + G_i \tag{2.88}$$

where Q_i represents the generalized forces, D_{ij} represents the inertia matrix, H_{ikm} represents the Coriolis and centrifugal terms, and G_i represents the gravity term. This equation governs the dynamics of the gimbal system.

2.4 Controlling of the Gimbal

In this section, MATLAB/SIMULINK is employed as a simulation tool to control a gimbal mounted on a quadcopter. The initial Simulink file provided in [80] serves as the foundation upon which additional functionalities, such as the gimbal mechanism and target observer, are incorporated into the system (Figure 2.6). The primary objective is to develop a system enabling the quadcopter to constantly track and follow a target object. To achieve this, a quadcopter path command is designed to align its trajectory with the target object's x and y directions. The output of the Target Observer block, which captures the target object's position, serves as the input for the position controller block. This, in turn, calculates the attitude commands, subsequently

utilized as the input for the attitude controller, thereby directing the quadcopter to pursue the target object autonomously.

Additionally, the state of the target object and the quadcopter are fed into the gimbal mechanism block, which calculates the appropriate gimbal angles to ensure that the gimbal mechanism's end effector remains constantly directed towards the target object. The target state block simulates the target object's changing position in the x, y, and z coordinates over time. The Target Observer block, which simulates the measurement system, introduces some measurement errors due to noise. These noise effects are mitigated using a Kalman filter, enabling the estimation of the target object's states. To evaluate the algorithm's performance, experiments involve making the quadcopter follow a target object moving along an arbitrary path. The system's effectiveness is assessed by examining how accurately the quadcopter tracks the target path and how precisely the gimbal mechanism's end effector maintains alignment with the target object's position.

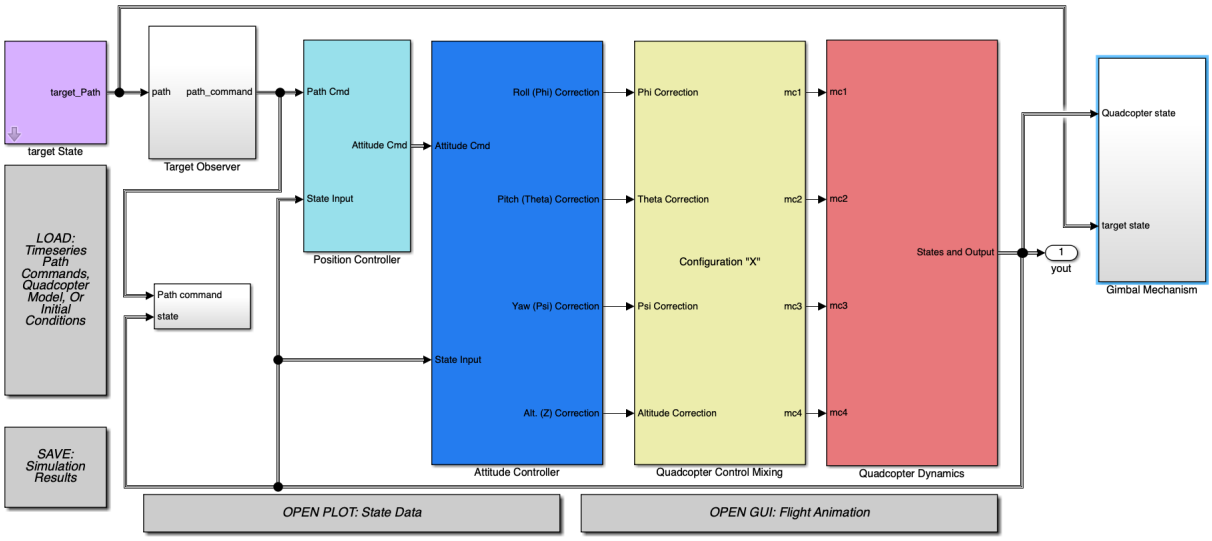


Figure 2.6: Simulink of the control system

Control techniques are employed to align the gimbal mechanism's joints to the desired angles using inverse kinematic equations. PID controllers are utilized for this purpose. The PID controller generates a control signal proportional to the error, the summation of the error, and the rate of change of the error. Let $\theta_{d,n}$ denote the desired angle obtained from the inverse kinematic equations, and θ_n represent the current an-

gle of the gimbal mechanism's joint n. The control signal for the n-th motor torque, denoted as τ_n, is given by the following equation:

$$\tau_n = K_{p,n}(\theta_{d,n} - \theta_n) + K_{i,n} \int_0^{t_f} (\theta_{d,n} - \theta_n)dt + K_{d,n} \frac{d(\theta_{d,n} - \theta_n)}{dt} \tag{2.89}$$

Here, $K_{p,n}$, $K_{i,n}$, and $K_{d,n}$ represent the proportional, integral, and derivative constants, respectively. These constants are determined through a manual tuning process, where their values are adjusted to achieve the desired control performance for the gimbal mechanism.

The PID controller utilizes feedback from the error between the desired and current angles to generate the control signal. The proportional term, proportional to the error, contributes to the control signal in a manner that is proportional to the deviation from the desired angle. The integral term considers the accumulated error over time, allowing the controller to address steady-state errors. Lastly, the derivative term captures the rate of change of the error, enabling the controller to respond effectively to sudden changes in the angle. By fine-tuning the PID controller constants, the gimbal mechanism can accurately and dynamically adjust the joint angles, aligning them with the desired positions dictated by the inverse kinematic equations. This control approach enables precise and responsive tracking of the target object, enhancing the gimbal mechanism's ability to maintain its orientation and direction.

2.4.1 Tracking a Target Object

Integrating a gimbal mechanism into the quadcopter's design enables precise control and manipulation of the end-effector's orientation. The end-effector, which carries the extruder, can be maneuvered by the gimbal mechanism to continuously point towards the target object, regardless of its movements or changes in position.

This part utilizes the gimbal mechanism and an end-effector with an extruder to ensure it consistently aligns with the target object. To achieve this, an algorithm that can receive either the location of the tracked object or the desired path that the quadcopter is developed. An algorithm that takes in either the target object's location or the predetermined path is designed to facilitate the tracking and following process. If the location is available, the algorithm generates commands for the quadcopter to move along that path, ensuring that the end effector remains pointed at the target through-

out the flight. By harnessing the capabilities of the gimbal mechanism, a system can enable precise and efficient tracking and following of target objects.

2.4.2 Object State Estimation by Using The Kalman Filter

In an ideal scenario, the UAV relies on sensor data to determine the target object's position, enabling the determination of its current state. Unfortunately, the measurement method employed introduces considerable noise due to environmental disturbances. Consequently, the UAV must rely on more than raw measurement data to ensure effective tracking of the target object. Thus, it becomes necessary to process the data obtained from the measuring instrument and eliminate the noise generated during the measurement process.

The Kalman filter, known for providing precise estimates of target object states, is applied in this phase to carry out the state estimation procedure to achieve accurate state estimation. Let's delve into the explanation of this procedure. The position of the target object in the x, y, and z directions is represented by the following discretized linear dynamic equation:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{2.90}$$

Here, $x(k+1)$ denotes the new state of the target object, and $x(k)$ represents its old state. A is the state transition matrix, B is the input matrix, $u(k)$ denotes the input to the system, and $w(k)$ accounts for disturbances in the system. The assumption is that any disturbance in the system follows white noise characteristics, having a mean value of 0 and a covariance, R_w, represented by an $n \times n$ matrix.

The measurement equation of the system can be described as follows:

$$y(k+1) = Cx(k) + Du(k) + v(k) \tag{2.91}$$

In this equation, $y(k+1)$ denotes the output of the measurement device, C is the matrix used to derive the measured state, D is the matrix demonstrating the influence of the input on the measurement output, and $v(k)$ represents the measurement noise introduced by the instruments used to measure the state. Similar to the disturbance, it is assumed that any measurement noise follows white noise characteristics with a mean value of 0 and a covariance matrix R_v.

However, the precise dynamic equation for a moving object is typically unknown. Hence, modeling techniques based on physics principles are employed to derive this equation. Consequently, the following notation is used to express the estimated discretized linear dynamic model for the target object:

$$\hat{x}(k+1|k) = \hat{A}\hat{x}(k) + \hat{B}u(k) \tag{2.92}$$

Here, $\hat{A}$ represents the estimated state transition matrix, which may differ from the target object's actual state transition matrix A. Similarly, $\hat{B}$ is the estimated input matrix, which may differ from the actual B. $\hat{x}(k+1|k)$ denotes the estimated new state of the target object given the estimated previous state $\hat{x}(k)$. Consequently, there exists a discrepancy between the estimated state and the actual state of the target object, with an associated covariance matrix P determined as follows:

$$P(k+1|k) = AP(k|k)A^T + R_w \tag{2.93}$$

Here, R_w represents the covariance matrix of the disturbance, and $P(k+1|k)$ is the estimated system covariance matrix derived using the previous system covariance matrix $P(k|k)$.

Once the new state of the target object has been estimated, the measurement device produces the following results:

$$e(k+1) = C\hat{x}(k+1|k) + Du(k) \tag{2.94}$$

The expected value denoted by $e(k+1)$ is obtained from the measurement device in this equation. As the state is estimated, the difference between the actual output of the measurement system and its estimated output can be computed using the following formula:

$$z = y(k+1) - e(k+1) \tag{2.95}$$

Here, z represents the error between the real state $y(k+1)$ and the estimated state $e(k+1)$. The covariance of this error can be calculated as:

$$Z = R_v + CP(k+1|k)C^T \tag{2.96}$$

In this equation, Z denotes the covariance of the error between the real state $y(k+1)$ and the estimated state $e(k+1)$, while R_v represents the covariance of the measurement noise. Next, the Kalman gain is computed using the following equation:

$$K = PC^T Z^{-1} = \frac{PC^T}{R_v + CP(k+1|k)C^T} \tag{2.97}$$

The Kalman gain allows for the update and correction of the estimated state, utilizing the following equation:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(y(k+1) - C\hat{x}(k+1|k)) \tag{2.98}$$

Here, $\hat{x}(k+1|k+1)$ represents the target object's new updated and corrected state. Additionally, the newly updated system covariance matrix is computed as:

$$P(k+1|k+1) = P(k+1|k) + KCP(k+1|k) \tag{2.99}$$

Figure 2.7 and Figure 2.8 serve as informative visual aids, presenting the outputs of both the target state measuring device and the Kalman filter. Specifically, Figure 2.7 demonstrates the production of X_{cmd} in conjunction with the output of the Kalman filter, enabling a direct comparison between the two. Similarly, in Figure 2.8, the production of Y_{cmd} is juxtaposed against the output of the Kalman filter, allowing for a comprehensive assessment of their relative performance.

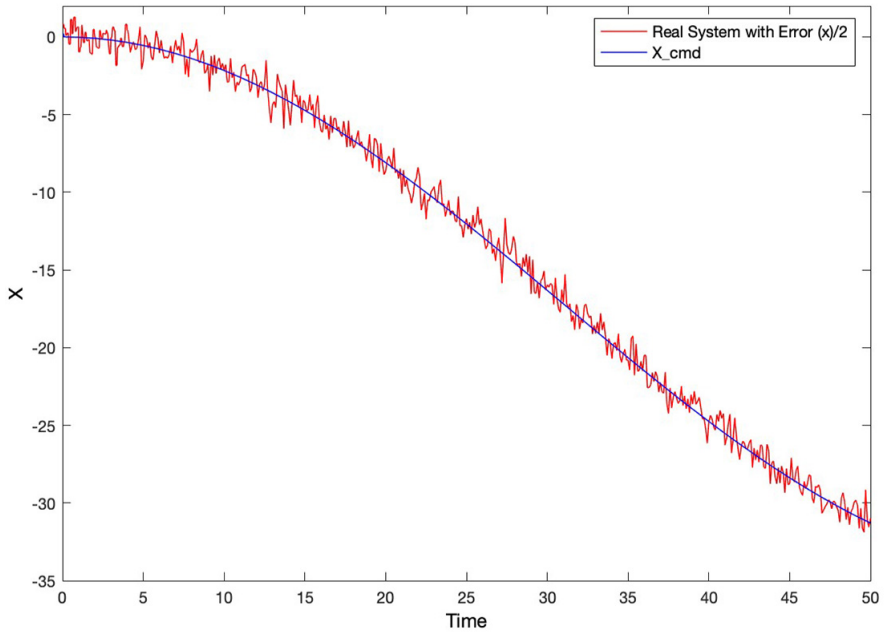


Figure 2.7: The output of the X_{cmd} with the output of Kalman filter

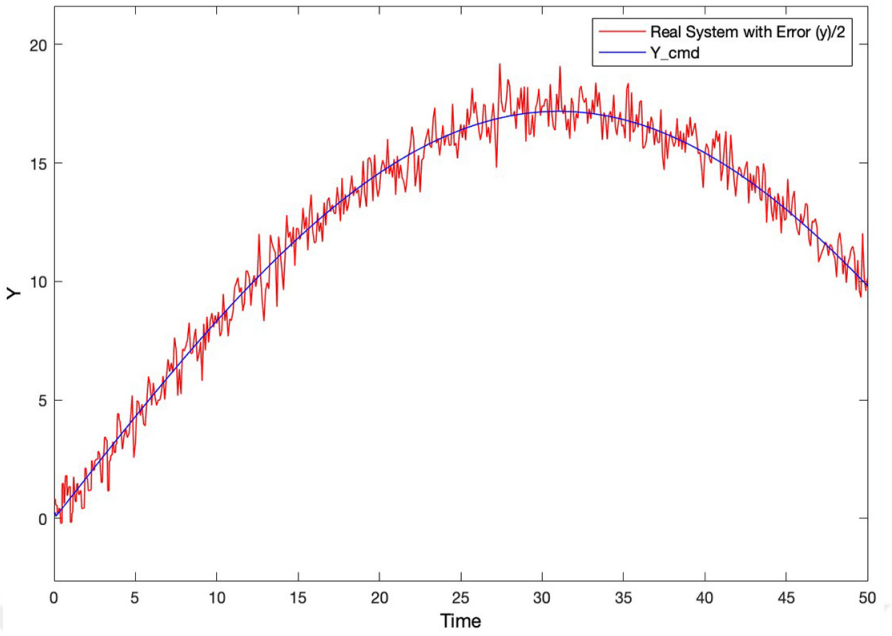


Figure 2.8: The output of the Y_{cmd} with the output of Kalman filter

The figures presented above serve a critical purpose by providing a concrete and visual depiction of the Kalman filter's effectiveness and accomplishments in precisely estimating and tracking the state of the target object. A direct visual comparison between the measurements acquired from the state measuring device and the outputs generated by the Kalman filter makes it considerably easier to successfully evaluate the filter's capability to counteract the impact of noise and environmental disturbances. This visual representation offers invaluable insights into the performance and robustness of the Kalman filter, further underscoring its significance as a dependable and trustworthy tool for state estimation and tracking in UAV applications.

By examining the figures, observers can directly witness the filter's ability to reduce tracking errors over time. Despite initially encountering significant deviations between the measured and estimated states, the Kalman filter consistently demonstrates its aptitude for minimizing these errors and aligning the estimated state with the actual state of the target object. This dynamic tracking performance showcases the filter's capacity to adapt and respond to changing conditions, resulting in a more accurate representation of the target object's state.

Furthermore, the figures effectively showcase the filter's ability to mitigate the adverse effects of noise and disturbances. Any discrepancies or deviations caused by

environmental factors become apparent by overlaying the measured and estimated states. Observers can observe how the Kalman filter effectively filters out noise and ensures that the estimated state closely aligns with the actual state of the target object. The insights gained from these figures emphasize the Kalman filter's crucial role in enabling precise state estimation and tracking in UAV applications. Its ability to handle noise and disturbances and its consistent reduction of tracking errors solidifies its position as a reliable and indispensable tool for ensuring accurate and reliable tracking of target objects. The figures offer a tangible representation of the Kalman filter's efficacy, giving researchers, engineers, and operators confidence in its performance and encouraging widespread adoption in UAV systems.

2.4.3 Target Object Tracking Algorithm

Figure 2.9 and 2.10 provide an insightful analysis of the quadcopter's performance, accurately tracking the target object along the X and Y. These graphs comprehensively evaluate the quadcopter's ability to track a target object above it effectively and give valuable insights into its tracking capabilities and performance in maintaining precision during the tracking process.

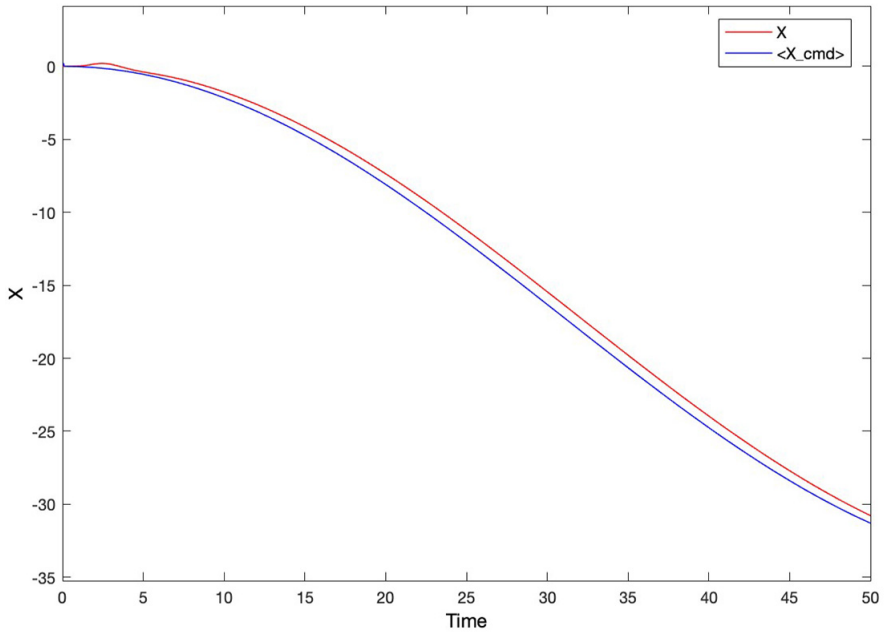


Figure 2.9: The performance of the quadcopter in tracking the target in X direction

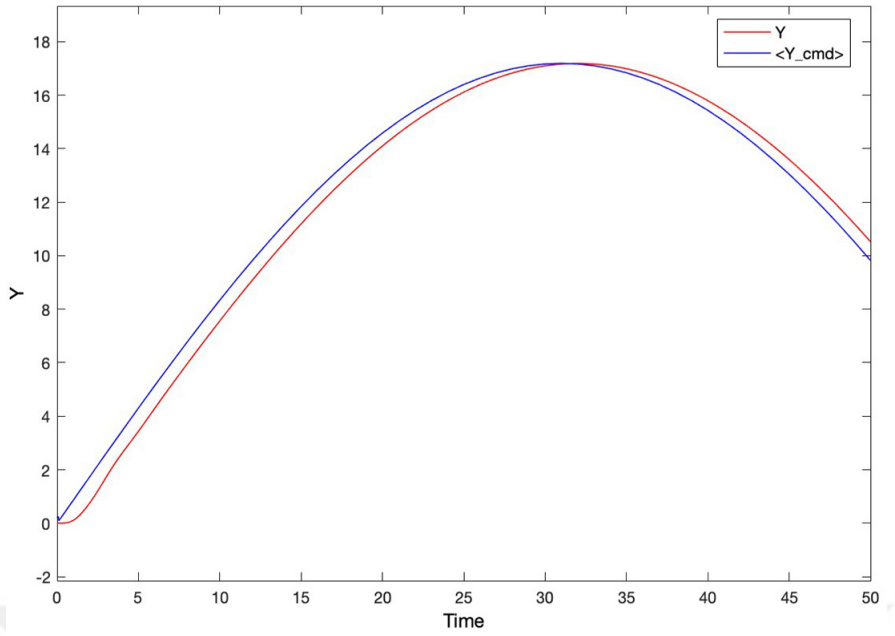


Figure 2.10: The performance of the quadcopter in tracking the target in Y direction

The graphs show that the quadcopter initially experiences a significant tracking error, resulting in a noticeable deviation between the desired target and the quadcopter's actual position. This initial error can be attributed to various factors such as system dynamics, external disturbances, or measurement inaccuracies.

It is noteworthy that achieving precise tracking in UAV systems, particularly for a quadcopter, presents inherent challenges due to system dynamics, aerodynamic effects, and environmental disturbances. Therefore, the level of performance exhibited by the quadcopter in these graphs can be deemed highly commendable. The quadcopter's ability to consistently improve its tracking accuracy reflects the effectiveness of its control system. It highlights the successful integration of the state estimation algorithm, such as the Kalman filter, into the tracking process.

While the quadcopter actively tracks the target object, the gimbal mechanism's end-effector must continuously align with the target. This alignment ensures that the extruder attached to the gimbal mechanism accurately captures the target object's position and remains focused on it throughout. The performance of the gimbal mechanism in precisely tracking the target object can be observed in Figure 2.11 and Figure 2.12.

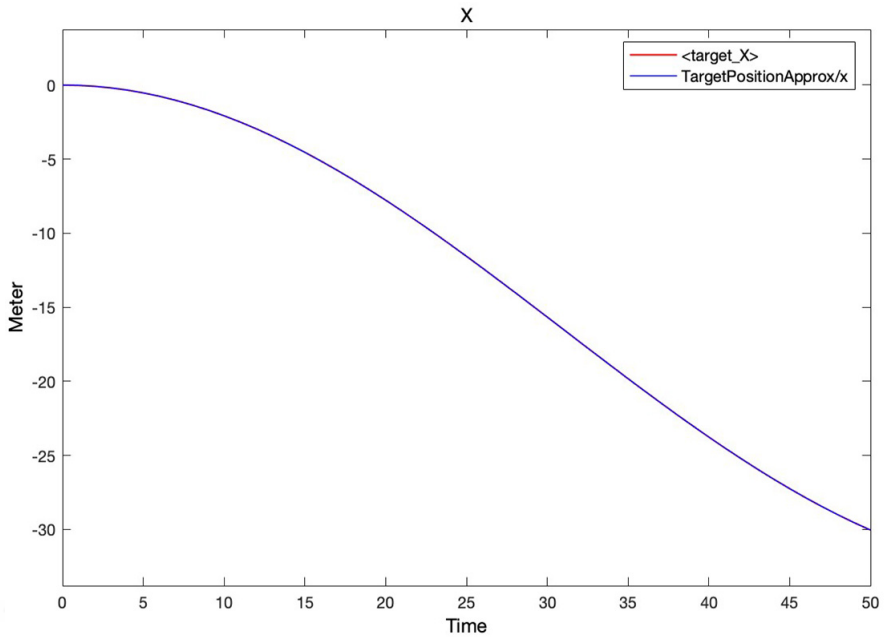


Figure 2.11: The performance of the gimbal in tracking the target in X direction

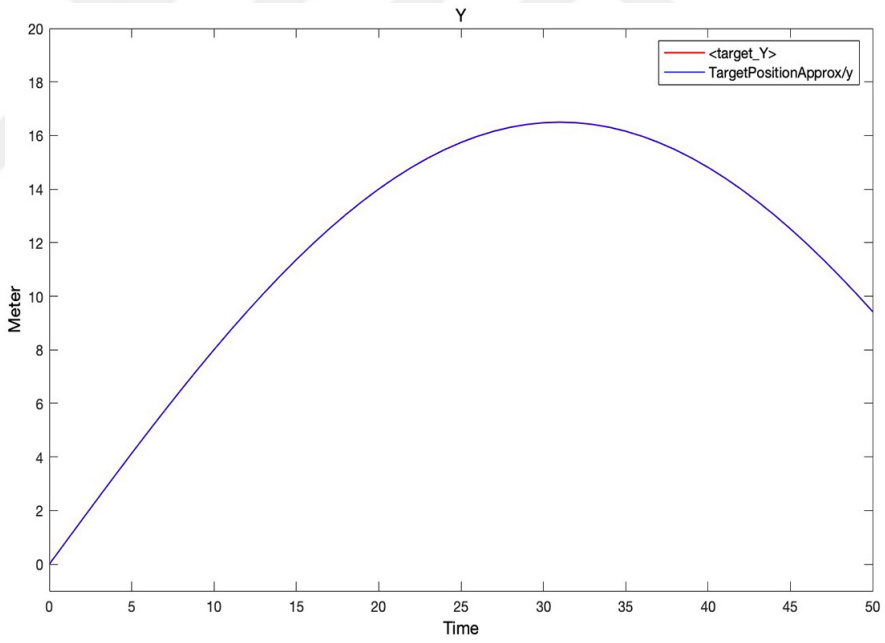


Figure 2.12: The performance of the gimbal in tracking the target in Y direction

Figure 2.11 visually represents the gimbal mechanism's tracking performance in the X direction. The graph exhibits the close correlation between the gimbal's output and the target object's position, showcasing the mechanism's ability to adjust and main-

tain alignment as the target moves. The consistent alignment depicted in the graph demonstrates the gimbal mechanism's proficiency in accurately tracking the target object's movement along the X direction.

Similarly, Figure 2.12 showcases the gimbal mechanism's tracking capability in the Y direction. The graph reveals the smooth and precise adjustments made by the gimbal mechanism to align itself with the target object's movement. The proximity between the gimbal's output and the target's position on the graph reflects the mechanism's effectiveness in accurately tracking the target object's motion in the Y direction.

These figures visually represent the gimbal mechanism's performance and its crucial role in maintaining consistent alignment with the target object. By effectively tracking the target object's movement, the gimbal mechanism ensures that the camera and extruder mounted on it remain focused on the target, enabling accurate and reliable visual tracking.

Furthermore, additional insights into the angle commands given to the gimbal mechanism and its angle-tracking performance can be derived from Figure 2.13, 2.14 and 2.15. These figures offer valuable information regarding the angle command provided to the gimbal mechanism. These graphs showcase the command's variations and how the machine responds to these commands, accurately adjusting the end-effector's orientation. This capability enables the gimbal mechanism to adapt to the target object's position continuously changes and maintain optimal alignment. The close correspondence between the gimbal's angle output and the target's angular position on the graph highlights the mechanism's accuracy in tracking the target object's angular movements.

These graphs confirm the gimbal mechanism's exceptional tracking performance in position and angle domains. Despite any errors or disturbances experienced by the quadcopter during the tracking process, the gimbal mechanism consistently and reliably tracks the target object's positions and orientations. This level of performance ensures that the camera and extruder attached to the gimbal's end-effector remain focused on the target object, undeterred by the quadcopter's movements and vibrations.

In conclusion, Figure 2.11, 2.12, 2.13, 2.14 and 2.15 collectively showcase the gimbal mechanism's exceptional tracking capabilities. The graphs provide tangible evidence of the mechanism's ability to align accurately with the target object's position and maintain a consistent viewing angle. This reliable and stable tracking ensures that the

target object remains focused, enabling the system to deliver dependable and precise visual tracking capabilities for various applications.

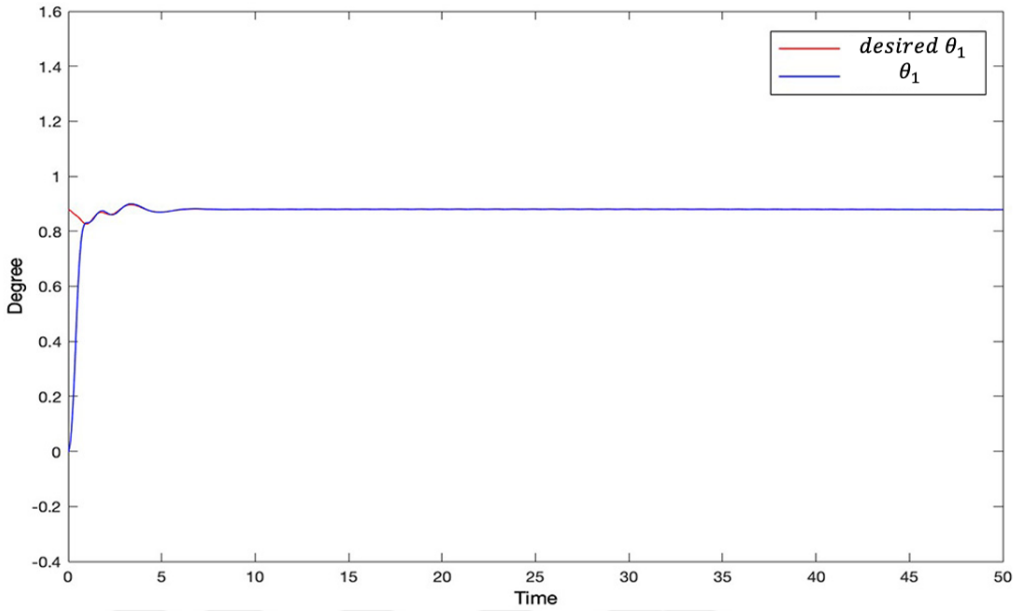


Figure 2.13: θ_1 of the gimbal mechanism

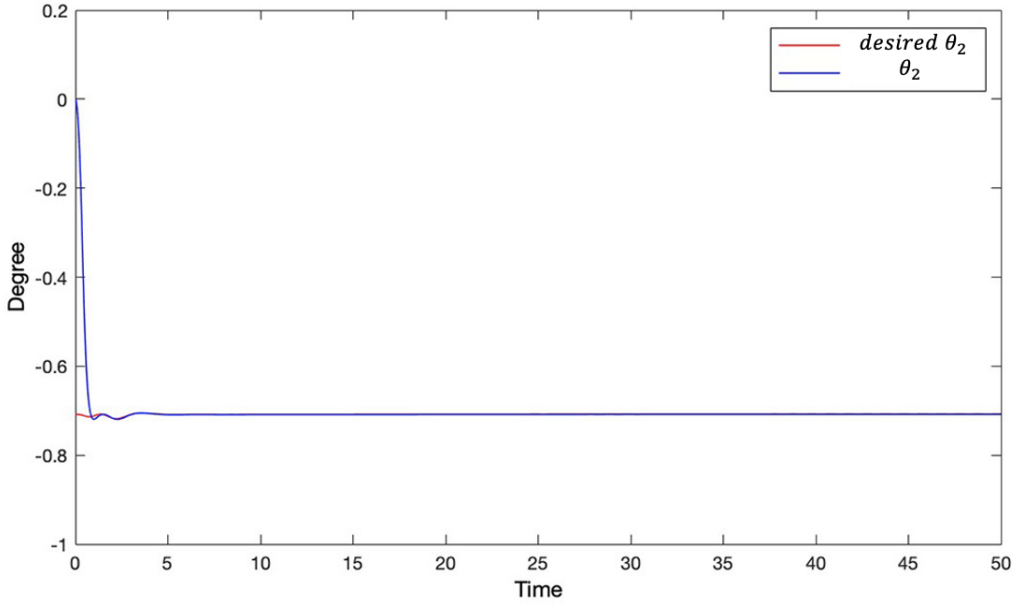


Figure 2.14: θ_2 of the gimbal mechanism

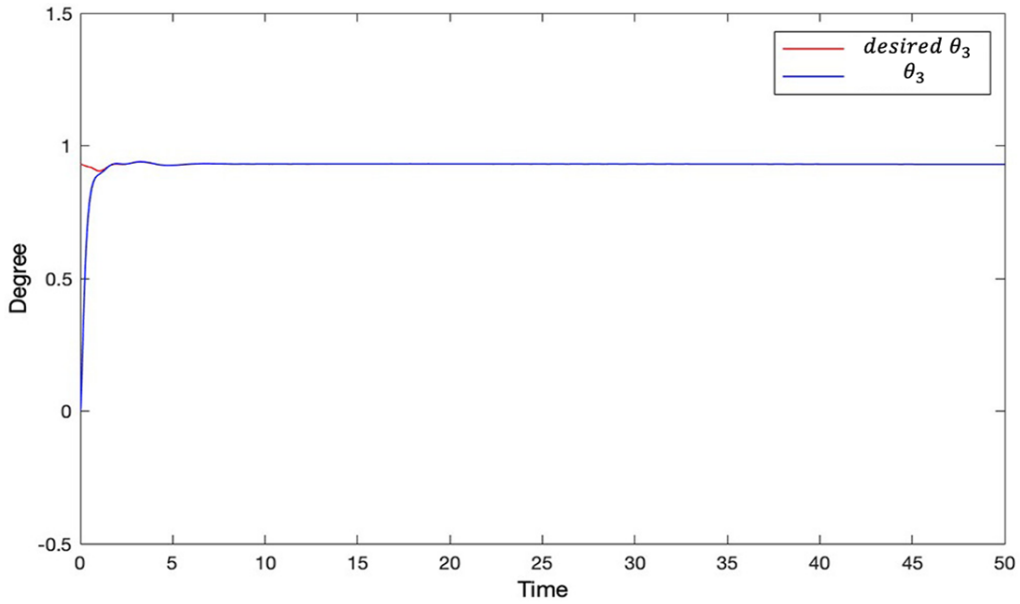


Figure 2.15: θ_3 of the gimbal mechanism

2.5 Conclusion

In conclusion, the presented results in Figures 2.9, 2.10, 2.11, 2.12, 2.13, 2.14 and 2.15 demonstrate the impressive tracking capabilities of both the quadcopter and the gimbal mechanism.

Analyzing the quadcopter's performance tracking the target object along the X and Y directions reveals an initial tracking error. Despite the challenges posed by system dynamics, external disturbances, and measurement inaccuracies, the quadcopter showcases adaptability and refinement in its tracking mechanism. This progressive reduction in tracking error highlights the effectiveness of the quadcopter's control algorithms, and feedback control mechanisms, making it suitable for various applications that require reliable and accurate tracking capabilities.

Furthermore, the results of the gimbal mechanism demonstrate its crucial role in maintaining consistent alignment with the target object. The mechanism exhibits precise adjustments and accurately tracks the target object's movements in both the position and angular domains. This ensures that the extruder and camera mounted on the gimbal mechanism remain focused on the target object, unaffected by the quad-

copter's motions and vibrations.

These results affirm the quadcopter's capability to track and follow a target object, reinforcing its viability and suitability for applications that demand reliable and accurate tracking capabilities. The exceptional performance of the gimbal mechanism further enhances the system's tracking capabilities, ensuring reliable and stable tracking of the target object's position and orientation.

In summary, the presented figures provide tangible evidence of the quadcopter's ability to minimize tracking errors and the gimbal mechanism's proficiency in maintaining accurate alignment. This combination of capabilities enables the system to deliver dependable and precise visual tracking, making it valuable for various real-world applications.

CHAPTER 3

CAMERA BASED MULTI QUADCOPTER SIMULATION

3.1 Introduction

The emergence of UAVs has opened new possibilities for 3D printing in unconventional environments and at larger scales. The mobility and versatility of UAVs as printing platforms create opportunities in various fields, including construction, disaster relief, infrastructure repair, and aerial art installations. Being able to fabricate objects in challenging or inaccessible environments holds significant potential in remote areas, regions affected by natural disasters, and even extraterrestrial missions.
Innovative approaches to explore this challenge is integrating an extruder mounted on a gimbal system that utilizes a camera to track and follow a predefined path for 3D printing. By mounting the extruder on a gimbal system, which provides rotational freedom, and employing a camera-based path-following technique, the printer can adapt to the surface geometry and maintain precise control over the printing process. Real-time visual feedback captured by the camera enables the system to adjust the gimbal's orientation accurately, ensuring that the desired path is faithfully followed.
Path-following algorithms are crucial in enabling autonomous flight and navigation for quadcopters. These algorithms leverage the visual feedback from a camera mounted on the quadcopter's gimbal to detect and track lines or paths with exceptional precision. Integrating a gimbal system enhances stability by compensating for the quadcopter's motion, vibrations, and external disturbances, ensuring the camera maintains a consistent view of the line throughout the flight. The precise control provided by the gimbal allows the quadcopter to closely follow the desired path and achieve accurate line deposition during 3D printing operations.
Simulating a quadcopter equipped with a gimbal and camera-based path-following

involves replicating the intricate interactions among the UAV, gimbal, camera, and environment. Realistic simulations can be achieved by accurately modeling the quadcopter's dynamics, the gimbal's control mechanisms, and the algorithms that process the camera's visual feedback and enable path-following. By leveraging simulations, potential issues and optimizations can be identified early on, saving time and resources in the development process.

This chapter delves into the intricacies and implications of simulating a quadcopter with a camera mounted on a gimbal to test line-following algorithms, and the simulation of the multi-UAV for tracking the line under the UAV is done. This line is generated as a path that UAV will be doing 3D printing on. In Section 3.2, the methodology of the line-following is explained. Several image processing methods are used in the simulation presented in Section 3.3. These methods are tested in the simulation part described in theSsection 3.4. In this section, the environment of the simulation part is explained. In Section 3.5, the result of the simulation of the line tracking with described image processing methods is discussed. And finally, the conclusions of this chapter are presented in Section 3.6.

3.2 Controlling of the Gimbal Using Camera

One of the main functions in this thesis is camera-based gimbal control. Using data and computer vision techniques, precise control and stabilization of the gimbal are provided, resulting in smooth and precise camera motions throughout quadcopter flight. Visual servoing is one technique for gimbal control using camera input. A closed-loop control method called visual servoing uses the camera's visual data to change the gimbal's orientation instantly. The system can determine the necessary corrections to maintain a specified camera attitude by evaluating the images and extracting pertinent characteristics or markers. The gimbal can now continuously track and stabilize the camera thanks to these modifications, which are converted into commands that turn on its motors. Furthermore, the control system can incorporate sophisticated computer vision algorithms like object tracking or motion estimation. These algorithms allow the gimbal to track objects independently or keep a constant reference point, even in chaotic situations. For instance, the gimbal can be set up to

follow a moving target or concentrate on a specific object in the camera's field of view.

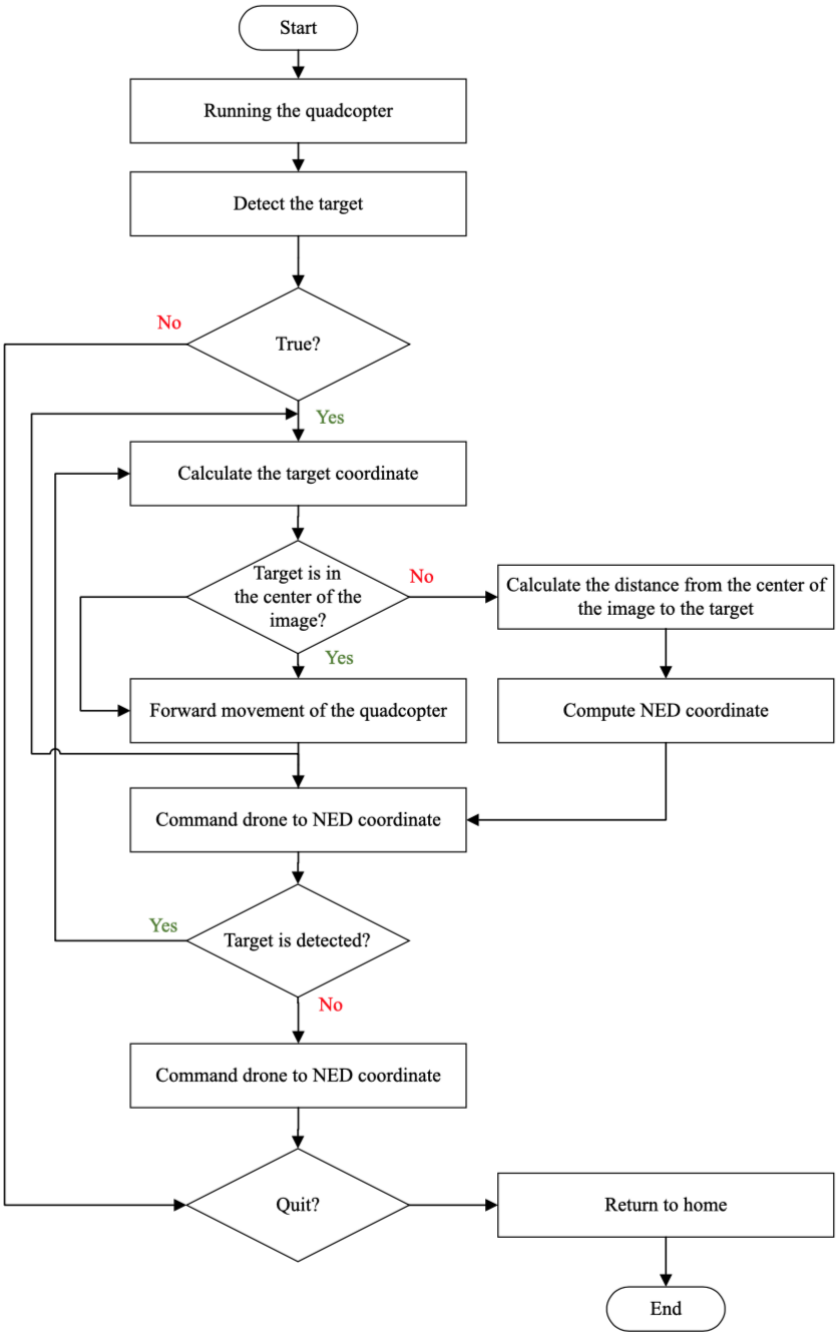


Figure 3.1: The flow diagram for the algorithm

Incorporating camera-based control techniques into the test platform ensures accurate and responsive gimbal control, resulting in smooth and stabilized camera footage.

With these capabilities, the gimbal's performance in precisely tracking objects, preserving desirable camera angles, and correcting outside disturbances like quadcopter motion or wind gusts can be assessed and demonstrated. The flow diagram for the core algorithm is shown in Figure 3.1.
First, the camera is configured, and a connection with the drone is established. Afterward, the drone's default coordinates are set.
This coordinate system uses cameras and image libraries to represent pixel coordinates in digital images. Its origin is in the top left corner of the picture. Meanwhile, the drone is controlled using the NED (North, East, Down) coordinate system. It is simple to convert between Cartesian and NED coordinate systems. Therefore, after receiving the camera image coordinates of the target object, these coordinates are converted to Cartesian coordinates. Pixel units are used for the camera image and Cartesian coordinates. The target point's location is relative to a "central rectangle" region created around the image frame's center and must be established after it is obtained in Cartesian coordinates. The target must be followed (that is, it must be re-centered in the image frame). If it is beyond the center rectangle, a new set of NED coordinates from the current Cartesian coordinates is calculated. The target is already centered if it is within the center rectangle, and the previously calculated NED coordinates will stay current. The image's origin is is in the top left corner of the image (3.2). Meanwhile, the drone is controlled using coordinates in the NED (North, East, Down) coordinate system.

The following steps are done to change the coordinate system and image processing steps:
Step 1: Storing RGB pixel values in a multidimensional array
$image_array = capture_image()$
Step 2: Conversion from the camera image to Cartesian coordinate system and Conversion from Cartesian to NED coordinate system
$x_center_position = (height/2) - 1$
$y_center_position = (width/2) - 1$
$x_cartesian = target_width - y_center_position$
$y_cartesian = x_center_position - target_height$
$East = x_cartesian * coordinate_system_conversion_factor$

$North = y_cartesian * coordinate_system_conversion_factor$

$Down = -hovering_altitude$

Step 3: Applying Gaussian Blurring to reduce noise

$blurred_image = gaussian_blur(image_array)$

Step 4: Converting image from RGB to HSV color space

$hsv_image = convert_to_hsv(blurred_image)$

Step 5: Thresholding to create a binary image

$binary_image = threshold(hsv_image)$

Step 6: Applying morphological operations for noise reduction

$processed_image = morphological_operations(binary_image)$

Step 7: Finding contour with the largest area and calculating enclosing circle radius

$largest_contour = find_largest_contour(processed_image)$

$radius = calculate_enclosing_circle_radius(largest_contour)$

Step 8: Applying Low-Pass Moving Average filter to centroid coordinates

$filtered_centroid = apply_low_pass_filter(centroid_coordinates)$

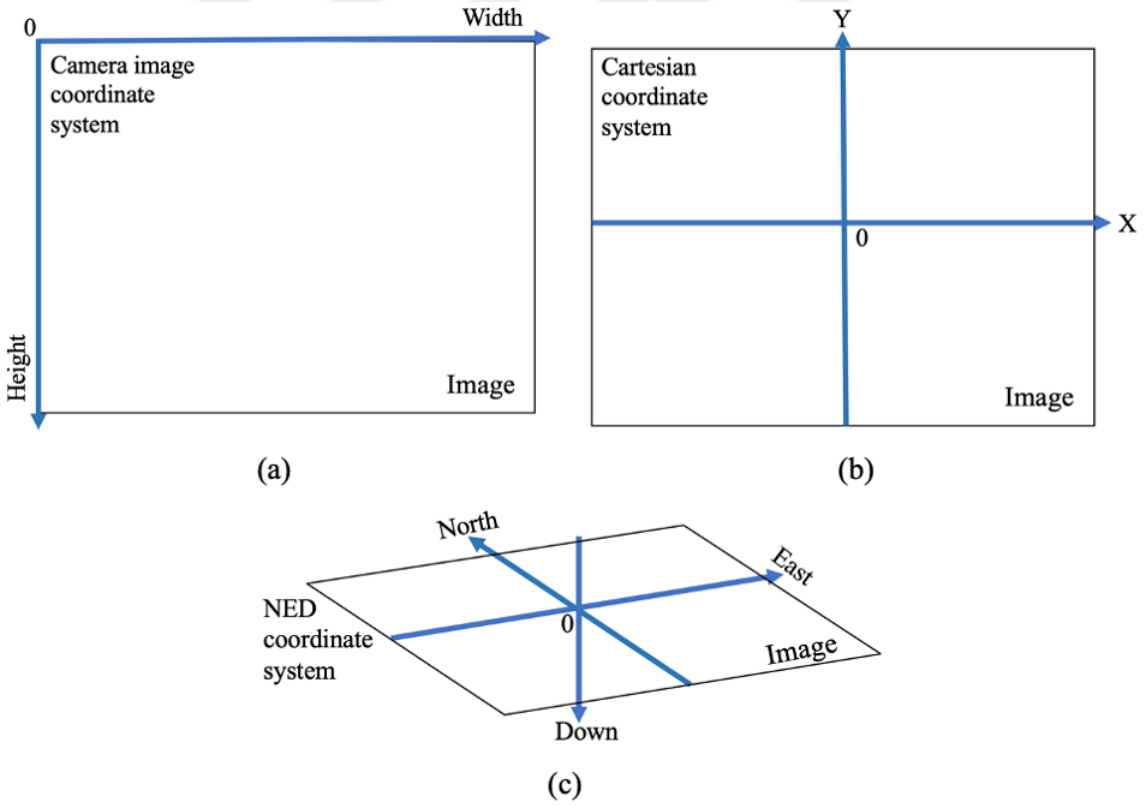


Figure 3.2: Coordinate system of the camera image, Cartesian and NED

3.3 Image Processing

All line-following algorithms aim to accurately identify and determine the center of the lane and subsequently guide the vehicle in that direction [81]. Typically, a two-step process is employed to achieve this goal. In the first step, the widely used OpenCV library calculates the center point of the largest contour detected in the captured image. OpenCV provides a robust set of image processing and computer vision algorithms that enable precise contour detection and analysis. By identifying the center point of the largest contour, the algorithm obtains a crucial reference point for guiding the vehicle along the desired path.
These calculations consider the distance between the center point and the vehicle's current position, the desired heading angle, and the control system dynamics. By converting the position information into rotation rates, the algorithm enables the quadcopter to adjust its steering and maintain alignment with the center of the lane.

3.3.1 Moment method

The Image Moment is a powerful mathematical concept used in image processing and analysis. It represents a precisely weighted average of pixel intensities within an image, enabling the extraction of valuable information about specific properties and characteristics of the image. By leveraging the Image Moment, various attributes such as the image's radius, area, centroid, and similar features can be determined.
It is common practice to convert the image into a binary format to utilize the Image Moment for centroid calculation. This conversion simplifies the analysis by representing the image as a collection of binary values (typically 0 for the background and 1 for the foreground). Once the image is in binary format, the centroid can be determined by locating its center. The centroid of an image can be computed using the following equation:

$$C_x = \frac{M_{10}}{M_{00}} \tag{3.1}$$

$$C_y = \frac{M_{01}}{M_{00}} \tag{3.2}$$

C_x and C_y represent the centroid's x and y coordinates, respectively. M_{10}, M_{01}, and M_{00} are the specific moments of the image obtained through calculations involving

the image's pixel intensities and their respective positions.
By applying this equation, the centroid of an image can be accurately determined, representing the geometric center or average position of the image's foreground pixels. This information can be utilized for various purposes in image processing applications, such as object tracking, image alignment, or feature extraction. The Image Moment and centroid calculation play a vital role in quantitative image analysis and provide valuable insights into the spatial distribution and characteristics of objects within an image.

3.3.2 Contour Line Detection

The identification and localization of contours within imagery pose significant challenges for computer vision systems, as contours serve as crucial descriptors of objects. Accurate contour detection is essential for various functional tasks, including object recognition and scene comprehension. Recognizing the importance of this problem, numerous researchers and academics have dedicated extensive time and effort to investigate and advance contour detection techniques, leading to substantial progress in the field [82]. Contour detection is fundamental in computer vision, enabling systems to extract meaningful information from visual data. Detecting contours involves identifying the boundaries or edges that outline distinct objects or regions within an image. These contours represent characteristics that encode the objects' basic shape and structural details.
In conclusion, the challenges associated with contour detection have attracted significant attention from the research community, leading to notable progress in the field. Ongoing efforts in this area continue to enhance the capabilities of computer vision systems, enabling them to achieve more sophisticated and reliable object recognition and scene comprehension tasks.

3.3.3 Hough Detection

Detecting lines within an image of contours entails identifying alignments of points that form linear structures. Among various methods available, the Hough transform

[83] stands out as the most commonly employed technique for line detection. The Hough transform utilizes a mathematical approach to analyze the image by transforming it from one spatial domain to another, often referred to as the Hough domain or parameter space.

The transformation from the spatial domain to the Hough domain is typically accomplished through a technique like the Fourier transform. This transformation allows the representation of the image information differently, highlighting the desired features for line detection. In the Hough domain, lines within the original image are converted into distinct points, creating a representation that simplifies line detection.

In summary, the Hough transform is a powerful technique for line detection in images of contours. By transforming the image from the spatial domain to the Hough domain, lines are represented as points, enabling the identification of line parameters. The Hough transform's versatility and robustness make it a popular choice in many domains, facilitating accurate line detection and enhancing the capabilities of computer vision systems.

3.4 Simulation Environment

The rapid development of software and applications for UAVs and the increasing number of UAV productions led to increased demand for simulation. Simulators for flying drones are widely available (Table 3.1).

Table 3.1: Open-Source Simulator

Platform	Implementation Language	Supported Operation System
AirSim	C++	Windows,Linux
Gazebo	C++	Linux,MacOS
Morse	Python	Linux
jMAVSim	Java	Windows,Linux,MacOS
New Paparazzi Simulator	C	Linux,MacOS
HackflightSim	C++	Windows,Linux

These frameworks offer a basis in physics, and sensor simulation makes it possible to further the development of robotic systems. In this study, Gazebo simulator with the Robot Operating System (ROS) are chosen to serve as the model for the algorithms and advancements. Because this simulator is able to simulate multi-UAV quadrotors using the MAVLink on ROS (MAVROS) communication protocol. MAVROS is the communication protocol that Pixhawk employs. Pixhawk is chosen to function as an autopilot within this system. The software and firmware supported by the Pixhawk hardware are PX4 and Ardupilot, respectively. The Gazebo simulator, ROS, and the Software-in-the-Loop (SITL) requirements can be met based on PX4.
Figure 3.3 visually represents the heading of the quadcopter and its position relative to the road in the Gazebo simulator.

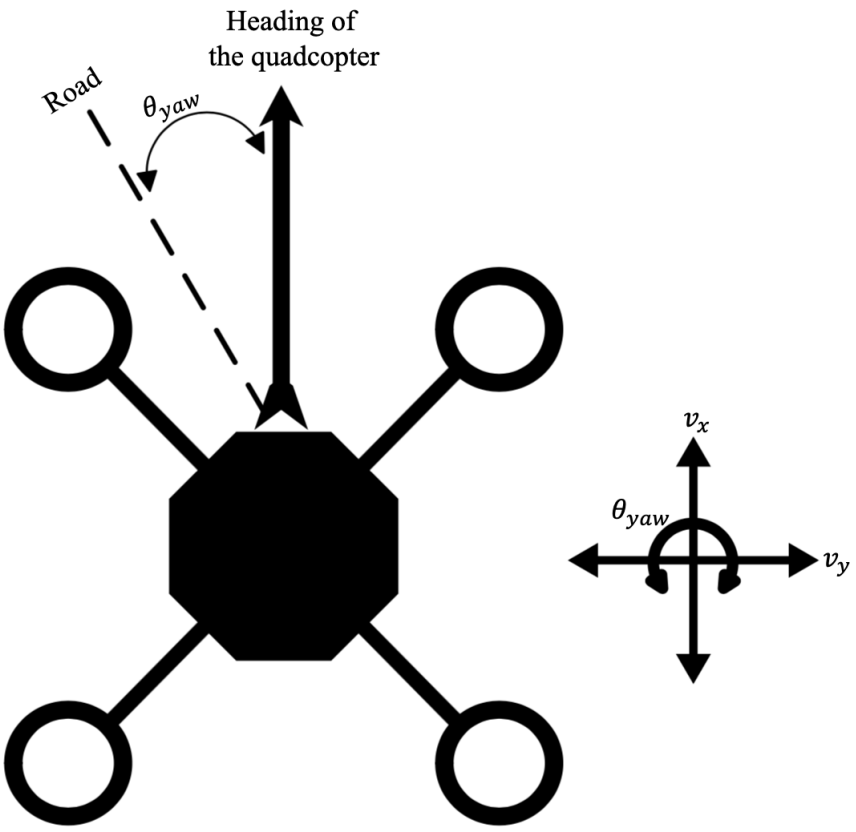


Figure 3.3: Heading of the quadcopter

To facilitate the simulation of line tracking, the employed approach revolves around an algorithm labeled as Algorithm 1. This algorithm systematically coordinates op-

erations that propel the line-tracking procedure forward. By strictly adhering to the tenets of this algorithmic framework, the seamless execution of the line-tracking simulation within the Gazebo simulator is realized.

Algorithm 1 Line Tracking Simulation

Initialize PID controller parameters
Initialize quadcopter's velocity and angular velocity
Divide captured image into three distinct parts {See Figure 3.4}
for each region of interest **do**
    Calculate center point of the road within the region
    Calculate $angle$ using:

$$angle = \arctan\left(\frac{y_{bot} - y_{min}}{x_{mid} - x_{bot}}\right)$$

    if $angle < 0$ **then**
        $angle = -(90 + angle)$
    else
        $angle = 90 - angle$
    end if
    Calculate linear displacement using:

$$linear\ displacement = x_{bot} - x_{center\ of\ image}$$

end for
Move quadcopter with constant velocity in x-direction
for each time step **do**
    Obtain error signals from previous step
    Calculate desired y-direction speed and yaw angular velocity using PID controller
    Set quadcopter's velocity in y-direction and angular velocity in z-direction
end for

Throughout the simulation, the quadcopter's position and heading undergo continuous updates derived from a combination of factors, including the analysis of captured images, the computation of error values, and the dynamic adjustments of a PID controller. This cohesive interplay guarantees the quadcopter's steadfast alignment with the road, resulting in precision and efficiency in its line-tracking capabilities. This

algorithm-driven methodology thus enables the quadcopter to navigate its course with exceptional accuracy and adeptness.

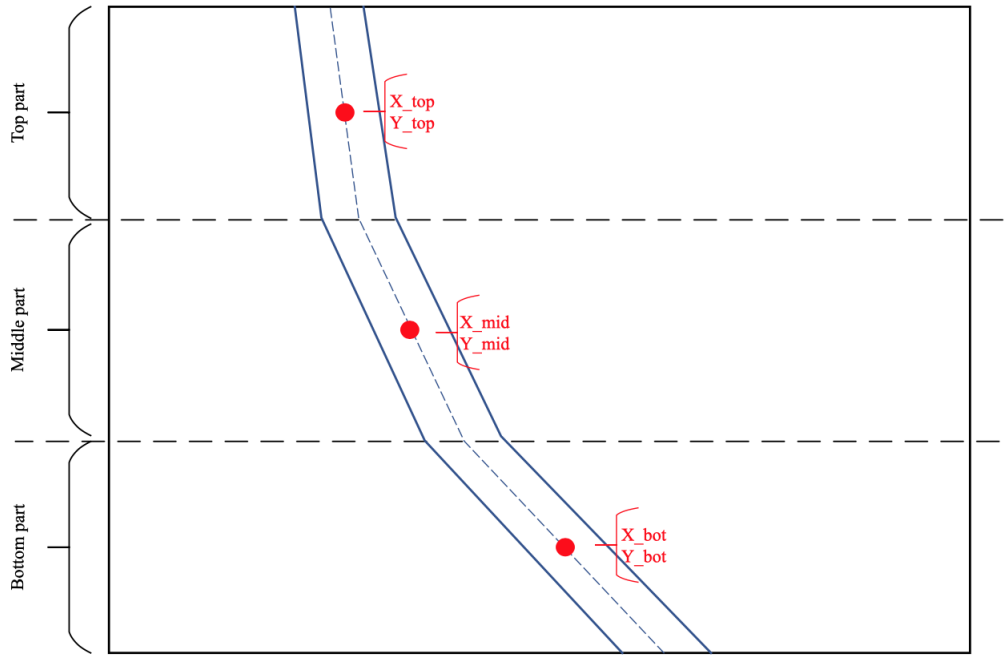


Figure 3.4: Road position in the image of gazebo

To test the algorithm, simulation involving three quadcopters is utilized. Figure 3.5 illustrates the ROS node's architecture flow chart. Each quadcopter is equipped with two cameras, serving different purposes. One camera is mounted in front of the quadcopter, while the other is positioned underneath. In the context of testing the line tracking methods detailed in Section 3.3, the camera situated under the quadcopter is employed."To comprehensively assess the algorithm's effectiveness, a comprehensive testing process involves the integration of three distinct quadcopters into the simulation framework. The architectural flow of the ROS nodes responsible for orchestrating this simulation is shown in Figure 3.5, offering a visual representation of the interconnected components.

Each quadcopter within the simulation is outfitted with a dual-camera configuration, each camera serving distinct functions to enhance the depth and breadth of data capture. One of the cameras is positioned meticulously positioned at the front of the quadcopter, while the second finds its location underneath the vehicle. In assessing

the viability of the line tracking methodologies elaborated upon in Section 3.3, the camera beneath the quadcopter is harnessed for data acquisition.
This simulation thus provides a comprehensive and controlled environment to rigorously evaluate the algorithm's performance, leveraging multiple quadcopters and strategically positioned cameras to yield a holistic assessment of its capabilities and adaptability in diverse scenarios.

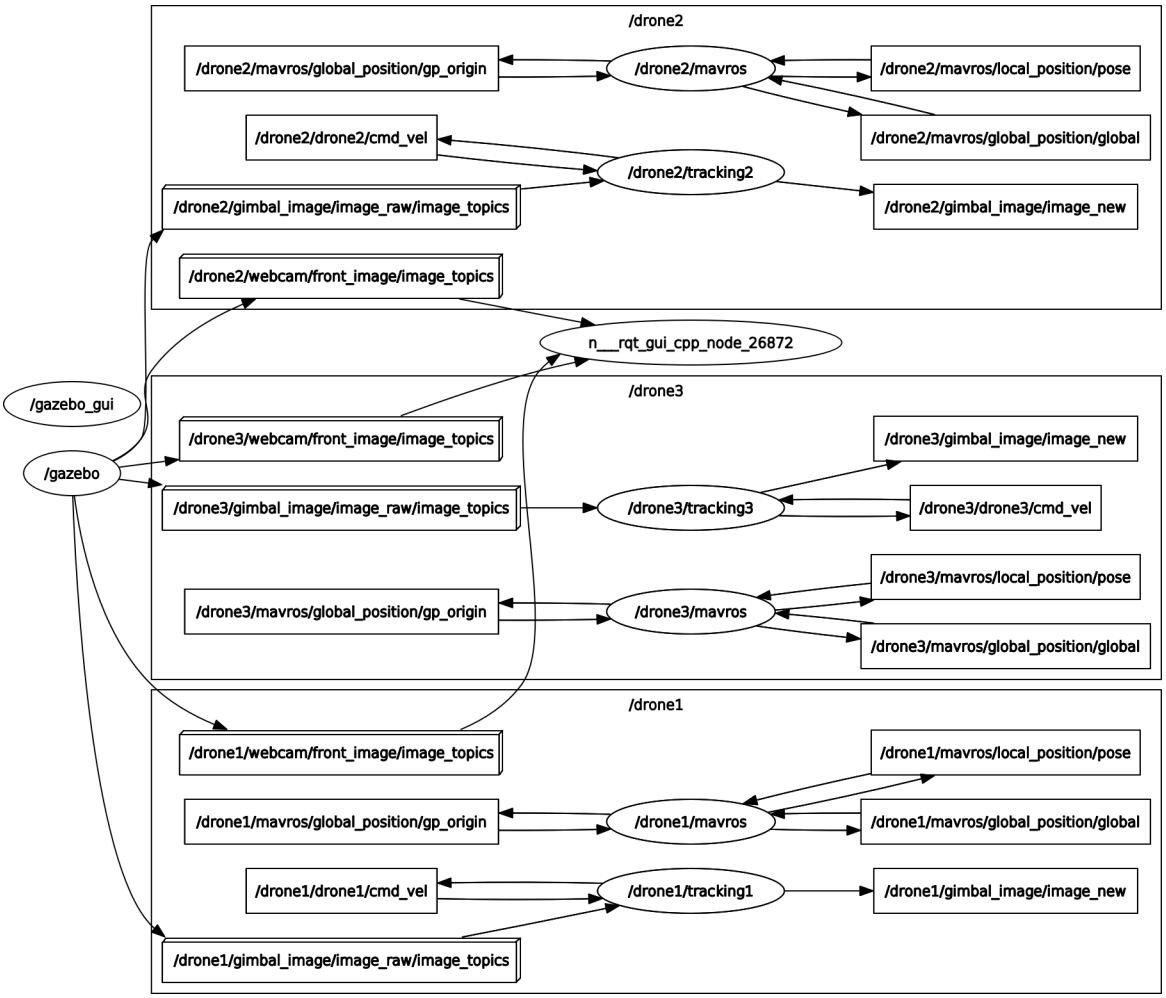
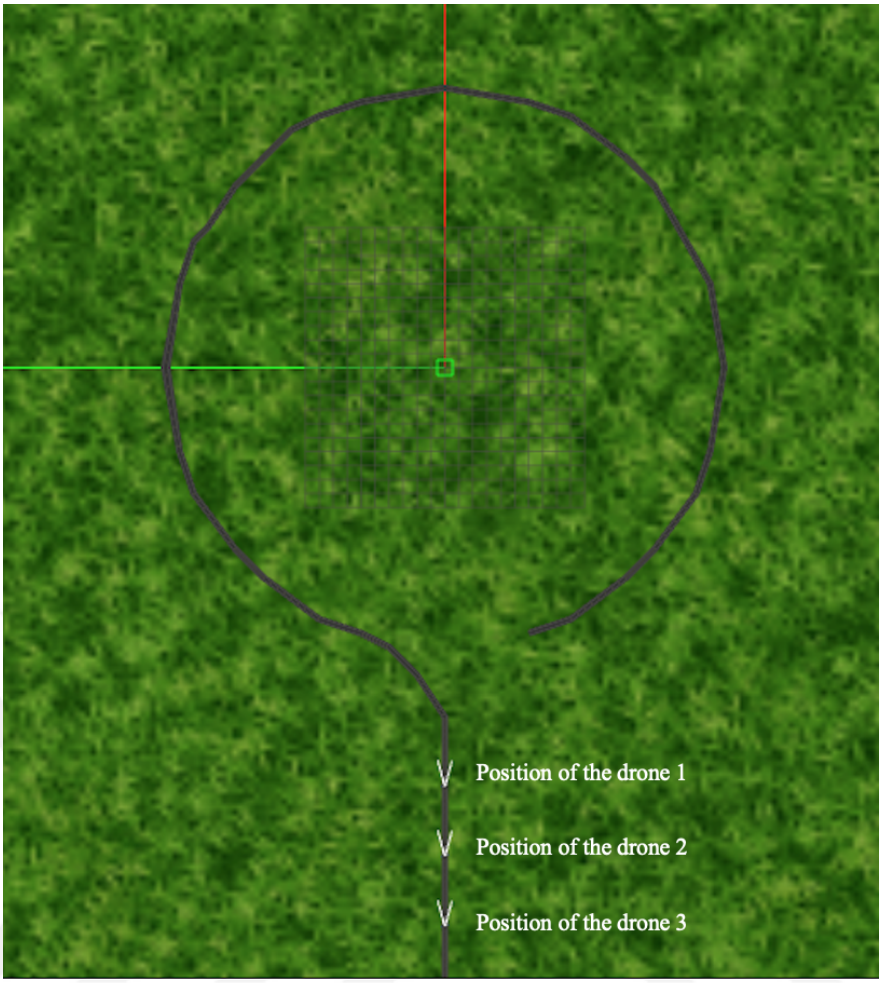
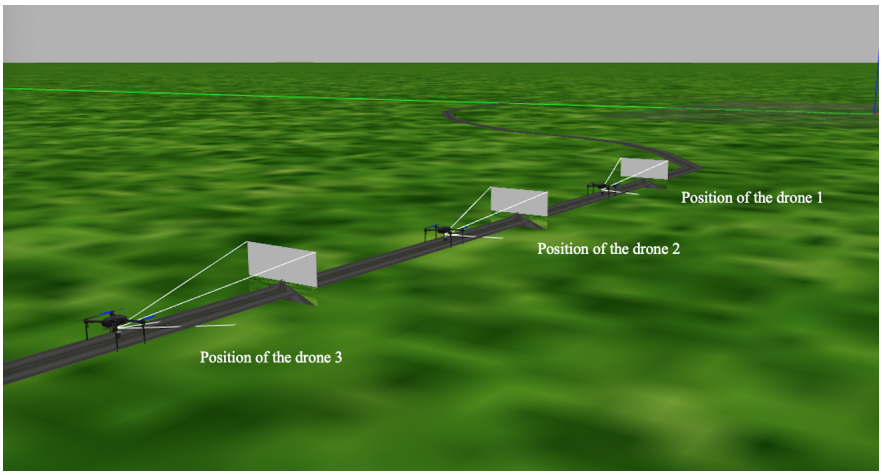


Figure 3.5: ROS Graph for multi-drone simulation

A circular road is generated within the Gazebo simulator to evaluate the line-tracking algorithms. Figure 3.6 visually encapsulates the landscape of this Gazebo world, showcasing the intricate details of the circular road and its surrounding elements.



(a) Gazebo world of the simulation environment



(b) Position of the drones in the simulation

Figure 3.6: Gazebo world for testing the multi-drone simulation

This circular road is depicted in Figure 3.6a. It provides a controlled and well-defined track for the quadcopters to navigate and perform line-tracking experiments. Figure 3.6b presents the quadcopters' positions within the Gazebo simulator, visually representing their spatial arrangement during the simulation tests. This enables observation and analysis of the quadcopters' movements and their interaction with the circular road.

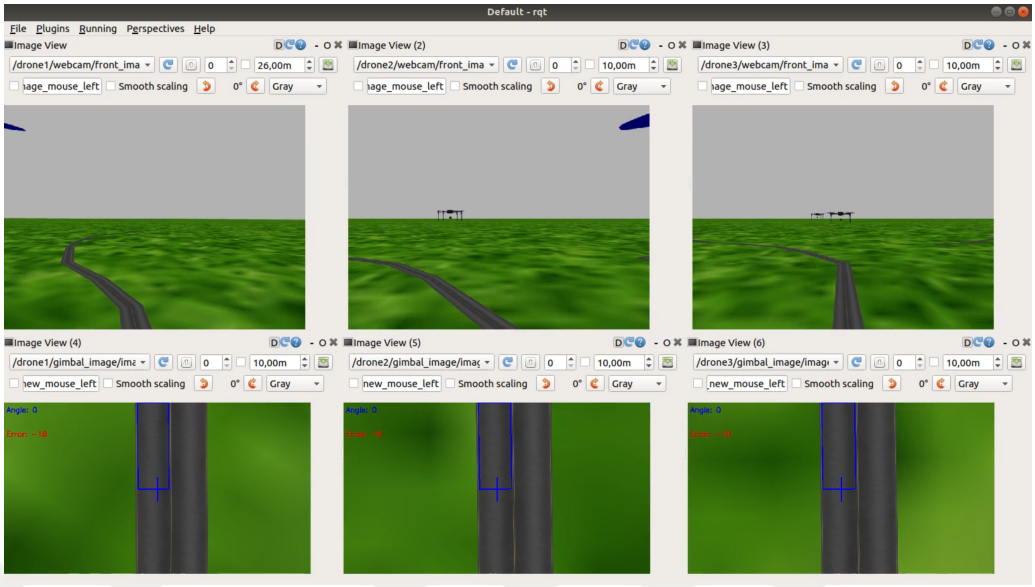
Using this setup, the simulation tests can be conducted effectively, leveraging the ROS framework, the specified camera configurations, and the carefully designed Gazebo environment. These components collectively contribute to the successful evaluation of the line tracking methods and enable the assessment of the quadcopters' performance in accurately following the circular road.

3.5 Results of the Simulation

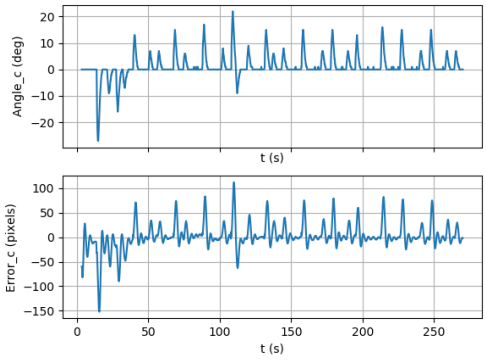
This section presents a comprehensive analysis of the outcomes obtained by implementing several line-tracking methods, precisely the contour, Hough, and moment techniques. The results obtained from applying these techniques are visualized and presented in Figure 3.7a, 3.8a, and 3.9a, offering a detailed assessment of their performance.

The figures showcase the imagery captured by each quadcopter's front and below cameras. These figures are organized in a top-row/bottom-row format, where the top row displays the front cameras of all quadcopters, while the bottom row presents the cameras mounted on the gimbal beneath the quadcopters. Each figure clearly illustrates the specific line-tracking method employed for analysis.

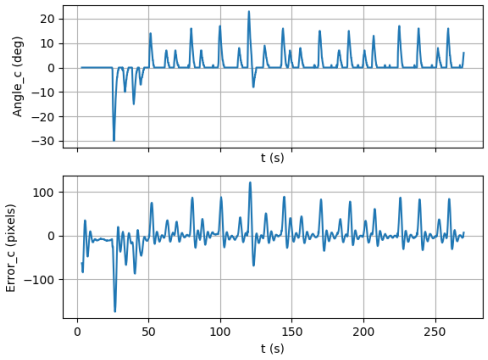
To comprehensively evaluate the performance of the implemented line-tracking methods, for contour method, the figures presented in Figure 3.7b, 3.7c, and 3.7d; for hough method, the figures presented in Figure 3.8b, 3.8c, and 3.8d; and for moment method, the figures presented in Figure 3.9b, 3.9c, and 3.9d; showcase the results of the error analysis in translational and rotational movements for all three quadcopters. These figures provide a detailed overview of the error patterns and fluctuations observed throughout the simulation, offering valuable insights into the accuracy and stability of each line-tracking technique.



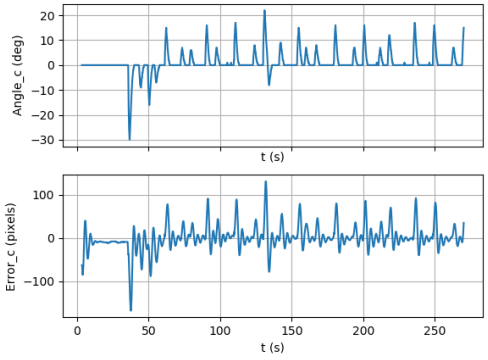
(a) Multi drone simulation using contour method



(b) Contour method for drone1

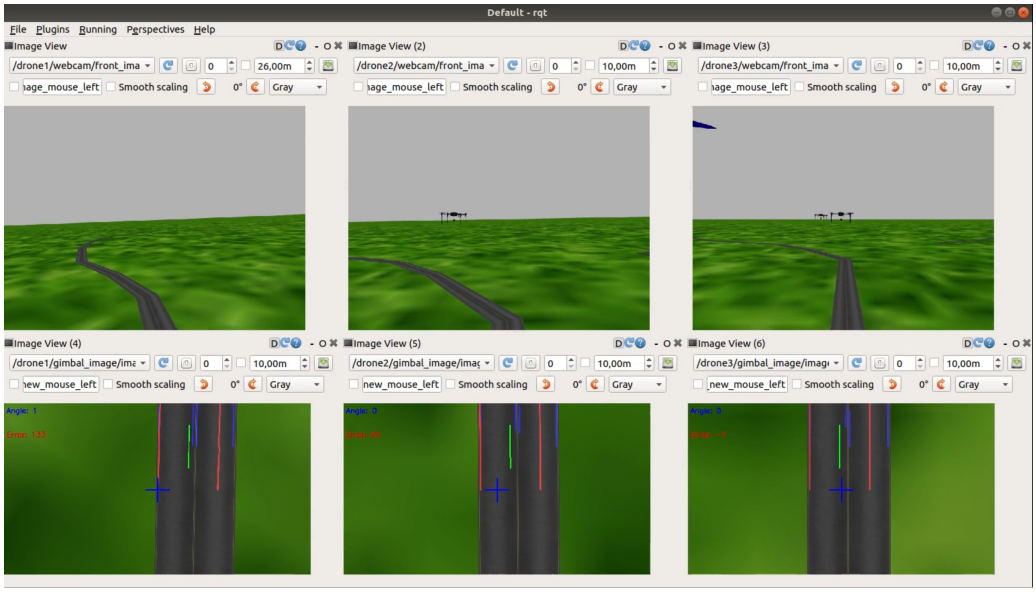


(c) Contour method for drone2

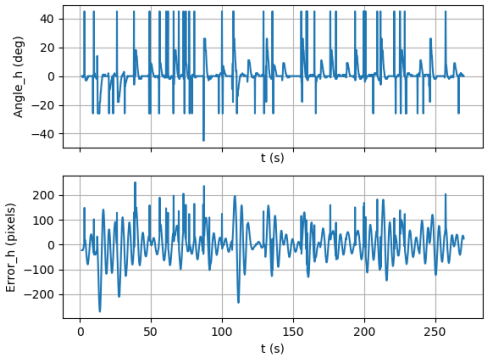


(d) Contour method for drone3

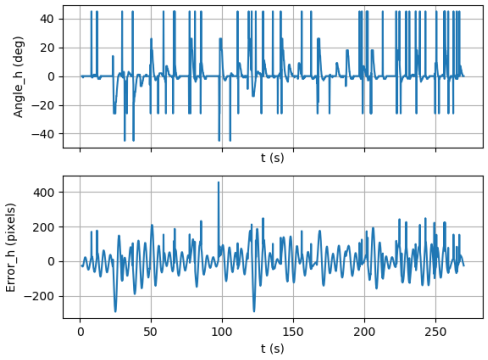
Figure 3.7: Contour method



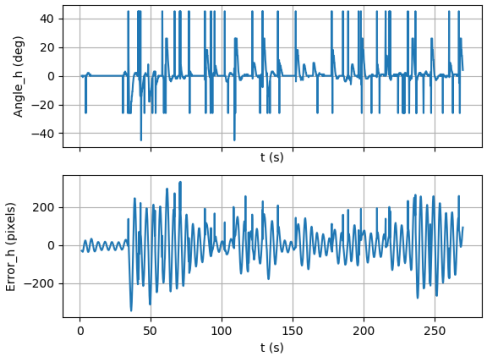
(a) Multi drone simulation using hough method



(b) Hough method for drone1

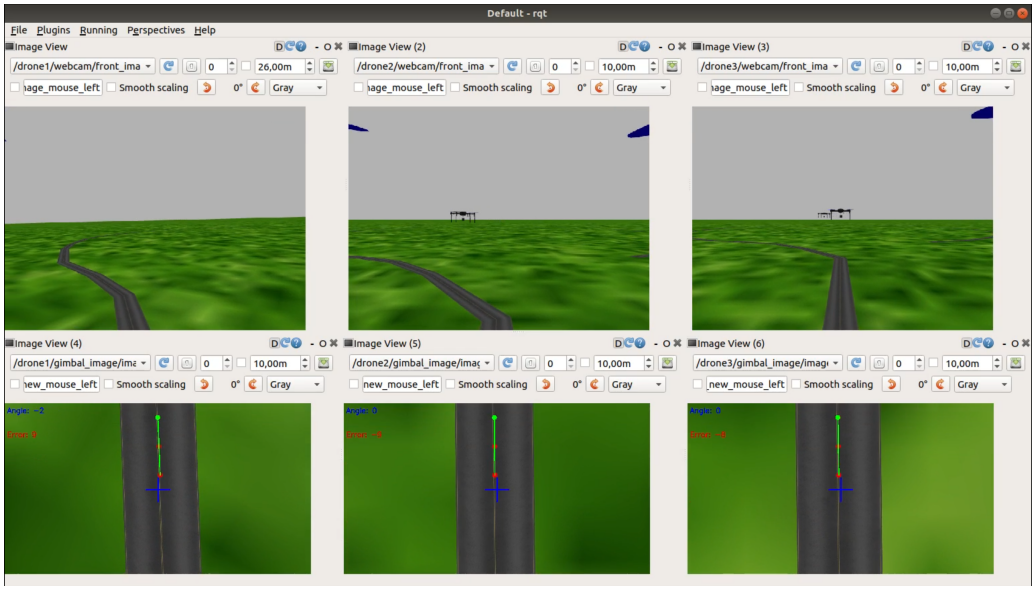


(c) Hough method for drone2

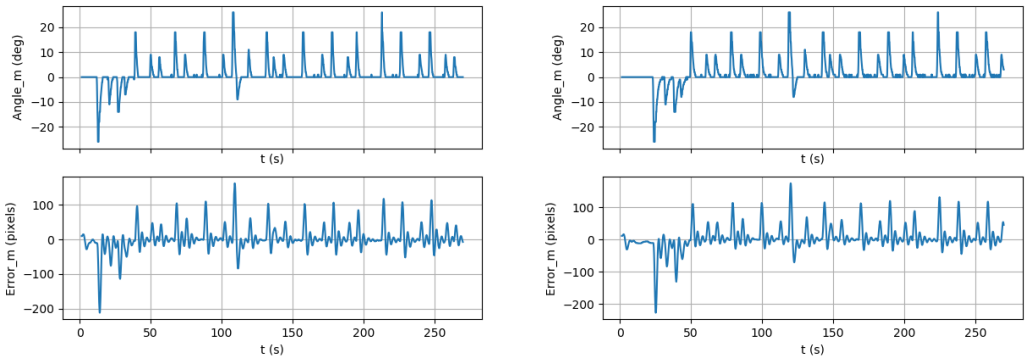


(d) Hough method for drone3

Figure 3.8: Hough method

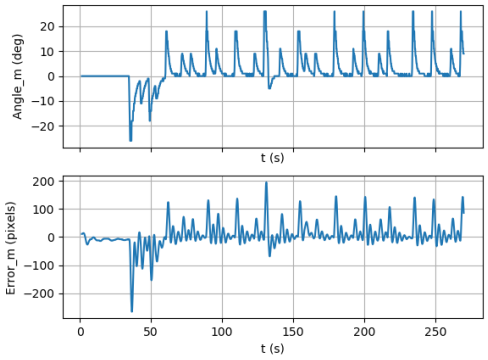


(a) Multi drone simulation using moment method



(b) Moment method for drone1

(c) Moment method for drone2



(d) Moment method for drone3

Figure 3.9: Moment method

To further supplement the visual representations, Table 3.2, 3.3, and 3.4 present the mathematical results obtained from the analysis of each quadcopter.

Table 3.2: Result of the Multi Drone Simulation using Contour Method

	Drone 1	Drone 2	Drone 3
Mean Error of the Translational Movement	2.07	1.46	1.23
Standard Deviation Error of the Translational Movement	23.09	23.09	25.66
Mean Error of the Rotational Movement	1.17	1.12	1.10
Standard Deviation Error of the Rotational Movement	4.13	4.15	4.23
Mean Absolute Error of the Translational Movement	21.39	20.12	22.76
Mean Absolute Error of the Rotational Movement	0.00	0.01	0.02

Table 3.3: Result of the Multi Drone Simulation using Hough Method

	Drone 1	Drone 2	Drone 3
Mean Error of the Translational Movement	2.35	1.93	1.41
Standard Deviation Error of the Translational Movement	60.17	61.11	6830
Mean Error of the Rotational Movement	1.50	1.51	1.47
Standard Deviation Error of the Rotational Movement	8.72	9.11	9.30
Mean Absolute Error of the Translational Movement	24.84	34.68	33.48
Mean Absolute Error of the Rotational Movement	0.00	0.59	0.58

Table 3.4: Result of the Multi Drone Simulation using Moment Method

	Drone 1	Drone 2	Drone 3
Mean Error of the Translational Movement	5.33	6.89	9.24
Standard Deviation Error of the Translational Movement	31.40	34.08	38.89
Mean Error of the Rotational Movement	1.26	1.56	2.14
Standard Deviation Error of the Rotational Movement	4.68	5.14	5.96
Mean Absolute Error of the Translational Movement	12.43	12.56	20.05
Mean Absolute Error of the Rotational Movement	0.00	0.08	0.43

These tables provide comprehensive information regarding the mean and standard deviation of each drone's translational and rotational movements during the simulation. The quantitative analysis presented in these tables allows for a more precise assessment of the performance and consistency of the line-tracking methods.

The combination of visual representations, error analysis figures, and quantitative results offered by the tables collectively enables a comprehensive evaluation of the line-tracking techniques utilized in this study. These evaluation components provide a solid foundation for assessing the effectiveness, accuracy, and reliability of the contour, Hough, and moment methods in accurately tracking lines. Moreover, they offer insights into the quantifiable errors associated with each method and provide a holistic understanding of the performance of the quadcopters during the simulation.

3.6 Conclusion

In this chapter, the integration of a camera-based gimbal control system for precise and stabilized camera motions during quadcopter flight was explored. The utilization of visual servoing and computer vision techniques to control the gimbal's orientation based on real-time visual feedback was discussed. The camera-based control techniques enhance the gimbal's functionality and enable it to adapt to various visual inputs, leading to accurate and responsive gimbal control.

In conclusion, the simulation environment is crucial in developing and evaluating UAV systems. The availability of open-source simulators provides a foundation for physics-based simulations and sensor simulations, enabling the advancement of robotic systems. This study chose Gazebo as the simulator due to its compatibility with the Pixhawk hardware, the PX4 and Ardupilot software and firmware, and the MAVROS communication protocol.

The line-tracking simulation in the Gazebo simulator used a well-defined methodology involving image processing and control algorithms. The captured image from the quadcopter's camera was divided into three regions of interest corresponding to the road. The error in linear displacement and angular rotation was calculated based on the coordinates of specific points in the image. A PID controller was employed to regulate the quadcopter's velocities, ensuring precise alignment with the road and

accurate line tracking.

The results of the simulation tests demonstrated the performance of multiple line tracking methods, including contour, Hough, and moment techniques. Visual representations of the simulation outcomes provided a detailed assessment of the quadcopters' movements and their interaction with the circular road. Error analysis figures and quantitative results regarding the mean and standard deviation of translational and rotational movements allowed for a comprehensive evaluation of the line-tracking methods.

Based on the evaluation components, it can be concluded that the contour, Hough, and moment methods exhibited varying degrees of effectiveness, accuracy, and reliability in accurately tracking lines. Each method had strengths and weaknesses, as evidenced by the visual representations and quantitative results. The contour method showed higher mean errors in translational and rotational movements than the Hough and moment methods. However, the standard deviation errors were comparable among the three methods.

The successful implementation and evaluation of the line-tracking methods in the Gazebo simulator, along with the comprehensive analysis of the simulation results, contribute to understanding the performance and limitations of these methods in real-world scenarios. The insights gained from this study can inform further developments in UAV line-tracking algorithms and assist in designing and testing autonomous navigation systems for UAVs.

CHAPTER 4

MULTI QUADCOPTER PATH PLANNING FOR AREA COVERAGE

4.1 Introduction

In recent years, the rapid development and widespread adoption of UAVs have opened new possibilities for various applications, including surveillance, monitoring, and exploration. One of the fundamental challenges in deploying multiple UAVs is efficiently covering a given area while minimizing redundancy and maximizing exploration effectiveness. Multi-UAV CPP addresses this challenge by enabling a team of UAVs to collaborate and navigate intelligently, optimizing coverage efficiency, and achieving comprehensive exploration. The utilization of multiple UAVs offers several advantages over single-agent approaches. Firstly, it significantly reduces the overall coverage time as multiple UAVs can work concurrently, exploring different regions of the area of interest. Secondly, cooperative CPP enables seamless coordination and collaboration among the UAVs, ensuring efficient allocation of tasks, sharing of information, and avoidance of redundant coverage. Furthermore, multi-UAV CPP enhances fault tolerance, as the failure of one UAV does not halt the entire operation but can be compensated by other functioning UAVs. The field of area coverage has seen a broad range of approaches in published research, and a recent work that provides a complete examination of numerous strategies can be found at [84]. The division of the target area into a more significant number of smaller parts or cells that need to be visited and covered is a component shared by all these tactics. Choset proposed a comprehensive strategy for deconstructing cellular structures created expressly for coverage problems [85]. This technology generates convex sections inside the environment that can be navigated by moving in a back-and-forth motion between two locations. To find a solution to the coverage issue, a graph search is carried out to es-

tablish the proper sequence in which the cells should be covered. After that, the UAV moves around inside each cell in a circular motion, covering all the available space by traveling from one side to the other in the previously specified order. This approach was improved by Acar et al. [86] to function with a smooth polygonal workspace. Their technique is designed to identify shifts in the connectivity inside the workspace rather than searching for individual vertices. They improved the system by categorizing cells as either narrow or wide [87]. Wide cells enable back-and-forth motions, and the width of each coverage row can be changed depending on the sensor footprint of the UAV.

On the other hand, the UAV modifies its behavior when it enters narrow cells and then follows the generalized Voronoi diagram that corresponds to it while utilizing its sensor to cover the area. The determination of the order of cell coverage is the primary task involved in optimization using these methods. Path optimization is a crucial feature that includes picking the path for back-and-forth motions within each cell. This is only one example of how path optimization can be used. This optimization seeks to reduce the overall number of turns along the route and limit the vehicle acceleration and deceleration associated with each turn [88]. This optimization was further expanded by Li et al. [89] considering cell connections. They demonstrated that a route with fewer turns is more efficient for UAVs in terms of the length, the time it takes to complete the route, and the energy required. Optimizing the sweep direction is another strategy that has been implemented in a variety of UAV systems [90]. When this occurs, each convex polygon is given to a different UAV, which covers it with an optimum back-and-forth pattern aligned with the sweep direction. This research suggests a strategy for deploying a swarm of UAVs to efficiently and thoroughly survey large land areas. The technique considers realistic deployment factors, including the few humans responsible for monitoring the UAVs. This is accounted for by including a setup time or initialization period, which is used to prepare each UAV for takeoff. When numerous UAVs cannot be launched simultaneously due to setup times, the total time for each UAV's setup is added. This study aims to minimize the mission time for the multi-UAV coverage task while utilizing the fewest possible vehicles. The approach automatically determines the optimal routes and the appropriate number of UAVs based on the available human operators. It also considers the constraint on the number of operators, as launching too many UAVs could increase

the total mission time due to cumulative setup time.

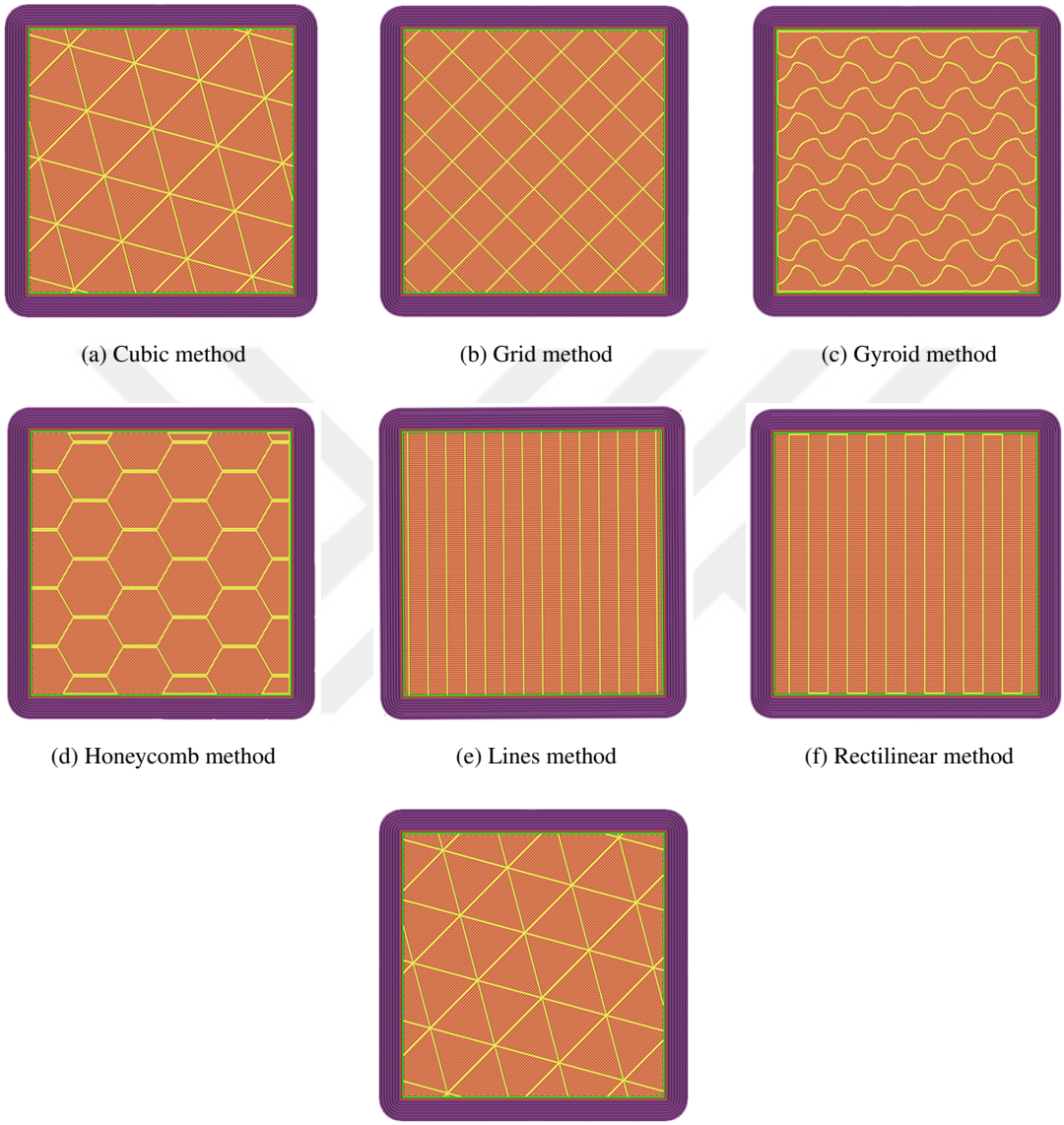
In this study, quadcopters must fly near the ground of the 3D printing area. Also, the speed of the quadcopters is slow due to the 3D printing; therefore, flight time is a crucial constraint of this study. The contributions of this chapter are as follows. Firstly, the approach incorporates practical deployment elements, such as the limited number of human operators to control the UAVs. Secondly, the approach minimizes the mission time for the multi-UAV coverage task while utilizing the fewest possible UAVs. And finally, the approach optimizes the sweep direction to reduce the number of rotations required during the operation.

Section 4.2 presents the problem statement and explains the situation in detail in this chapter. Section 4.3 outlines the process and strategy for solving the problem and gives the suggested solution. Section 4.4 displays the result of the proposed method in simulation. and finally, Section 4.5 explains the proposed method's conclusion.

4.2 Problem Definition

It is possible to adapt the infill slicing pattern to utilize multiple UAVs for 3D printing while considering CPP to meet the requirements of coordinating and collaborating with multiple UAVs. Infill slicing is a commonly used technique in 3D printing that generates the path for shaping an object. Infill refers to the internal structure of a 3D-printed item, providing support, strength, and stability. Infill slicing involves determining the pattern and paths for depositing material within the object's internal structure. The choice of infill pattern for slicing can vary depending on the specific needs of the printed object. Various infill patterns are illustrated in Figure 4.1, each with its characteristics in terms of strength, material usage, and print time. Once the infill pattern is selected, a path planning algorithm determines the paths of the 3D printer's nozzle to deposit the material. These paths aim to ensure a continuous and efficient deposition of infill material within the designated areas, guaranteeing proper adhesion and structural integrity. In multi-UAV 3D printing with CPP, the infill slicing pattern enables efficient and synchronized printing across multiple UAVs. The infill pattern is adjusted to facilitate coordinated movement of the UAVs and optimize the printing process. One approach to incorporating multi-UAV coordination is to divide

the printing area into regions or zones. Each UAV is assigned specific zones to cover, and the infill slicing pattern is customized accordingly. Planning infill lines within each zone strategically minimizes the overlap between UAVs and ensures complete coverage of the printing area.



(a) Cubic method (b) Grid method (c) Gyroid method

(d) Honeycomb method (e) Lines method (f) Rectilinear method

(g) Triangles method

Figure 4.1: Infill Pattern Types

In the context of CPP, choosing an appropriate path is crucial in ensuring efficient and compelling exploration of a target area. In this study, the CPP is approached using the rectilinear method, as depicted in Figure 4.2. This method involves selecting a path

of straight-line segments that systematically traverse the area of interest, facilitating comprehensive coverage.

A slicing shape of the square is generated to determine the spacing between the lines comprising the rectilinear path. This slicing shape divides the square into parallel lines of uniform distance, ensuring a consistent and evenly distributed coverage pattern. By calculating the distances between these lines, the rectilinear path can be precisely defined, providing a structured framework for the CPP.

The choice of the rectilinear method for the CPP offers several advantages. First, its simplicity allows for straightforward implementation and planning. Additionally, the rectilinear path facilitates efficient coverage of the target area, as it systematically traverses the region without unnecessary overlaps or gaps. Moreover, the rectilinear method enables easy integration with various applications, including mapping, surveillance, or inspection tasks.

In Section 4.3, a detailed explanation of the CPP methodology will be presented, encompassing the rectilinear path and its associated lines. The utilization of the rectilinear method and the calculated line spacing form a critical component of the methodology employed in this research. This approach enables precise planning, execution, and analysis of the coverage path, ultimately contributing to accurate data collection and evaluation of the overall coverage performance.

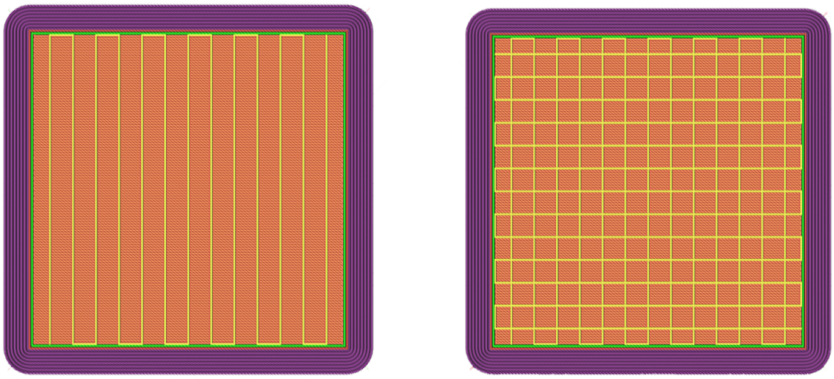


Figure 4.2: Infill patterns for the rectilinear method

The pattern is customized based on the assigned zones, communication between UAVs, and the capabilities of individual UAVs, ensuring optimal performance and successful completion of the printing process.

In this chapter, the scenario assumes a collection of UAVs, comprising k units, flying at the same altitude and covering a polygonal convex region in $\mathbb{R}^2$ space defined by a set of points P, taking off from a common base station. The essential factor to consider when determining the parameters for each operation is the subsequent waypoints, as illustrated in Figure 4.3. Each UAV's maximum flight time is fixed and known in advance. Strategies to address these challenges will be explored in Section 4.3.

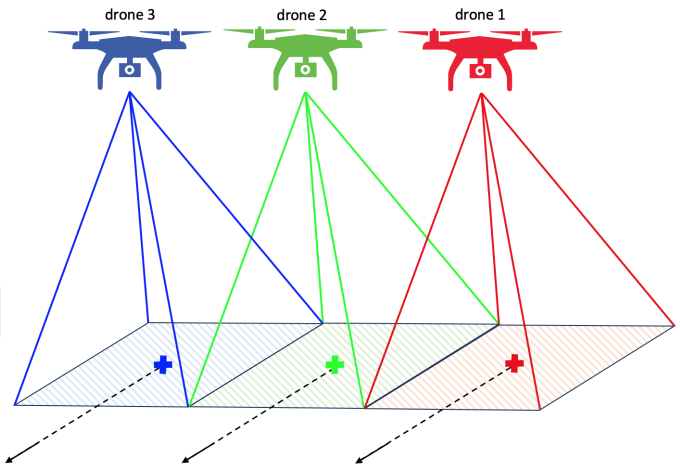


Figure 4.3: Path of the multi-drone

4.3 Methodology

The approach to addressing the previously mentioned problem consists of two main components. Firstly, a similar methodology as described in [91] (Figure 4.4) is employed to divide the region into sweeping rows. This step involves partitioning the area into rows to be covered. Secondly, inspiration is drawn from a solution to a Vehicle Routing Problem (VRP) [92]. In this context, the problem is transformed into a VRP by treating the group of customers as the endpoints of the coverage rows and the set of UAVs as vehicles. Each UAV is considered a vehicle, and each endpoint of the coverage rows is modeled as a customer to convert the problem outlined in Section 4.2 into a VRP. One UAV is permanently assigned to follow the coverage rows, as additional constraints ensure that the vehicles must utilize specific graph edges in their routes. Ultimately, solving the VRP will provide each UAV's path to traverse.

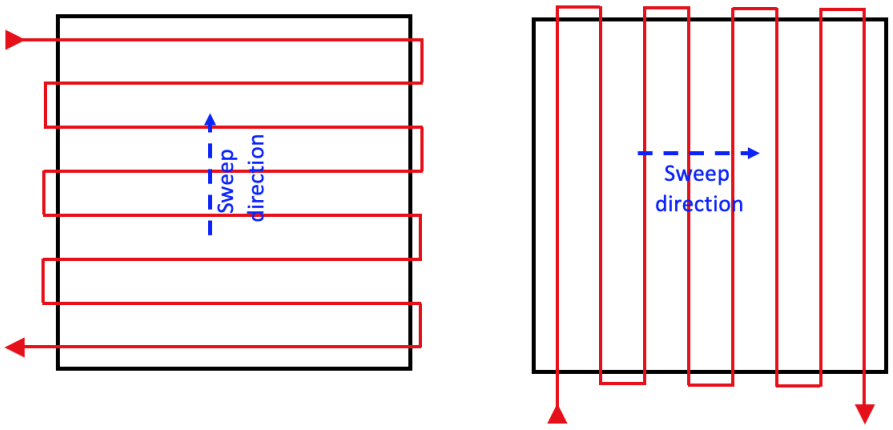


Figure 4.4: Sweep direction

Additional constraints can be introduced to ensure that at least one of the deployed UAVs adheres to the coverage rows. This is achieved by enforcing that vehicles follow specific graph edges during their paths. Ultimately, by solving the VRP, the optimal routes for each UAV can be determined. The following constraints are taken into consideration.

1. **Constraints with Time:** The objective is to minimize the required time for complete area coverage. The constraints ensure the coverage tasks are performed efficiently without unnecessary delays or redundant efforts. The time taken by UAV K to fly its path, as stated by the [93], is determined mathematically by:

$$T_k = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{C_{ij}}{V_{ij}^k} X_{ij}^k + d_k \tag{4.1}$$

where constant C_{ij} is the edge's traversal expense between nodes i and j, the binary variable $X_{ij}^k \in \{0, 1\}$ indicates whether or not the k^{th} UAV will fly from vertex i to vertex j. The k^{th} UAV's flight speed from vertex i to vertex j is represented by constant $V_{ij}^k \in \mathbb{R}$. The additional time required to launch UAV k is represented by the variable d_k.

Minimizing mission time stands as the primary goal. To achieve this, the longest path taken by all UAVs should have the shortest travel time. Consequently, the objective is to minimize the maximum T_k in a min-max problem. Introducing an additional variable ν, representing the longest UAV route, con-

verts the min-max problem into a linear one. The purpose of the optimization is minimize the value of the ν. The fundamental optimization issue can be formulated as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{C_{ij}}{V_{ij}^{k}}X_{ij}^{k}+d_{k}\leq\nu,k=1,...,M \tag{4.2}$$

$$t_{s}\left[\frac{k}{O}\right]\sum_{j=1}^{N}X_{1j}^{k}=d_{k},k=1,...,M \tag{4.3}$$

The first of these constraints is given by:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{C_{ij}}{V_{ij}^{k}}X_{ij}^{k}+d_{k}\leq L_{k},k=1,...,M \tag{4.4}$$

The flight time of UAV k is limited by its battery duration, L_k. It is assumed that the battery charge remains constant during setup or is minimally affected. Consequently, each UAV's flying time is determined by Equation (4.2) minus the setup time d_k.

Two more constraints are needed to ensure that a single UAV visits each graph node only once:

$$\sum_{k=1}^{M}\sum_{j=1}^{M}X_{1i}^{k}=m \tag{4.5}$$

$$m\leq M \tag{4.6}$$

2. **Constraints on Area Coverage:** Each UAV must cover a specific area within the overall target region. The coverage tasks of UAVs must be executed serially to ensure complete coverage of the target area. The coverage tasks must be distinct; each zone should be covered only once. Two additional constraints are required to ensure that a single UAV only ever visits each node in the graph:

$$\sum_{k=1}^{N}\sum_{i=1}^{N}X_{ij}^{k}=1,j=2,3,4,...,N \tag{4.7}$$

$$\sum_{i=1}^{N}X_{ip}^{k}-\sum_{j=1}^{N}X_{pj}^{k}=0,p=1,2,3...,N,k=1,2,3...,M \tag{4.8}$$

According to the equation (4.7), no more than one UAV can travel between nodes other than the home node (Node 1). However, equation (4.8) guarantees that the same UAV visits and leaves each node.

3. **Constraints on Routing:** The routing decisions for UAVs must be made sequentially to optimize the overall time required for coverage. The routing algorithm should consider the serial constraints to ensure that each UAV visits its designated area in a logical order. The order of UAV visits should be determined based on factors such as distance, proximity, and optimal path planning. Two possible restrictions can be added to the optimization problem to prohibit UAVs from leaving the coverage area along an edge perpendicular to the coverage rows.

$$\sum_{k=1}^{M} X_{i,i+1}^{k} = \sum_{k=1}^{M} \sum_{j=\{1,3,...\}/\{i+1\}}^{M} X_{i+1,i}^{k}, i = 2, 4, 6..., N \tag{4.9}$$

$$\sum_{k=1}^{M} X_{i,i-1}^{k} = \sum_{k=1}^{M} \sum_{j=\{1\}\cup\{2,4,...\}/\{i-1\}}^{M} X_{i-1,i}^{k}, i = 3, 5, 7..., N \tag{4.10}$$

4.4 Simulations

This section comprehensively demonstrates and evaluates the proposed multi-UAV CPP methodology using simulations. The main objective is to assess the algorithms' effectiveness, efficiency, and robustness under various scenarios and constraints. The simulation process encompasses three distinct parts, each contributing to the validation and evaluation of the proposed methodology.

The part 4.4.1 focuses on calculating the CPP while considering the critical constraints outlined in Section 4.3. This part ensures that the CPP is optimized and tailored to the specific requirements and limitations of the given scenario.

In the part 4.4.2, the proposed CPP methodology is evaluated using simulations in the Gazebo simulator, which provides a realistic virtual platform. Various scenarios are tested, considering wind speed, gusts, wind direction, and the number of visible satellites. These simulations enable us to assess the efficiency and performance of the CPP algorithm in a controlled environment. The results obtained from the simulations are analyzed and compared to validate the methodology's effectiveness.

Part 4.4.3 involves implementing the CPP methodology on three actual drones in a real-world scenario to validate further and practically evaluate the proposed algorithms. The drones are equipped with specific modifications and additions to replicate

practical conditions. Real-world tests assess the algorithms' performance and behavior under real-time constraints. The paths followed by the drones during the coverage operation are visualized and analyzed to verify the CPP algorithm's effectiveness and reliability in practical settings.

This section aims to validate the proposed multi-UAV CPP methodology, ensuring its effectiveness, feasibility, and practicality through simulations and real-world tests. The insights gained from these comprehensive evaluations contribute to the overall understanding and applicability of the CPP approach in real-world scenarios.

4.4.1 CPP Algorithm with Constraints

To visualize the path used in the simulations, Figure 4.5 illustrates the trajectory generated by the CPP algorithm. This visual representation depicts the path followed during the simulation experiments, providing a reference for further analysis and comparison. Through the combined efforts of the simulations, virtual testing, and real-world experiments, this section aims to validate the proposed methodology, ensuring its effectiveness, feasibility, and practicality. The insights gained from these comprehensive evaluations contribute to the overall understanding and applicability of the CPP approach in real-world scenarios.

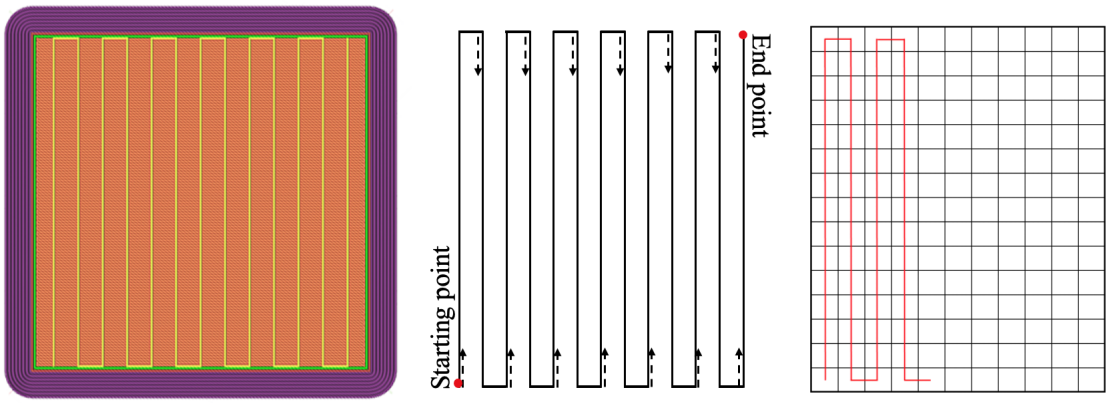


Figure 4.5: Path in the square shape

The `multi_UAV_CPP` algorithm (Algorithm 2) is a sophisticated and efficient solution designed to handle a range of inputs, such as the geographical region that needs

to be covered, the number of available UAVs, maximum flight time, and setup time for each UAV.

Algorithm 2 multi_UAV_CPP

Require: $region$: The region to be covered by UAVs

Require: num_UAVs: The number of UAVs available

Require: max_flight_time: The maximum flight time for each UAV

Require: $setup_time$: The setup time for each UAV

$waypoints \leftarrow []$

$max_route_time \leftarrow 0$

$sweeping_rows \leftarrow$ divide_into_sweeping_rows$(region, num_UAVs)$

$VRP_solution \leftarrow$ solve_VRP $(sweeping_rows, num_UAVs, max_flight_time, setup_time)$

for UAV **in** num_UAVs **do**

    $route \leftarrow VRP_solution[UAV]$

    $waypoints$.append$(route)$

    $max_route_time \leftarrow \max(max_route_time,$ calculate_total_flight_time$(route))$

end for

return $waypoints, max_route_time$

By intelligently leveraging the `divide_into_sweeping_rows` algorithm (Algorithm 3), it ensures an optimal distribution of the coverage area among the UAVs, effectively dividing the region into sweeping rows that the UAVs can efficiently cover.

Algorithm 3 divide_into_sweeping_rows

Require: $region$: The region to be divided

Require: num_UAVs: The number of UAVs to distribute

{Implementation of the sweeping row division algorithm}

return $sweeping_rows$

The `solve_VRP` algorithm is vital in generating optimal routes for each UAV, as it considers factors like distance, time constraints, and the specific requirements of each vehicle to ensure the UAVs complete their assigned tasks most efficiently.

Algorithm 4 solve_VRP

Require: $sweeping_rows$: The region divided into sweeping rows

Require: num_UAVs: The number of UAVs available

Require: max_flight_time: The maximum flight time for each UAV

Require: $setup_time$: The setup time for each UAV

$covered_cells \leftarrow []$

{Implementation of the VRP algorithm with constraints}

for UAV **in** num_UAVs **do**

  $total_flight_time \leftarrow$ calculate_total_flight_time($UAV.route$)

  if $total_flight_time + setup_time > max_flight_time$ **then**

    return "Infeasible Solution"

  end if

  for $cell$ **in** $UAV.route$ **do**

    if $cell$ **in** $covered_cells$ **then**

      return "Infeasible Solution"

    end if

    $covered_cells$.append($cell$)

  end for

  for i **in range** (len($UAV.route$) $- 1$) **do**

    if not is_edge_allowed(UAV.route[i], UAV.route[i+1]) **then**

      return "Infeasible Solution"

    end if

  end for

end for

return $VRP_solution$

To accurately calculate the total flight time for each UAV's route, the algorithm 3 utilizes the `calculate_total_flight_time` (Algorithm 5). This step is crucial in ensuring that the UAVs can complete their missions within the specified maximum flight time, helping to manage resources effectively and meet operational requirements.

Algorithm 5 calculate_total_flight_time

Require: $route$: The route for which to calculate total flight time

1: $total_time \leftarrow 0$
2: **for** i **in range** $(\text{len}(route) - 1)$ **do**
3: $total_time \leftarrow total_time +$ calculate_time_between_nodes$(route[i], route[i+1])$
4: **end for**
5: **return** $total_time$

Furthermore, the algorithm implements the `is_edge_allowed` algorithm (Algorithm 6) function to rigorously check the feasibility of each edge in the UAV's route based on the problem constraints. By thoroughly evaluating each edge, the algorithm can avoid potential issues or conflicts during the execution of the routes, ensuring that the UAVs can safely navigate and accomplish their objectives.

Algorithm 6 is_edge_allowed

Require: $node1$: The first node of the edge

Require: $node2$: The second node of the edge

1: {Implementation of the constraint on routing algorithm}
2: **return True or False**

All the computations for the CPP are performed using MATLAB, utilizing its algorithm development and simulation capabilities. The assumptions of this methodology are such that the number of UAVs is 3, the number of the operator is 1, the UAV setup time is 6, and the UAV flight time is 10 minutes. The result of the CPP process, considering the defined constraints, is illustrated in Figure 4.6.

To achieve consistent measurements within the scanned polygonal region, all UAVs must maintain the same altitude, ensuring uniform coverage. Additionally, maintaining a consistent distance between lanes is crucial. By employing these strategies and leveraging the capabilities of MATLAB for calculation and analysis, an efficient and optimized coverage path plan can be generated for the UAVs, enabling effective surveillance or data collection within the designated area.

A sequence of 3 waypoints can represent the coverage path for the UAV denoted as

$\{\omega_1, \omega_2, \omega_3\}$. Each waypoint is ω_i encapsulates information such as latitude, longitude, and altitude, serving as navigation instructions for the vehicle, such as initiating takeoff, changing speed, or moving to specific areas. The problem can be simplified by assuming that each cell's center corresponds to a waypoint since these waypoints contain all the necessary localization data to guide the vehicle effectively.

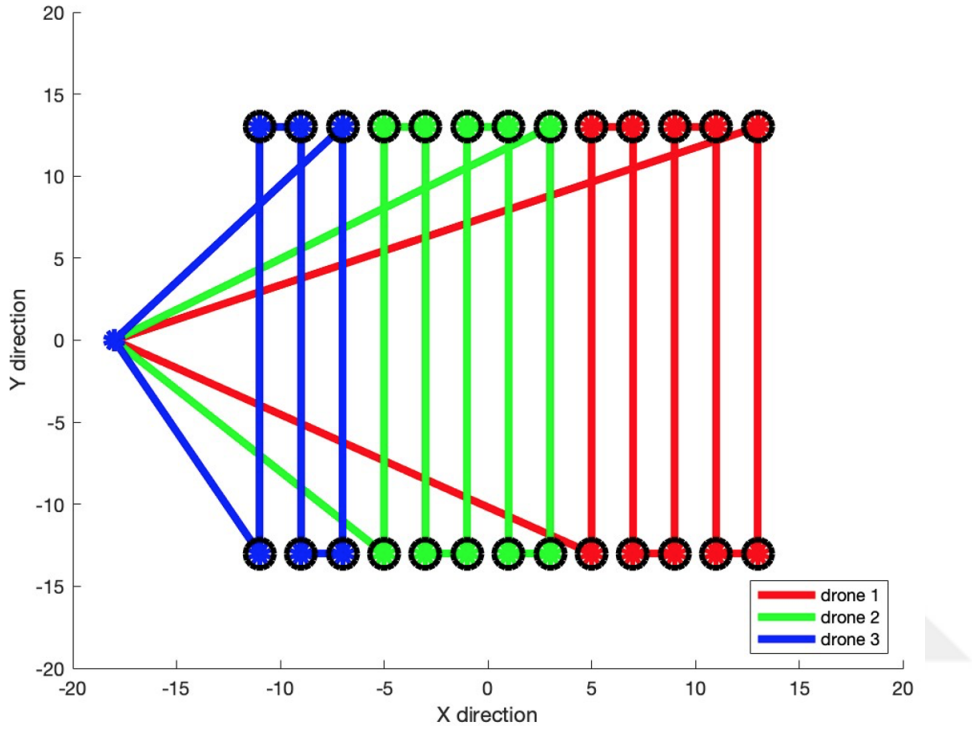


Figure 4.6: The result of the CPP with constraints

A crucial aspect of the operation is preventing potential collisions among the UAVs. To address this, each UAV is assigned a specific altitude for takeoff, travel to the mission start point, and Return to Launch (RTL) to maintain separation and avoid conflicts during the mission (Figure 4.7). However, the problem cannot be solved by dividing the area by the number of UAVs due to two significant factors: the distance of each UAV from the mission's starting point and the allocated altitude. These variables complicate the problem, as the UAVs must maintain safety and avoid flight path disruptions throughout their assigned tasks. Overcoming these challenges requires developing a sophisticated system that can efficiently calculate the optimal routes for each UAV, considering the unique constraints of distance, altitude, and safety. Factors

such as speed, endurance, payload capacity, and battery life must be carefully considered, and optimized to ensure the mission is accomplished within the designated time frame. Designing a system that can efficiently compute the UAVs' routes, account for their structural components, and ensure adherence to the specified criteria poses the primary difficulty in successfully achieving the mission objectives.

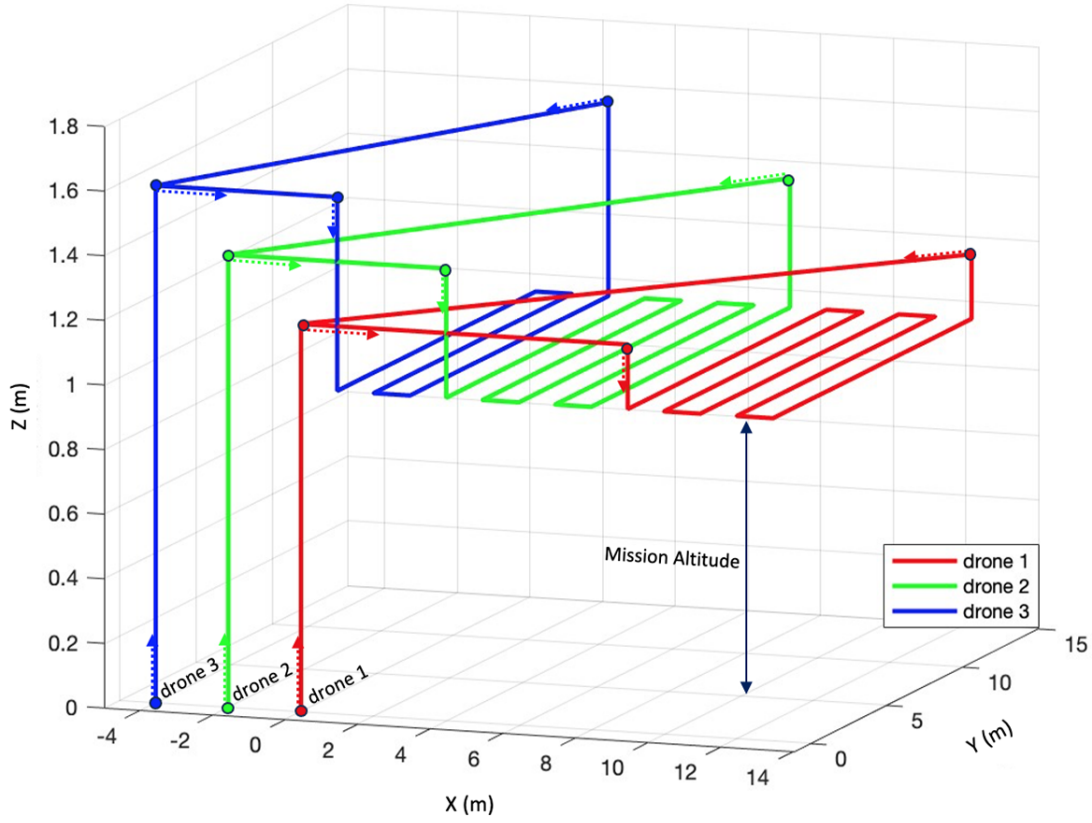


Figure 4.7: Operation plan. Each UAV takes off, operates at mission altitude, and RTLs at the allotted altitude.

4.4.2 Simulations in Gazebo

The simulation environment incorporates Gazebo and ROS to facilitate the simulation process. Figure 4.8 is shown the position of the three quadcopters in the Gazebo. Figure 4.9 presents a visualization of the results obtained from a simulation in Gazebo, which involved three quadcopters and implemented the CPP algorithm with constraints.

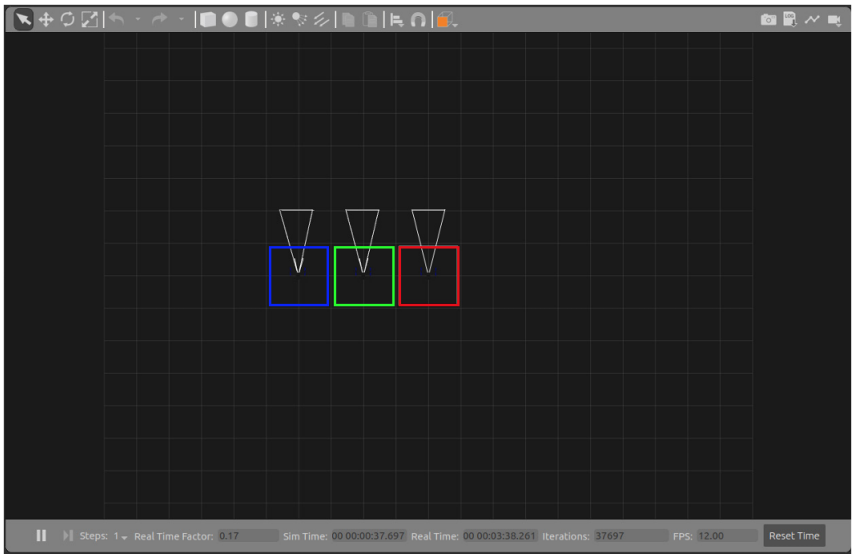


Figure 4.8: Position of the three drones in Gazebo simulation

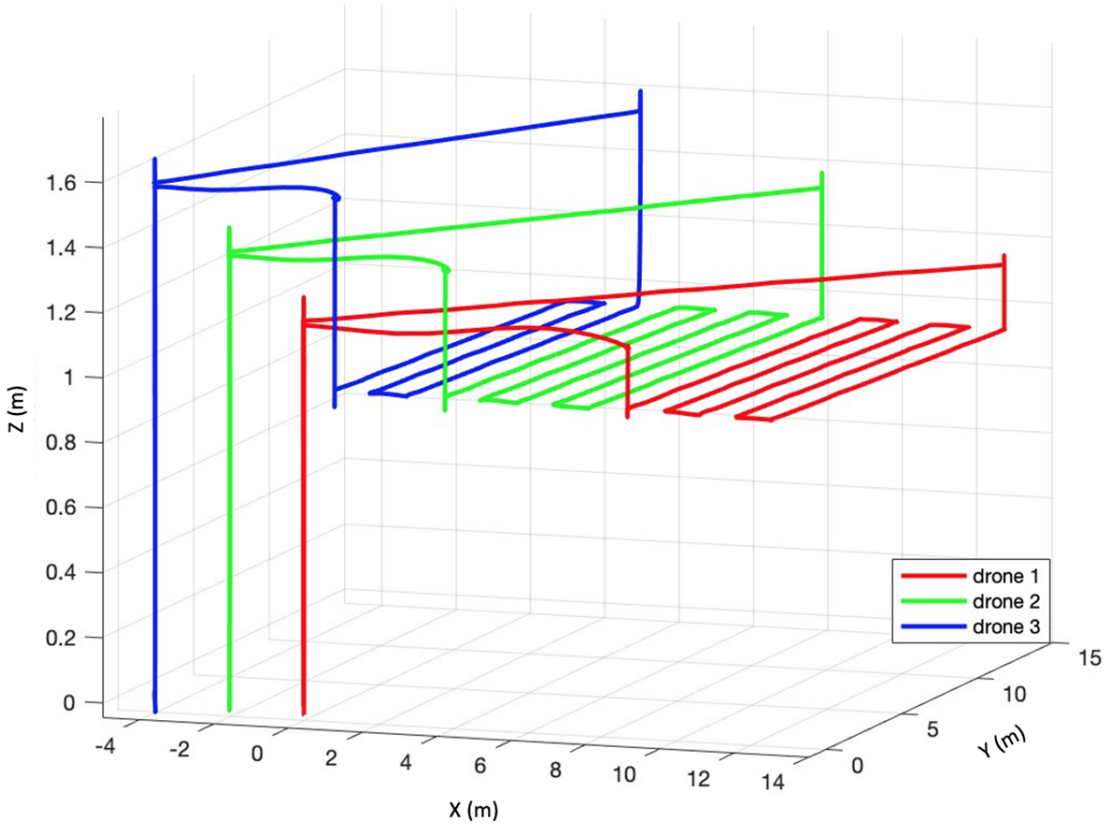


Figure 4.9: The result of the CPP in Gazebo simulation

The figure visually represents the outcome, showcasing the CPP algorithm's effective-

ness and the quadcopters' coordinated behavior. The simulation result in the figure provides valuable insights into the performance of the implemented CPP methodology. It demonstrates the successful execution of the algorithm within the simulated environment, highlighting the capability of the quadcopters to navigate and cover the designated areas effectively. Observing the paths followed by the quadcopters in the figure shows that they can traverse the environment in a coordinated manner while adhering to the specified constraints. This coordinated behavior is a desirable outcome, as it indicates the successful coordination and cooperation among the quadcopters during the coverage tasks.

4.4.3 Real-World Tests

To complement the simulation environment and comprehensively assess the proposed algorithms, a test was conducted using three actual drones in a real-world scenario. This experimental setup aimed to validate and verify the simulation results and practically evaluate the proposed algorithms' effectiveness. Real-world tests were conducted under the following conditions:

1. Coordinates: The test was performed near Ankara, Turkey, with coordinates 39.828479 latitude and 32.559464 longitude.

2. Wind Speed: The wind speed during the test was measured at 3 km/h. This represents the average speed of the wind.

3. Gusts Speed: Additionally, gusts of wind were observed during the test, reaching a 15 km/h speed. Gusts are short-duration bursts of higher wind speed than average wind speed.

4. Wind Direction: The wind was blowing from the north to the south direction. This means that the wind was coming from the northern direction and moving towards the southern direction.

5. Visible Satellites: During the test, 13 satellites were visible. The visibility of satellites is important for positioning and navigation systems that rely on

satellite signals. These conditions assessed the performance and behavior of the system or device tested in real-world scenarios near Ankara, Turkey.

6. Temperature: The test was conducted at 24°C, representing the ambient air temperature.

Figure 4.10 visually represents the real-world test setup. The illustration showcases the drones within the test environment, equipped with specific modifications and additions tailored to meet the experimental requirements.



Figure 4.10: Real-world test for CPP

The purpose of this setup was to closely mimic real-world conditions and ensure the fidelity of the experimental results. By incorporating modifications and additions to the drones, the testing environment aimed to replicate practical scenarios and accurately represent the challenges and constraints encountered in real-world applications. Furthermore, to evaluate the performance of the real-world CPP implementation, Figure 4.11 presents the outcomes of the actual CPP process. This visualization effectively showcases the paths followed by each drone during the coverage operation, offering valuable insights into the real-time execution and effectiveness of the pro-

posed CPP algorithm in a practical setting. The paths traced by the drones visually represent their coordinated behavior and demonstrate the successful implementation of the CPP methodology in a real-world scenario.

The combination of comprehensive tests using actual drones, along with the visual representations in Figures 4.10 and 4.11, enhances the credibility and reliability of the proposed algorithms. The real-world experiments validate the algorithms' performance under realistic conditions, ensuring their practical effectiveness beyond the simulated environment.

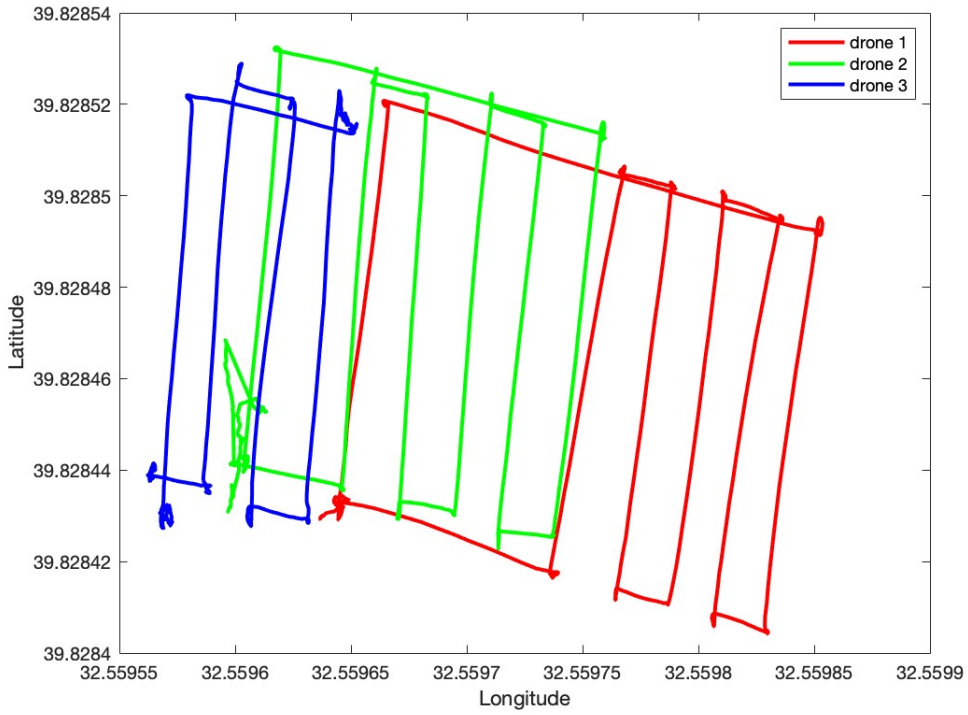
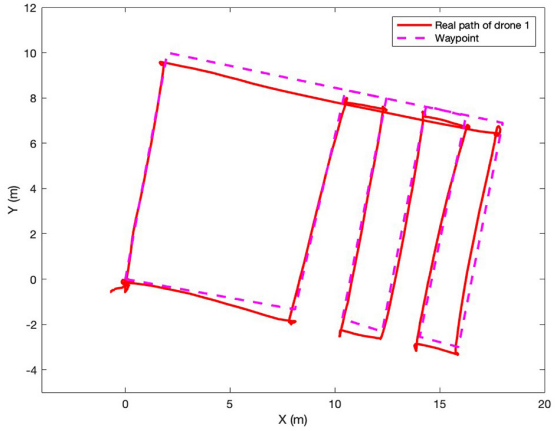
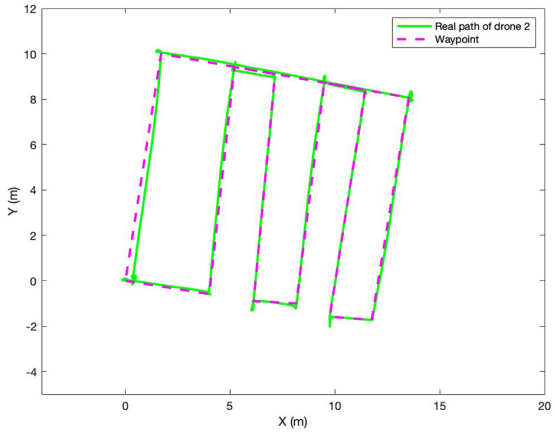


Figure 4.11: The result of the real-world CPP

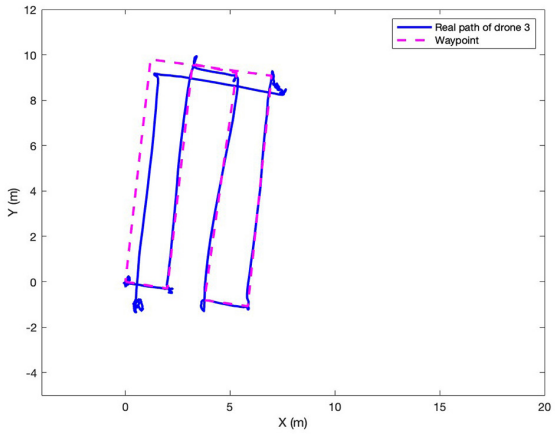
In this study, a comprehensive analysis is presented regarding the real-world performance demonstrated by three distinct drones, as depicted in Figures 4.12a, 4.12b, and 4.12c. These visual representations offer a detailed insight of each drone. Drones follow the pre-defined path which is generated in the CPP process. However, because of the number of GPS satellites detected by drones and using the standard GPS, not Real Time Kinematic (RTK) GPSs, each drone has an error of a maximum of 50 cm.



(a) Comparing the real path of the drone 1 with waypoint



(b) Comparing the real path of the drone 2 with waypoint



(c) Comparing the real path of the drone 3 with waypoint

Figure 4.12: Performance of drones

4.5 Conclusion

This chapter presents the proposed methodology for efficiently addressing the CPP problem using UAVs. The process comprises two main components. Firstly, a similar approach as described in [91] was employed to divide the region into sweeping rows, following the methodology shown in Figure 4.4. This step involved partitioning the area into rows to be covered. Secondly, inspiration was drawn from a solution to a VRP [92], transforming the problem into a VRP by considering the customers as the endpoints of the coverage rows and the set of UAVs as vehicles. Each UAV was treated as a vehicle, and each endpoint of the coverage rows was modeled as a customer. Solving the VRP could determine the optimal routes for each UAV.

To ensure efficient coverage and adherence to the specific constraints, additional constraints in the proposed optimization problem are introduced. These constraints included considerations of time, area coverage, and routing. The objective is to minimize the required time for complete area coverage, and each UAV's time to fly its path is mathematically formulated. Also, constraints are introduced to ensure that each UAV covers a specific area within the target region and that routing decisions are made sequentially to optimize the overall time required for coverage.

To validate and evaluate the proposed methodology, simulations using the Gazebo simulator and real-world tests using three actual drones are conducted. The simulations allowed us to assess the proposed methodology's efficiency, accuracy, and robustness under various conditions and scenarios. The real-world tests provided insights into the practical applicability and performance of the method.

These comprehensive evaluations demonstrate the proposed CPP approach's effectiveness, feasibility, and practicality in simulated and real-world settings. The results obtained from the simulations and real-world tests validated the proposed methodology and provided valuable insights for further analysis and comparison.

In conclusion, the proposed methodology for efficient coverage path planning using UAVs offers a practical and effective solution for optimizing the time required for complete area coverage. Controlling the region into sweeping rows and transforming the problem into a VRP allows for generating optimal routes for each UAV, considering various constraints. The methodology has been successfully validated through simulations and real-world tests, confirming its applicability and performance. By

leveraging these insights, the approach can contribute to the efficient deployment of UAVs in surveillance, data collection, and other applications that require complete coverage of a target area.

CHAPTER 5

TEST PLATFORMS

5.1 Introduction

The growing popularity of quadcopter technology has increased demand for UAVs that excel in photography and videography, driving the need for dependable and effective aerial platforms [94, 95, 96, 97]. In this context, the gimbal emerges as a crucial piece of quadcopter equipment, playing a vital role in providing stable and fluid movement of the camera during flight. The gimbal's primary function is to ensure stabilization, effectively counteracting the effects of vibrations, gusts of wind, or sudden movements, which can compromise the quality of captured footage.

This chapter focuses on designing, constructing, and applying a specialized test bench to evaluate the performance of gimbals attached to quadcopters. However, understanding the context in which test benches are employed for assessing quadcopter components is vital. Evaluating various components of quadcopter systems requires using test benches that come in multiple designs and combinations, each serving a specific function [98]. A stationary test bench is a form of test bench that is quite widespread. This type of test bench offers a controlled environment to assess individual components extensively. According to [99, 100, 101], stationary test benches make it possible to take precise measurements and conduct tests in a manner that can be repeated, ensuring an accurate and trustworthy evaluation of performance metrics. These multifunctional test benches may test various quadcopter components, including motors, propellers, flight controllers, sensors, batteries, and overall system stability. Faundes et al. [102] presented a platform in a controlled environment for testing a UAV. This test platform measures the pitch, yaw, and roll angles; acceleration; power consumption, and the force exhausted by the UAV in various environmental condi-

tions. The authors have proposed the test side-based platform solutions in [103] to have a comprehensive study. In this platform, a developer can simulate the actual environmental conditions for UAVs, where the objective is to take water samples from a river. Different test platforms are permitted free-flight tests with conventional user algorithms in the literature [99, 100, 101, 102, 104, 105, 106, 107, 108, 109].
Although stationary test benches have seen much use in quadcopter research and development, the specific application of these benches when it comes to measuring gimbal performance is still not well explored. This research addresses this knowledge gap by presenting a test bench explicitly built to evaluate the performance of gimbals mounted on quadcopters. Such testing helps validate the gimbal's performance and suitability for diverse applications requiring precise camera control and stable aerial footage. Additionally, the test platform can be used for hardware-in-the-loop testing on the algorithms of the onboard system. With the diagnostic tools available on the platform, it is possible to evaluate object tracking, line tracking, stabilization, control, and navigation algorithms.
This chapter is organized as follows. Section 5.2 describes the test bench platform. Section 5.3 presents the system control of the test platform and its working method. In Section 5.4, the design of the gimbal, which is used for 3D printing, is explained, and in Section 5.5, the result of the simulation and 3D printing, which is obtained in this work, is described.

5.2 Design of the Test Platform

To conduct experimental experiments, a test bench platform was designed to imitate the quadcopter's flight in a secure setting. This test platform is constructed on a robust structure replicating the quadcopter and integrating different gimbal types easily. To guarantee stability throughout testing, the structure is built from solid components. The test platform incorporates an adjustable mounting system for precise gimbal positioning. This enables testing different conditions and recording the full range of gimbal performance. The test platform has a motorized gimbal movement system fitted, which provides controlled motion in numerous axes. The ability of the gimbal to retain stability and suppress unwanted oscillations is determined by preci-

sion sensors on the test platform, which continuously measure and analyze vibrations. Additionally, the testing platform features data monitoring and acquisition tools that enable real-time data capture. These systems record essential variables, such as angular position, rotational speeds, motor currents, and control signals, providing a sizable dataset for performance analysis.

The test platform created offers a flexible and regulated testing environment for evaluating the performance of gimbals mounted on quadcopters. Its features, such as adjustable mounting, motorized gimbal movement, vibration analysis, and data gathering systems, enable extensive and accurate performance evaluation of the gimbal in a controlled laboratory setting. A three-degrees-of-freedom experimental test platform for testing control algorithms in quadcopter systems is shown in Figure 5.1. The specifications of this test platform are shown in Figure 5.2.

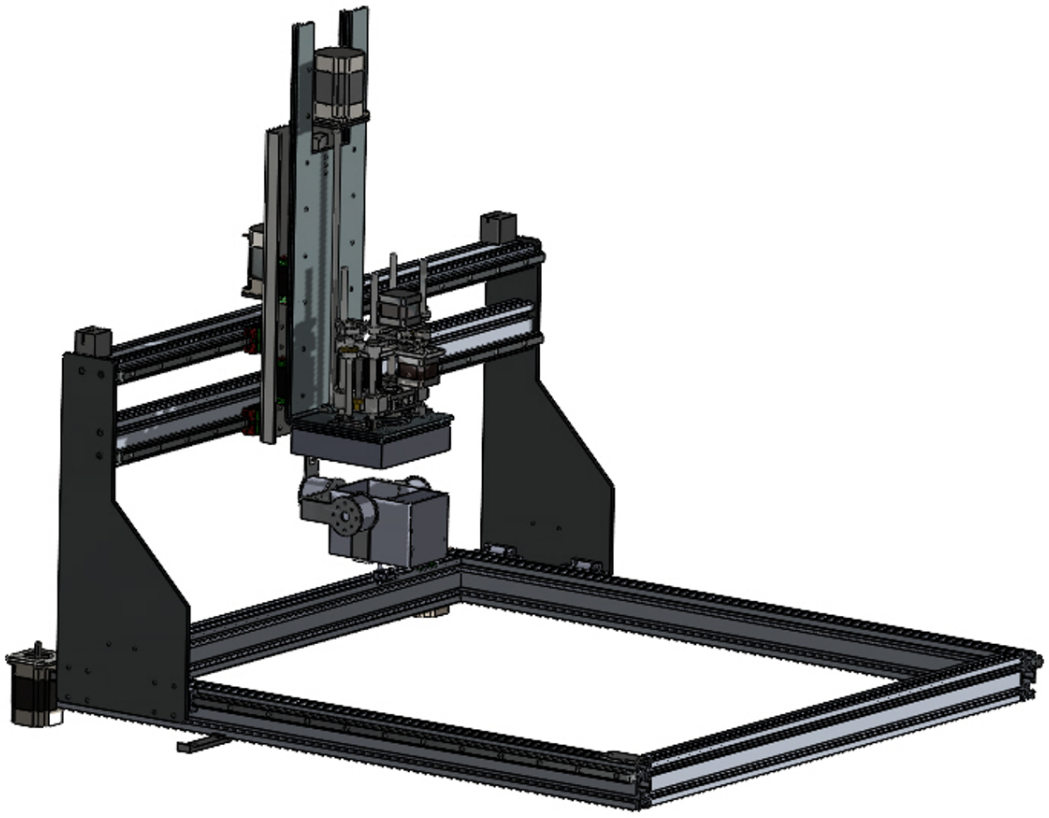
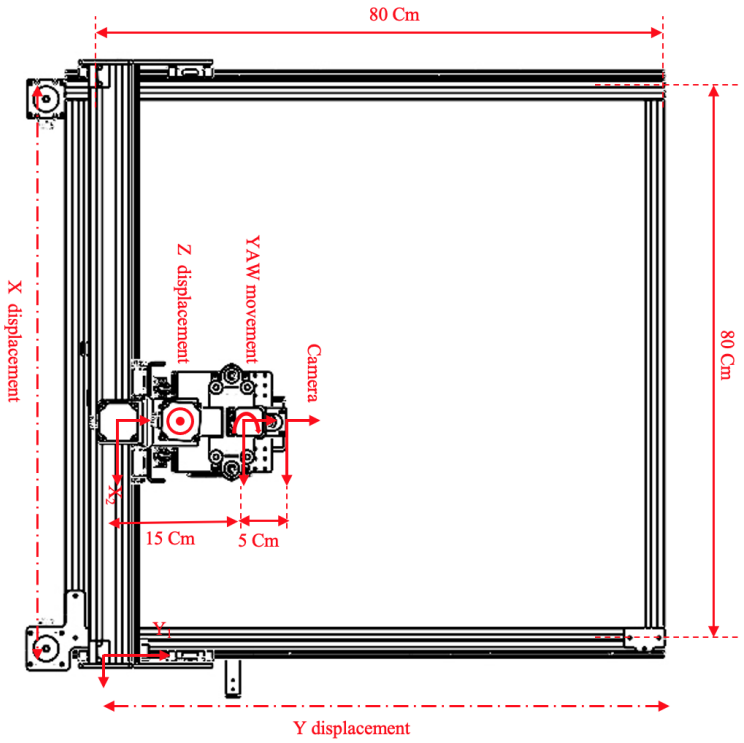
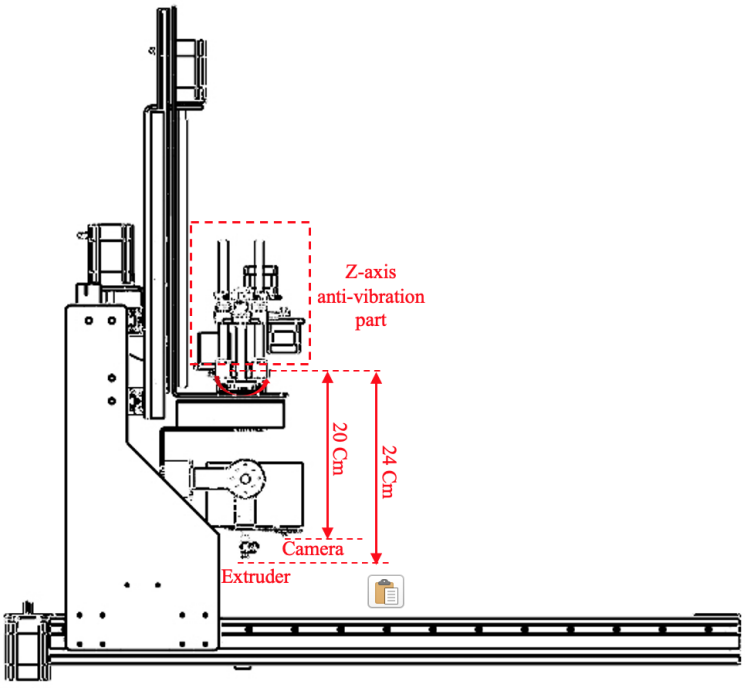


Figure 5.1: Test platform with mounted gimbal



(a) Top view



(b) Side view

Figure 5.2: Dimension of the test platform

5.3 System Model

When the quadcopter is flying, the total number of revolutions around the quadcopter's roll, pitch, and yaw axes is referred to as the quadcopter's instantaneous attitude variations. The position of the motors in the quadcopter is shown in Figure 5.3. With the information of the motor speeds from Pixhawk, forward, left, right, and backward movement of the test bench platform is adjusted.

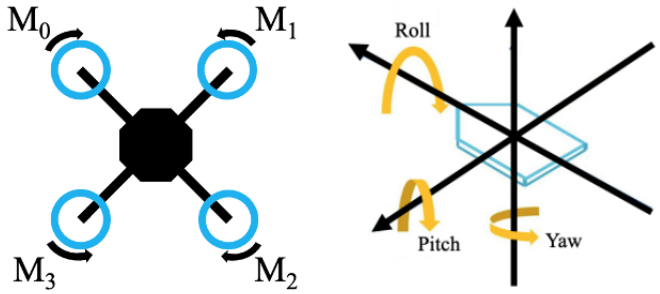


Figure 5.3: The quadcopter's instantaneous variations

Pitching the quadcopter will control its forward and backward movements. This may be done by delivering greater power to the rear two props, simultaneously limiting the power in front props for forward motion. Vice-versa is done for reverse motion. The cause is the increase in momentum on a side, generating the imbalance of momentum; therefore, the quadcopter moves forward or backward. Rolling the quadcopter will regulate its rightward and leftward motion or bend. This is done by delivering greater power to left props while lowering power to right props for rightward motion. Vice-versa is done for leftward motion. Yaw is just the twisting of the quadcopter around its vertical axis. This is done to alter the orientation of the front of the quadcopter in whichever direction the pilot wishes or possibly an autopilot desire. Two opposed props are powered more while diminishing the power of the other two. Several characteristics of quadcopter flying behavior are shown in Figure 5.4.

As shown in Figure 5.4, all the rotors must simultaneously revolve at the same increasing speed to gain altitude. Similarly, to lower the height, all the rotors must revolve at the same decreasing speed. Hovering is accomplished when all the rotors revolve at the same rate. Forward movement occurs when the rear rotors rotate

faster than the front rotors, and backward movement occurs when the front and rear rotor speeds contradict one another. The pitch angle is affected by these two motions. Whenever the two left or suitable rotors rotate faster than the other two, a strafe movement to the left or right is accomplished, affecting the roll angle. Suppose the combined speed of all the rotors traveling clockwise is higher than all the rotors traveling counterclockwise (or vice versa). In that case, a rotation along the Z-axis is generated, affecting the yaw angle. To simulate the quadcopter's forward, backward, left, right, and orientation movement, the motor output of the Pixhawk is used.

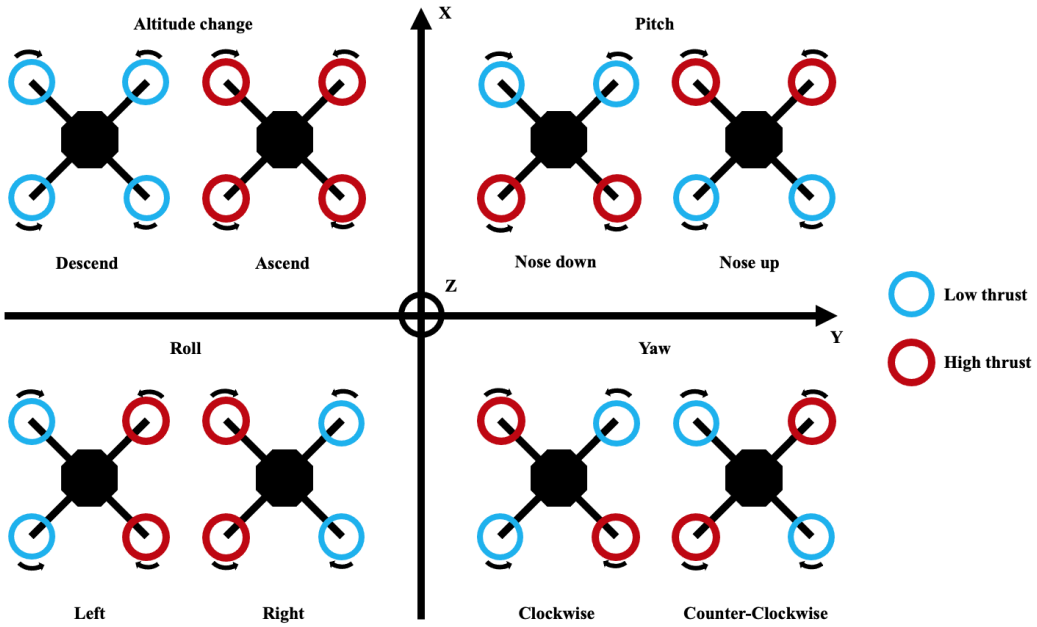


Figure 5.4: Basic principles of quadcopter maneuvering

5.3.1 Possible Errors in the System

Successful quadcopter operations depend on precise and accurate placement in the XY plane (horizontal plane) and Z-direction (vertical axis). It is crucial to comprehend the potential faults in these dimensions for performance optimization, safety, and accurate flight results. Sensor limitations, outside disturbances, and control system dynamics can bring on errors in the XY plane. The quadcopter's location estimation may be off the intended position due to sensor faults, drift, or calibration mistakes. Maintaining a steady position in the XY plane can be difficult when exter-

nal influences like wind or air turbulence introduce lateral drift. The dynamics of the control system can also influence errors in position control, such as response delays or less-than-ideal control gains. Similar errors can also happen in the Z-direction, the vertical axis of the quadcopter. The limitations of altitude sensors and changes in atmospheric conditions might cause inaccuracies in altitude estimation. The height measurement accuracy may be affected by drift over time in barometric pressure sensors used for altitude estimation. Due to aerodynamic changes, the ground effect phenomena that occur when flying close to the ground might cause mistakes in the quadcopter's vertical position. Additionally, the quadcopter's stability may be impacted by the weight of payloads or equipment, leading to inaccuracies or variations in keeping a constant height. For measures to reduce their influence, it is essential to comprehend these possible faults. The methods that can help reduce errors in both the XY plane and Z-direction include sensor calibration, improved control system tuning, environmental monitoring, and taking payload impacts into account. Utilizing cutting-edge locating methods, such as GPS or vision-based systems, can help improve accuracy and reduce errors.

5.3.1.1 Possible Errors in the XY Plane

Errors in the quadcopter's XY (horizontal) plane might result from various sources. These faults may impact the stability and accuracy of the quadcopter's placement. The followings are some typical causes of mistakes in the XY plane.

1. Position Estimation Errors: To determine their position in the XY plane, quadcopters commonly use onboard sensors like accelerometers, gyroscopes, and GPS. Sensor noise, constraints, and inaccuracy can cause errors in position estimates. For instance, accelerometer noise can cause the predicted position to wander over time.

2. Sensor Fusion and Filtering: To enhance position estimation, sensor fusion techniques, such as using a Kalman filter, combine data from several sensors. inaccuracies in the XY plane can still result from biases or inaccuracies in the sensor data that affect the fused position estimation.

3. External Disturbances: The XY plane can be disturbed by external causes such as wind, air turbulence, or abrupt changes in air pressure. These disturbances may cause the quadcopter to misestimate its position, compromising its ability to hold a stable position.

4. Dynamics of the control system: A quadcopter's control system, including its flight controller, is essential to preserving a stable location in the XY plane. Position control mistakes can result from dynamics in the control system, such as reaction delays or inaccurate control inputs. Errors in maintaining the correct position can also be caused by improperly tuned control gains or control algorithm constraints.

5. GPS Signal Errors: GPS is frequently used to place quadcopters outside. But GPS signals can contain mistakes because of signal multipath, weather, or satellite geometry. These mistakes may impact the quadcopter's location estimation precision in the XY plane.

6. Accurate sensor alignment and calibration: For accurate position estimation in the XY plane, onboard sensors must be correctly aligned and calibrated. Biases can be introduced, and the position estimation distorted by misalignment or calibration faults in sensors like gyroscopes or magnetometers.

Instead of immediately reducing inaccuracies in the XY plane position of a quadcopter, a gimbal is primarily intended to stabilize and regulate the camera's or Extruder's orientation. Indirectly, the stabilization offered by a gimbal can help enhance the quadcopter's overall stability and control, which can help lessen some mistakes in the XY plane. Here are some ways a gimbal can help.

1. Increased Stability: A gimbal keeps the camera and extruder oriented, minimizing unintentional movements and vibrations.

2. Improved Flight Control: When a quadcopter is fitted with a gimbal, the control system frequently includes stabilization feedback. By lowering position errors in the XY plane, this feedback can aid the quadcopter's overall flight control and stability.

3. Vibration Damping: To reduce or eliminate vibrations and disturbances, the stabilization system of the gimbal may contain vibration-damping elements. The overall stability and precision of the quadcopter's position estimation and control in the XY plane can be improved by decreasing vibrations.

Remembering that a gimbal may not immediately address all potential inaccuracies in the XY plane, even though it can sometimes help stabilize the quadcopter. Additional aspects like sensor accuracy, control system dynamics, and external disturbances still need to be considered and dealt with independently to improve position estimation and control in the XY plane. A well-designed gimbal can help decrease mistakes and enhance the quadcopter's overall performance in the XY plane when combined with suitable sensor fusion, control system tuning, and environmental monitoring. Here is how a gimbal might impact the quadcopter's vibrations.

1. The gimbal's stabilization function can, inadvertently, lessen the effect of vibrations on the camera and extruder. A gimbal often includes mechanical parts that assist in separating the camera and extruder from the quadcopter's frame, such as dampers or isolation mounts. These mechanical components can reduce or absorb the transfer of vibrations from the quadcopter to the camera and extruder, isolating them and reducing the impact of quadcopter vibrations on 3D printing quality.

2. Stabilization: The gimbal's motors and control system enables active stabilization, which helps reduce external disturbances like vibrations. The gimbal's sensors track the movements caused by the quadcopter's vibrations and take appropriate action to steady the camera and extruder. This stabilizing technique reduces the noticeable impacts of vibrations on 3D printing quality.

3. Additional Vibration Dampening functions: Some gimbals include extra vibration dampening functions. Specialized mounts, shock absorbers, or vibration isolation systems are some characteristics that further lessen vibrations transferred to the camera and extruder. Smoother and more stable images result from the gimbal's ability to reduce vibration amplitude.

5.3.1.2 Possible Errors in the Z-Direction

The performance of gimbals might be negatively affected by quadcopter vibrations that introduce undesired motion. Several things, such as propeller imbalances, engine vibrations, or outside disturbances, may cause these vibrations. Inevitable mistakes could happen when flying a quadcopter in the Z-direction (vertical axis). Here are some typical mistakes to think about:

1. Sensor noise and inaccuracies: To detect the Z-axis location, quadcopters commonly utilize altitude sensors such as barometric pressure, altimeters, or ultrasonic sensors. These sensors could measure with noise and inaccuracies that would make it difficult to determine the true height.

2. Limitations of barometric pressure sensors: Barometric pressure sensors rely on changes in air pressure to determine altitude. Weather, temperature, and humidity variations can cause errors in altitude estimation.

3. Ground effect and height estimation errors: When a quadcopter flies close to the ground, the ground effect phenomena cause aerodynamic changes. As a result of the compressed air between the quadcopter and the ground affecting pressure readings, the ground effect might affect height estimation.

4. Payload implications on Z-axis stability: Quadcopter gimbal testing sometimes includes the deployment of payloads like cameras or sensors. Payloads may alter the quadcopter's center of gravity and general stability, which could result in inaccuracies when trying to maintain a stable Z-axis position.

5. Dynamics and tuning of the control system as they relate to Z-axis control: For a quadcopter to maintain stable flight and precise Z-axis orientation, its control system is essential. Control system dynamics, tuning, and reaction characteristics can affect the quadcopter's ability to maintain the correct Z-axis position.

Additional methods can be used to maintain stability and minimize vibrations, specifically for the gimbal system in a quadcopter. Here are several vibration-dampening strategies pertinent to maintaining the gimbal's stability.

1. Mechanical vibration isolation systems: By physically separating the gimbal from the quadcopter's frame using vibration-damping materials, shock absorbers, or isolators, vibration transmission is minimized.

2. Active vibration control techniques: Using active control systems that detect vibrations and exert opposing forces or motions to cancel them out, enhancing gimbal stability.

3. Software-based vibration compensation techniques: By creating algorithms that assess sensor data and correct for vibrations in real-time, the gimbal can counteract the disruptions brought on by vibrations.

While a gimbal is an effective tool for minimizing the effects of vibrations on a camera and extruder, it is essential to acknowledge that it may not eliminate all the vibrations. This should be considered when utilizing a gimbal for stabilization purposes. The effectiveness of vibration reduction depends on several factors, including the gimbal's design, the quality of its stabilization mechanism, and the magnitude of vibrations generated by the quadcopter.

It is crucial to address various variables to optimize the reduction of vibrations. Firstly, the design of the gimbal itself plays a significant role. A well-engineered gimbal with robust stabilization capabilities is more likely to provide superior vibration reduction performance.

Additionally, the intensity of vibrations produced by the quadcopter, such as those caused by motor vibrations or propeller imbalances, can impact the gimbal's effectiveness. Therefore, employing balanced propellers and minimizing motor imbalances can reduce vibrations at their source.

Moreover, maintaining the quadcopter in proper working condition is essential. Regular maintenance, such as inspecting and tightening components, checking for damaged or worn-out parts, and ensuring the quadcopter is well-balanced, can help mitigate vibrations and enhance the gimbal's performance.

5.3.1.3 Resistance during 3D Printing with Quadcopters

Resistance during 3D printing with a quadcopter pertains to the counteracting force encountered by the quadcopter during the material extrusion process and the subsequent formation of a three-dimensional structure. The resistance encountered can be attributed to multiple aspects related to the material deposition process, the contact between the material and the ground, and the propulsion system of the quadcopter. The following section elucidates some essential facets of resistance.

1. Material Adhesion to Ground: The material adhesion to the ground occurs when the 3D printing material is extruded and attaches to the surface. The presence of adhesion results in substantial resistance as the quadcopter seeks to advance. The substance exhibits a braking effect, thereby hindering the quadcopter's motion.

2. Shear Strength of Material: The shear strength of the deposited material is a significant factor in determining the force necessary to induce its movement. Materials possessing greater shear strength will provide increased opposition to displacement.

3. Extrusion Force: The resistance experienced during extrusion can be influenced by the force used to push the material through the nozzle. When confronted with a material that necessitates a substantial extrusion force, the propulsion system of the quadcopter is compelled to exert more significant effort to surmount this resistance.

4. Printing Speed: The velocity at which the quadcopter traverses throughout the material extrusion process impacts the level of resistance experienced. Elevated velocities can result in heightened resistance due to the deposited substance's augmented momentum.

5. Ground Interaction: Ground interaction refers to the phenomenon where uneven or rocky terrain can intensify the resistance experienced by a substance as it encounters surface irregularities. The quadcopter is required to manage these variances, contributing to its moving task's complexity.

Integrating a 3D printing extruder onto a gimbal is a potential remedy for addressing challenges related to resistance that arise during the implementation of 3D printing with a quadcopter. The integration of a gimbal for the 3D printing extruder has the potential to address resistance concerns effectively.

1. Stabilization Enhancement: Implementing a gimbal system for mounting the 3D printing extruder offers enhanced stabilizing capabilities. The motorized adjustments of the gimbal serve to uphold the extruder's alignment by compensating for the quadcopter's motions and external disturbances. The improved stabilization system reduces abrupt changes in the extruder's position, preventing disturbances in material deposition resulting from the quadcopter's movement.

2. Precise Material Deposition: Stabilizing the gimbal guarantees the maintenance of a level and consistent extruder nozzle during the quadcopter's meeting with resistance from the printed material. The enhanced precision of the process leads to a higher degree of accuracy in the deposition of materials, hence minimizing the occurrence of uneven layers or misaligned structures caused by deviations resulting from resistance.

3. Controlled Printing Process: Implementing a gimbal to stabilize the extruder enhances the control of the 3D printing process. The extruder's motions are effectively coordinated with the quadcopter's maneuvers, facilitating material deposition with enhanced precision and consistency. This control mechanism aims to reduce resistance and non-uniform material flow disturbances.

4. Mitigation of External Forces: The mitigation of external forces is achieved through the active compensation of variations in orientation and resistance by the gimbal as the quadcopter progresses in its printing process. The reduction of external pressures, such as material adhesion, aids in enhancing the quadcopter's navigational efficiency and facilitates a steady trajectory, resulting in an overall improvement in movement performance.

5. Enhanced Flight Dynamics: The flight dynamics of the quadcopter during 3D printing are enhanced due to the stability supplied by the gimbal. The proposed

mechanism serves to mitigate abrupt tilting or oscillation, which has the potential to result in heightened resistance and disruption of materials. The quadcopter exhibits enhanced maneuverability within the printing environment.

6. Optimized Extrusion Speed: Combining the gimbal and extruder enhances stability, facilitating more precise control over extrusion speeds. The quadcopter can modulate its velocity to synchronize with the extrusion rate, thereby mitigating obstacles associated with resistance and guaranteeing proper adhesion of the material to the printed structure.

7. Reduced Reprints and Errors: Implementing a gimbal-extruder system effectively reduces reprints and errors in the printing process. This setup minimizes disturbances caused by resistance, enhancing the printed output's precision and quality. The mitigation of uneven layers, misalignments, and material disturbances minimizes the necessity for reprints and promotes the overall efficacy of the printing process.

In conclusion, incorporating a gimbal into the 3D printing extruder can significantly enhance the quadcopter's maneuverability during the 3D printing process by offering stabilization and control. The integration process employed in this context mitigates the adverse effects of resistance, facilitating more precision in material deposition, smoother motion, and improved overall quality of the printed output. Nevertheless, despite the numerous advantages presented by this strategy, it is crucial to meticulously measure the benefits against the additional factors of weight, power consumption, and system complexity.

5.4 Experimental Results

Figure 5.5 and 5.6 show the experimental test platform used in this study. The experimental results presented in this work are divided into two control tests, each designed to address specific aspects of the quadcopter's movement.
The experimental results aim to showcase the gimbal's performance and effectiveness in maintaining stability, accurate line-tracking, and vibration reduction by conducting these two control tests. The outcomes of these experiments provide valuable insights

for further development and improvement of gimbal systems, ultimately contributing to the advancement of UAV technology.

The first control test aims to demonstrate and simulate the quadcopter's movement in a test platform designed explicitly for line following. The camera is mounted on the gimbal and placed underneath the test bench platform in this setup. This test evaluates the gimbal's ability to accurately follow a predetermined line under the platform. This test validates the gimbal's performance and its capability to maintain stability and precision.

The second control test assesses the gimbal's ability to mitigate vibrations and disturbances while the quadcopter is in flight. This control test examines the gimbal's response to external factors, such as wind gusts or rapid changes in the quadcopter's velocity. The experiment measures the gimbal's effectiveness in stabilizing and reducing unwanted movements by subjecting the quadcopter and gimbal to controlled vibration sources.

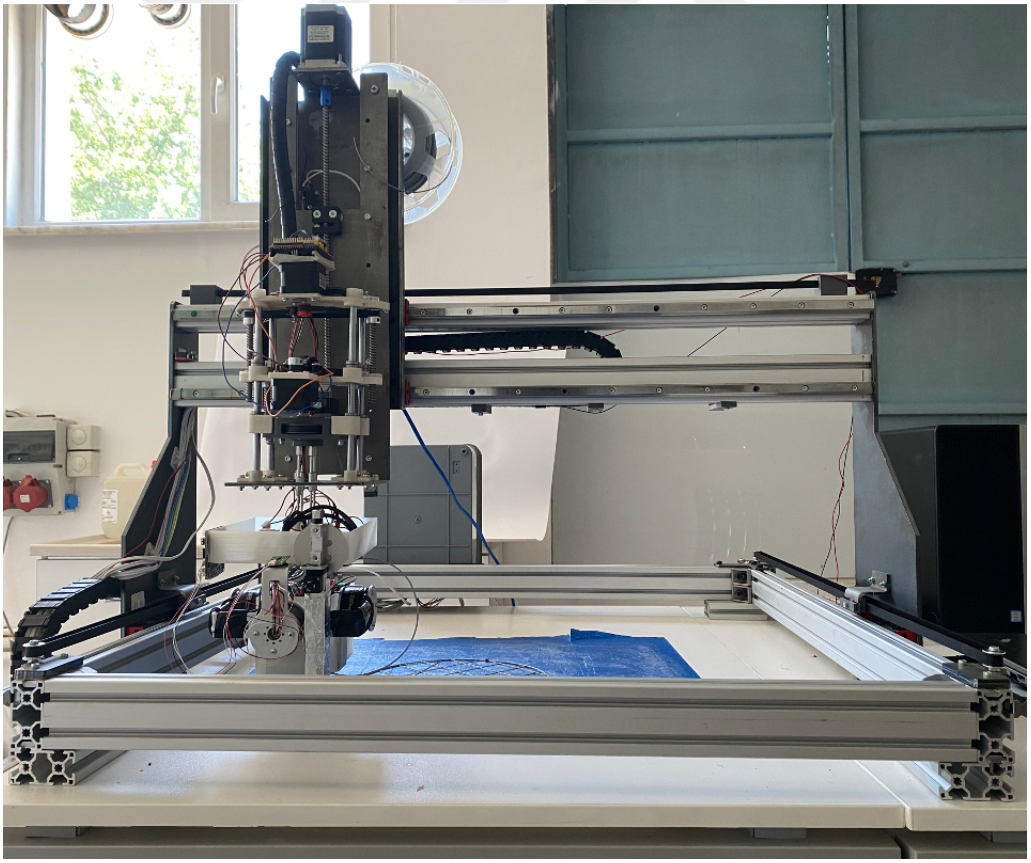


Figure 5.5: Test Platform

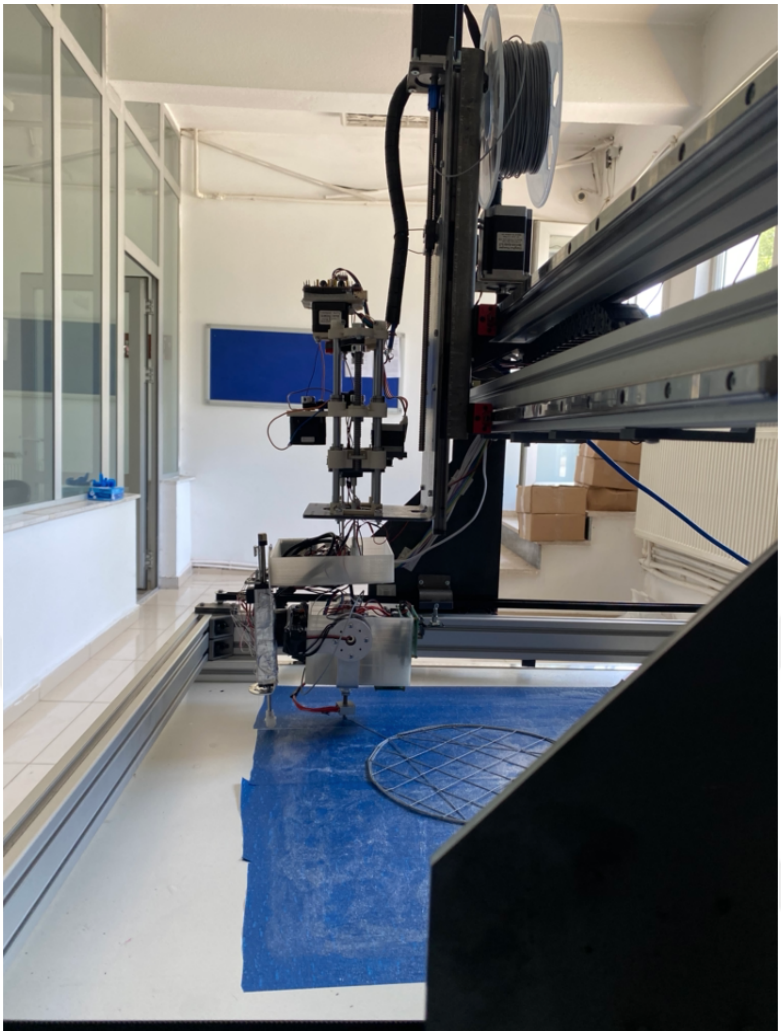


Figure 5.6: Test Platform - Right view

5.4.1 Line Following Test

Once the entire system has been brought under control, a 3D printing test is initiated. This test examines the movement of the platform in an X or Y direction, as well as inclined lines and the rotation of the platform.
The system is provided with the coordinates of each point in the map. Ultimately, the 3D printing process is restarted from the beginning. Because of the vibrations in the system, mistakes can occur whenever the platform moves and the endpoint will not be in the correct position. The gimbal provides control and corrects all these problems. The concluding test and its outcomes are in the figure at 5.7.

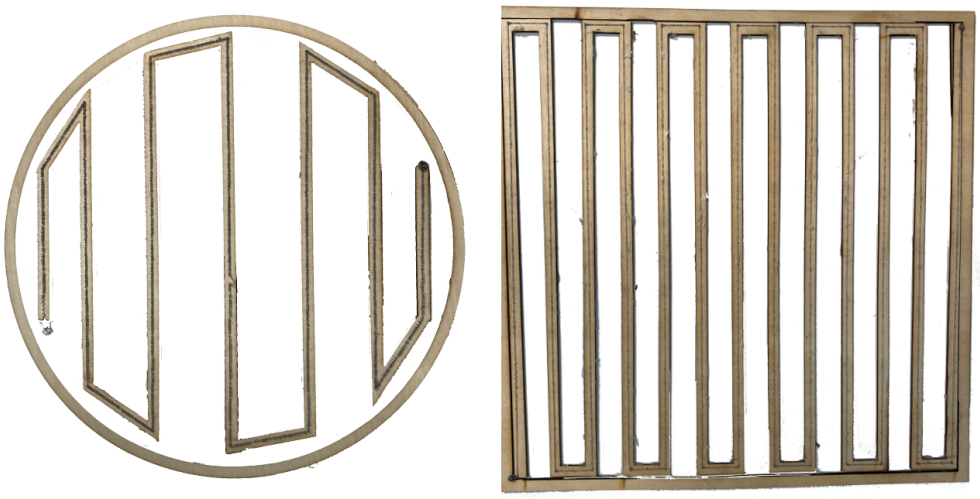


Figure 5.7: The result of the circle and square shape single layer 3D printing

5.4.2 Gimbal Performance Test

To effectively reduce potential system abnormalities, the functioning of the gimbal must be evaluated using a thorough testing process. This evaluation process is carefully planned to unfold in two stages, allowing for a rigorous investigation of the gimbal's capabilities in several aspects. Each evaluation stage is precisely engineered to target and correct problems that manifest differently in their respective domain, acting as strategic checkpoints.

The first stage of this comprehensive evaluation delves into the intricate realm of the XY plane, where the gimbal's dexterity in rectifying errors related to lateral movement and orientation is scrutinized with precision. This test probes the gimbal's responsiveness and adaptability in counteracting variances and disturbances, ensuring that any deviations from the intended trajectory are promptly corrected. By compensating for errors within this plane, the gimbal fortifies its ability to maintain stability and accuracy in movements that traverse the horizontal dimensions.

The second evaluation stage focuses on the Z-direction, encompassing the vertical axis of movement. The gimbal's proficiency in rectifying altitude-related discrepancies comes to the fore in this dimension. This test aims to ensure that the gimbal's corrections remain steadfast and precise, guaranteeing that deviations from the in-

tended altitude are minimized and swiftly corrected.

Collectively, this dual-staged evaluation encapsulates an exhaustive exploration of the gimbal's performance, ranging from lateral stability to vertical precision. The testing procedure is an unmistakable testament invested in perfecting the gimbal's efficacy by addressing errors in the XY plane and the Z-direction. This systematic approach to evaluation ultimately engenders a heightened level of confidence in the gimbal's ability to seamlessly operate within the quadcopter system, effectively mitigating errors and bolstering its overall performance envelope.

5.4.2.1 Gimbal Performance in XY Plane

The initial step focuses on rectifying errors within the XY plane by employing two IMUs; one is strategically positioned near the extruder, controlling the gimbal, while the second one is mounted before the roll motor. This configuration enables the precise calculation of system errors directly preceding the functioning of the gimbal itself. Figure 5.8 is shown the positioning of the IMUs on the platform.

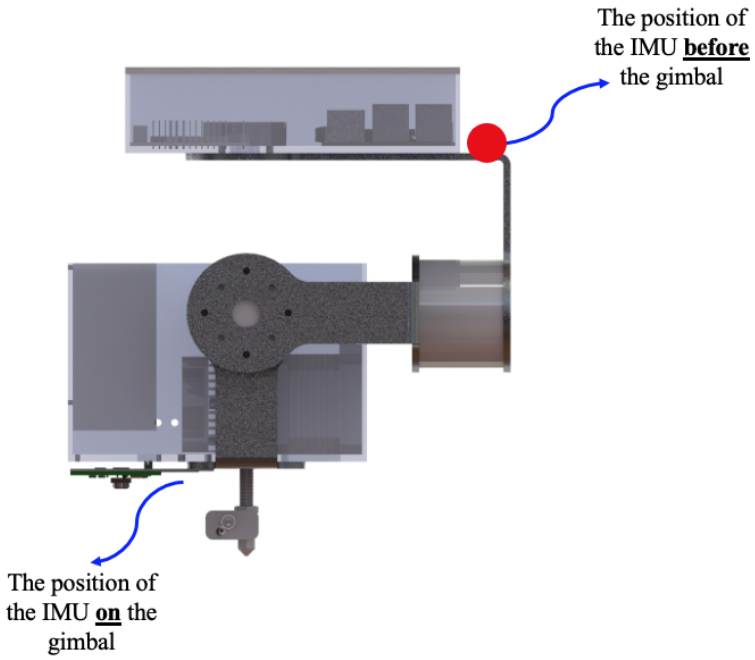


Figure 5.8: Position of the IMUs on the platform

Furthermore, Figure 5.9 provides a visual representation of the IMU data obtained from the two IMUs during a circular motion of the platform. This visualization offers valuable insights into the comparative behavior of the IMUs and assists in analyzing their performance in different scenarios. By examining this data, the effectiveness of the gimbal's error compensation mechanisms can be accurately assessed, and any necessary adjustments can be made accordingly.

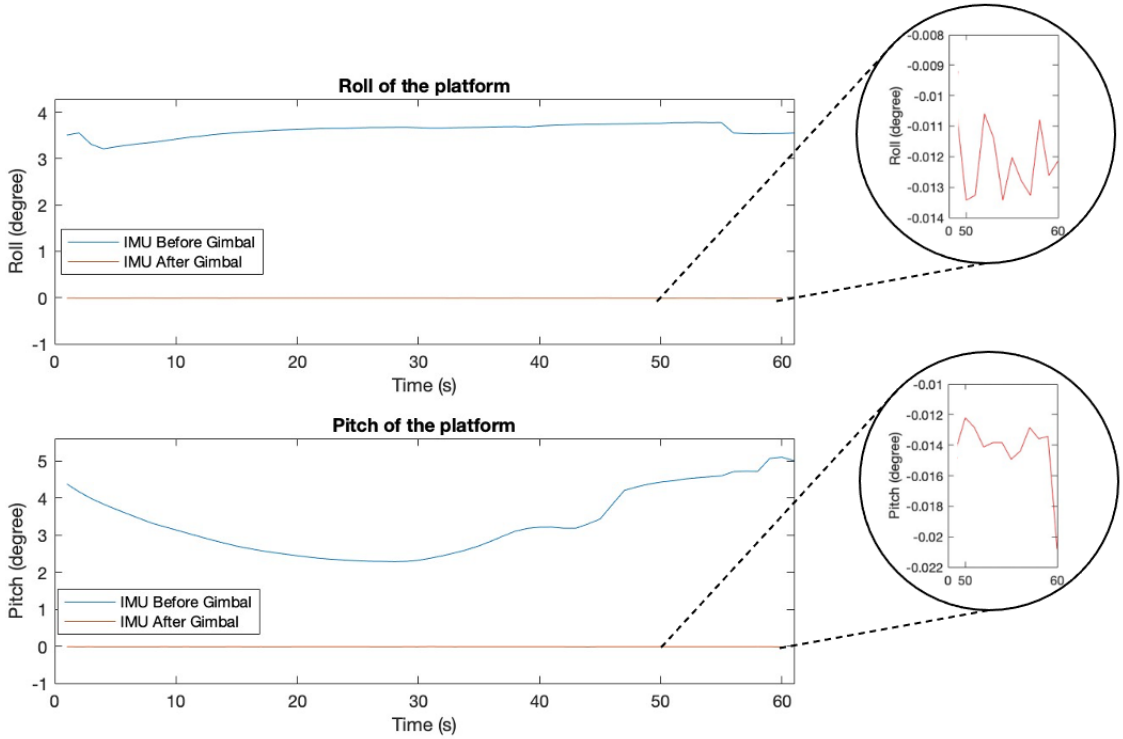


Figure 5.9: The effect of the gimbal

5.4.2.2 Gimbal Performance in Z-direction

In scenarios where a quadcopter system is tasked with the identification and precise quantification of alterations in surface elevation or ground altitude, the integration of a specialized mechanism takes on paramount importance. This mechanism is designed to meticulously carry out the task of detecting and gauging any shifts in the level of the terrain. A visual depiction of this indispensable component is provided in Figure 5.10.

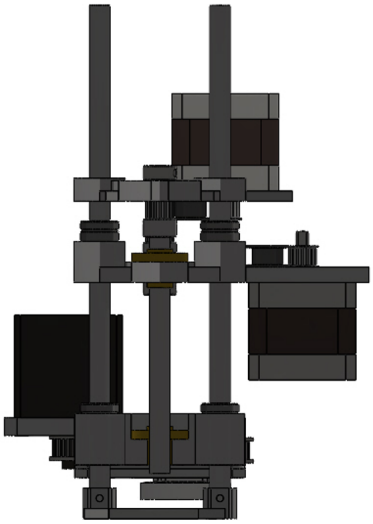


Figure 5.10: Z axis disturbance generator

This mechanism is essential for maintaining precise altitude control and compensating for errors in the z-direction induced by fluctuations in the surface. By accurately detecting and measuring changes in ground height, the quadcopter system can make necessary adjustments to ensure accurate altitude control and stable flight performance.

Figure 5.11 provides a comprehensive description of the altitude test procedure employed on the platform. This test examines the platform's performance when its Z-axis altitude remains fixed while traversing uneven terrain. Instead of adjusting the entire system's altitude, this test focuses on the gimbal's ability to adapt and modify its altitude in response to uneven ground conditions.

By altering the gimbal's altitude independently, the system can maintain a more stable and level printing surface despite the variations in the terrain. This innovative approach allows for greater precision and control during the 3D printing process, minimizing the impact of uneven ground on the quality and accuracy of the printed object.

To further validate the effectiveness of this mechanism, the 3D printing outcome utilizing the proposed approach shown in Figure 5.11 is presented. This figure serves as visual evidence of the successful implementation of the suggested method, demonstrating the resulting 3D printed object and highlighting the benefits of incorporating

the gimbal system and its altitude adjustment capability. The figure not only showcases the final print but also emphasizes the importance of the suggested mechanism in achieving desirable printing outcomes even when the platform encounters uneven surfaces.

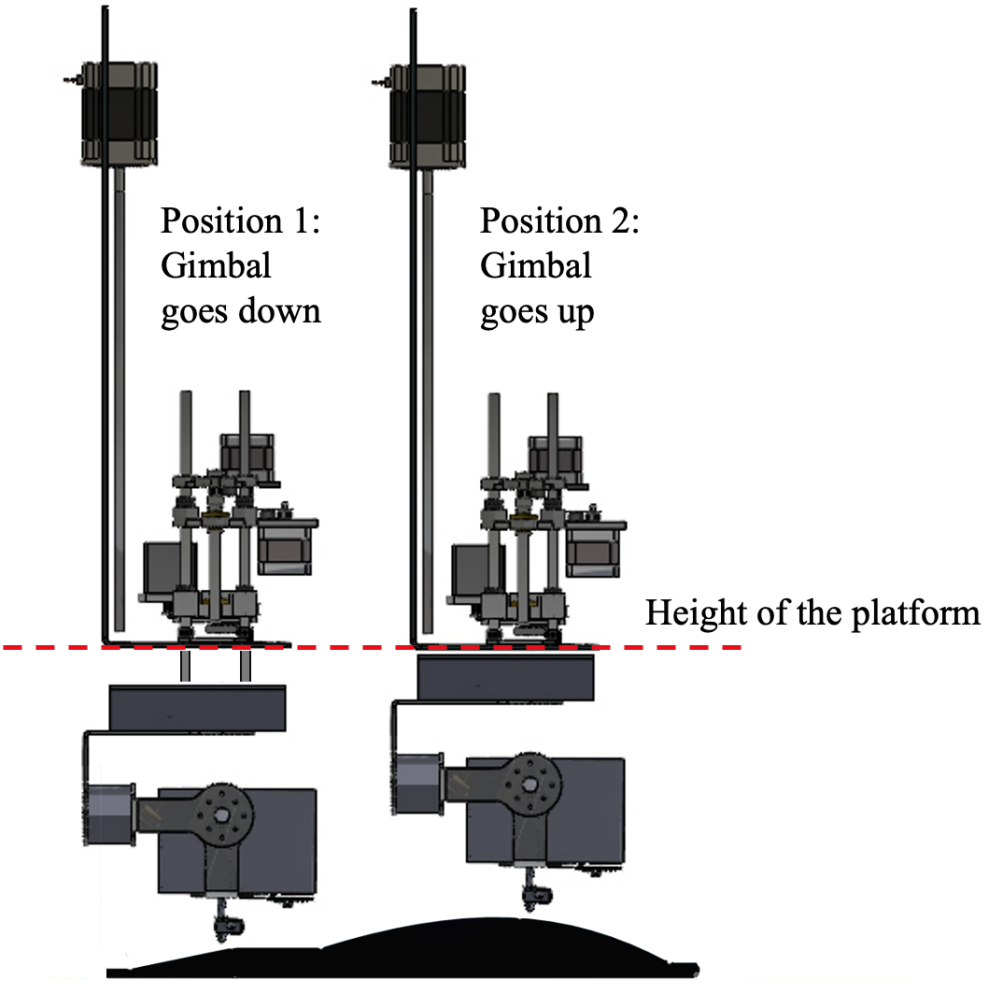


Figure 5.11: Position of the Z axis, vibration damping, and gimbal



Figure 5.12: Result of the vibration damping part test

5.4.2.3 Gimbal Performance during 3D Printing

The gimbal system's remarkable stabilization, smooth rotations across three axes, and precise brushless motors demonstrate its noteworthy capability. This ingenious system ensures harmonious movement and incredible stability.

However, intentionally stopping the gimbal system motors during the complex 3D printing process causes undesirable effects that disturb the overall harmony. This move disrupts the complex planning, casting a shadow over the entire process. The gimbal's motor symphony offers stability, but stopping it disrupts the precisely regulated rotations. This stillness increases the risk of unexpected motions, compromising the precision alignment needed for 3D printing. Thus, the delicate balance between precision and motion breaks, allowing mismatched layers and output aberrations. The gimbal system loses its reliability, leaving the intricate network of movements exposed to external forces.

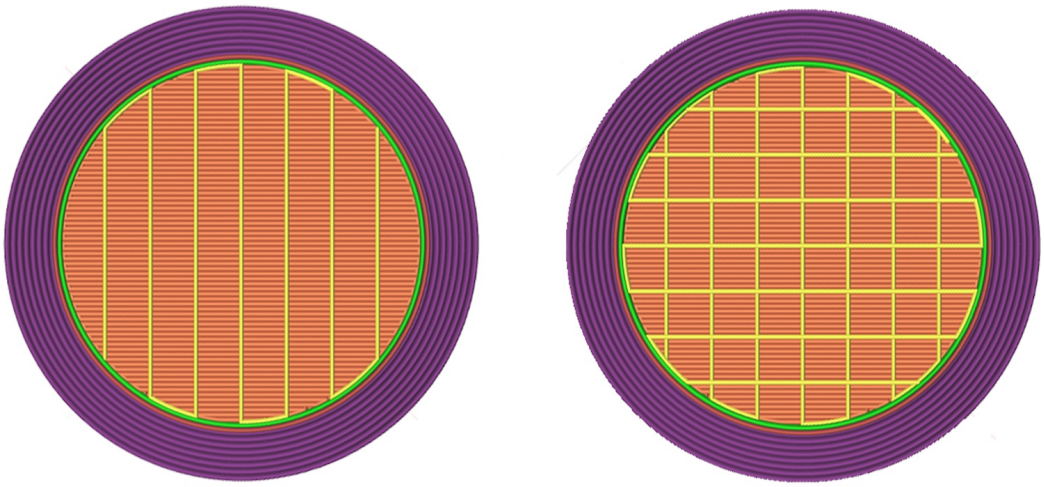
Deactivating the gimbal's motors during 3D printing has many negative effects that weaken the precision and stability that the system was rigorously engineered to achieve. Firstly, the loss of motorized control hampers the ability to dynamically adjust and fine-tune the printing parameters during the printing operation. This absence of automated control significantly hinders the flexibility and adaptability of the printing process.

Furthermore, disabling the gimbal system can potentially give rise to extrusion problems. The gimbal's motorized control plays a crucial role in ensuring consistent and accurate extrusion of the printing material. Without it, there is a higher likelihood of uneven extrusion, material flow disruptions, or even extrusion failures, leading to compromised print quality and structural integrity.

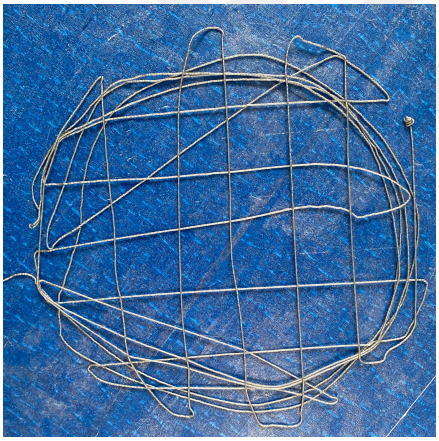
Overall, the decision to utilize a gimbal system for 3D printing should be weighed against its drawbacks. While adopting a gimbal system may offer certain advantages, such as improved stability or reduced vibrations, it is vital to acknowledge and address the abovementioned limitations. Particularly, when the goal is to achieve precise and controlled movement, these drawbacks become more significant, and alternative strategies or solutions may need to be explored to ensure optimal printing results.

Figure 5.13 and Figure 5.14 provide a comprehensive overview of the outcomes achieved in the printing test platform. For a more detailed examination of the printing

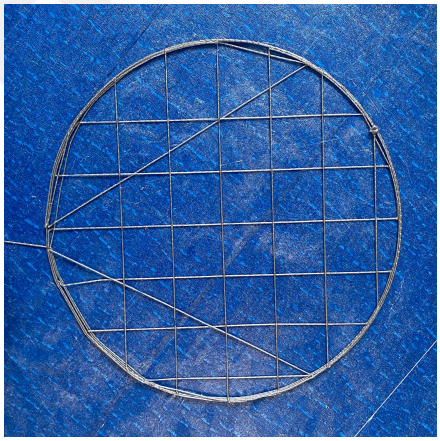
process, Figure 5.13a illustrates a cylindrical shape's 10% infill pattern. In comparison, Figure 5.14a showcases a cubic shape's 10% infill pattern, allowing for a closer inspection of the individual layers.



(a) Circle's layers; Left: First layer - Right: Second layer



(b) circle - gimbal off

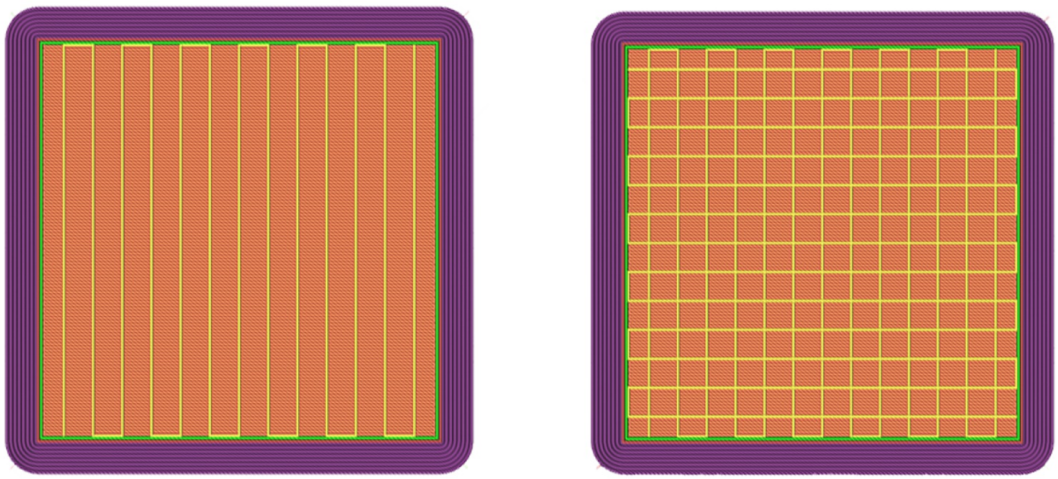


(c) circle - gimbal on

Figure 5.13: Circle method

To assess the influence of the gimbal system, a series of experiments were conducted with the system turned on and off. The corresponding results for the cylindrical shape are presented in Figures 5.13b (gimbal system turned off) and 5.13c (gimbal system turned on). In contrast, the results for the cubic shape are showcased in Figures 5.14b (gimbal system turned off) and 5.14c (gimbal system turned on). These figures serve as valuable references for evaluating the influence of the gimbal system on the 3D

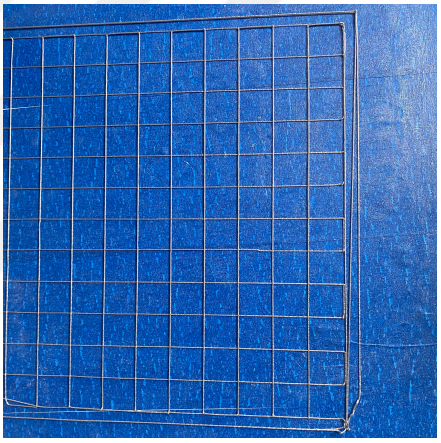
printing process and the resulting quality of both cylindrical and cubic shapes.



(a) Square's layers; Left: First layer - Right: Second layer



(b) square - gimbal off



(c) square - gimbal on

Figure 5.14: Square method

By conducting a comparative analysis between the results obtained with and without the implementation of the gimbal system, a profound understanding of the gimbal system's significance in enhancing or altering the outcome of the 3D printing process can be gained. Such comparisons enable the assessment of the extent to which the gimbal system contributes to improving the overall quality and precision of the printed objects. Furthermore, the gimbal system's role in compensating for tilting or misaligning the printing surface is another aspect that can greatly impact print quality. It allows for automatic adjustments and compensation for any irregularities or

variations in the printing surface, ensuring a consistent and uniform deposition of the printing material. This gimbal system capability can significantly reduce print defects, such as uneven layers, warping, or distortion.

In addition to print quality, the gimbal system may also play a crucial role in enhancing printing speed and efficiency. By providing stabilization and control, it enables faster and more precise movements of the print head or the printing platform, facilitating the rapid and accurate deposition of the printing material. This can increase productivity and reduce printing time, improving overall process optimization.

5.5 Conclusion

This chapter covered the experimental setup and tests carried out to assess the functionality of a quadcopter system with a gimbal for 3D printing applications.

In conclusion, the experimental results presented in this chapter demonstrate the effectiveness of the gimbal system in mitigating system errors and improving the quality and precision of 3D printing on a test platform. The experiments were divided into two control tests to evaluate the gimbal's performance.

The first test focused on simulating the quadcopter's movement on a line following a scenario using a camera mounted on the gimbal. The results showed that the gimbal followed the line under the extruder, ensuring accurate 3D printing. The second test examined the gimbal's performance in maintaining the extruder and controlling the platform's movement. Various movements were tested, including straight, inclined, and platform rotation. The gimbal effectively compensated for system vibrations and errors, resulting in precise movements and improved 3D printing outcomes. The final test demonstrated the successful 3D printing of circular and square shapes with minimal deviations from the desired patterns.

Two IMUs were strategically positioned on the platform to evaluate the gimbal's performance. The visual representation of IMU data during circular motion provided insights into the comparative behavior of the IMUs and assisted in assessing the gimbal's effectiveness.

In summary, the gimbal system significantly improves stability and reduces vibrations. Future research should address the challenges associated with the absence of

motorized control and explore alternative strategies to ensure precise and controlled movement during 3D printing. The findings presented in this chapter contribute to understanding the gimbal system's effect on 3D printing and provide insights for further optimization and refinement of the printing process.

CHAPTER 6

CONCLUSIONS

This thesis explores the integration of multirotor UAVs with 3D extruders to overcome the limitations of traditional 3D printers regarding size and production volume. The motivation behind this research is to enable the production of large and complex structures without the need for costly expansions of fixed-size 3D printers. The limitations of existing 3D printers and industrial robots in size and working area are discussed, highlighting the potential of UAVs, particularly multirotor UAVs, in overcoming these limitations. Multirotor UAVs equipped with 3D extrusion systems offer the advantage of producing or repairing structures without being constrained by size or positional conditions, making them especially useful in hazardous or inaccessible environments.

The thesis addresses various aspects related to 3D printing with multirotor UAVs. It emphasizes the design considerations for lightweight manipulator systems, such as gimbals that can be attached to UAVs to perform the 3D printing process. The need for highly accurate flight controllers to compensate for unwanted movements of the UAVs during printing operations is also highlighted. Furthermore, the concept of swarm UAVs is explored, where multiple UAVs work together autonomously to perform a specific task, offering advantages such as time savings, reduced labor, and increased operational efficiency.

Based on the conclusions drawn from each chapter, it is evident that integrating multirotor UAVs with 3D extruders holds significant potential for expanding the capabilities of 3D printing. The research findings demonstrate the feasibility and advantages of using UAVs for 3D printing, including precise and autonomous operations. The thesis contributes to the field of additive manufacturing by showcasing the potential of multirotor UAVs in overcoming the limitations of traditional 3D printers.

Future research in this area could focus on further improving the accuracy and reliability of the 3D printing process with UAVs. This could involve advancements in lightweight manipulator designs, more robust flight control algorithms, and enhanced navigation and obstacle detection systems. Additionally, exploring the scalability and coordination of swarm UAVs for additive manufacturing could be an exciting avenue for future investigation.

REFERENCES

[1] P. Latteur, S. Goessens, J.-S. Breton, J. Leplat, Z. Ma, and C. Mueller, "Drone-based additive manufacturing of architectural structures," in *Proceedings of IASS Annual Symposia*, vol. 2015, pp. 1–12, International Association for Shell and Spatial Structures (IASS), 2015.

[2] R. Geiser, *Explorations in architecture: teaching, design, research*. Birkhäuser, 2008.

[3] X. Zhang, M. Li, J. H. Lim, Y. Weng, Y. W. D. Tay, H. Pham, and Q.-C. Pham, "Large-scale 3D printing by a team of mobile robots," *Automation in Construction*, vol. 95, pp. 98–106, 2018.

[4] R. SUGIURA, N. NOGUCHI, K. ISHII, and H. TERAO, "The development of remote sensing system using unmanned helicopter," in *Automation Technology for Off-Road Equipment Proceedings of the 2002 Conference*, p. 120, American Society of Agricultural and Biological Engineers, 2002.

[5] K. Schutte, H. Sahli, D. Schrottmayer, M. Eisl, F. J. Varas, M. Bajic, M. Uppsall, and E. den Breejen, "ARC: A camcopter based mine field detection system," in *Fifth International Airborne Remote Sensing Conference*, p. 20, Citeseer, 2001.

[6] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[7] M. Kovac, "Learning from nature how to land aerial robots," *Science*, vol. 352, no. 6288, pp. 895–896, 2016.

[8] N. Kuntz and P. Oh, "Autonomous cargo transport system for an unmanned aerial vehicle, using visual servoing," *Systemics, Cybernetics and Informatics*, vol. 7, no. 6, 2009.

[9] S. Wright, "Uavs in community police work," in *Infotech@ Aerospace*, p. 6955, 2005.

[10] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*, vol. 200. Springer, 2008.

[11] M. Kovac, R. Stuart-Smith, S. Leuttenegger, M. Srinivasan, R. Ball, C. Williams, and P. Shepherd, "Aerial ABM: Aerial Additive Building Manufacturing 3D Printed Construction Using a Swarm of Aerial Robots," 2016.

[12] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2668–2673, IEEE, 2011.

[13] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "The yale aerial manipulator: grasping in flight," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2974–2975, IEEE, 2011.

[14] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspiration & biomimetics*, vol. 9, no. 2, p. 025010, 2014.

[15] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems*, pp. 545–558, Springer, 2013.

[16] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 227–240, 2013.

[17] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor uavs equipped with a servo robot arm," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 4014–4020, IEEE, 2015.

[18] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics

framework: General formulation, stability analysis, and experimental results," *Frontiers in Robotics and AI*, vol. 3, p. 16, 2016.

[19] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Coordinated control of aerial robotic manipulators: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1406–1413, 2017.

[20] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, 2014.

[21] A. Mirjan, F. Augugliaro, R. D'Andrea, F. Gramazio, and M. Kohler, "Building a bridge with flying robots," in *Robotic fabrication in architecture, art and design 2016*, pp. 34–47, Springer, 2016.

[22] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," *Proc. Robotics: Science & Systems VII*, vol. 7, 2011.

[23] B. Dams, S. Sareh, K. Zhang, P. Shepherd, M. Kovac, and R. J. Ball, "Aerial additive building manufacturing: three-dimensional printing of polymer structures using drones," *Proceedings of the Institution of Civil Engineers-Construction Materials*, vol. 173, no. 1, pp. 3–14, 2020.

[24] P. Chermprayong, K. Zhang, F. Xiao, and M. Kovac, "An integrated delta manipulator for aerial repair: A new aerial robotic system," *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 54–66, 2019.

[25] J. Willmann, F. Augugliaro, T. Cadalbert, R. D'Andrea, F. Gramazio, and M. Kohler, "Aerial robotic construction towards a new field of architectural research," *International journal of architectural computing*, vol. 10, no. 3, pp. 439–459, 2012.

[26] G. Hunt, F. Mitzalis, T. Alhinai, P. A. Hooper, and M. Kovac, "3d printing with flying robots," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 4493–4499, IEEE, 2014.

[27] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *2011 IEEE international conference on robotics and automation*, pp. 2491–2498, IEEE, 2011.

[28] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4645–4651, IEEE, 2011.

[29] A. Q. Keemink, M. Fumagalli, S. Stramigioli, and R. Carloni, "Mechanical design of a manipulation system for unmanned aerial vehicles," in *2012 IEEE international conference on robotics and automation*, pp. 3147–3152, IEEE, 2012.

[30] L. Marconi, R. Naldi, and L. Gentili, "Modelling and control of a flying robot interacting with the environment," *Automatica*, vol. 47, no. 12, pp. 2571–2583, 2011.

[31] L. Gentili, R. Naldi, and L. Marconi, "Modeling and control of vtol uavs interacting with the environment," in *2008 47th IEEE Conference on decision and control*, pp. 1231–1236, IEEE, 2008.

[32] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4990–4995, IEEE, 2013.

[33] J. L. Scholten, M. Fumagalli, S. Stramigioli, and R. Carloni, "Interaction control of an uav endowed with a manipulator," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4910–4915, IEEE, 2013.

[34] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3532–3537, IEEE, 2012.

[35] C. Korpela, M. Orsag, and P. Oh, "Hardware-in-the-loop verification for mobile manipulating unmanned aerial vehicles," *Journal of intelligent & robotic systems*, vol. 73, no. 1, pp. 725–736, 2014.

[36] T. W. Danko and P. Y. Oh, "Evaluation of visual servoing control of aerial manipulators using test gantry emulation," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 821–829, IEEE, 2014.

[37] T. W. Danko and P. Y. Oh, "A hyper-redundant manipulator for mobile manipulating unmanned aerial vehicles," in *2013 international conference on unmanned aircraft systems (ICUAS)*, pp. 974–981, IEEE, 2013.

[38] K. Kondak, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach, M. Laiacker, I. Maza, A. Rodriguez-Castano, and A. Ollero, "Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 145, 2013.

[39] F. Huber, K. Kondak, K. Krieger, D. Sommer, M. Schwarzbach, M. Laiacker, I. Kossyk, S. Parusel, S. Haddadin, and A. Albu-Schäffer, "First analysis and experiments in aerial manipulation using fully actuated redundant robot arm," in *2013 IEEE/RSJ international conference on intelligent robots and systems*, pp. 3452–3457, IEEE, 2013.

[40] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero, "Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 2107–2112, IEEE, 2014.

[41] V. Lippiello and F. Ruggiero, "Exploiting redundancy in cartesian impedance control of uavs equipped with a robotic arm," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3768–3773, IEEE, 2012.

[42] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a uav with a robotic arm," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 704–709, 2012.

[43] M. Kobilarov, "Nonlinear trajectory control of multi-body aerial manipulators," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 679–692, 2014.

[44] I. Palunko and R. Fierro, "Adaptive control of a quadrotor with dynamic

changes in the center of gravity," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2626–2631, 2011.

[45] C. Korpela, M. Orsag, M. Pekala, and P. Oh, "Dynamic stability of a mobile manipulating unmanned aerial vehicle," in *2013 IEEE international conference on robotics and automation*, pp. 4922–4927, IEEE, 2013.

[46] F. Pierrot, C. Reynaud, and A. Fournier, "Delta: a simple and efficient parallel robot," *Robotica*, vol. 8, no. 2, pp. 105–109, 1990.

[47] D. Carp-Ciocardia *et al.*, "Dynamic analysis of clavel's delta parallel robot," in *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, vol. 3, pp. 4116–4121, IEEE, 2003.

[48] C.-S. Y. C.-S. Yoo and I.-K. A. I.-K. Ahn, "Low cost gps/ins sensor fusion system for uav navigation," in *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*, vol. 2, pp. 8–A, IEEE, 2003.

[49] M. Barczyk and A. F. Lynch, "Integration of a triaxial magnetometer into a helicopter uav gps-aided ins," *IEEE transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2947–2960, 2012.

[50] B. Yun, K. Peng, and B. M. Chen, "Enhancement of gps signals for automatic control of a uav helicopter system," in *2007 IEEE International Conference on Control and Automation*, pp. 1185–1189, IEEE, 2007.

[51] G. Qingbo, L. Nan, and L. Baokui, "The application of gps/sins integration based on kalman filter," in *Proceedings of the 31st Chinese Control Conference*, pp. 4607–4610, IEEE, 2012.

[52] Z. Tao and W. Lei, "Sins and gps integrated navigation system of a small unmanned aerial vehicle," in *2008 International Seminar on Future BioMedical Information Engineering*, pp. 465–468, IEEE, 2008.

[53] C. V. Angelino, V. R. Baraniello, and L. Cicala, "High altitude uav navigation using imu, gps and camera," in *Proceedings of the 16th International Conference on Information Fusion*, pp. 647–654, IEEE, 2013.

[54] Y. Jiong, Z. Lei, D. Jiangping, S. Rong, and W. Jianyu, "Gps/sins/baro integrated navigation system for uav," in *2010 International Forum on Information Technology and Applications*, vol. 3, pp. 19–25, IEEE, 2010.

[55] H. Nakanishi, S. Kanata, and T. Sawaragi, "Gps-ins-baro hybrid navigation system taking into account ground effect for autonomous unmanned helicopter," in *2012 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pp. 1–6, IEEE, 2012.

[56] H. Nakanishi, S. Kanata, and T. Sawaragi, "Measurement model of barometer in ground effect of unmanned helicopter and its application to estimate terrain clearance," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 232–237, IEEE, 2011.

[57] A. Beyeler, J.-C. Zufferey, and D. Floreano, "3d vision-based navigation for indoor microflyers," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1336–1341, IEEE, 2007.

[58] S. Hrabar, "3d path planning and stereo-based obstacle avoidance for rotorcraft uavs," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 807–814, IEEE, 2008.

[59] Y. Gao, X. Ai, J. Rarity, and N. Dahnoun, "Obstacle detection with 3d camera using uv-disparity," in *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, pp. 239–242, IEEE, 2011.

[60] S. Saha, A. Natraj, and S. Waharte, "A real-time monocular vision-based frontal obstacle detection and avoidance for low cost uavs in gps denied environment," in *2014 IEEE international conference on aerospace electronics and remote sensing technology*, pp. 189–195, IEEE, 2014.

[61] I. Golightly and D. Jones, "Visual control of an unmanned aerial vehicle for power line inspection," in *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pp. 288–295, IEEE, 2005.

[62] B. Kada and Y. Ghazzawi, "Robust pid controller design for an uav flight control system," in *Proceedings of the World congress on Engineering and Computer Science*, vol. 2, pp. 1–6, 2011.

[63] G. T. Martin, “Modelling and control of the parrot ar.drone,” 2012.

[64] S. Suzuki, T. Ishii, N. Okada, K. Iizuka, and T. Kawamura, “Autonomous navigation, guidance and control of small electric helicopter,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 54, 2013.

[65] D. Lee, T. Ryan, and H. J. Kim, “Autonomous landing of a vtol uav on a moving platform using image-based visual servoing,” in *2012 IEEE international conference on robotics and automation*, pp. 971–976, IEEE, 2012.

[66] G. Limnaios and N. Tsourveloudis, “Fuzzy logic controller for a mini coaxial indoor helicopter,” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 187–201, 2012.

[67] M. Bergerman, O. Amidi, J. R. Miller, N. Vallidis, and T. Dudek, “Cascaded position and heading control of a robotic helicopter,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 135–140, IEEE, 2007.

[68] G. H. Elkaim, F. A. P. Lie, and D. Gebre-Egziabher, “Principles of guidance, navigation, and control of uavs,” *Handbook of unmanned aerial vehicles*, pp. 347–380, 2015.

[69] M. A. Olivares-Méndez, I. F. Mondragón, P. Campoy, and C. Martínez, “Fuzzy controller for uav-landing task using 3d-position visual estimation,” in *International Conference on Fuzzy Systems*, pp. 1–8, Ieee, 2010.

[70] M. Andrew, W. Sanders, and F. Leavenworth, “Drone swarms—a monograph by school of advanced military studies,” 2017.

[71] P. Vincent and I. Rubin, “A framework and analysis for cooperative search using uav swarms,” in *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 79–86, 2004.

[72] N. J. P. Scerri and T. Sugawara, “Massively multi-agent technology,”

[73] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, “Decentralized control of unmanned aerial vehicle collab-

orative sensing missions," in *2007 American Control Conference*, pp. 4672–4677, IEEE, 2007.

[74] F. Koenig and D. Wong, "Real-time kinematics global positioning system (gps) operation and setup method for the synchronous impulse reconstruction (sire) radar," tech. rep., ARMY RESEARCH LAB ADELPHI MD SENSORS AND ELECTRON DEVICES DIRECTORATE, 2010.

[75] R. N. Jazar, *Theory of applied robotics: kinematics, dynamics, and control*. Springer., 2010.

[76] Y. Song, *Intelligent PID controller based on fuzzy logic control and neural network technology for indoor environment quality improvement*. PhD thesis, University of Nottingham, 2014.

[77] H. ElKholy, "Dynamic modeling and control of a quadrotor using linear and nonlinear approaches," *American University in Cairo*, 2014.

[78] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *2007 IEEE/RSJ international conference on intelligent robots and systems*, pp. 153–158, Ieee, 2007.

[79] H. ElKholy, "Dynamic modeling and control of a quadrotor using linear and nonlinear approaches," *American University in Cairo*, 2014.

[80] D. Hartman, K. Landis, M. Mehrer, S. Moreno, J. Kim, and B. Chang, "Quad-copter dynamic modeling and simulation (quad-sim)," *Senior DesignProject, Drexel University*, 2014.

[81] G. Papari and N. Petkov, "Edge and line oriented contour detection: State of the art," *Image and Vision Computing*, vol. 29, no. 2-3, pp. 79–103, 2011.

[82] G. Papari and N. Petkov, "Edge and line oriented contour detection: State of the art," *Image and Vision Computing*, vol. 29, no. 2-3, pp. 79–103, 2011.

[83] P. V. Hough, "Method and means for recognizing complex patterns," *US patent*, vol. 3, no. 6, 1962.

[84] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[85] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[86] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The international journal of robotics research*, vol. 21, no. 4, pp. 331–344, 2002.

[87] E. U. Acar, H. Choset, and J. Y. Lee, "Sensor-based coverage with extended range detectors," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 189–198, 2006.

[88] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 1, pp. 27–32, IEEE, 2001.

[89] Y. Li, H. Chen, M. J. Er, and X. Wang, "Coverage path planning for uavs based on enhanced exact cellular decomposition method," *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011.

[90] I. Maza and A. Ollero, "Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*, pp. 221–230, Springer, 2007.

[91] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 1, pp. 27–32, IEEE, 2001.

[92] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

[93] G. S. Avellar, G. A. Pereira, L. C. Pimenta, and P. Iscold, "Multi-uav routing for area coverage and remote sensing with minimum time," *Sensors*, vol. 15, no. 11, pp. 27783–27803, 2015.

[94] E. Kuantama, R. Tarca, S. Dzitac, I. Dzitac, T. Vesselenyi, and I. Tarca, "The design and experimental development of air scanning using a sniffer quadcopter," *Sensors*, vol. 19, no. 18, p. 3849, 2019.

[95] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (uavs) for traffic monitoring," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 221–234, IEEE, 2013.

[96] A. Olejnik, R. Rogólski, Ł. Kiszkowiak, and M. Szcześniak, "Specific problems of selecting and integrating equipment components in the course of developing a technology demonstrator for the mini-uav," in *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 284–289, IEEE, 2019.

[97] A. Olejnik, Ł. Kiszkowiak, R. Rogólski, G. Chmaj, M. Radomski, M. Majcher, and Ł. Omen, "Precise remote sensing using unmanned helicopter," in *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 544–548, IEEE, 2019.

[98] M. Waliszkiewicz, K. Wojtowicz, and Z. Rochala, "Experimental method of controller tuning for quadcopters," in *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 165–170, IEEE, 2019.

[99] B. Brzozowski, P. Daponte, L. De Vito, F. Lamonaca, F. Picariello, M. Pompetti, I. Tudosa, and K. Wojtowicz, "A remote-controlled platform for uas testing," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 8, pp. 48–56, 2018.

[100] P. Daponte, F. Lamonaca, F. Picariello, M. Riccio, L. Pompetti, and M. Pompetti, "A measurement system for testing light remotely piloted aircraft," in *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 397–402, IEEE, 2017.

[101] P. Daponte, L. De Vito, F. Lamonaca, F. Picariello, M. Riccio, S. Rapuano, L. Pompetti, and M. Pompetti, "Dronesbench: An innovative bench to test

drones," *IEEE Instrumentation & Measurement Magazine*, vol. 20, no. 6, pp. 8–15, 2017.

[102] N. Faundes, V. Wunsch, S. Hohnstein, B. Glass, and M. Vetter, "Research paper on the topic of different uav drive train qualification and parameter sets," in *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, pp. 6E3–1, IEEE, 2013.

[103] B. Smith, B. Stark, T. Zhao, and Y. Chen, "An outdoor scientific data drone ground truthing test site," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 436–443, IEEE, 2015.

[104] J. P. How, B. Behihke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 51–64, 2008.

[105] V. Vladimerou, A. Stubbs, J. Rubel, A. Fulford, J. Strick, and G. Dullerud, "A hovercraft testbed for decentralized and cooperative control," in *Proceedings of the 2004 American Control Conference*, vol. 6, pp. 5332–5337, IEEE, 2004.

[106] D. Howard and T. Merz, "A platform for the direct hardware evolution of quadcopter controllers," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4614–4619, IEEE, 2015.

[107] I. Sharf, M. Nahon, A. Harmat, W. Khan, M. Michini, N. Speal, M. Trentini, T. Tsadok, and T. Wang, "Ground effect experiments and model validation with draganflyer x8 rotorcraft," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1158–1166, IEEE, 2014.

[108] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (starmac)," in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576)*, vol. 2, pp. 12–E, IEEE, 2004.

[109] H. J. Wright, R. Strydom, and M. V. Srinivasan, "A generalized algorithm for tuning uas flight controllers," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1369–1378, IEEE, 2018.