Backstepping Sliding Mode-based Trajectory Control of a Quadrotor UAV

Yohannes Lisanewerk Mulualem

A Thesis Submitted to the department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

July, 2023

Adama, Ethiopia

Backstepping Sliding Mode-based Trajectory Control of a Quadrotor UAV

Yohannes Lisanewerk Mulualem

Advisor: Prof. Gang Gyoo Jin

A Thesis Submitted to the department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

July, 2023

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "**Backstepping Sliding Mode-based Trajectory Control of a Quadrotor UAV**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Name of the student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Backstepping Sliding Mode-based Trajectory Control of a Quadrotor UAV**" prepared under my guidance by **Yohannes Lisanewerk** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Electrical Power and Control Engineering (Control Engineering). Therefore, I recommend the submission of revised version of the thesis to the department following the applicable procedures.

____________________	_________________	_________
Major Advisor	Signature	Date

APPROVAL PAGE

I, the advisor of the thesis entitled "**Backstepping Sliding Mode-based Trajectory Control of a Quadrotor UAV**" and developed by **Yohannes Lisanewerk**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

____________________	________________	________
Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by **Yohannes Lisanewerk** have read and evaluated the thesis entitled "**Backstepping Sliding Mode-based Trajectory Control of a Quadrotor UAV**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Electrical Power and Control Engineering (Control Engineering).

______________________	________________	________________
Chairperson	Signature	Date
______________________	________________	________________
Internal Examiner	Signature	Date
______________________	________________	________________
External Examiner	Signature	Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

______________________	________________	________________
Department Head	Signature	Date
______________________	________________	________________
School Dean	Signature	Date
______________________	________________	________________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENTS

Firstly, I express my gratitude to the Almighty God for granting me the fortitude, chance, and enabling me to embark on and successfully accomplish this thesis. Additionally, I extend my appreciation to Adama Science and Technology University, specifically the School of Electrical Engineering and Computing, as well as the Power and Control Engineering department, for providing me with the opportunity to participate in this Master's program.

I am filled with immense joy as I convey my profound appreciation to **Professor Gang Gyoo Jin**, my advisor and mentor, for his sincere support and continuous guidance. His unwavering assistance, insightful advice, and constant inspiration have been of immeasurable value throughout the completion of this thesis, and I am extremely grateful for his contributions.

Thank you.

Yohannes Lisanewerk Mulualem

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

Abbreviations	Description
ADRC	Active Disturbance Rejection Control
ASTU	Adama Science and Technology University
BSC	Backstepping Control
BSMC	Backstepping Sliding Mode Control
COG	Center of Gravity
DC	Direct Current
DOF	Degree of Freedom
ESC	Electronic Speed Controller
FBL	Feedback Linearization
GPS	Global Positioning System
IAE	Integral Absolute Error
IBC	Integral Backstepping Control
ISE	Integral Squared Error
ITAE	Integral Time Absolute Error
ITSE	Integral Time Squared Error
LiPo	Lithium Polymer
LMI	Linear Matrix Inequality
LNMPC	Linear-Nonlinear Model Predictive Control
LQG	Linear Quadratic Gaussian

LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MIMO	Multiple Input Multiple Output
MPC	Model Predictive Control
NMPC	Non-Linear Model Predictive Control
PC	Personal Computer
PD	Proportional Derivative
PD^2	Second Order Differentiator
PID	Proportional, Integral, and Derivative
PSO	Particle Swarm Optimization
QUAV	Quadrotor type Unmanned Aerial Vehicle
RBF	Radial Basis Function
SMC	Sliding Mode Controller
STI	Space Technology Institute
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicles
VSC	Variable Structure Control
VTOL	Vertical Take-Off Landing
WG	Wind Gust

NOMENCLUTURE

C_D	Aerodynamic drag coefficient
C_D	Aerodynamic thrust coefficient
Ω_i	Angular velocity of ith rotor
η^I	Attitude Vector of Quadrotor in inertial frame
K_D	Drag constant
ϕ, θ, ψ	Euler angles
$\dot{\phi}, \dot{\theta}, \dot{\psi}$	Euler angles rates
θ	Euler pitch angle
ϕ	Euler roll angle
ψ	Euler yaw angle
f_i	Force generated by ith rotor
g	Gravitational acceleration
G	Gravitational force expressed in body frame
τ_{gyro}	Gyroscopic moments due to rotor inertia
I	Inertia matrix
x_i	ith system state
l	Length of the moment arm
m	Mass of the quadrotor
τ_{net}^{B}	Net moment generated by the propeller
ξ^I	Position Vector of Quadrotor in inertial frame

Ω_r	Relative speed of the propellers
R	Rotation matrix from body to inertial frame
R^{-1}	Rotation matrix from inertial to body frame
J_r	Rotor inertia
K_T	Thrust constant
τ_M	Torque generated by i^{th} rotor
T_{net}^{B}	Total thrust force generated by the propeller
e_i	Tracking error for ith state
T_M	Transformation matrix
V^B	Translational linear velocity vector in body frame
P, Q, R	Angular velocities in body frame
T, τ_ϕ, τ_θ, τ_ψ	Control inputs
v_x, v_y, v_z	Linear velocities in body frame
I_{xx}, I_{yy}, I_{zz}	Moments of inertia about principal axis in body frame

ABSTRACT

Technological advancements and cost reductions have enabled the emergence of unmanned aerial vehicles (UAVs), which are aircraft capable of flying without the need for a pilot. These UAVs have opened up various possibilities such as aerial photography, video recording, mapping, pollution and land monitoring, power-line inspection, firefighting, agricultural applications, military operations and more. Due to its unique characteristics, such as strong coupling of subsystems, unidentified physical properties, nonparametric uncertainties in inputs, and external disruptions, this vehicle is particularly well-liked in the scientific community. In this research work, a dynamic model of a quadrotor UAV and actuators is developed and backstepping sliding mode control (BSMC) technique is applied to control trajectory of a quadrotor UAV and asymptotic stability of the designed controller is achieved using the Lyapunov stability analysis. The designed control structure incorporates both a backstepping and sliding mode control method to adequately reduce the chattering effect of sliding mode control by differential iteration, the switching gain of sliding mode control is developed during the backstepping design process. Particle Swarm Optimization (PSO) is used to optimize the controller parameters. Finally, the performance of the designed controller is tested under variable load and time varying wind gust for a different trajectory in MATLAB software tool. The simulation results show that the quadrotor can handle a load varies from 0.5kg to 18 kg under normal conditions and a load varies from 0.5kg to 12kg when the quadrotor is subjected to a time varying wind gust. Also, the chattering effect due to SMC is reduce effectively. Additionally, performance of BSMC is compared with SMC and BSC to show the effectiveness of the proposed controller.

Keywords: - *BSC, BSMC, Lyapunov stability, PSO, Quadrotor, SMC, UAVs.*

CHAPTER ONE

1. INTRODUCTION

1.1 BACKGROUND OF THE STUDY

UAV stands for Unmanned Aerial Vehicle. It is also commonly known as a drone. UAVs are aircraft that are flown without a human pilot onboard, instead being controlled remotely by a pilot on the ground or through pre-programmed instructions. They can range in size from small, handheld models to large, military-grade drones. UAVs have become increasingly popular in recent years due to their versatility and ability to perform tasks that may be difficult or dangerous for human pilots (Idrissi, Salami, and Annaz 2022a).

Some of the key components of a UAV include a flight control system, a navigation system, and a communication link between the ground station and the aircraft. The flight control systems use sensors and software to maintain the UAVs stability and control its movement, while the navigation system uses GPS and other sensors to guide the aircraft to its destination. The communication link allows the pilot to control the UAV and receive real-time data and video from the aircraft.

A quadrotor is a type of rotary-winged unmanned aerial vehicle that is capable of taking off and landing vertically. In order to provide propulsion, thrust, this aerial vehicle has four motors that send power to propellers. The engine speed is changed to control and balance this vehicle. Researchers and engineers have recently paid a lot of attention to quadrotors because of its multiple potential uses in both military and civilian sectors. The quadrotor offers a broad range of abilities despite being intensively nonlinear, MIMO, highly coupled, and underactuated, a precise and practical model is crucial to control the vehicle which seems to be simple to operate. As a rotorcraft, the dynamics of a quadrotor is mainly dominated by the complicated aerodynamic effects of the rotors (Rinaldi, Primatesta, and Guglieri 2023).

Their broad range of abilities includes VTOL, hovering, and performing daring acrobatics. Additionally, its compactness, lightness, and capacity to transport payloads that are substantially heavier than its weight. Quadrotors have a variety of uses, including surveillance (Patel et al. 2013), search and rescue (Dev and Murthy 2015), building inspections (Teixeira et al. 2014), the military (Gadda, Jinay S., and Rajaram D. Patil 2013), and more. Quadrotor type UAVs have been the subject of extensive research and development across various domains. These vehicles, with their unique capabilities and maneuverability, have gained significant attention in both academic and industrial research communities and due to the sharp increase in quadrotor applications, a strong enough controller is essential to allow the quadrotor to properly complete the specified tasks.

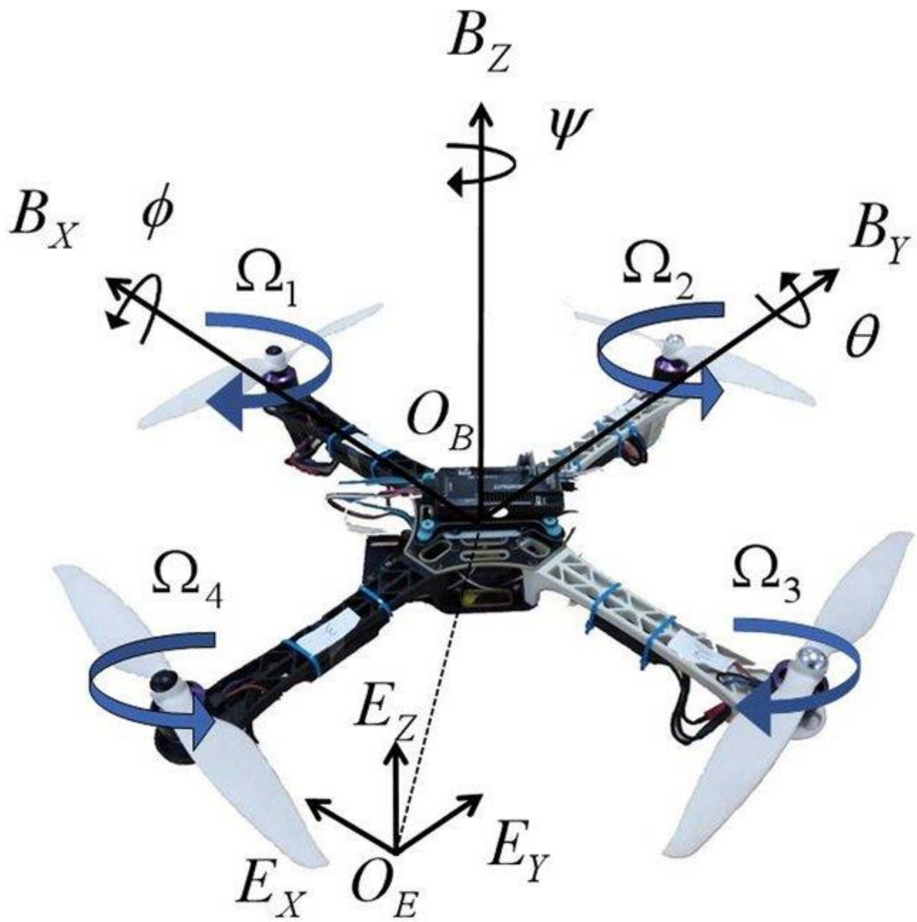


Figure 1.1 Quadrotor system [L1]

It is possible to model the dynamics of quadrotors using either the scalar Euler-Lagrange formulation or the vectorial Newton-Euler formulation. The ASTU Space Technology Institute (STI) center of excellence is currently conducting drone research, and its staff is designing quadrotor structural drones. The design of an appropriate control system is the primary difficulty and complex system of any UAV. Numerous studies utilizing both traditional and contemporary control techniques have been conducted on quadrotor altitude, attitude, and position controls (Lidiya Girma 2021).

From the perspective of control theory, the design of quadrotor controllers must overcome at least two significant obstacles. First, MIMO unstable nonlinear systems, such as quadrotors, have many inputs and numerous outputs. Due to its six degrees of freedom (DOFs) and four actuators, such a control system is coupled and under-actuated. Second, like other UAV types, quadrotors are constantly vulnerable to both internal and external disturbances, model uncertainty, and parametric perturbations. Robust controllers are frequently created to lessen the effects of disturbances and uncertainties in order to guarantee the stability of quadrotors.

Extensive research and practical implementation have focused on developing effective control techniques for quadrotors, resulting in various applications. To simplify the process, quadrotors are typically linearized around specific equilibrium conditions, such as a hover point or a tracking point. Consequently, numerous linear control algorithms, including proportional-integral-derivative (PID) control and linear quadratic regulator/Gaussian (LQR/LQG) control, have been utilized to achieve control objectives.

Traditional linear controllers for quadrotors, which employ linear approximations of the system dynamics based on one or multiple operational points, are highly desirable in practical autopilot designs. However, these approaches encounter performance degradation when the aircraft deviates from its designated operating points. This is due to the inherent nonlinearity and strong coupling characteristics of quadrotors, which impose several limitations on traditional linear control techniques (Hawi Aboma 2022). When it comes to minimizing the impacts of nonlinearity and having the capacity to reject disturbances, advanced controllers like sliding mode controllers (SMC) are useful. However, because of the chattering effect, it has limits.

In this research work, a backstepping sliding mode control (BSMC) is implemented to design a trajectory control system for stabilizing, tracking, and position control of a quadrotor aerial robot subjected to time varying disturbances and parametric uncertainties by considering the dynamic model of the motors. The proposed system is implemented in the MATLAB software tool.

1.2 STATEMENT OF THE PROBLEM

The capacity to fly in any direction, take off and land vertically, and hover at the proper altitude are the main benefits of quadrotor UAVs. Because it has unique characteristics such as strongly linked non-linear difficulties, instability, multivariable nature, potentially non-minimum phase, underactuated, presence of parameter uncertainties, and external disturbances, it makes it difficult to control quadrotor UAV.

Among these techniques, sliding mode control (SMC) was commonly employed in the development of the trajectory control systems for a quadrotor UAVs. The reason for its popularity is its ability to react quickly and its insensitivity to model flaws, uncertainties in parameters, and other disturbances. However, SMC suffers from certain drawbacks, such as the chattering effect, which leads to reduced control precision, increased wear on moving mechanical components, and significant heat control precision, increased wear on moving mechanical components, and significant heat losses on electrical power circuits. Furthermore, previous research often neglected the consideration of actuator dynamics.

Therefore, a trajectory control system based on backstepping sliding mode control is designed which takes the actuator dynamics and limitations in to account. This approach aims to minimize the chattering effect induced by SMC. The proposed controller combines the sliding mode controller with the backstepping technique to reduce input chattering and dynamically estimate the upper bound of the sliding mode switching gain in real-time. Finally, to show the effectiveness of the proposed controller comparison with backstepping and sliding mode control is done.

1.3 OBJECTIVES

1.3.1 General Objectives

The overall goal of this research work is to design a trajectory control for a quadrotor UAV employing a backstepping sliding mode technique.

1.3.2 Specific Objectives

The specific objectives of this thesis are:

- To model the dynamics of quadrotor UAV including the actuators.
- To design a backstepping sliding mode control for a quadrotor UAV system.
- To simulate the feedback system and draw relevant conclusions and recommendations from the results of the simulation.

1.4 SCOPE OF THE THESIS

The scope of this research work is to drive a mathematical model of a quadrotor system and the actuator and design a BSMC for trajectory control of a quadrotor considering time varying external disturbances and parametric uncertainties in a mass using MATLAB software tool.

1.5 LIMITATION OF THE THESIS

The thesis will focus exclusively on the mathematical design and simulation of the system using the MATLAB software tool. Its purpose is to showcase the complex interactions between the various components of the system. The decision to limit the scope to simulation arises from challenges associated with the actual deployment of the system, such as the significant time and cost implications involved.

1.6 MOTIVATION OF STUDY

The advancements and introduction of quadrotor drone technology have opened up a plenty of opportunities in different domains. These versatile vehicles have demonstrated their efficacy in fields such as photography, search and rescue, agriculture, and infrastructure inspection. The design of sophisticated control algorithms plays a crucial role in maximizing the potential of these advanced systems, enabling precise control, efficient operation, and safe execution of tasks.

1.7 SIGNIFICANCE OF THE STUDY

The study of controlling the position, altitude, and orientation of unmanned aerial vehicles (UAVs) through control systems is highly fascinating. The research programs focusing on this area are captivating due to the diverse range of applications and advantages associated with multirotor UAVs. Many countries are currently engaged in conducting these research endeavors.

Additionally, the ASTU-based Space Technology Institute (STI) is currently engaged in the advancement of quadrotor drones. The key focus of their work lies in creating a robust controller capable of effectively handling external disturbances and uncertainties in parameters. This research holds significant importance as it addresses the control challenges that arise during the drone's development stage.

1.8 THESIS ORGANIZATION

The structure of the thesis is as follows:

Chapter 2: This section provides overview of UAVs, quadrotors, nonlinear controllers and finally examines prior researches that has been done to control quadrotors with their merits and demerits.

Chapter 3: In this section using the Euler-Newton formalism, the nonlinear dynamic model of the quadrotor system is constructed with its actuator dynamics. Subsequently, the modelled system is represented in state space form and simulation the open-loop model of the quadrotor system is done.

Chapter 4: In this section design of the backstepping sliding mode controller (BSMC) for the nonlinear dynamic model of the quadrotor system is done to make sure that the quadrotor is following the given trajectory.

Chapter 5: This section presented findings from the simulation along with a comprehensive examination of the outcomes.

Chapter 6: The final section of this thesis presents a summary and suggests future directions for the researchers.

CHAPTER TWO

2. LITERATURE REVIEW

In this chapter, a brief description, history, and classification of the UAV is presented. Then the history, components and operation of the quadrotor-type UAVs are discussed. Additionally, the concept and theoretical background of backstepping and sliding mode control with Particle Swarm Optimization (PSO) technique is presented. Finally, closely related works along with their merits and demerits are reviewed and discussed.

2.1 UNMANNED AERIAL VEHICLES (UAV)

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are aircraft that operate without a human pilot on board. They are controlled either remotely by a human operator or autonomously through pre-programmed flight plans or artificial intelligence algorithms. UAVs come in various sizes and configurations, ranging from small handheld devices to large military-grade aircraft.

Unmanned aerial vehicles (UAVs) are also a part of unmanned aircraft systems (UAS), which also feature the addition of a controller on the ground and a communications network with the UAV. UAVs' flight can be controlled remotely by a human pilot, or it can have varying degrees of autonomy, such as autopilot support, up to autonomous aircraft that do not allow for human interference (Hawi Aboma 2022).

2.1.1 History of UAV

The aerial torpedoes, which were first produced in 1916 by Lawrence and Sperry from the United States, were initially known as UAVs (Unmanned Aerial Vehicles). These UAVs, depicted in Figure 2.1, demonstrated a flight range of 30 miles. It is believed that Lawrence and Sperry employed a gyroscope to maintain the stability of the aircraft (Lidiya Girma 2021).

Figure 2.1 Lawrence [L2]

2.1.2 Classification of UAVs

Depending on their size, use, range, and capabilities, unmanned aerial vehicles (UAVs) can be categorized in a variety of ways (Singhal, Bansod, and Mathew 2018). Here are a few typical categories for UAVs:

- Size-based classification:
    - Micro UAVs: Less than 100 grams in weight.
    - Mini UAVs: Between 100 grams and 2 kilograms in weight.
    - Small UAVs: Between 2 and 25 kilograms in weight.
    - Medium UAVs: Between 25 and 150 kilograms in weight.
    - Large UAVs: More than 150 kilograms in weight.
- Purpose-based classification:
    - Civilian UAVs: Used for commercial and non-military purposes such as aerial photography, surveillance, mapping, and agriculture.

    - Military UAVs: Used for military purposes such as reconnaissance, surveillance, target acquisition, and attack.

- Range-based classification:
    - Close-range UAVs: Operate within a range of 1-2 km.
    - Short-range UAVs: Operate within a range of 2-25 km.
    - Medium-range UAVs: Operate within a range of 25-650 km.
    - Long-range UAVs: Operate beyond a range of 650 km.
- Capabilities-based classification:
    - Fixed-wing UAVs: Resemble traditional airplanes, they are typically more efficient and have longer ranges than other UAV types.
    - Multirotor UAVs: Resemble helicopters, can hover in place, and take off and land vertically, but they usually have shorter ranges than fixed-wing UAVs.
    - Hybrid UAVs: Combine features of both fixed-wing and multirotor UAVs.
    - VTOL (Vertical Takeoff and Landing) UAVs: can take off and land vertically like multirotor UAVs but fly horizontally like fixed-wing UAVs.
- Autonomy-based classification:
    - Manual UAVs: Operated by human pilots using remote control.
    - Semi-autonomous UAVs: Able to fly a pre-programmed route but require human control for tasks like takeoff and landing.
    - Autonomous UAVs: Capable of performing tasks without human intervention, such as navigating, detecting obstacles, and performing missions.

These classifications are not mutually exclusive, and different UAVs may fit into multiple categories.

2.2 QUADROTORS

Quadrotors, also known as quadcopters, are a type of unmanned aerial vehicle (UAV) that features four rotors arranged in a square or "X" configuration. They have gained significant popularity in recent years due to their agility, stability, and maneuverability.

2.2.1 History of Quadrotors

The concept of quadrotors dates back to the early 20th century, but significant advancements occurred in the 1990s and early 2000s. In 1990, Étienne Oehmichen developed the "Gyroplane No. 1," which is considered one of the earliest examples of a quadrotor. However, it was not until the 2000s that quadrotors began to attract attention for their potential applications (Norouzi Ghazbi et al. 2016).

Since then, quadrotors have become a prominent area of research, leading to the development of various types and applications.

2.2.2 Quadrotor Configuration

There are several configurations or variations of quadrotors based on the arrangement and orientation of their rotors (Thu and Gavrilov 2017). Here are some commonly known quadrotor configurations:

- Plus (+) Configuration: In this configuration, the four rotors are arranged in the shape of a plus sign (+). Two rotors are placed in the front and two at the back, forming a cross-like structure. This configuration provides stability and maneuverability in all directions.

- X Configuration: The X configuration has a similar setup to the + configuration, but with the rotors positioned diagonally. The front left and rear right rotors form one diagonal, while the front right and rear left rotors form the other diagonal. This configuration also offers stability and agility.

- H Configuration: The H configuration resembles the letter "H," with two rotors placed at each end of a horizontal bar. The front and rear rotors are aligned, while the left and right rotors are placed on the sides. This configuration allows for efficient payload carrying and is commonly used in industrial applications.

- Y Configuration: The Y configuration features a central rotor and three additional rotors positioned at equal distances, forming a Y shape. This configuration provides stability and redundancy, as it can continue flying even if one of the rotors fails.

- V Configuration: The V configuration resembles the letter "V," with two forward-facing rotors and two rear-facing rotors. The forward rotors are typically larger than the rear rotors. This configuration is often utilized in long-range or heavy-lift quadrotors.

- Tiltrotor Configuration: In a tiltrotor configuration, the rotors can be tilted to transition between vertical takeoff and landing (VTOL) and horizontal flight. This configuration allows for the advantages of both helicopters and fixed-wing aircraft, providing increased speed and range.

- Coaxial Configuration: In a coaxial configuration, two sets of rotors are stacked on top of each other, with one rotating in the opposite direction to the other. This configuration eliminates the need for a tail rotor and provides enhanced stability and control.

Out of the quadrotor configurations mentioned above, the most common ones are the "plus" (+) and "cross" (×) configurations. Among them, the X-configuration quadrotor is generally regarded as more stable than the "plus" configuration, which is known for its more acrobatic nature.

2.2.3 Applications of Quadrotors

Quadrotors are applicable in different areas (Idrissi, Salami, and Annaz 2022b). Some of the applications are listed below:

- Aerial Photography and Videography: Quadrotors equipped with cameras offer a unique perspective for aerial photography and videography. They are used in filmmaking, sports coverage, real estate photography, and more.

- Inspection and Surveillance: Quadrotors provide a convenient platform for inspecting infrastructure, such as buildings, power lines, pipelines, and wind turbines. They can access difficult-to-reach areas and provide visual data for analysis. Quadrotors are also used for surveillance in various domains.

- Search and Rescue Operations: Quadrotors equipped with thermal cameras and other sensors can assist in search and rescue operations. They can quickly survey large areas, locate missing persons, and provide real-time situational awareness to rescue teams.

- Precision Agriculture: Quadrotors equipped with sensors and cameras are used in agriculture for monitoring crop health, detecting pest infestations, and optimizing irrigation. They provide farmers with valuable data for precision agriculture techniques.

- Delivery Services: Companies like Amazon and UPS are exploring the use of quadrotors for package delivery. These drones can transport small packages to customers' doorsteps quickly and efficiently.

- Research

2.2.4 Components of Quadrotor

Quadrotor is powered by four rotors. Each rotor consists of several components working together to enable the flight and control of the quadrotor. The main components of a quadrotor are (Norouzi Ghazbi et al. 2016):

- Frame: The frame provides the structure and support for the quadrotor. It is typically made of lightweight materials like carbon fiber or aluminum to reduce weight while maintaining durability.
- Motors: A quadrotor has four brushless DC electric motors, one attached to each rotor. These motors generate the necessary thrust to lift and maneuver the quadrotor in the air.
- Propellers: Each motor is connected to a propeller, which consists of two or more blades that spin rapidly to create the airflow needed for lift and propulsion. The propellers are usually made of lightweight materials such as plastic or carbon fiber.
- Electronic Speed Controllers (ESCs): The ESCs are devices that control the speed and direction of the motors. They receive signals from the flight controller (see next point) and adjust the power supplied to each motor to achieve the desired flight behavior.

- Flight Controller: The flight controller is the "brain" of the quadrotor. It is a circuit board with an integrated microcontroller that processes data from various sensors and uses algorithms to stabilize the quadrotor, control its flight modes, and execute user commands. It also communicates with other components of the quadrotor system.
- Sensors: Quadrotors are equipped with various sensors to gather data about their orientation, altitude, and speed. The most common sensors include gyroscopes, accelerometers, magnetometers, barometers, and sometimes even GPS (Global Positioning System) receivers.
- Battery: A rechargeable lithium polymer (LiPo) battery provides the electrical power required to operate the quadrotor's motors, flight controller, and other electronic components. The battery is usually mounted on the frame and connected to the ESCs and flight controller.
- Radio Transmitter/Receiver: To control the quadrotor remotely, a radio transmitter is used by the pilot or operator. The transmitter sends commands to the quadrotor, which are received by the onboard radio receiver. The receiver relays these commands to the flight controller, which then adjusts the quadrotor's behavior accordingly.
- Onboard Computer/Companion Computer: In more advanced quadrotor systems, an onboard computer or companion computer may be used to handle additional computational tasks, such as image processing for computer vision applications or autonomous flight control algorithms.

These are the main components of a typical quadrotor system, but variations and additional features can exist depending on the specific design, purpose, and sophistication of the quadrotor.

2.2.5 Operation of Quadrotors

The X configuration is a popular configuration for quadrotors, which refers to the arrangement of the four rotors in the shape of an "X." In this configuration, two rotors are placed at the front in an "X" formation, and the other two rotors are placed at the rear, also forming an "X" shape. Each rotor produces lift and is individually controlled to enable stable flight and maneuverability.

Figure 2.2 illustrates the general mechanisms involved in controlling the movements of a quadrotor. By adjusting the electric current supplied to its four individual motors, the quadrotor's force and torque distributions are modified, enabling it to perform basic maneuvers. Altering the overall torque while maintaining a constant motor speed allows the quadrotor to ascend or descend. To maintain the overall upward thrust, the speeds of the right and left motors must remain unchanged, while the front motor's speed needs to be reduced and the back motor's speed increased for forward flight. For backward flight, the front and back motors must increase and decrease their speeds, respectively, in opposite directions. Likewise, lateral movements to the right or left are achieved by changing the speeds of the right and left motors, while keeping the front and back motor speeds constant. The quadrotor's yaw motion is generated when the front and back motors rotate in the same direction and speed, creating a torque that counters the opposing torque produced by the left and right motors, which have the same speed but rotate in the opposite direction compared to the front and back motors.

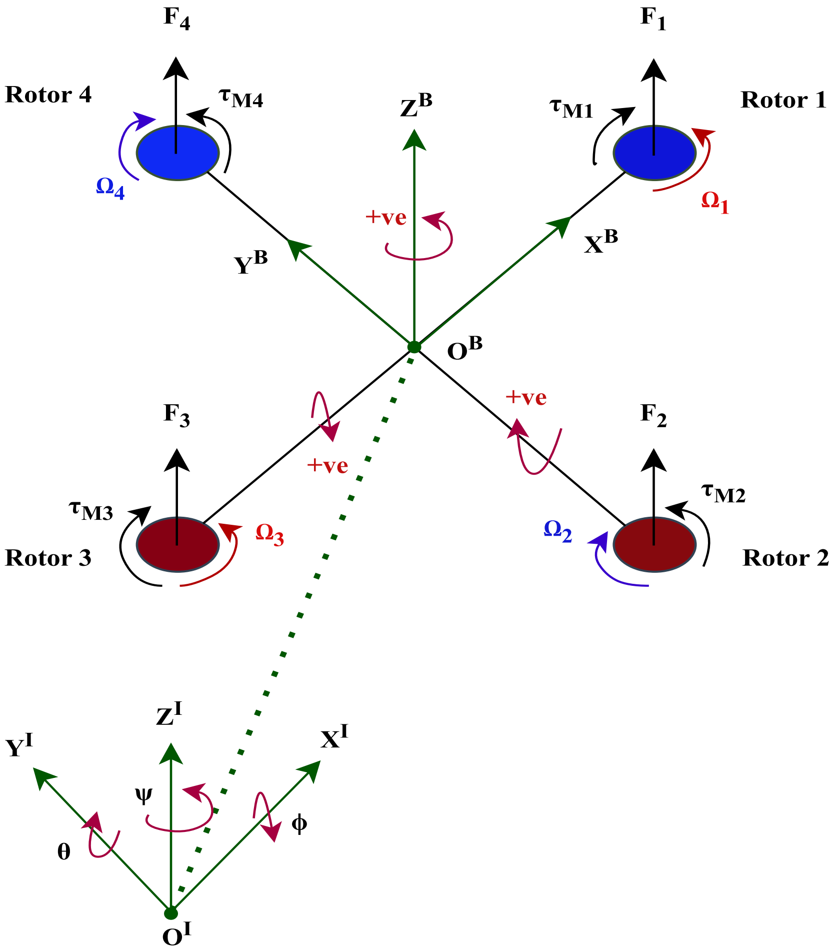


Figure 2.2 Schematic of X configuration Quadrotor

It is important to note that quadrotors with an X configuration are highly maneuverable and versatile but can be more mechanically complex compared to other configurations. They require careful control and stabilization algorithms to maintain stability and respond accurately to control inputs. Table 2-1 presents all motions of quadrotor related with the motor speed.

Table 2-1 Combination of rotor speeds to make an appropriate movement

Movements	Rotor 1	Rotor 2	Rotor 3	Rotor 4
Counter Clockwise	Increase	Decrease	Increase	Decrease
Clockwise	Decrease	Increase	Decrease	Increase
Up	Increase	Increase	Increase	Increase
Down	Decrease	Decrease	Decrease	Decrease
Hovering	Equal	Equal	Equal	Equal
Left	Decrease	Increase	Increase	Decrease
Right	Increase	Decrease	Decrease	Increase
Forward	Decrease	Decrease	Increase	Increase
Backward	Increase	Increase	Decrease	Decrease

2.3 OVERVIEW OF SLIDING MODE CONTROL

Emelyanov and a group of coresearchers from the Soviet Union initially introduced and explained Variable Structure Control (VSC) in combination with Sliding Mode Control (SMC) during the early 1950s. In recent decades, there has been considerable attention within the control research community towards VSC (Variable Structure Control) and SMC (Sliding Mode Control). SMC has gained considerable popularity and has been implemented in various fields, including robotics, power systems, aerospace systems, automotive systems, and industrial processes. Its effectiveness in dealing with uncertainties and its robustness make it a suitable choice for systems that operate in challenging and unpredictable environments (Jinkun Liu 2017).

The fundamental idea behind sliding mode control is to create a sliding manifold in the system state space, on which the system dynamics behave in a desired manner. The sliding manifold is typically defined as a hyperplane or a lower-dimensional surface. The goal is to drive the system state onto this manifold and keep it there. The key concept in sliding mode control is the sliding variable, which quantifies the deviation of the system state from the sliding manifold.

The control action is designed to drive the sliding variable to zero, which ensures that the system state stays on the sliding manifold. The main advantages of sliding mode control include its robustness to uncertainties and disturbances, its ability to handle nonlinear systems, and its simplicity in design and implementation. It can provide excellent tracking performance and robustness in the face of model uncertainties, parameter variations, and external disturbances.

2.4 OVERVIEW OF BACKSTEPPING CONTROL

Backstepping control is an advantageous method in control systems engineering that emerged in the early 1990s courtesy of Peter V. Kokotovic and a team of researchers. It serves as a valuable approach for creating stabilizing controls specifically tailored to a particular group of nonlinear dynamical systems. Backstepping control is a nonlinear control technique used to design controllers for complex and highly nonlinear systems. It is particularly effective for systems with uncertain or unknown dynamics. The key idea behind backstepping control is to decompose the control problem into a series of simpler subsystems and design controllers for each subsystem in a recursive manner (’Ahmad 2021).

Backstepping control is known for its ability to handle complex nonlinear systems and uncertainties. It allows for systematic controller design by decomposing the control problem into simpler subsystems. However, it does require a good understanding of the system dynamics and Lyapunov theory to ensure stability and performance.

2.5 OVERVIEW OF PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle Swarm Optimization (PSO) is a population-based stochastic optimization algorithm inspired by the social behavior of bird flocking or fish schooling. It was first introduced by Eberhart and Kennedy in 1995. PSO is commonly used to solve optimization problems in various domains (Gad 2022).

2.5.1 Parameters and Mathematical Model of PSO

The main parameters used to model the PSO are:

- $S(n)=\{s_1,s_2,...,s_n\}$: a swarm of n-particles
- s_i : an individual in the swarm with a position p_i and velocity $v_i, i \in [|1,n|]$
- p_i : the position of a particle s_i
- v_i : the velocity of a particle p_i
- $pbest_i$: the best solution of a particle
- $gbest$: the best solution of the swarm (Global)
- f : the best solution of the swarm (Global)
- c_1, c_2 : acceleration constants (cognitive and social parameters)
- r_1, r_2 : random numbers between 0 and 1
- t : the iteration numbers

Two main equations are involved in the PSO algorithm. The first equation is the velocity equation, where each particle in the swarm updates its velocity using the computed values of the individual and global best solutions and its current position which is written as follows:

$$v_i^{t+1} = v_i^t + c_1 r_1 \left(pbest_i^t - p_i^t \right) + c_2 r_2 \left(gbest^t - p_i^t \right) \tag{2.1}$$

The second equation is the position equation, where each particle updates its position using the newly calculated velocity:

$$p_i^{t+1} = p_i^t + v_i^{t+1} \tag{2.2}$$

2.5.2 Flowchart of PSO Algorithm

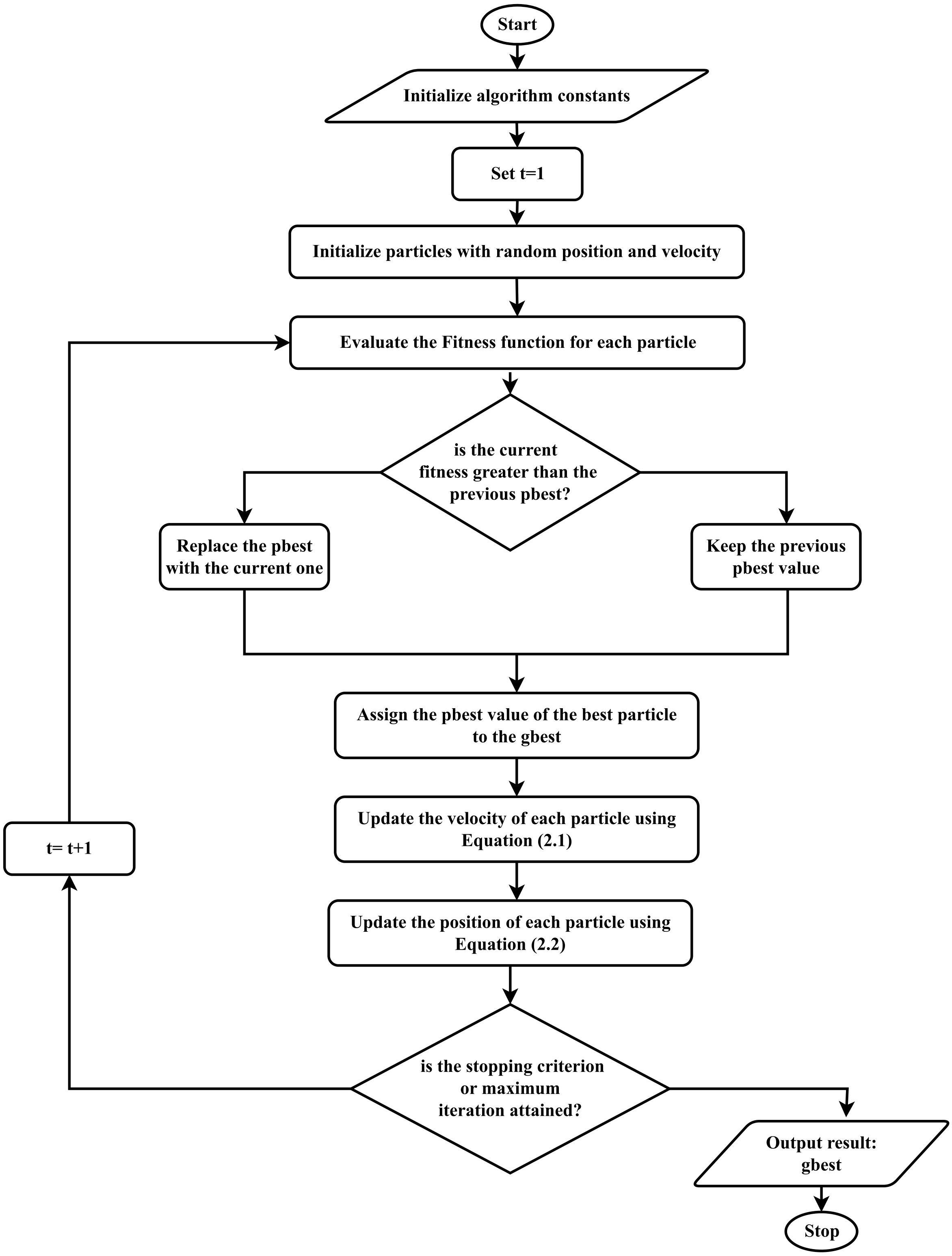


Figure 2.3 Flowchart of PSO algorithm

2.6 REVIEW OF RELATED WORKS

The backstepping control method was developed to address a collection of nonlinear dynamical systems. It is a strategy that focuses on utilizing feedback to stabilize the system right from its initial state. When applied to an under-actuated system, this approach demonstrates the ability to effectively handle external uncertainties. In (Li, Chen, and Peng 2023) A method called integral backstepping control (IBC) is introduced for the purpose of tracking the trajectory of a quadrotor. This approach includes an examination of stability using the Lyapunov theory, considering both matched and mismatched disturbances. Unlike traditional methods, the analytical approach presented here avoids the use of numerical differentiation in the virtual control rule, which helps address the problem noise amplification. The proposed IBC control algorithm offers performance in tracking trajectories and generates control signals that are reasonable and realistic. As a result, the analytical solution yields smoother control signals and prevents control saturation. Simulation results are provided to demonstrate the efficiency and practicality of the suggested control algorithm. It is important to note that this technique is applied to quadrotor with three degrees of freedom (3-DOF).

(Hawi Aboma 2022) utilizes a control scheme that relies on Lyapunov stability criteria and Grey Wolf Optimization. This control scheme is designed to effectively control altitude, attitude, and horizontal motion control for a quadrotor. By conducting simulations with different initial values for all inputs, the system ensures global stability, robustness, and improved trajectory tracking. As a result, the quadrotor is able to accurately reach its desired position. However, it does not consider the actuator dynamics and integral backstepping control relies on an accurate model of the system dynamics. If there are significant discrepancies between the actual system and the model used in the control design, the controller performance can deteriorate. Model uncertainties or variations in the system parameters can lead to suboptimal or unstable control performance.

In (Saibi, Boushaki, and Belaidi 2022) the quadrotors dynamic control is achieved through the utilization of backstepping control. In this study, control principles for maintaining stability in a 6-degree-of-freedom quadrotor were devised and implemented based on the models developed. Initially, a mathematical model specific to the quadrotor was constructed. Subsequently, the backstepping control approach was employed to control the quadrotors position and orientation. To evaluate the efficacy of the backstepping method, numerous simulations were conducted, and the results obtained from these simulations clearly exhibited the efficiency of the proposed control approach. However, effectiveness and robustness of the suggested controllers were not demonstrated in the presence of model inaccuracies and disturbances.

Another type of robust nonlinear control is Sliding Mode Control, which provides a simple framework, controller settings that are easy to adjust, and the ability to handle uncertainties in the system model. However, the practical application of this control method did not match the initial expectations due to the use of a discontinuous control rule that caused an undesired chattering effect. In (Chandra and Lal 2022) a reliable control technique for a quadrotor system carrying a suspended load is proposed. The aim is to eliminate load oscillation and ensure precise tracking of the quadrotor. To achieve this, a higher-order sliding mode controller was developed, incorporating the super twisting algorithm. This combination effectively eliminated undesirable chattering that occurred due to the control input from the standard sliding mode controller. Simulation results clearly indicate the superior performance of the super twisting controller compared to traditional sliding mode control methods. However, the effectiveness and robustness of the suggested controller under model inaccuracies and external disturbances were not evaluated.

(Hou, Yu, and Lu 2022) introduces a novel flight controller for quadrotors, known as the chattering minimized terminal sliding mode control-based quadrotor disruption estimator. The newly developed controller demonstrates superior performance compared to existing sliding mode control-based methods in terms of its suitability for engineering applications. It employs a finite-time convergent technique that effectively reduces chattering while maintaining robustness. Experimental tests conducted on quadrotors utilizing PX4 and Pixracer platforms validate the theoretical results. However, this study does not consider the appropriate variation of the load in its analysis.

In (Zhu and Cao 2019) The issue of a quadrotor experiencing rotor failure is addressed through fault-tolerant control. It is recommended to employ a fault-tolerant controller that utilizes the back-stepping approach and sliding mode control method. The objective of the sliding mode approach is to suppress specific disturbances in order for the system to reach the sliding surface. By disabling the faulty rotor, stable control is achieved. The simulation results show that the quadrotor can effectively and rapidly reach the desired position while maintaining a reasonably stable orientation, unfortunately with compromised control over the yaw angle. As a result, this technique proves to be beneficial for practical engineering applications. (Xu et al. 2020) focused on addressing the problem of controlling the trajectory of a quadrotor unmanned aerial vehicle (UAV). Initially, a cascade Active Disturbance Rejection Control (ADRC) scheme is explained for the attitude subsystem. Building upon this idea, a backstepping Sliding Mode Control (SMC) method is developed for the position subsystem. The stability of the entire system is analyzed using the Lyapunov theory. The research shows that the proposed control strategy can significantly minimize tracking errors in both attitude and position, bringing them to very small and precise ranges. The effectiveness and resilience of the chosen control approach are supported by experimental and numerical results. In (Zaidi, Kazim, and Wang 2022) a reliable solution for controlling a group of drones against external disruptions is presented. Their approach combines a fuzzy-based system with the sliding mode control method to create a controller that is free from chattering. By integrating a fuzzy rule-based system, the sliding mode tracking controller can generate smooth and chatter-free control signals, effectively addressing the chattering problem caused by the abrupt switching function in the sliding mode control approach. The stability of the entire system is demonstrated using the Lyapunov stability theory, showing its asymptotic stability. To evaluate the effectiveness of our control design, numerical simulations are performed. However, in all of the above backstepping-based sliding mode control design for a quadrotor does not take into account time-varying disturbances and changes in payloads.

In (Wang et al. 2022) a novel control system that combines trajectory tracking and obstacle avoidance using NMPC (Nonlinear Model Predictive Control) is presented. It utilizes Lyapunov's direct technique and backstepping control methodology to establish stability constraints, forming an auxiliary controller. To enhance the tracking cost function and facilitate obstacle avoidance, a carefully designed potential field-based cost term is incorporated. This innovative NMPC formulation effectively addresses the challenge of controlling commercial quadrotors in real-world scenarios, enabling them to navigate while simultaneously avoiding obstacles. In (Bui, Van Nguyen, and Phung 2022) A new Linear-Nonlinear Model Predictive Controller (LNMPC) was developed for controlling the flight paths of unmanned aerial vehicles (UAVs). The controller's effectiveness and stability were shown through the application of a contraction condition and the implementation of a sliding mode control. The results indicate that the suggested controller outperforms other established nonlinear controllers in terms of response time, accuracy of trajectory tracking, and the smoothness of control signals. This is particularly evident in challenging situations where significant adjustments in control are required.

In (Singh and Kumar 2023) MATLAB environment has been utilized to exhibit and illustrate the QUAV 6-DoF dynamics and its ability to track different trajectories. The impact of external atmospheric disturbances, such as wind gusts, has been effectively addressed through the development and implementation of feedback linearization and MPC controller for translational and rotational dynamics. In order to minimize the computational burden associated with obtaining the control action, the two controllers are integrated. Additionally, to enhance the tracking accuracy of the MPC controller, both controllers work together and operate at different frequencies. The performance of both controllers is evaluated using a combination of helical and circular trajectories. The model predictive controller faces challenges related to a significant computational burden. Nonetheless, when compared to nonlinear model predictive control (MPC), the linear MPC design necessitates less computational resources. However, because of the continuous optimization of the quadratic cost function, the linear MPC still requires careful consideration regarding its computational workload. Consequently, in order to implement the suggested controller in real-time, a dedicated PC with efficient performance is necessary. To achieve a satisfactory equilibrium between computational load and controller effectiveness, the MPC horizon is selected accordingly. Furthermore, MPC relies on a reliable model, making the controller's performance dependent on the accuracy of the system model.

In general, the literature examined in this part reviews and discusses the effectiveness of various control system strategies used to control the altitude, position, and orientation of a quadrotor. The quadrotors positions (X and Y) are uncommon. The majority of earlier research focused solely on the time varying disturbances and variation of payloads which is related directly to model parametric uncertainties and was based on attitude altitude control. Additionally, the mathematical model used is not incorporate the effect of transformation matrix and actuator dynamics is not considered. These impacts need to be considered in order to verify and assess the performance of the controllers. In order to address saturation of control input (actuator limit), external disturbance (such as wind gust), parametric uncertainties (such as mass change), and other pertinent factors, a trajectory control system employing backstepping sliding mode controller is designed and simulation is done to show the effectiveness of the proposed controller in this research study.

CHAPTER THREE

3. MODELLING OF A QUADROTOR SYSTEM

3.1 INTRODUCTION

This chapter discusses the procedures used to accomplish this research study as well as the development of the quadrotor system's kinematics and dynamic mathematical modelling based on the Euler-Newton approach including actuator dynamics.

3.2 MATERIALS

Software like MATLAB, Microsoft office, Draw.io and Math Type are utilized in this thesis work. The computer programme known as MATLAB includes Simulink and technical toolboxes. Editing of the thesis documentation is done using Microsoft office. A variety of diagrams are created using the drawing tool Draw.io. For composing mathematical equations and formulae in Microsoft office word and PowerPoint presentation tools, math type is utilized.

3.3 METHODS

In order to attain the objectives of this thesis, the following methods and techniques are followed.

- Data is gathered through reviewing previous studies and literatures.
- The gathered data is theoretically and visually examined to create system modelling.
- Analysis is done on the quadrotor's dynamics modelling.
- A BSMC is designed to control the modelled quadrotor system.
- The performance of the designed controller is analyzed by subjecting the system to a time varying external disturbance and parametric uncertainties.

- Discussion about the results will be made, evaluation and analysis on the performance of the proposed controller will be made and finally relevant conclusions and recommendation of future work will be given.

The block diagram in Figure 3.1 summarizes the method followed throughout the research work.

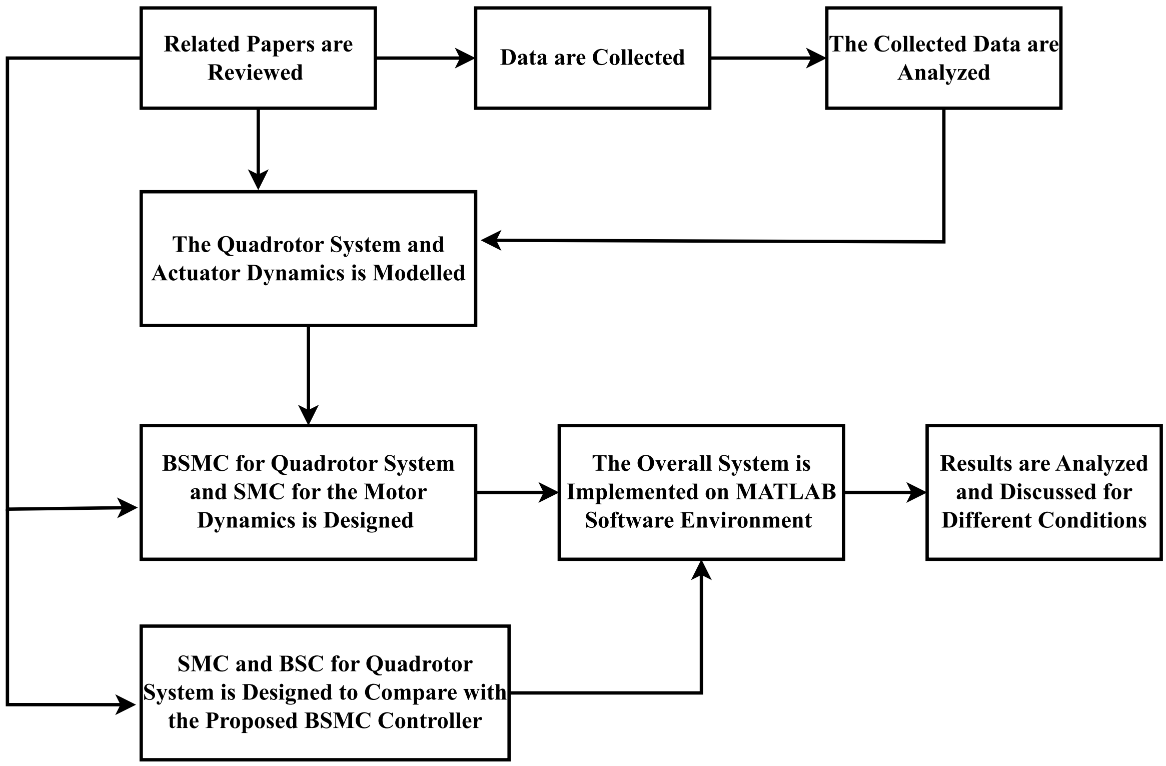


Figure 3.1 Block diagram of the research methodology

3.4 MODELLING APPROACH OF QUADROTOR SYSTEM

When it comes to model a quadrotor system, there are various approaches that can be taken depending on specific requirements and level of complexity. Here is a common modeling approach for a quadrotor system (Chovancová et al. 2014):

1. Rigid Body Dynamics: At the core of a quadrotor model is its rigid body dynamics. This involves representing the quadrotor as a rigid body and applying Newton's laws of motion to describe its motion. The key variables in this model are the position, orientation, linear velocity, and angular velocity of the quadrotor.

2. Kinematics: To describe the motion of the quadrotor, it needs to determine the relationships between the quadrotor's position, orientation, linear velocity, and angular velocity. This can be done using mathematical tools such as rotation matrices or quaternions and transformation matrices.

3. Actuator Dynamics: The next step is to model the dynamics of the actuators that control the quadrotor's motion, typically the four motors. This involves relating the control inputs (such as the motor speeds) to the resulting forces and torques generated by the propellers.

4. Aerodynamics: To account for the aerodynamic effects on the quadrotor, we can include simplified aerodynamic models. These models capture the forces and torques acting on the quadrotor due to the interaction between the propellers and the surrounding air. Factors like thrust, drag, and lift are considered.

5. Control System: A quadrotor is typically controlled using a feedback control system. The control system takes into account the desired quadrotor behavior and the feedback from sensors (e.g., accelerometers, gyroscopes) to compute appropriate control inputs.

6. Sensor Modeling: To simulate the quadrotor's behavior realistically, we can model the sensors that provide feedback to the control system. This may include modeling the noise, biases, and delays associated with sensors like accelerometers, gyroscopes, and position sensors.

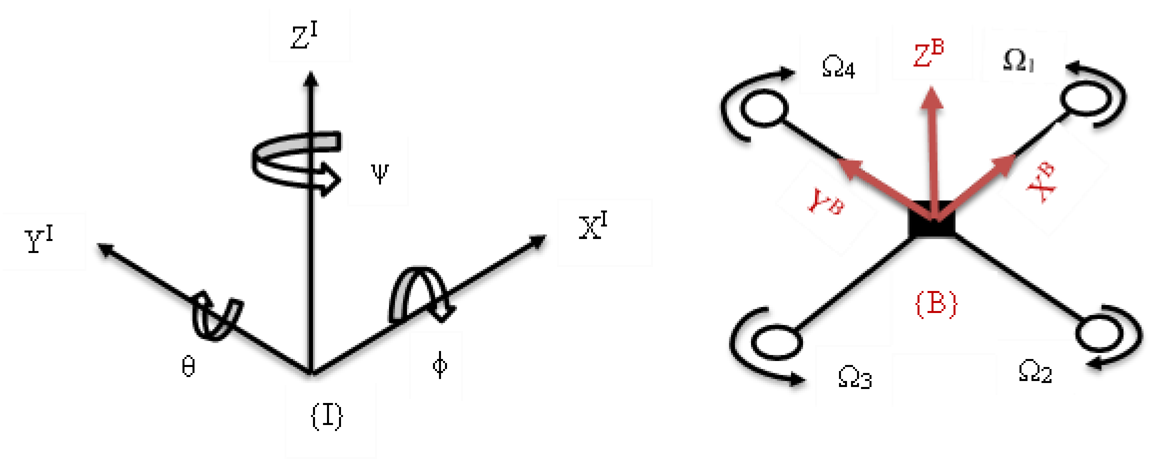
7. Environmental Effects: Depending on the application and level of detail desired, it might need to consider environmental effects such as wind, ground effects, or other external disturbances that can affect the quadrotor's motion.

From above mentioned modelling approaches kinematics, rigid body dynamics, actuator dynamics, environmental effects, and the control system part are discussed.

3.5 KINEMATIC MODEL

3.5.1 Reference Frame Alignment

Because the quadrotor can move in six different ways, it requires six variables to describe its position and orientation in space. To understand how a quadrotor behaves, two reference frames are necessary: the body frame and the inertial frame. The quadrotor has various sensors, such as a gyroscope and accelerometer, which provide measurements relative to the body frame to determine linear and angular velocity. Additionally, there are sensors like a magnetometer and GPS that provide measurements relative to the inertial frame to determine linear and angular position. To derive system equations using a single frame, a transformation method is required. The body framework of the quadrotor is connected to its center of gravity (COG), which serves as the origin. The quadrotor's COG is positioned at the center of its symmetrical design, consisting of a central core and four identical motors attached to its arms (Belsty 2021).



(a) Inertial frame (b) Body fixed frame

Figure 3.2 Reference frames

The absolute linear position of the center of mass of the quadrotor is expressed in the inertial frame as:

$$\xi^I = [X \ Y \ Z]^T \tag{3.1}$$

The attitude or the angular position is defined in the inertial frame with the 'roll-pitch-yaw' Euler angles.

$$\eta^I = [\phi \ \theta \ \psi]^T \tag{3.2}$$

The linear velocity V^B and angular velocity v^B are defined as:

$$V^B = [\mathrm{v}_X \ \mathrm{v}_Y \ \mathrm{v}_Z]^T \tag{3.3}$$

$$v^B = [P \ Q \ R]^T \tag{3.4}$$

The net thrust acting on the quadrotor T^B_{net} and net rotational moment τ^B_{net} are defined in the body fixed frame as:

$$T^B_{net} = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{3.5}$$

$$\tau^B_{net} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{3.6}$$

3.5.2 Rotation and Transformation Matrix

A rotation matrix exists that transforms the components of the quadrotor's net thrust and net rotational moment, as experienced in its own body frame, into the equivalent values in the inertial reference frame. This transformation involves performing consecutive rotations around the principal axis *Z*, *Y* and *X*.

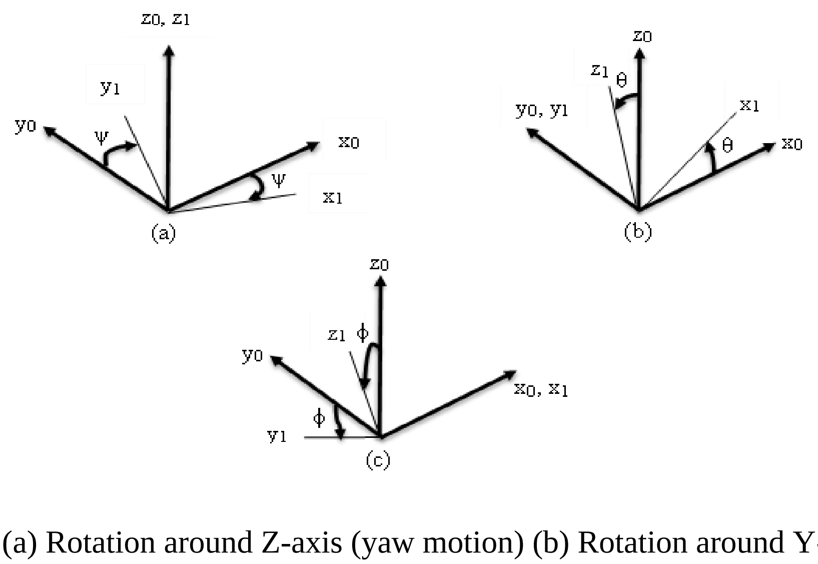


Figure 3.3 (a) Rotation around Z-axis (yaw motion) (b) Rotation around Y-axis (pitch motion) (c) Rotation around X-axis (roll motion)

In three dimensions each axis of the frame $O_1x_1y_1z_1$ is projected onto coordinate frame $O_0x_0y_0z_0$. The resulting rotation matrix is given by:

$$R_1^0 = \begin{bmatrix} x_1.x_0 & y_1.x_0 & z_1.x_0 \\ x_1.y_0 & y_1.y_0 & z_1.y_0 \\ x_1.z_0 & y_1.z_0 & z_1.z_0 \end{bmatrix} \tag{3.7}$$

$$A_k.B_k = |A_k||B_k|\cos(\gamma) \tag{3.8}$$

Where A, $B = (x, y, z)$ $k = 0,1$ and γ is an angle between A_k and B_k. From this rotation about z-axis with angle ψ ($R_{z,\psi}$), rotation about y-axis with angle θ ($R_{y,\theta}$), and rotation about x-axis with angle ϕ ($R_{x,\phi}$) are represented as follows:

$$R_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.10}$$

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.11}$$

Thus, as shown in (Teppo Luukkonen 2014) the rotation matrix is given by:

$$\begin{aligned} R &= R_{z,\psi} \times R_{y,\theta} \times R_{x,\phi} \\ R &= \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \end{aligned} \tag{3.12}$$

Where $C_k = cos(k)$ and $S_k = sin(k)$. The rotation matrix R is orthogonal and used to convert from body reference frame to inertial reference frame and $R^{-1}=R^T$ for $\frac{-\pi}{2} < \phi, \theta < \frac{\pi}{2}$ and $-\pi < \psi < \pi$ which is the rotation matrix from the inertial frame to the body frame.

Also, a transformation matrix T_M to convert angular velocities from the inertial frame to body frame and T_M^{-1} to convert angular velocities from body frame to inertial frame is needed.

As shown in (Alderete n.d.) the transformation matrix T_M is calculated as follows:

$$T_M = R_{x,\phi} R_{y,\theta} \begin{bmatrix} 0 \\ 0 \\ \dfrac{d\psi}{dt} \end{bmatrix} + R_{x,\phi} \begin{bmatrix} 0 \\ \dfrac{d\theta}{dt} \\ 0 \end{bmatrix} + \begin{bmatrix} \dfrac{d\phi}{dt} \\ 0 \\ 0 \end{bmatrix} \tag{3.13}$$

Thus, by substituting Equation (3.9), (3.10), and (3.11) into Equation (3.13) T_M and its inverse T_M^{-1} are given as follows:

$$T_M = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.14}$$

$$T_M^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \cos\theta & \cos\phi / \cos\theta \end{bmatrix} \tag{3.15}$$

By using Equation (3.14) and (3.15) it is possible to convert the angular velocities from inertial frame to body frame similarly from the body frame to inertial frame as follows:

$$v^B = T_M \dot{\eta}^I, \quad \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = T_M \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.16}$$

$$\dot{\eta}^I = T_M^{-1} v^B, \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_M^{-1} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{3.17}$$

3.6 RIGID BODY DYNAMICS

The movement of the quadrotor consists of two components: rotational motion and translational motion. The translational motion relies on the alignment of the quadrotor, which is determined by its rotational movement. The rotational subsystem of the quadrotor is fully controllable. In the following sections, the equations that describe the quadrotor's rotational and translational motion is derived.

To represent the dynamics of the quadrotor system, the subsequent assumptions are taken into account (Rinaldi, Primatesta, and Guglieri 2023):

- The structure is perfectly symmetrical.
- The structure is perfectly rigid.
- The quadrotor’s center of mass coincides with the origin of the body fixed frame.
- Sophisticated phenomena that are challenging to simulate, such as the ground effect, blade flapping, and other smaller aerodynamic or inertial factors that are not explicitly considered, are disregarded.

3.6.1 Rotational Equation of Motion

In the body frame, according to (Teppo Luukkonen 2014) the rotational equation of the quadrotor motion is defined as: -

$$I\dot{v}^B + v^B \times (Iv^B) + \tau_{gyro} = \tau_{net}^B \tag{3.18}$$

where $I\dot{v}^B$ is an angular acceleration of the inertia, $v^B \times Iv^B$ is a centripetal force, τ_{gyro} is a gyroscopic moment due to rotors inertia and τ_{net}^B is net moments acting on the quadrotor due to the propellers movement.

The reasons behind deriving the rotational equations of motion in the body frame and not in the inertial frame are:

- The inertia matrix is time-invariant in body frame and
- Advantage of body symmetry can be taken to simplify the equations.

1. **Inertia matrix:** also known as the inertia tensor or moment of inertia matrix, is a mathematical representation of an object's resistance to changes in rotational motion. For a quadrotor, which is a symmetric object, the inertia matrix takes a specific form. The quadrotor is assumed to have symmetric structure with the four arms aligned with the body *x* and *y* axis. Thus, the inertia matrix is diagonal matrix *I* in which $I_{xx}=I_{yy}$.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.19}$$

Where I_{xx}, I_{yy} and I_{zz} are the moments of inertia about the principal axis in the body frame.

2. **Gyroscopic moments:** In a quadrotor, the gyroscopic moments play a crucial role in its stability and control. The gyroscopic moments arise from the rotation of the propellers and their interaction with the surrounding air. When the propellers rotate, they generate a torque that opposes any change in the quadrotor's orientation. This torque is known as the gyroscopic moment. The gyroscopic moments are dependent on various factors such as the rotational speed of the propellers, the moment of inertia of the quadrotor, and the geometry of the propellers. These factors influence the magnitude and direction of the gyroscopic moments. To effectively control a quadrotor, these gyroscopic moments need to be taken into account in the design and control algorithms. According to (Teppo Luukkonen 2014) gyroscopic torque is given as follows:

$$\begin{aligned} \tau_{gyro} &= J_r v^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Omega_r \\ &= J_r \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4) \end{aligned} \tag{3.20}$$

Where J_r and Ω_r are rotors inertia and rotors relative speed respectively.

3. **Moments and forces acting on the quadrotor:** Each propeller rotates at the angular velocity Ω_i producing the corresponding force f_i directed upwards and the counteracting torque τ_M directed opposite to the direction of the angular velocity Ω_i.

As shown in Figure 3.4 the propellers with the angular speed Ω_1 and Ω_3 spin counter-clockwise and the other two spin clockwise.

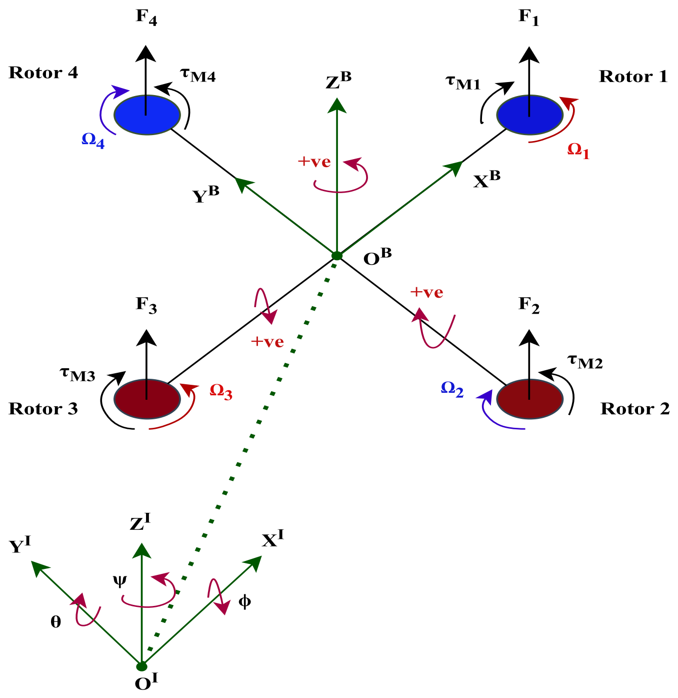


Figure 3.4 Schematic diagram which shows the movement of a quadrotor

The thrust generated by the rotor i is proportional to the square of the angular speed of the rotor and thrust constant K_T, which includes the air density ρ, the radius of the propeller r, and the thrust coefficient C_T, that depends on the blade rotor characteristics (cube of the rotor blade radius, number of blades, and chord length of the blade) (Chovancová et al. 2014).

$$f_i = C_T \rho \pi r^4 \Omega_i^2 = K_T \Omega_i^2 \tag{3.21}$$

The torque developed around the rotor axis is given as the product of the square of the angular velocity and the drag coefficient K_D, which depends on the same factors as K_T and also on the moment of inertia of rotor J_r and on the angular acceleration of rotor i (Chovancová et al. 2014).

$$\tau_{Mi} = C_D \rho \pi r^5 \Omega_i^2 + J_r \dot{\Omega}_i = K_D \Omega_i^2 \tag{3.22}$$

The combined forces of rotors create thrust T in the direction of the body z-axis. Torque τ_{net}^B consists of the torques τ_ϕ, τ_θ and τ_ψ in the direction of the corresponding body frame angles.

$$T_{net}^{B}=\begin{bmatrix}0\\0\\T\end{bmatrix}=\begin{bmatrix}0\\0\\\sum_{i=1}^{4}f_i\end{bmatrix}=\begin{bmatrix}0\\0\\K_T\sum_{i=1}^{4}\Omega_i^2\end{bmatrix}=\begin{bmatrix}0\\0\\K_T(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2)\end{bmatrix} \tag{3.23}$$

$$\tau_{net}^{B}=\begin{bmatrix}\tau_\phi\\\tau_\theta\\\tau_\psi\end{bmatrix}=\begin{bmatrix}l(-f_2+f_4)\\l(-f_1+f_3)\\\sum_{i=1}^{4}\tau_{Mi}\end{bmatrix}=\begin{bmatrix}lK_T(-\Omega_2^2+\Omega_4^2)\\lK_T(-\Omega_1^2+\Omega_3^2)\\K_D(\Omega_1^2-\Omega_2^2+\Omega_3^2-\Omega_4^2)\end{bmatrix} \tag{3.24}$$

The distance between the center of mass of the quadrotor and the rotor is represented by l. Consequently, the quadrotor achieves a positive roll movement by reducing the angular velocity of the second rotor and increasing the angular velocity of the fourth rotor. Likewise, for positive pitch movement, the angular velocity of the first rotor is decreased while the angular velocity of the third rotor is increased. Yaw movement is attained by increasing the angular velocities of two opposite rotors while decreasing the velocities of the remaining two rotors.

Thus, by combining Equations (3.19), (3.20), (3.23) and (3.24) Equation (3.18) can be written as follows:

$$I\dot{v}^B+v^B\times(Iv^B)+\tau_{gyro}=\tau_{net}^{B}$$

$$\begin{bmatrix}I_{xx}&0&0\\0&I_{yy}&0\\0&0&I_{zz}\end{bmatrix}\begin{bmatrix}\dot{P}\\\dot{Q}\\\dot{R}\end{bmatrix}=-\begin{bmatrix}P\\Q\\R\end{bmatrix}\times\begin{bmatrix}I_{xx}P\\I_{yy}Q\\I_{zz}R\end{bmatrix}-J_r\begin{bmatrix}P\\Q\\R\end{bmatrix}\times\begin{bmatrix}0\\0\\1\end{bmatrix}\Omega_r+\begin{bmatrix}\tau_\phi\\\tau_\theta\\\tau_\psi\end{bmatrix} \tag{3.25}$$

By rearranging Equation (3.25) the rotational equation of quadrotor motion in the body frame is derived as follows:

$$
\begin{bmatrix}\dot{P}\\ \dot{Q}\\ \dot{R}\end{bmatrix}=\begin{bmatrix}\frac{(I_{yy}-I_{zz})qr}{I_{xx}}\\ \frac{(I_{zz}-I_{xx})pr}{I_{yy}}\\ \frac{(I_{xx}-I_{yy})pq}{I_{zz}}\end{bmatrix}-\begin{bmatrix}\frac{qJ_r\Omega_r}{I_{xx}}\\ \frac{-pJ_r\Omega_r}{I_{yy}}\\ 0\end{bmatrix}+\begin{bmatrix}\frac{\tau_\phi}{I_{xx}}\\ \frac{\tau_\theta}{I_{yy}}\\ \frac{\tau_\psi}{I_{zz}}\end{bmatrix}
$$

$$
=\begin{bmatrix}\frac{(I_{yy}-I_{zz})QR}{I_{xx}}-\frac{QJ_r\Omega_r}{I_{xx}}+\frac{\tau_\phi}{I_{xx}}\\ \frac{(I_{zz}-I_{xx})PR}{I_{yy}}+\frac{PJ_r\Omega_r}{I_{yy}}+\frac{\tau_\theta}{I_{yy}}\\ \frac{(I_{xx}-I_{yy})PQ}{I_{zz}}+\frac{\tau_\psi}{I_{zz}}\end{bmatrix} \tag{3.26}
$$

Using transformation matrix T_M^{-1} it is possible to convert body frame to inertial reference frame as follows:

$$
\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=T_M^{-1}\begin{bmatrix}P\\ Q\\ R\end{bmatrix}
$$

$$
=\frac{d\left(T_M^{-1}\begin{bmatrix}P\\ Q\\ R\end{bmatrix}\right)}{dt}=\frac{d(T_M^{-1})}{dt}\begin{bmatrix}P\\ Q\\ R\end{bmatrix}+T_M^{-1}\begin{bmatrix}\dot{P}\\ \dot{Q}\\ \dot{R}\end{bmatrix} \tag{3.27}
$$

3.6.2 Translational Motion of Equation

The quadrotor is assumed to be rigid body and thus Newton-Euler equations can be used to describe its translational dynamics (Teppo Luukkonen 2014). In the body frame,

$$
m\dot{V}^B+v^B\times(mV^B)=R^TG+T_{net}^B \tag{3.28}
$$

where $m\dot{V}^B$ is force required for the acceleration of the mass, $v^B\times(mV^B)$ is a centrifugal force, R^TG is a gravitational force with respect to body frame and T_{net}^B is a net thrust produced due to the rotor's movement.

Centrifugal force can be observed as an outward force in a rotating frame of reference, but it is nonexistent when describing a system relative to an inertial frame. All measurements of position and velocity require a reference frame. Consequently, in an inertial frame, the centrifugal force is canceled out, and the equation of motion for translational motion is determined as follows (Teppo Luukkonen 2014):

$$\begin{aligned}
&R(m\dot{V}^B + v^B \times (mV^B)) = R(R^T G + T_{net}^B) \\
&m\ddot{\xi} = G + RT_{net}^B \\
&m\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + T\begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix}
\end{aligned} \tag{3.29}$$

By rearranging the above equation, the translational equation of quadrotor motion is written as:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} \dfrac{T(C_\psi S_\theta C_\phi + S_\psi S_\phi)}{m} \\ \dfrac{T(S_\psi S_\theta C_\phi - C_\psi S_\phi)}{m} \\ -g + \dfrac{TC_\theta C_\phi}{m} \end{bmatrix} \tag{3.30}$$

Finally, the equation of motion for a quadrotor is described by

$$\begin{cases}
\ddot{X} = \dfrac{T(C_\psi S_\theta C_\phi + S_\psi S_\phi)}{m} \\
\ddot{Y} = \dfrac{T(S_\psi S_\theta C_\phi - C_\psi S_\phi)}{m} \\
\ddot{Z} = -g + \dfrac{TC_\theta C_\phi}{m} \\
\dot{P} = \dfrac{(I_{yy} - I_{zz})QR}{I_{xx}} - \dfrac{QJ_r\Omega_r}{I_{xx}} + \dfrac{\tau_\phi}{I_{xx}} \\
\dot{Q} = \dfrac{(I_{zz} - I_{xx})PR}{I_{yy}} + \dfrac{PJ_r\Omega_r}{I_{yy}} + \dfrac{\tau_\theta}{I_{yy}} \\
\dot{R} = \dfrac{(I_{xx} - I_{yy})PQ}{I_{zz}} + \dfrac{\tau_\psi}{I_{zz}}
\end{cases} \tag{3.31}$$

3.7 Dynamic Model of the Actuator

Quadrotors commonly employ brushless DC motors, known for their ability to generate substantial torque with minimal friction. In the subsequent analysis, we make the assumption that the rotors lack gearing and exhibit a rigid mechanical connection between the motors and the propellers. At a stable condition, the behavior of a brushless DC motor aligns with that of a traditional DC motor (Derafa1, Madani2, and Benallegue2 2006).

$$\begin{cases} V = Ri_a + L\dfrac{di_a}{dt} + K_e\Omega \\ \zeta_m = J_r\dfrac{d\Omega}{dt} + C_s + \zeta_r \end{cases} \tag{3.32}$$

where

$$\begin{cases} \zeta_m = K_m i_a \\ \zeta_r = K_r\Omega^2 \end{cases} \tag{3.33}$$

By taking a small motor with a very low inductance, then the second order DC motor dynamics can be approximated as

$$\begin{gathered} \frac{a_0 + a_1\Omega + a_2\Omega^2 + \dot{\Omega}}{d} = V \\ a_0 = \frac{C_s}{J_r}, a_1 = \frac{K_e K_m}{J_r R}, a_2 = \frac{K_r}{J_r}, d = \frac{K_m}{J_r R} \end{gathered} \tag{3.34}$$

where

V – motor input voltage
Ω – angular speed
$K_e K_m$ – electrical and mechanical torque constant
K_r – load torque constant
J_r – rotor inertia
C_s – solid friction

And a_0=189.63, a_1=6.0612, a_2=0.0122 and d=280.19 (Derafa1, Madani2, and Benallegue2 2006).

3.8 State Space Representation of the Dynamic Model

The rotational and translational equations of motion, once derived, can be alternatively expressed in a state-space format. In the case of a quadrotor system, this state-space representation consists of 12 states, and the state vector can be defined as follows:

$$x = \left[x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8 \; x_9 \; x_{10} \; x_{11} \; x_{12} \right]^T \tag{3.35}$$

Euler angles and angular velocities in body frame is used in the rotational equations of motion and the orientation of the quadrotor is directly defined with *X*, *Y* and *Z* for translational dynamics. Thus, the states can be written as:

$$x = \left[X \;\; \dot{X} \;\; Y \;\; \dot{Y} \;\; Z \;\; \dot{Z} \;\; P \;\; Q \;\; R \;\; \phi \;\; \theta \;\; \psi \;\; \right]^T \tag{3.36}$$

In order to form the state space model of the system the control input vector should also be defined. Therefore, control vector $\bar{U}$ can be represented as:

$$\bar{U} = [T \;\; \tau_\phi \;\; \tau_\theta \;\; \tau_\psi]^T \tag{3.37}$$

By obtaining the control input expressions from Equations (3.23) and (3.24), the control vector $\bar{U}$ can be written as follows:

$$\bar{U} = \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -lK_T & 0 & lK_T \\ -lK_T & 0 & lK_T & 0 \\ K_D & -K_D & K_D & -K_D \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3.38}$$

From Equation (3.31) the desired speed to the motor is calculated as follows:

$$\begin{bmatrix} \Omega_{1des}^2 \\ \Omega_{2des}^2 \\ \Omega_{3des}^2 \\ \Omega_{4des}^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_T} & 0 & -\frac{1}{2lK_T} & \frac{1}{4K_D} \\ \frac{1}{4K_T} & -\frac{1}{2lK_T} & 0 & -\frac{1}{4K_D} \\ \frac{1}{4K_T} & 0 & \frac{1}{2lK_T} & \frac{1}{4K_D} \\ \frac{1}{4K_T} & \frac{1}{2lK_T} & 0 & -\frac{1}{4K_D} \end{bmatrix} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{3.39}$$

Finally, the state space representation of the overall quadrotor system is given as follows:

$$\dot{X} = f(x,\bar{U},t) = \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{u_x T}{m} + D_x \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \dfrac{u_y T}{m} + D_y \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -g + \dfrac{u_z T}{m} + D_z \\ \dot{x}_7 = a_1 x_8 x_9 - a_2 x_8 \Omega_r + b_1 \tau_\phi + D_\phi \\ \dot{x}_8 = a_3 x_7 x_9 + a_4 x_7 \Omega_r + b_2 \tau_\theta + D_\theta \\ \dot{x}_9 = a_5 x_7 x_8 + b_3 \tau_\psi + D_\psi \\ \dot{x}_{10} = x_7 + x_8 \sin(x_{10}) \tan(x_{11}) + x_9 \cos(x_{10}) \tan(x_{11}) \\ \dot{x}_{11} = x_8 \cos(x_{10}) - x_9 \sin(x_{10}) \\ \dot{x}_{12} = \dfrac{x_8 \sin(x_{10})}{\cos(x_{11})} + \dfrac{x_9 \cos(x_{10})}{\cos(x_{11})} \end{cases} \tag{3.40}$$

where

$$\begin{aligned} a_1 &= \frac{I_{yy} - I_{zz}}{I_{xx}}, a_2 = \frac{J_r}{I_{xx}}, b_1 = \frac{1}{I_{xx}} \\ a_3 &= \frac{I_{zz} - I_{xx}}{I_{yy}}, a_4 = \frac{J_r}{I_{yy}}, b_2 = \frac{1}{I_{yy}} \\ a_5 &= \frac{I_{xx} - I_{yy}}{I_{zz}}, b_3 = \frac{1}{I_{zz}} \\ u_x &= \cos_\psi \sin_\theta \cos_\phi + \sin_\psi \sin_\phi \\ u_y &= \sin_\psi \sin_\theta \cos_\phi - \cos_\psi \sin_\phi \\ u_z &= \cos_\theta \cos_\phi \end{aligned}$$

And D_x, D_y, D_z, D_ϕ, D_θ and D_ψ are the combined effects of unknown model parameter uncertainties and unknown disturbances (such as aerodynamic effects which are not modelled accurately).

Similarly for the actuator dynamics by defining:

$$\Omega = x_m \tag{3.41}$$

The state space representation of the actuator dynamics is given by:

$$\dot{x}_m = f(x_m, V, t) = -(a_0 + a_1 x_m + a_2 x_m^2) + dV \tag{3.42}$$

3.9 CALCULATION OF DESIRED ROLL AND PITCH ANGLES

As shown in Equation (3.40) the virtual controls U_x and U_y are given by:

$$U_x = \cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi) \tag{3.43}$$

$$U_y = \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) \tag{3.44}$$

Multiply Equation (3.43) by $sin(\psi)$ it becomes as follows:

$$U_x \sin(\psi) = \cos(\psi)\sin(\psi)\sin(\theta)\cos(\phi) + \sin^2(\psi)\sin(\phi) \tag{3.45}$$

Rearranging Equation (3.45) gives

$$\begin{aligned} &U_x \sin(\psi) = \cos(\psi)\left[U_y + \cos(\psi)\sin(\phi)\right] + \sin^2(\psi)\sin(\phi) \\ &\sin^2(\psi)\sin(\phi) + \cos^2(\psi)\sin(\phi) = U_x \sin(\psi) - U_y \cos(\psi) \\ &\sin(\phi)\left[\sin^2(\psi) + \cos^2(\psi)\right] = U_x \sin(\psi) - U_y \cos(\psi) \end{aligned} \tag{3.46}$$

Thus, from Equation (3.46) the resulting desired roll angle is given by:

$$\phi_{desired} = \sin^{-1}\left(U_x \sin(\psi) - U_y \cos(\psi)\right) \tag{3.47}$$

Similarly, the desired pitch angle is given by:

$$\theta_{desired} = \tan^{-1}\left(\frac{U_x \cos(\psi) + U_y \sin(\psi)}{u_z}\right) \tag{3.48}$$

where $u_z = cos(\phi)cos(\theta)$.

3.10 MODEL VALIDATION

After developing the mathematical model of the quadrotor system verification of the model is done by simulating the open-loop response in MATLAB software environment.

The open-loop dynamic demonstrates the effect of input on the output state by using rotor speed for calculating the total thrust, roll torque, pitch torque and yaw torque.

3.10.1 Movement in the Positive Z direction

Movement along positive Z axis (altitude change) is obtained when the total thrust generated by the rotors is greater than the downward gravitational force due to the total mass of the quadrotor. In order to generate this thrust, the speed of the rotor must be greater than the nominal (equilibrium) speed.

At the ground level the total thrust generated by the rotors is equal to the gravitational force due to the total mass of the quadrotor system.

$$T = mg \tag{3.49}$$

By substituting Equation (3.23) into Equation (3.43) the minimum speed which creates movement along the positive Z axis is given by:

$$\Omega_e = \sqrt{\frac{mg}{4K_T}} \tag{3.50}$$

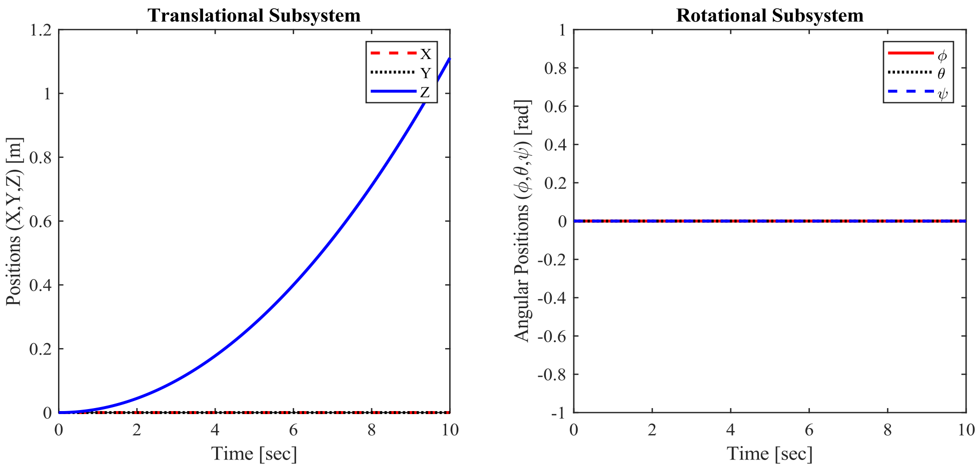


Figure 3.5 Open-loop response of a quadrotor at $\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4 = 287$ rad/sec

As we have seen from the open-loop responses in Figure 3.5 when the speed of all the rotors is equal there is only a movement along positive Z direction this is because when all the rotors have the same speed only the control input T has a value.

3.10.2 Movement along the Positive X direction

To create the movement along the positive *X* direction we have to create a pitch torque which creates a positive pitch angle by increasing the speed of rotor 3 and decreasing speed of rotor 1 and by giving the other two equal speed.

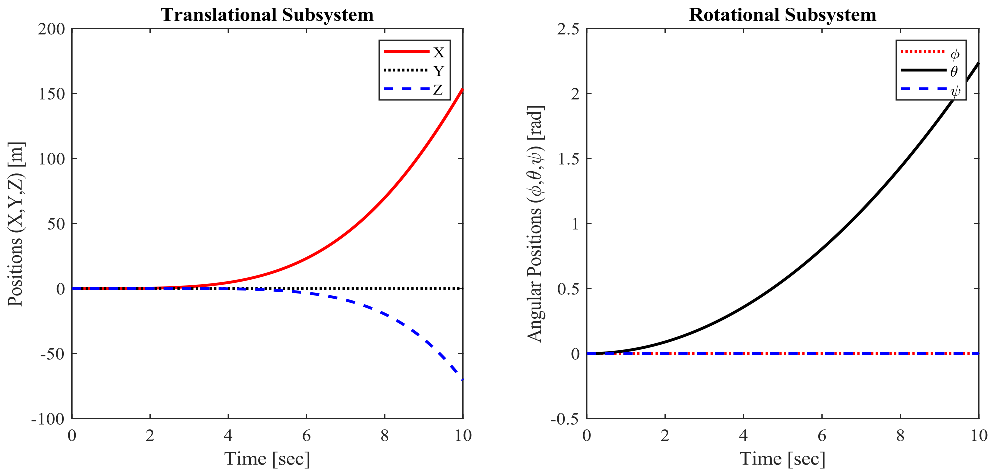


Figure 3.6 Open-loop response of a quadrotor at Ω_1=286.99, Ω_3= 287.01 and Ω_2=Ω_4= 287 rad/sec

3.10.3 Movement along Positive Y direction

To create the movement along the positive *Y* direction we have to create a roll torque which creates a negative roll angle by increasing the speed of rotor 2 and decreasing speed of rotor 4 and by giving the other two equal speed.

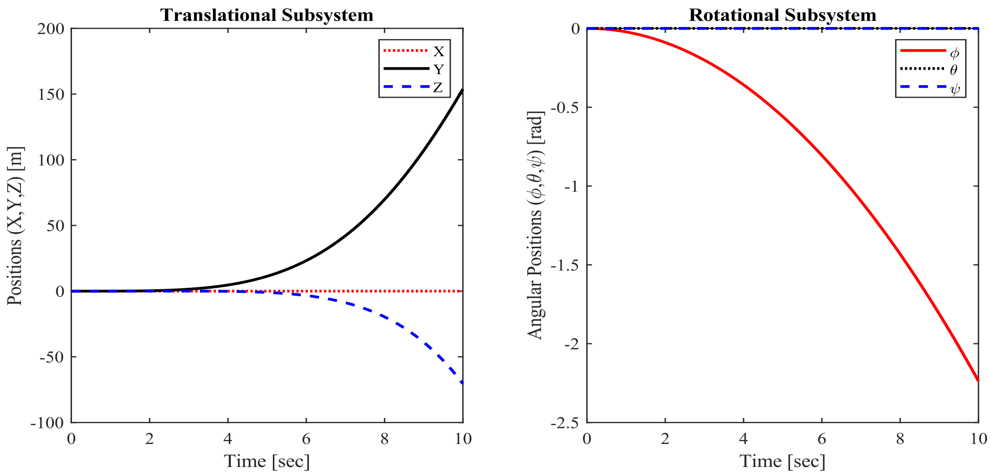


Figure 3.7 Open-loop response of a quadrotor at Ω_4=286.99, Ω_2= 287.01 and Ω_1=Ω_3= 287 rad/sec

CHAPTER FOUR

4. DESIGN OF A CONTROLLER FOR THE QUADROTOR

4.1 INTRODUCTION

In this chapter a BSMC for the quadrotor system is designed and stability of the designed controller is checked in Lyapunov stability analysis. Finally, PSO is applied to find the control parameters with a discussion of objective function used.

4.2 DESIGN OF A BACKSTEPPING SLIDING MODE CONTROLLER

In this section, a quadrotor's nonlinear controller, which incorporates height control, attitude control and trajectory control for stabilization is considered. The primary goal of the designed controller is to make sure that the quadrotor's location X, Y, Z and ψ asymptotically follows the required trajectories X_{des}, Y_{des}, Z_{des} and ψ_{des}. Additionally, the responsibility of the attitude controller is to give the control inputs τ_ϕ, τ_θ and τ_ψ that makes sure the Euler angles ϕ, θ and ψ are following the desired attitude trajectories ϕ_{des}, θ_{des} and ψ_{des} within the prescribed time.

The overall control block diagram is shown in Figure 4.1 and stability analysis of the designed controller is verified by Lyapunov stability theorem.

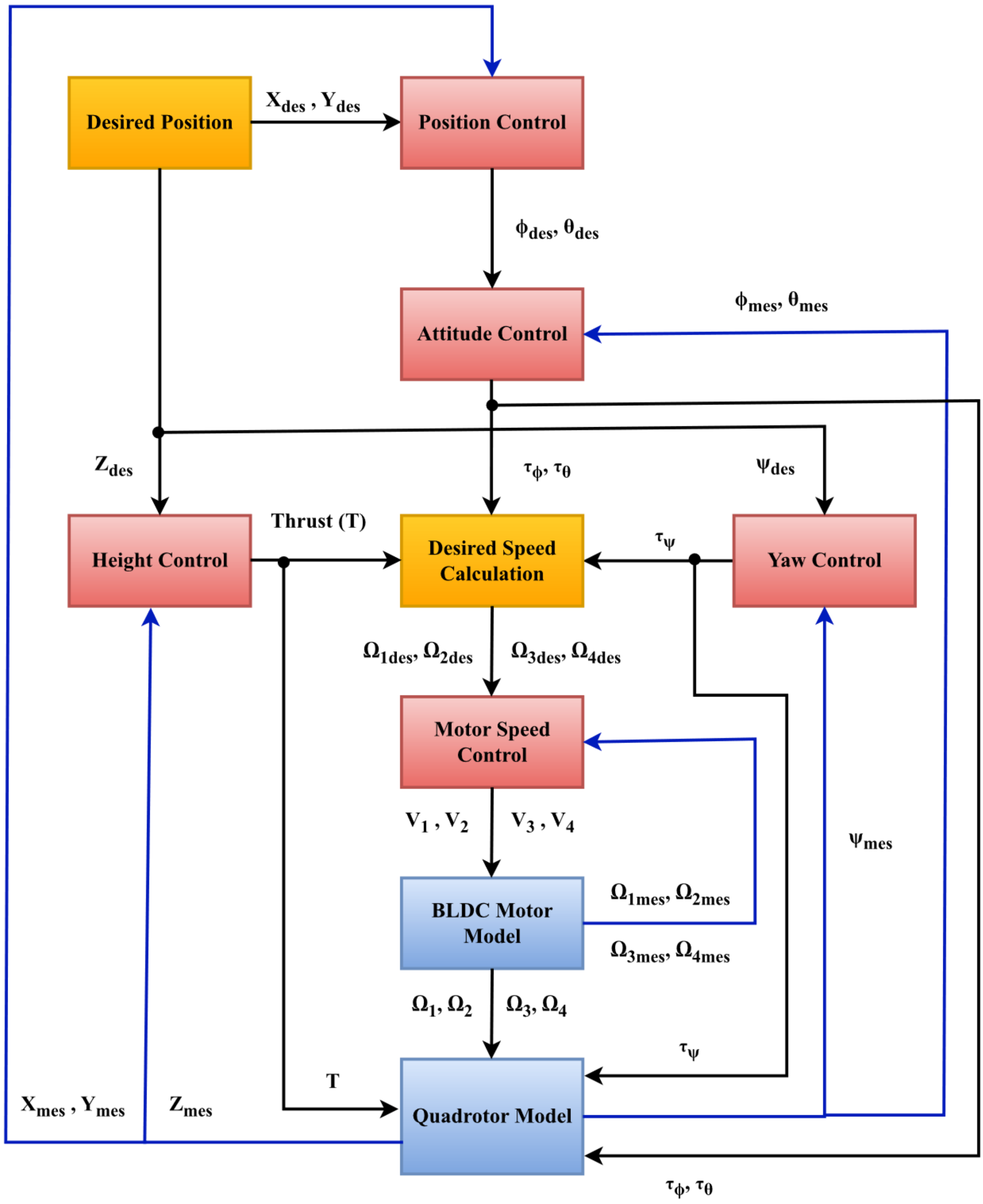


Figure 4.1 Overall Control Block Diagram

Theorem 1: Consider a system given by:

$$\dot{x} = f(x,t), x_0 \tag{4.1}$$

It is said to be Lyapunov stable if a Lyapunov function *V(x)* satisfying the following conditions exists:

1. $V(x) \geq 0$ for $\forall x$, and $V(x) = 0$ for $x = 0$,

2. $\dot{V}(x) < 0$ for $\forall x \neq 0$, and $\dot{V}(x) = 0$ for $x = 0$.

4.2.1 Height Control

The thrust force control input *T*, which is the sum of each rotor's thrust force is used to control the height subsystem. So, the height control law is formulated as follows to control the height (distance from the ground to the center of the quadrotor body) of the quadrotor.

From the state space representation of the quadrotor system of Equation (3.40) we have

$$\begin{aligned} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -g + \frac{u_z T}{m} \end{aligned} \tag{4.2}$$

Step 1: - define tracking error e_z and its derivative $\dot{e}_z$ in position *Z* as follows:

$$\begin{aligned} e_z &= x_5 - z_{des} \\ \dot{e}_z &= \dot{x}_5 - \dot{z}_{des} = x_6 - \dot{z}_{des} \end{aligned} \tag{4.3}$$

By choosing a virtual control of $\tilde{u}_1$ and a virtual error e_{2z} as follows:

$$\begin{aligned} \tilde{u}_1 &= \dot{z}_{des} - c_z e_z \\ e_{2z} &= x_6 - \tilde{u}_1 = x_6 - \dot{z}_{des} + c_z e_z \end{aligned} \tag{4.4}$$

Where c_z is a positive constant control parameter, and the error e_{2z} is converges to zero when the tracking error e_z is zero. Then, the candidate Lyapunov function V_{1z} to stabilize the height control system can be defined using these error definitions as follows:

$$V_{1z} = \frac{1}{2} e_z^2 \tag{4.5}$$

Then, the Lyapunov candidate's time derivative can be represented by:

$$\begin{aligned} \dot{V}_{1z} &= e_z \dot{e}_z = e_z (x_6 - \dot{z}_{des}) \\ &= e_z \dot{e}_z = e_z (e_{2z} - c_z e_z) \\ &= e_z e_{2z} - c_z e_z^2 \end{aligned} \tag{4.6}$$

If we set e_{2z}=0, then $V_{1z} = -c_z e_z^2 \leq 0$ which satisfies the Lyapunov stability criterion. Hence, let us proceed to the next step.

Step 2: - define a switching surface (sliding surface) δ_z and its derivative $\dot{\delta}_z$ as follows:

$$\begin{aligned} \delta_z &= k_z e_z + e_{2z} \\ \dot{\delta}_z &= k_z \dot{e}_z + \dot{e}_{2z} \end{aligned} \tag{4.7}$$

And the augmented Lyapunov function V_{2z} is given by:

$$V_{2z} = V_{1z} + \frac{1}{2}\delta_z^2 \tag{4.8}$$

Then, the time derivative of V_{2z} is given by:

$$\dot{V}_{2z} = \dot{V}_{1z} + \delta_z \dot{\delta}_z \tag{4.9}$$

From Equation (4.4) $\dot{e}_z$ and e_{2z} are represented as follows:

$$\begin{aligned} \dot{e}_z &= \dot{e}_{2z} - c_z \dot{e}_z \\ \dot{e}_{2z} &= \dot{x}_6 - \dot{\ddot{u}}_1 \end{aligned} \tag{4.10}$$

Substituting Equation (4.3-4.4) and Equation (4.10) into Equation (4.9) gives: -

$$\begin{aligned} \dot{V}_{2z} &= e_z e_{2z} - c_z e_z^2 + \delta_z (k_z \dot{e}_z + \dot{e}_{2z}) \\ &= e_z e_{2z} - c_z e_z^2 + \delta_z (k_z (e_{2z} - c_z e_z) + \dot{x}_6 - \dot{\ddot{u}}_1) \\ &= e_z e_{2z} - c_z e_z^2 + \delta_z (k_z (e_{2z} - c_z e_z) - g + \frac{u_z T}{m} - \dot{\ddot{u}}_1) \end{aligned} \tag{4.11}$$

In order to satisfy Lyapunov stability criterion $\dot{V}_{2z} \leq 0$ choose thrust force control input T as follows:

$$T = -\frac{m}{u_z}\left[(k_z + c_z)(e_{2z} - c_z e_z) - g - \ddot{z}_{des} + h_z \delta_z + \mu_z sign(\delta_z)\right] \tag{4.12}$$

where k_z, c_z, h_z, $\mu_z > 0$ and $u_z = cos(\phi)cos(\theta)$.

Let us check the stability of the designed controller by substituting Equation (4.12) into Equation (4.11).

$$\begin{aligned}
\dot{V}_{2z} &= e_z e_{2z} - c_z e_z^2 + \delta_z (k_z (e_{2z} - c_z e_z) - g + \frac{u_z T}{m} - \dot{\hat{u}}_1) \\
&= e_z e_{2z} - c_z e_z^2 + \delta_z [-c_z (e_{2z} - c_z e_z) + \ddot{z}_{des} - h_z \delta_z - \mu_z sign(\delta_z) - \dot{\hat{u}}_1] \\
&= e_z e_{2z} - c_z e_z^2 + \delta_z [-c_z e_{2z} + c_z^2 e_z + \ddot{z}_{des} - h_z \delta_z - \mu_z sign(\delta_z) - \ddot{z}_{des} + c_z \dot{e}_z] \\
&= e_z e_{2z} - c_z e_z^2 + \delta_z [-c_z e_{2z} + c_z^2 e_z - h_z \delta_z - \mu_z sign(\delta_z) + c_z (e_{2z} - c_z e_z)] \\
&= e_z e_{2z} - c_z e_z^2 + \delta_z [-h_z \delta_z - \mu_z sign(\delta_z)] \\
&= e_z e_{2z} - c_z e_z^2 - h_z \delta_z^2 - \mu_z \delta_z sign(\delta_z) \\
&= e_z e_{2z} - c_z e_z^2 - h_z \delta_z^2 - \mu_z |\delta_z|
\end{aligned} \tag{4.13}$$

Since c_z, h_z, $\mu_z > 0$ thus if we set $e_{2z}=0$ then the time derivative of Lyapunov candidate is semi-negative definite as follows:

$$\dot{V}_{2z} = -c_z e_z^2 - h_z \delta_z^2 - \mu_z |\delta_z| \leq 0 \tag{4.14}$$

Thus, the system is asymptotically stable.

4.2.2 Attitude Control

Rotational (attitude) subsystem of a quadrotor consists roll angle, pitch angle and yaw angle which is the crucial part for stabilization and tracking control problem quadrotor.

From the state space representation of the quadrotor system of Equation (3.40) we have

$$\begin{aligned}
\dot{x}_7 &= a_1 x_8 x_9 - a_2 x_8 \Omega_r + b_1 \tau_\phi \\
\dot{x}_8 &= a_3 x_7 x_9 + a_4 x_7 \Omega_r + b_2 \tau_\theta \\
\dot{x}_9 &= a_5 x_7 x_8 + b_3 \tau_\psi \\
\dot{x}_{10} &= x_7 + x_8 \sin(x_{10}) \tan(x_{11}) + x_9 \cos(x_{10}) \tan(x_{11}) \\
\dot{x}_{11} &= x_8 \cos(x_{10}) - x_9 \sin(x_{10}) \\
\dot{x}_{12} &= \frac{x_8 \sin(x_{10})}{\cos(x_{11})} + \frac{x_9 \cos(x_{10})}{\cos(x_{11})}
\end{aligned} \tag{4.15}$$

Let us define states in a matrix form to shorten the control design procedures as follows:

$$X_{Rot}^{Earth} = X_R^E = \begin{bmatrix} x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{4.16}$$

$$X_{Rot}^{Body} = X_R^B = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4.17}$$

From Equation (3.15) a relation between the body frame angular velocity to earth or inertial frame velocity is represented as follows:

$$T_M = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{4.18}$$

$$T_M^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{4.19}$$

$$\dot{X}_R^E = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_M^{-1} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4.20}$$

$$X_R^B = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = T_M \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.21}$$

Where T_M is a transformation matrix and T_M^{-1} is its inverse.

Thus, the updated state space representation for the rotational (attitude) subsystem is given by

$$\begin{aligned} \dot{X}_R^E &= \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_M^{-1} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = T_M^{-1} X_R^B \\ \dot{X}_R^B &= \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} a_1 x_8 x_9 - a_2 x_8 \Omega_r \\ a_3 x_7 x_9 + a_4 x_7 \Omega_r \\ a_5 x_7 x_8 \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \\ \dot{X}_R^B &= A + BU_2 \end{aligned} \tag{4.22}$$

where

$$A = \begin{bmatrix} a_1 x_8 x_9 - a_2 x_8 \Omega_r \\ a_3 x_7 x_9 + a_4 x_7 \Omega_r \\ a_5 x_7 x_8 \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \tag{4.23}$$

$$U_2 = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

Now let us start formulating the attitude controller by defining the rotational (attitude) tracking error E_{1R} and its derivative $\dot{E}_{1R}$ as follows:

$$\begin{gathered} E_{1R} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} - \begin{bmatrix} \phi_{des} \\ \theta_{des} \\ \psi_{des} \end{bmatrix} = X_R^E - X_{Rdes}^E \\ \dot{E}_{1R} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} \dot{\phi}_{des} \\ \dot{\theta}_{des} \\ \dot{\psi}_{des} \end{bmatrix} = \dot{X}_R^E - \dot{X}_{Rdes}^E \end{gathered} \tag{4.24}$$

Considering a candidate Lyapunov function of V_{1R} to stabilize the attitude control system:

$$V_{1R} = \frac{1}{2} E_{1R} E_{1R}^T \tag{4.25}$$

The time derivative of V_{1R} is given by:

$$\begin{aligned} \dot{V}_{1R} &= E_{1R}^T \dot{E}_{1R} \\ &= E_{1R}^T \left(\dot{X}_R^E - \dot{X}_{Rdes}^E \right) \end{aligned} \tag{4.26}$$

By choosing a virtual control of $\tilde{U}_2$ and a virtual control error of E_{2R} which satisfies the semi negative definiteness of the candidate Lyapunov function $\dot{V}_{1R} \leq 0$ as follows:

$$\begin{aligned} \tilde{U}_2 &= \dot{X}_{Rdes}^E - C_R E_{1R} \\ E_{2R} &= \dot{X}_R^E - \tilde{U}_2 = \dot{X}_R^E - \dot{X}_{Rdes}^E + C_R E_{1R} \end{aligned} \tag{4.27}$$

By substituting Equation (4.28) into Equation (4.27) we get:

$$\begin{aligned}\dot{V}_{1R} &= E_{1R}^T \dot{E}_{1R} \\ &= E_{1R}^T \left(E_{2R} + \dot{X}_{Rdes}^{E} - C_R E_{1R} - \dot{X}_{Rdes}^{E} \right) \\ &= E_{1R}^T \left(E_{2R} - C_R E_{1R} \right) \\ &= E_{1R}^T E_{2R} - E_{1R}^T C_R E_{1R} \\ &= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 \end{aligned} \tag{4.28}$$

If we set E_{2R} =0, then $\dot{V}_{1R} = -C_R^T E_{1R}^2$ which satisfies the Lyapunov stability criterion. Hence, let us proceed to the next step by defining a sliding surface δ_R and its derivative $\dot{\delta}_R$ as follows:

$$\begin{aligned}\delta_R &= K_R E_{1R} + E_{2R} \\ \dot{\delta}_R &= K_R \dot{E}_{1R} + \dot{E}_{2R}\end{aligned} \tag{4.29}$$

And the augmented Lyapunov function V_{2R} is given by:

$$V_{2R} = V_{1R} + \frac{1}{2}\delta_R \delta_R^T \tag{4.30}$$

From Equation (4.25) and (4.28) we have: -

$$\begin{aligned}\dot{E}_{1R} &= \dot{X}_{R}^{E} - \dot{X}_{Rdes}^{E} \\ &= E_{2R} + \dot{X}_{Rdes}^{E} - C_R E_{1R} - \dot{X}_{Rdes}^{E} \\ &= E_{2R} - C_R E_{1R}\end{aligned} \tag{4.31}$$

Then, the time derivative of V_{2R} is given by:

$$\begin{aligned}\dot{V}_{2R} &= \dot{V}_{1R} + \delta_R^T \dot{\delta}_R \\ &= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \dot{\delta}_R \\ &= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(K_R \dot{E}_{1R} + \dot{E}_{2R} \right) \\ &= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(K_R \dot{E}_{1R} + \ddot{X}_{R}^{E} - \dot{\hat{U}}_2 \right) \\ &= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(\ddot{X}_{R}^{E} + \ddot{X}_{Rdes}^{E} + \left(K_R - C_R \right) \dot{E}_{1R} \right)\end{aligned} \tag{4.32}$$

In order to satisfy the Lyapunov stability criterion $\dot{V}_{2R}$ in the Equation (4.33) the derivative of sliding surface $\dot{\delta}_R$ must have to satisfy $\dot{\delta}_R \leq 0$ which formulated as follows:

From sliding mode theorem,

$$\dot{\delta}_R = \ddot{X}_R^E + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} = -H_R\delta_R - \beta_R \tag{4.33}$$

where

$$\beta_R = \begin{bmatrix} \mu_\phi & 0 & 0 \\ 0 & \mu_\theta & 0 \\ 0 & 0 & \mu_\psi \end{bmatrix} \begin{bmatrix} sign(\delta_\phi) \\ sign(\delta_\theta) \\ sign(\delta_\psi) \end{bmatrix} \tag{4.34}$$

Thus, the control input which satisfy Equation (4.34) is formulated as follows:

$$\begin{aligned}
&\ddot{X}_R^E + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} = -H_R\delta_R - \beta_R \\
&\ddot{X}_R^E = -\ddot{X}_{Rdes}^E - \left(K_R - C_R\right)\dot{E}_{1R} - H_R\delta_R - \beta_R = G_R \\
&\dot{T}_M^{-1}X_R^B + T_M^{-1}\dot{X}_R^B = G_R \\
&\dot{T}_M^{-1}X_R^B + T_M^{-1}\left(A + BU_2\right) = G_R \\
&T_M^{-1}\left(A + BU_2\right) = G_R - \dot{T}_M^{-1}X_R^B \\
&A + BU_2 = T_M\left(G_R - \dot{T}_M^{-1}X_R^B\right) \\
&U_2 = B^{-1}\left[T_M\left(G_R - \dot{T}_M^{-1}X_R^B\right) - A\right]
\end{aligned} \tag{4.35}$$

Where $C_R, K_R, H_R \in \mathbb{R}^{3\times3}$ are positive definite diagonal matrices and G_R is given by:

$$G_R = -\ddot{X}_{Rdes}^E - \left(K_R - C_R\right)\dot{E}_{1R} - H_R\delta_R - \beta_R \tag{4.36}$$

Let us check the stability of the designed controller by substituting Equation (4.36) into Equation (4.33):

$$
\begin{aligned}
\dot{V}_{2R} &= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(\ddot{X}_R^E + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} \right) \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(\dot{T}_M^{-1} X_R^B + T_M^{-1} \dot{X}_R^B + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} \right) \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(\dot{T}_M^{-1} X_R^B + T_M^{-1} \left(A + BU_2\right) + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} \right) \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \begin{pmatrix} \dot{T}_M^{-1} X_R^B + T_M^{-1}\left(A + B\left(B^{-1}\left[T_M\left(G_R - \dot{T}_M^{-1} X_R^B\right) - A\right]\right)\right) \\ + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} \end{pmatrix} \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(G_R + \ddot{X}_{Rdes}^E + \left(K_R - C_R\right)\dot{E}_{1R} \right) \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \begin{pmatrix} -\ddot{X}_{Rdes}^E - \left(K_R - C_R\right)\dot{E}_{1R} - H_R\delta_R - \beta_R + \ddot{X}_{Rdes}^E \\ + \left(K_R - C_R\right)\dot{E}_{1R} \end{pmatrix} \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 + \delta_R^T \left(-H_R\delta_R - \beta_R \right) \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 - \delta_R^T H_R \delta_R - \delta_R^T \beta_R \\
&= E_{1R}^T E_{2R} - C_R^T E_{1R}^2 - H_R^T \delta_R^2 - \delta_R^T \beta_R
\end{aligned}
\tag{4.37}
$$

Since C_R, K_R and H_R are positive definite diagonal matrices thus if we set $E_{2R} = 0$ then the time derivative of Lyapunov candidate is semi-negative definite as follows:

$$
\dot{V}_{2R} = -C_R^T E_{1R}^2 - H_R^T \delta_R^2 - \delta_R^T \left|\beta_R\right| \le 0 \tag{4.38}
$$

Thus, the system is asymptotically stable.

4.2.3 Trajectory Control

In trajectory control the positions X and Y are controlled to enable the quadrotor follow the given desired trajectories effectively.

From the state space representation of the quadrotor system of Equation (3.40) we have: -

$$
\begin{aligned}
\dot{x}_1 &= \dot{x}_2 \\
\dot{x}_2 &= \frac{U_x T}{m} \\
\dot{x}_3 &= \dot{x}_4 \\
\dot{x}_4 &= \frac{U_y T}{m}
\end{aligned}
\tag{4.39}
$$

Since the quadrotor system is underactuated it is not possible to control position *X* and *Y* independently. So, we need to design virtual control U_x and U_y as follows:

Let us start by defining tracking error and its derivative for position *X* and the procedures is same for position *Y*.

$$
\begin{aligned}
e_x &= x_1 - x_{des} \\
\dot{e}_x &= \dot{x}_1 - \dot{x}_{des} = x_2 - \dot{x}_{des}
\end{aligned} \tag{4.40}
$$

By choosing a virtual control of $\tilde{u}_x$ and a virtual error e_{2x} as follows:

$$
\begin{aligned}
\tilde{u}_x &= \dot{x}_1 - c_x e_x \\
e_{2x} &= x_2 - \tilde{u}_x = x_2 - \dot{x}_1 + c_x e_x
\end{aligned} \tag{4.41}
$$

Where c_x is a positive constant control parameter, and the error e_{2x} is converges to zero when the tracking error e_x is zero. Then, the candidate Lyapunov function V_{1x} to stabilize the position X control system can be defined using these error definitions as follows:

$$
V_{1x} = \frac{1}{2} e_x^2 \tag{4.42}
$$

Then, the Lyapunov candidate's time derivative can be represented by:

$$
\begin{aligned}
\dot{V}_{1x} &= e_x \dot{e}_x = e_x (x_2 - \dot{x}_{des}) \\
&= e_x \dot{e}_x = e_x (e_{2x} - c_x e_x) \\
&= e_x e_{2x} - c_x e_x^2
\end{aligned} \tag{4.43}
$$

If we set e_{2x}=0, then $\dot{V}_{1x} = -c_x e_x^2 \leq 0$ which satisfies the Lyapunov stability criterion. Hence, let us proceed to the next step by defining a sliding surface δ_x and its derivative $\dot{\delta}_x$ as follows:

$$
\begin{aligned}
\delta_x &= k_x e_x + e_{2x} \\
\dot{\delta}_x &= k_x \dot{e}_x + \dot{e}_{2x}
\end{aligned} \tag{4.44}
$$

And the augmented Lyapunov function V_{2x} is given by:

$$
\dot{V}_{2x} = \dot{V}_{1x} + \delta_x \dot{\delta}_x \tag{4.45}
$$

From Equation (4.17) we get $\dot{e}_x$ and e_{2x} as follows:

$$
\begin{aligned}
\dot{e}_x &= \dot{e}_{2x} - c_x \dot{e}_x \\
&= \dot{x}_2 - \dot{\tilde{u}}_x
\end{aligned} \tag{4.46}
$$

Substituting Equation (4.16-4.17) and Equation (4.22) into Equation (4.21) gives:

$$\begin{aligned}\dot{V}_{2x} &= e_x e_{2x} - c_x e_x^2 + \delta_x (k_x \dot{e}_x + \dot{e}_{2x}) \\ &= e_x e_{2x} - c_x e_x^2 + \delta_x (k_x (e_{2x} - c_x e_x) + \dot{x}_2 - \dot{\tilde{u}}_x) \\ &= e_x e_{2x} - c_x e_x^2 + \delta_x (k_x (e_{2x} - c_x e_x) + \frac{U_x T}{m} - \dot{\tilde{u}}_x)\end{aligned} \tag{4.47}$$

In order to satisfy Lyapunov stability criterion $\dot{V}_{2x} \le 0$ choose virtual control input U_x as follows:

$$U_x = -\frac{m}{T}\left[(k_x + c_x)(e_{2x} - c_x e_x) - \ddot{x}_{des} + h_x \delta_x + \mu_x sign(\delta_x)\right] \tag{4.48}$$

Let us check the stability of the designed controller by substituting Equation (4.24) into Equation (4.23):

$$\begin{aligned}\dot{V}_{2x} &= e_x e_{2x} - c_x e_x^2 + \delta_x (k_x (e_{2x} - c_x e_x) + \frac{U_x T}{m} - \dot{\tilde{u}}_x) \\ &= e_x e_{2x} - c_x e_x^2 + \delta_x [-c_x (e_{2x} - c_x e_x) + \ddot{x}_{des} - h_x \delta_x - \mu_x sign(\delta_x) - \dot{\tilde{u}}_x] \\ &= e_x e_{2x} - c_x e_x^2 + \delta_x [-c_x e_{2x} + c_x^2 e_x + \ddot{x}_{des} - h_x \delta_x - \mu_x sign(\delta_x) - \ddot{x}_{des} + c_x \dot{e}_x] \\ &= e_x e_{2x} - c_x e_x^2 + \delta_x [-c_x e_{2x} + c_x^2 e_x - h_x \delta_x - \mu_x sign(\delta_x) + c_x (e_{2x} - c_x e_x)] \\ &= e_x e_{2x} - c_x e_x^2 + \delta_x [-h_x \delta_x - \mu_x sign(\delta_x)] \\ &= e_x e_{2x} - c_x e_x^2 - h_x \delta_x^2 - \mu_x \delta_x sign(\delta_x) \\ &= e_x e_{2x} - c_x e_x^2 - h_x \delta_x^2 - \mu_x \left|\delta_x\right|\end{aligned} \tag{4.49}$$

Since c_x, h_x, μ_x>0 thus if we set e_{2x}=0 then the time derivative of Lyapunov candidate is semi-negative definite as follows:

$$\dot{V}_{2x} = -c_x e_x^2 - h_x \delta_x^2 - \mu_x \left|\delta_x\right| \le 0 \tag{4.50}$$

Thus, the system is asymptotically stable.

Using the same procedures used to get the virtual control input U_x similarly U_y is given by:

$$U_y = -\frac{m}{T}\left[(k_y + c_y)(e_{2y} - c_y e_y) - \ddot{y}_{des} + h_y \delta_y + \mu_y sign(\delta_y)\right] \tag{4.51}$$

In general, the control inputs to stabilize the translational subsystems are given by:

$$
\begin{aligned}
T &= -\frac{m}{u_z}\left[(k_z + c_z)(e_{2z} - c_z e_z) - g - \ddot{z}_{des} + h_z\delta_z + \mu_z sign(\delta_z)\right] \\
U_x &= -\frac{m}{T}\left[(k_x + c_x)(e_{2x} - c_x e_x) - \ddot{x}_{des} + h_x\delta_x + \mu_x sign(\delta_x)\right] \\
U_y &= -\frac{m}{T}\left[(k_y + c_y)(e_{2y} - c_y e_y) - \ddot{y}_{des} + h_y\delta_y + \mu_y sign(\delta_y)\right]
\end{aligned} \tag{4.52}
$$

Where T is a thrust force which drives the quadrotor up and down, U_x is a virtual control of position X and U_y is a virtual control of position Y.

Finally, the control parameters are optimized using particle swarm optimization (PSO) algorithm.

4.3 Tuning of the Controller Parameters

The parameters of the designed backstepping sliding mode controller (BSMC) for the quadrotor system are tuned using particle swarm optimization algorithm. There are 24 parameters for the quadrotor system. These are: -

$$k_\phi, k_\theta, k_\psi, k_x, k_y, k_z, c_\phi, c_\theta, c_\psi, c_x, c_y, c_z, h_\phi, h_\theta, h_\psi, h_x, h_y, h_z, \mu_\phi, \mu_\theta, \mu_\psi, \mu_x, \mu_y, \mu_z$$

PSO employs a method that operates at a population level, utilizing a collective of particles to create a swarm. Each individual particle corresponds to a potential solution. The collection of candidate solutions exists and collaborates concurrently. Within the swarm, each particle traverses the search space, seeking the optimal solution to converge upon. Thus, the search space encompasses the range of feasible solutions, while the dynamic assemblage of airborne particles symbolizes the evolving set of solutions (Gad 2022).

Over successive iterations, every particle within the group maintains records of its own best solution and the best solution achieved by the entire group. Subsequently, two key parameters, namely flying speed, and position, are adjusted. More specifically, each particle dynamically adapts its flying speed based on its own experiences as well as those of its neighboring particles. Similarly, it attempts to alter its position by considering its current position, velocity, the distance between its current position and personal best solution, and the distance between its current position and the best solution achieved by the entire group. The swarm of particles, akin to a flock of birds, continues to move towards a promising region until it reaches the global optimum, thereby solving the optimization problem. The lower and upper bounds used to initialize PSO are chosen based on the Lyapunov stability criterion and actuator limit respectively.

The control parameters are optimized under the following control input constraints:

$$
\begin{aligned}
&0 \le T \le 4K_T\Omega_{\max}^2 \\
&-K_T l\Omega_{\max}^2 \le \tau_\phi, \tau_\theta \le K_T l\Omega_{\max}^2 \\
&-2K_D\Omega_{\max}^2 \le \tau_\psi \le 2K_D\Omega_{\max}^2
\end{aligned}
\tag{4.53}
$$

Where Ω_{max} is the maximum speed of the motors (actuators).

The proposed method enables the evaluation of how well the quadrotor responds to optimization efforts through a performance index. This index, also referred to as the objective function, aims to minimize the disparities between the desired and actual output responses of the quadrotor. The focus of these theses is to decrease the errors in the quadrotor's trajectory tracking, specifically in terms of position and orientation. The control engineering field favors performance indexes such as Integral Squared Error (ISE), Integral Time Squared Error (ITSE), Integral Absolute-Error (IAE), and Integral Time Absolute Error (ITAE), which are presented below (Belsty 2021):

$$
\begin{aligned}
&ISE = \int_0^\infty e^2(t)dt \\
&ITSE = \int_0^\infty te^2(t)dt \\
&IAE = \int_0^\infty |e(t)|\,dt \\
&ITAE = \int_0^\infty t\,|e(t)|\,dt
\end{aligned}
\tag{4.54}
$$

Each performance index listed in Equation (4.54) possesses distinct characteristics. The ISE index is utilized to give significant penalties to significant errors and lighter penalties to smaller errors. When a system is designed based on this criterion, it tends to exhibit a quick reduction in a large initial error, resulting in a fast and oscillatory response. On the other hand, the ITSE index focuses on initial errors and heavily penalizes errors that occur later in the transient response of the system. In contrast, the IAE index penalizes control errors. Systems designed using the ITAE index display minimal overshoots and damped oscillations. While a large initial error is penalized lightly, errors occurring later in the response are heavily penalized in ITAE, which leads to penalties for extended settling time and control errors. The research study in question employs the IAE performance index.

The following values are assigned for controller parameters optimization:

1. Swarm size = 40.
2. The number of maximum iterations = 100.
3. Search ranges = [0, 100].
4. Simulation time = 20seconds.

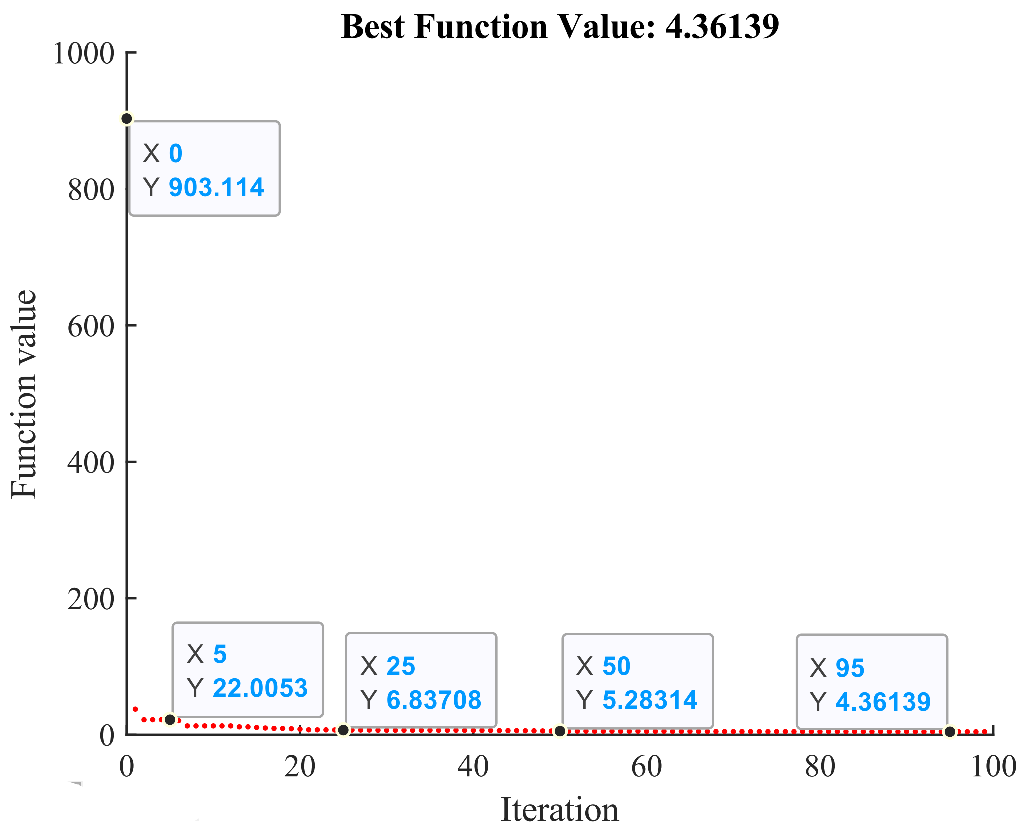


Figure 4.2 Convergence curve for the PSO algorithm

CHAPTER FIVE

5. SIMULATION RESULT AND ANALYSIS

In this chapter, the validity and stabilization of the proposed backstepping sliding mode controller (BSMC) for the Quadrotor dynamics are simulated using MATLAB Software. Different trajectory tracking is done to test the controller performance.

5.1 PARAMETERS FOR SIMULATION

For the simulation of the proposed system, the following parameters shown in Table 5-1 are used which is collected from STI of ASTU and (Derafa1, Madani2, and Benallegue2 2006).

Table 5-1 Physical parameters of the quadrotor

Parameters	Value	Unit
Arm length (l)	0.5	m
Moment inertia of the body in the x-axis (I_{xx})	3.8278×10^{-3}	$Nm/(rad/sec^2)$
Moment inertia of the body in the y-axis (I_{yy})	3.8278×10^{-3}	$Nm/(rad/sec^2)$
Moment inertia of the body in the z-axis (I_{zz})	7.1345×10^{-4}	$Nm/(rad/sec^2)$
Moment inertia of the rotor (J_r)	2.8385×10^{-5}	Kg/m^2
Total mass of the quadrotor (m)	0.5 - 18	Kg
Gravitational acceleration (g)	9.81	m/sec^2
Lift constant (K_T)	2.9842×10^{-5}	$N/(rad/sec)$
Drag constant (K_D)	3.232×10^{-7}	$Nm/(rad/sec)$
Maximum speed of the motors (Ω_{max})	$400\ \pi$	rad/sec

5.2 TUNED PARAMETER SETTING

PSO tuned parameters of the BSMC controller using 0 as a lower bound and 100 as an upper bound are given below in Table 5-2

Table 5-2 PSO tuned parameters of the controller

Parameters	Values	Parameters	Values
c_ϕ	4.6534	c_x	2.5487
c_θ	90.3252	c_y	1.7297
c_ψ	46.0854	c_z	0.6906
k_ϕ	80.9636	k_x	14.7240
k_θ	87.8250	k_y	0.1348
k_ψ	3.7395	k_z	62.2074
h_ϕ	52.8354	h_x	2.0858
h_θ	86.9526	h_y	0
h_ψ	9.6185	h_z	0.7383
μ_ϕ	87.0379	μ_x	0
μ_θ	31.0951	μ_y	3.2361
μ_ψ	86.7513	μ_z	65.7034

5.3 SIMULATION RESULT UNDER NORMAL CONDITIONS

In this section different desired trajectories are given to the quadrotor to follow, without subjecting the quadrotor either to an external disturbance or parametric uncertainties.

5.3.1 Set Point Tracking Performance at m= 1.5kg

Under this tracking problem the quadrotor is commanded to hover at the position of X_{des}=2, Y_{des}=2, Z_{des}=4 and ψ_{des}=0 where the position *X*, *Y* and *Z* are measured in meter and the Euler angle ψ is measured in radians.

The resulting *X*, *Y* and *Z* position trajectories are shown in Figure 5.1 thus as shown from the figure the quadrotor is able to follow the given *X* and *Y* position command with a rise time of 1.0656 and 1.428 seconds respectively and a settling time of 1.9375 and 2.226 seconds respectively with no overshoot. Similarly, the quadrotor is hovered at the commanded height which is 4 meters from the ground with a rise time of 1.3767 seconds and settling time of 1.7643 seconds respectively and with an overshoot of 0.0379 meter. The integral absolute value error in X, Y and Z position trajectories are 1.1025, 1.7096 and 2.9337 respectively over the given simulation time.

The roll and the pitch angle that capable of creating the commanded Y and X positions respectively and the commanded psi angle are shown in the Figure 5.2 as shown from the figure the quadrotor generates the roll and pitch angle that used to create Y and X position perfectly and the given yaw angle command is achieved within 0.2129 seconds. The integral absolute value error in roll, pitch and yaw Euler angles are 0.0207, 0.0220 and 0.0385 respectively over the given simulation time.

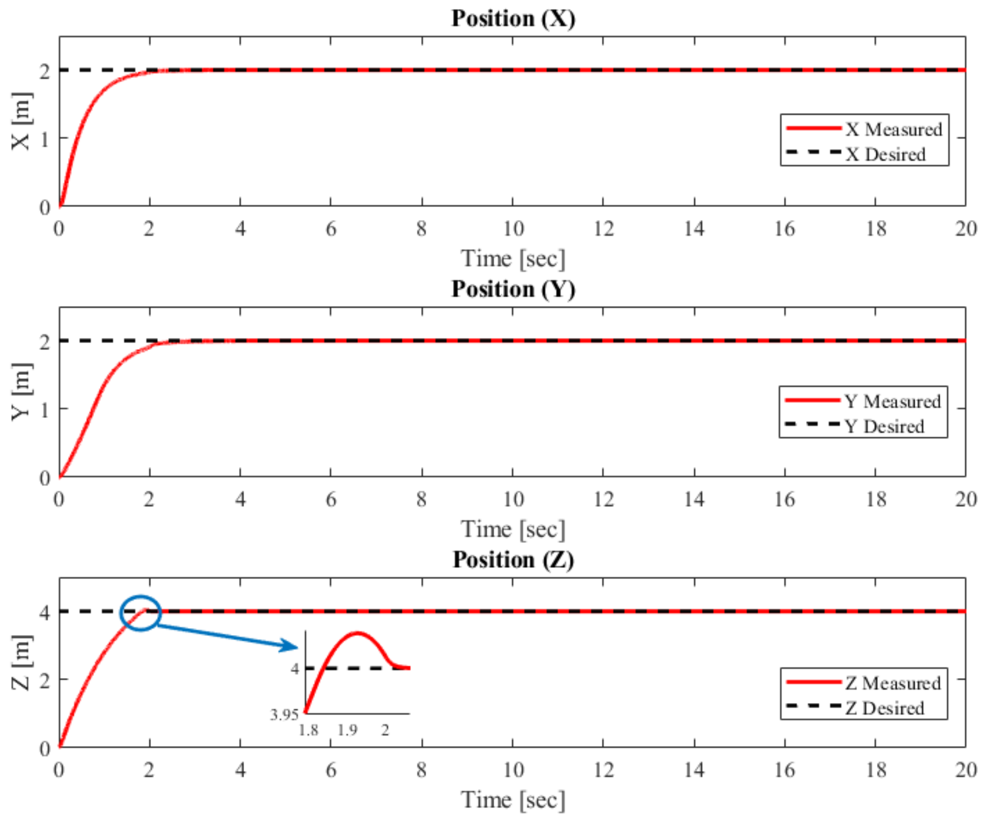


Figure 5.1 Response of translational subsystem for set point tracking

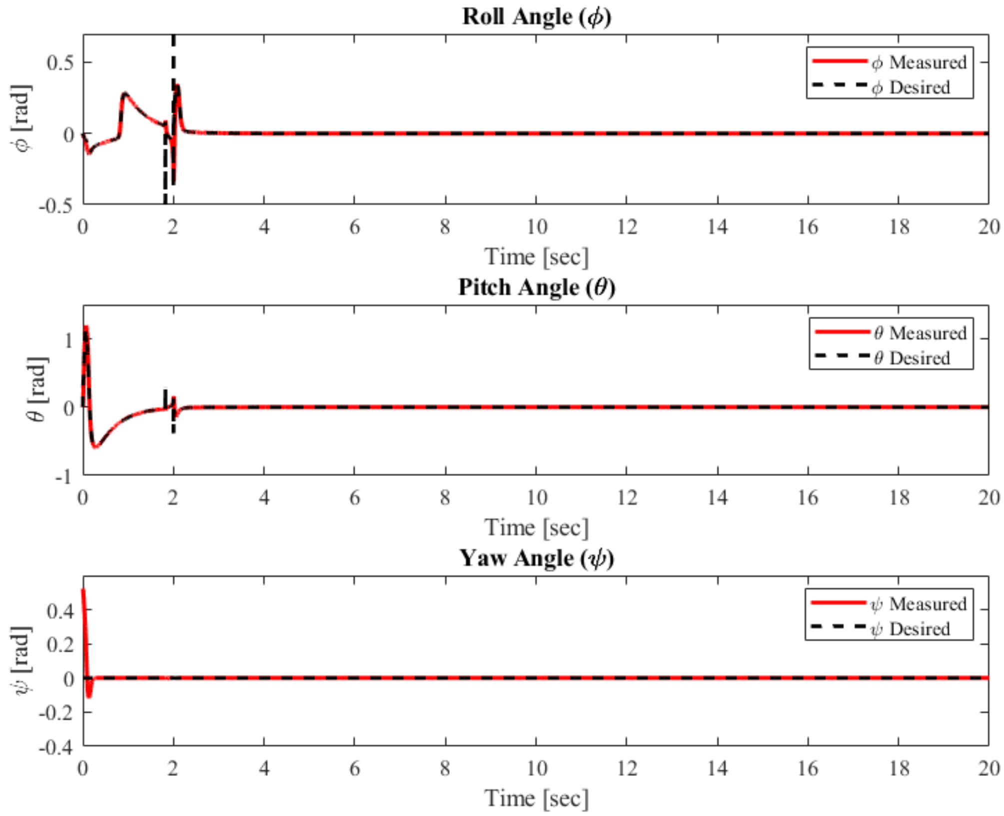


Figure 5.2 Response of rotational subsystem for set point tracking

5.3.2 Circular Trajectory Tracking Performance at m= 1.5kg

Under this tracking problem the quadrotor is commanded to track the trajectory generated by X_{des}=0.5$cos(\pi t/20)$, Y_{des}=0.5$sin(\pi t/20)$ + 0.5, Z_{des}=3 and ψ_{des}=0 which is a circle at the height of 3 meter from the ground.

The resulting *X*, *Y* and *Z* position trajectories are shown in Figure 5.3 thus as shown from the figure the quadrotor is able to follow the given *X* and *Y* position command effectively. The integral absolute value error in *X* and *Y* position trajectories are 0.2785 and 0.3222 respectively over the given simulation time. Similarly, the quadrotor is hovered at the commanded height which is 3 meters from the ground with a rise time of 1.1855 seconds and settling time of 1.5034 seconds respectively and with an overshoot of 0.0380 meter. The integral absolute value error in altitude *Z* is 1.9418.

The roll and the pitch angle that capable of creating the commanded *Y* and *X* positions respectively and the commanded psi angle are shown in the Figure 5.4 as shown from the figure the quadrotor generates the roll and pitch angle that used to create Y and X position perfectly and the given yaw angle command is achieved within 1.9095 seconds. The integral absolute value error in roll, pitch and yaw Euler angles are 0.0078, 0.0052 and 0.0386 respectively over the given simulation time.

In Figure 5.5 the resulting trajectory with its desired trajectory is plotted in 3D view.

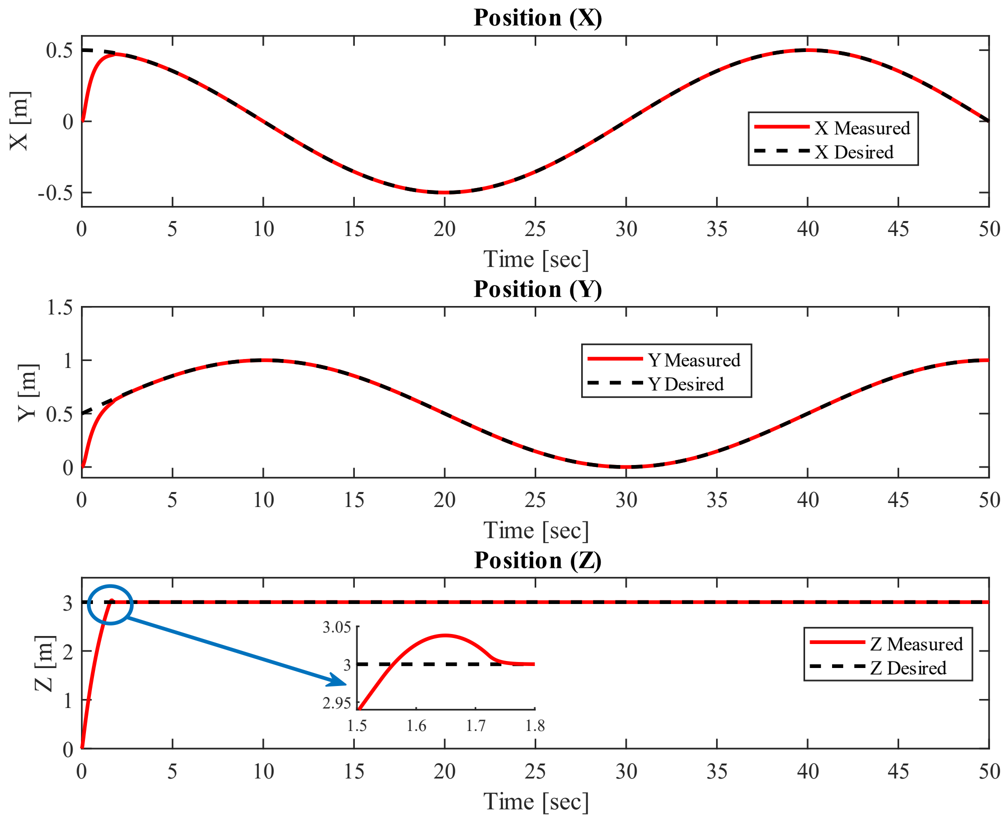


Figure 5.3 Response of translational subsystem for circular trajectory tracking

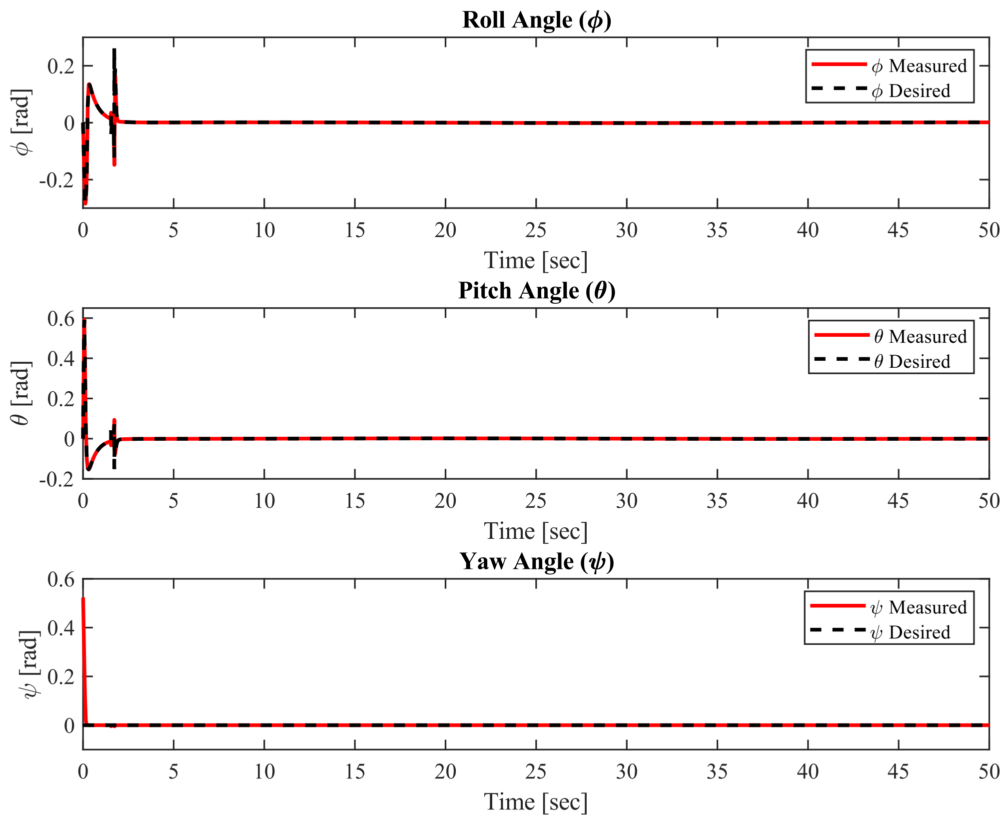


Figure 5.4 Response of rotational subsystem for circular trajectory tracking

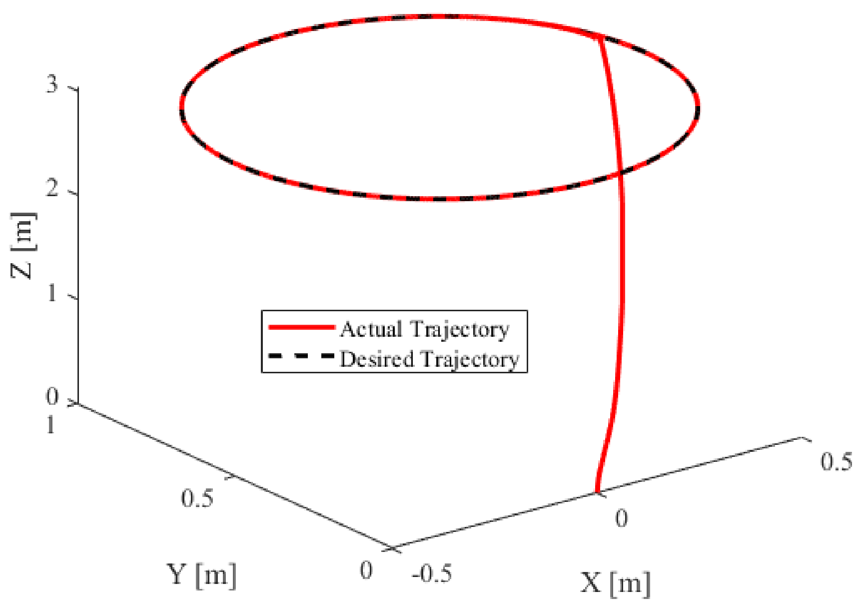


Figure 5.5 3D view of circular trajectory tracking

5.3.3 Infinite Eight Shape Trajectory Tracking at m= 1.5kg

Under this tracking problem the quadrotor is commanded to track the trajectory generated by $X_{des}=cos(t)$, $Y_{des}=0.5sin(2t)$, $Z_{des}=2$ and $\psi_{des}=0$ which is an infinite eight shape at the height of 2 meter from the ground.

The resulting X, Y and Z position trajectories are shown in Figure 5.6 thus as shown from the figure the quadrotor is able to follow the given X and Y position command effectively. The integral absolute value error in X and Y position trajectories are 0.5381 and 0.1071 respectively over the given simulation time. Similarly, the quadrotor is hovered at the commanded height which is 2 meters from the ground with a rise time of 0.9326 seconds and settling time of 1.1721 seconds respectively and with an overshoot of 0.0379 meter. The integral absolute value error in altitude Z is 1.0575.

The roll and the pitch angle that capable of creating the commanded Y and X positions respectively and the commanded psi angle are shown in the Figure 5.7 as shown from the figure the quadrotor generates the roll and pitch angle that used to create Y and X position perfectly and the given yaw angle command is achieved within 1.5036 seconds. The integral absolute value error in roll, pitch and yaw Euler angles are 0.0419, 0.0394 and 0.0539 respectively over the given simulation time.

In Figure 5.8 the resulting trajectory with its desired trajectory is plotted in 3D view.

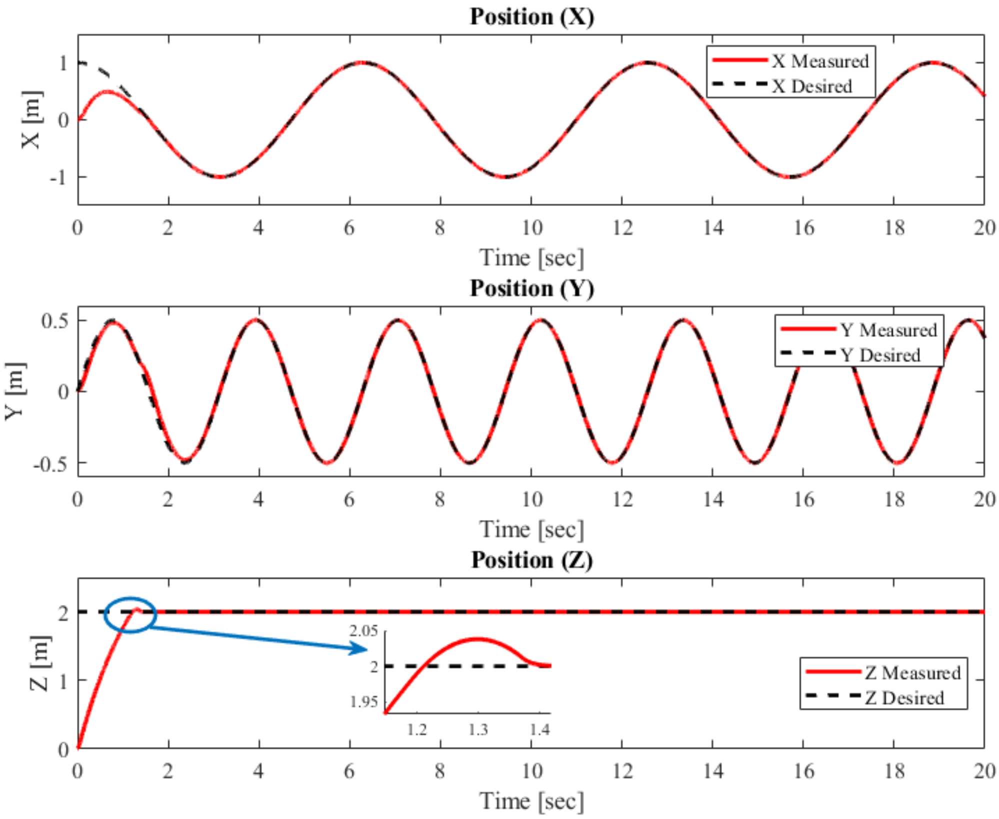


Figure 5.6 Response of translational subsystem for infinite eight shape trajectory tracking

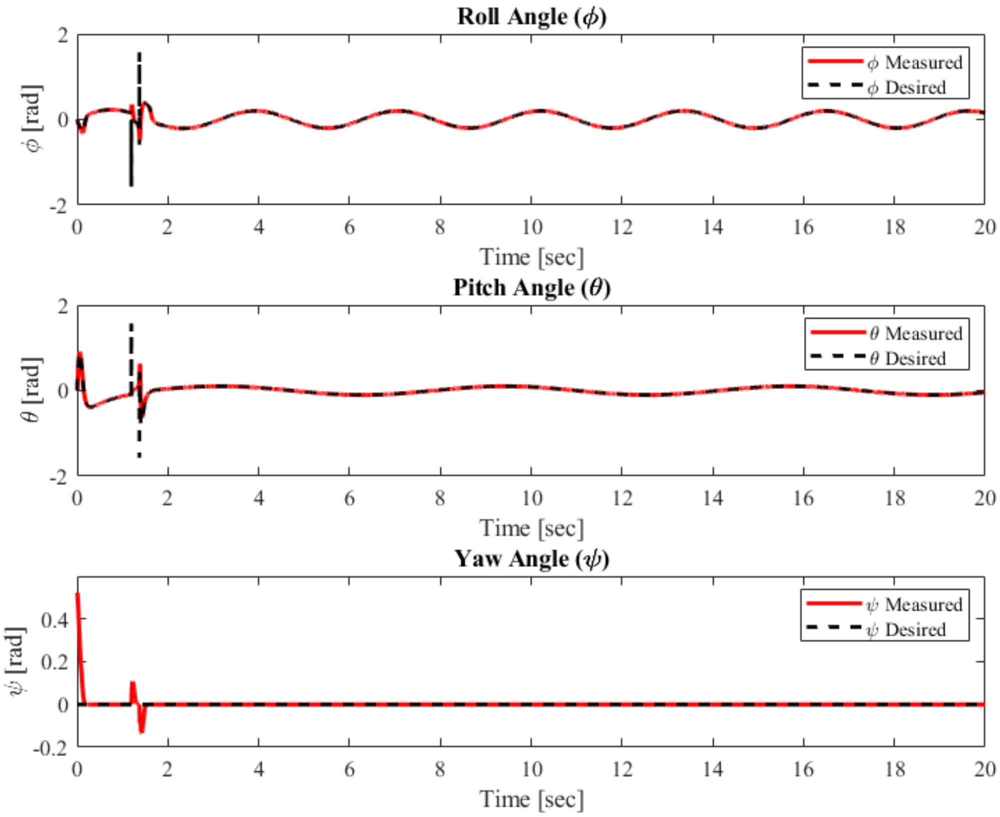


Figure 5.7 Response of rotational subsystem for infinite eight shape trajectory tracking

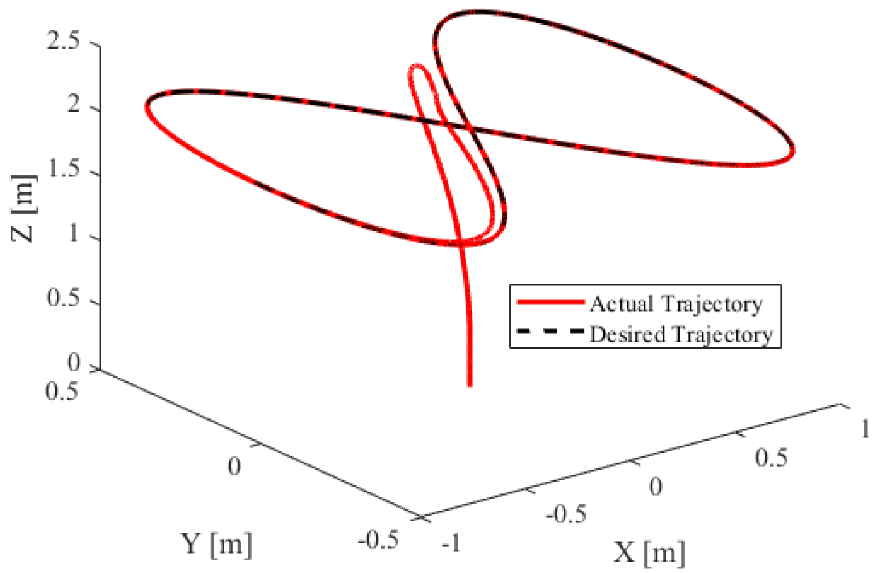


Figure 5.8 3*D* view of the infinite eight shape trajectory

5.3.4 Complex Helix Trajectory Tracking at m= 1.5kg

Under this tracking problem the quadrotor is commanded to track the trajectory generated by X_{des}=cos (0.05t) – cos^3(0.05t), Y_{des}= sin (0.05t) – sin^3(0.05t), Z_{des}=0.3t and ψ_{des}=0 which is a complex helix shape.

The resulting X, Y and Z position trajectories are shown in Figure 5.9 thus as shown from the figure the quadrotor is able to follow the given X and Y position command effectively. The integral absolute value error in X and Y position trajectories are 0.0054 and 0.0061respectively over the given simulation time. Similarly, the quadrotor follows the given ramp height with an integral absolute value error of 0.06.

The roll and the pitch angle that capable of creating the commanded Y and X positions respectively and the commanded psi angle are shown in the Figure 5.10 as shown from the figure the quadrotor generates the roll and pitch angle that used to create Y and X position perfectly and the given yaw angle command is achieved within 0.1691 seconds. The integral absolute value error in roll, pitch and yaw Euler angles are 0.0027, 0.0001 and 0.0391 respectively over the given simulation time.

In Figure 5.11 and Figure 5.12 the resulting trajectory with its desired trajectory is plotted in 3D and 2D view respectively.

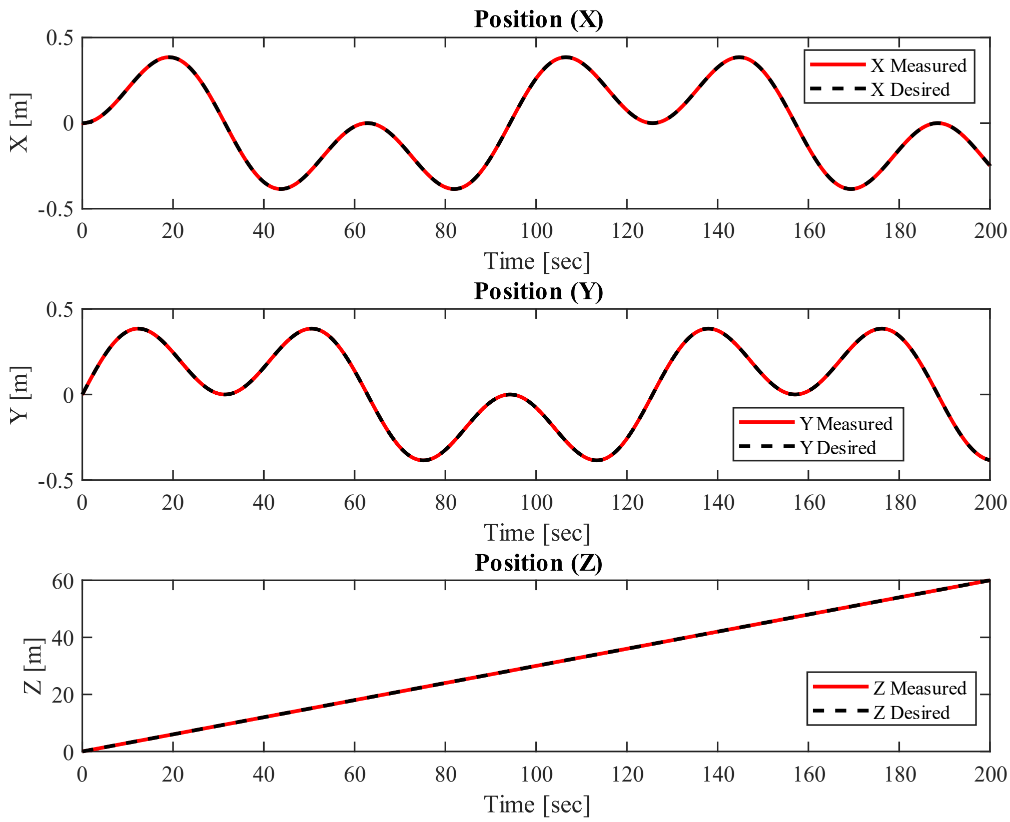


Figure 5.9 Response of translational subsystem for complex helix trajectory tracking

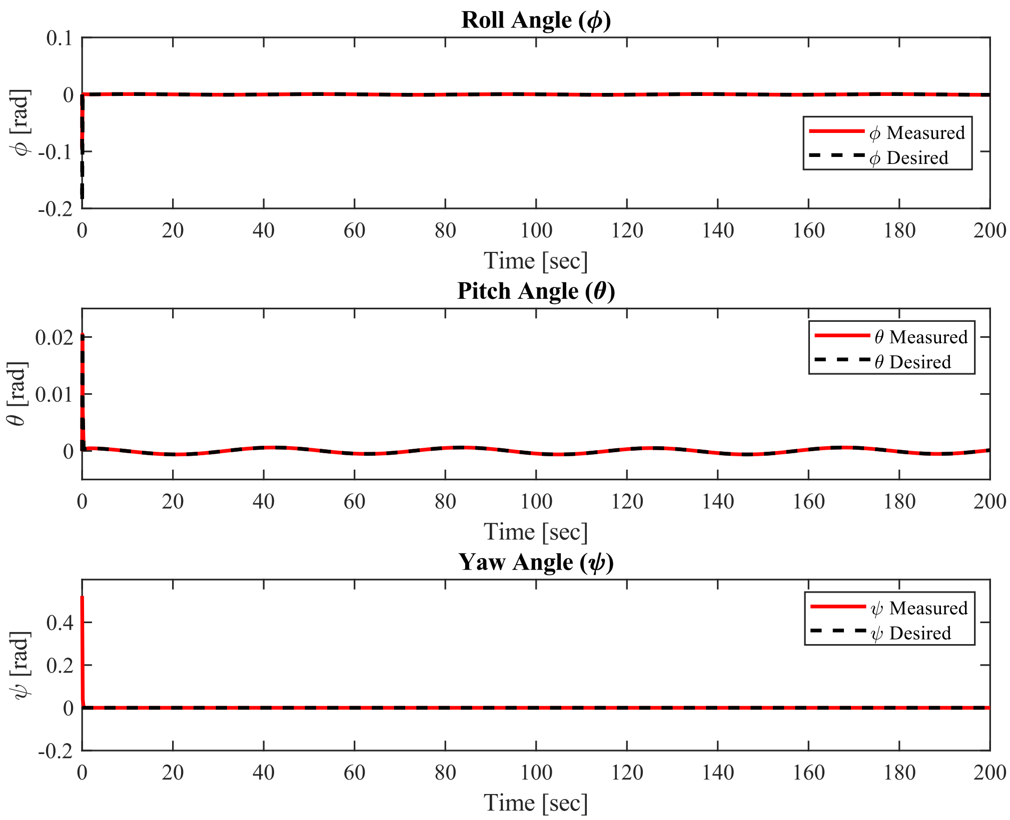


Figure 5.10 Response of rotational subsystem for complex helix trajectory tracking

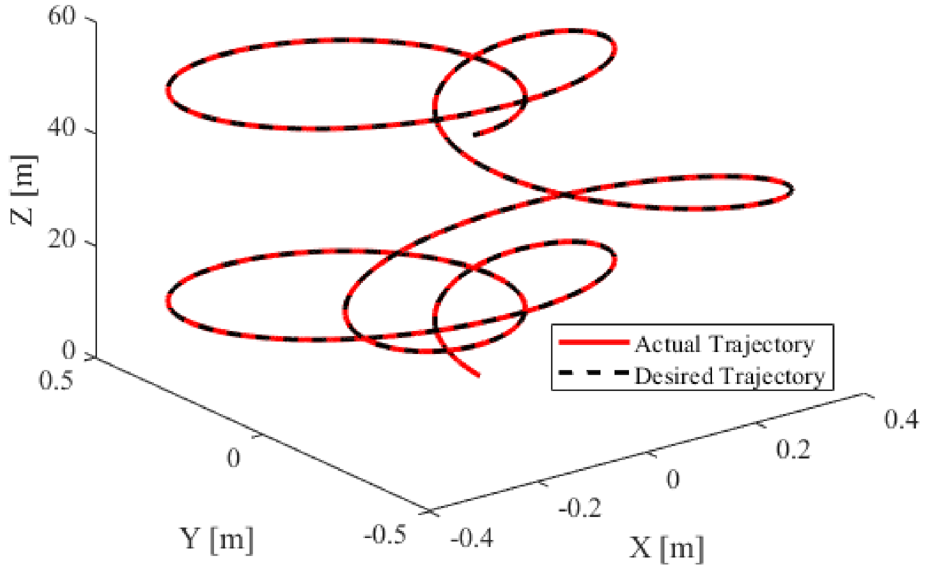


Figure 5.11 3*D* view of complex helix trajectory

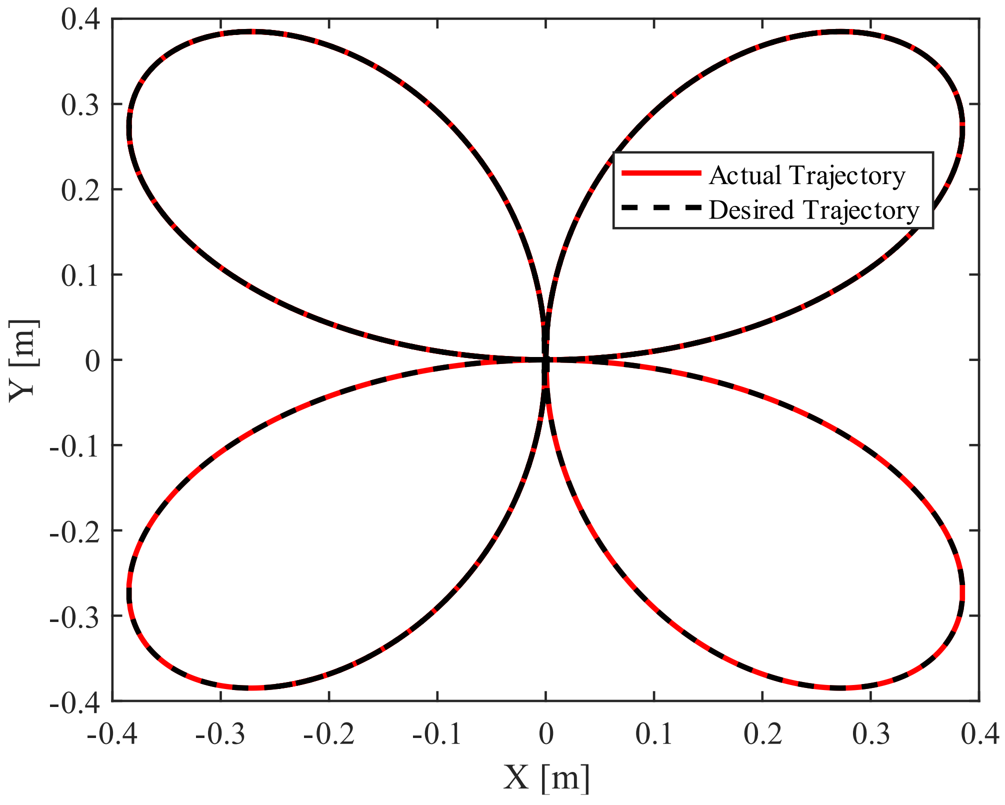


Figure 5.12 2*D* view of complex helix trajectory tracking on *X* and *Y* plane

5.3.5 Control Efforts

From Figure 5.13 to Figure 5.16 the control efforts that are needed to create the desired trajectories are given which shows a very small chattering which prevents the quadrotor from vibrating and also from heating.

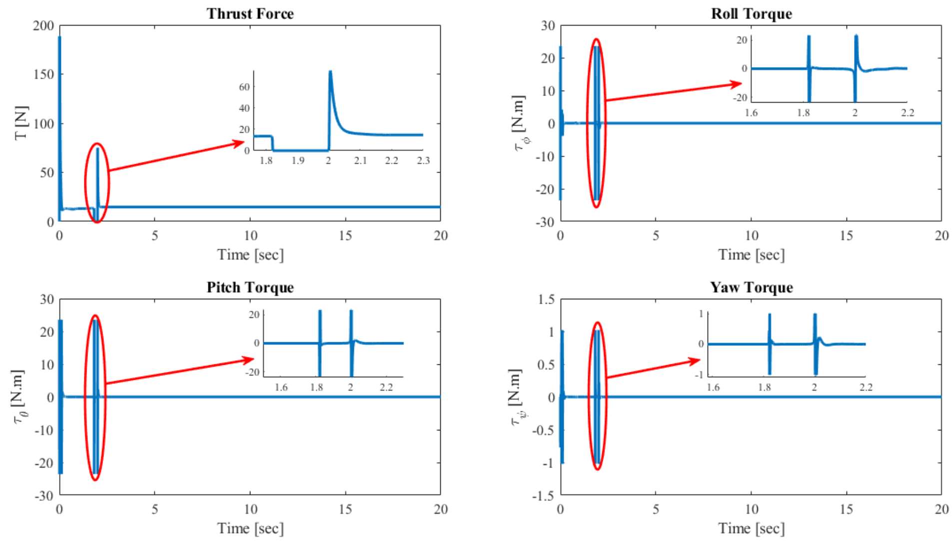


Figure 5.13 Control inputs to create set point trajectory tracking

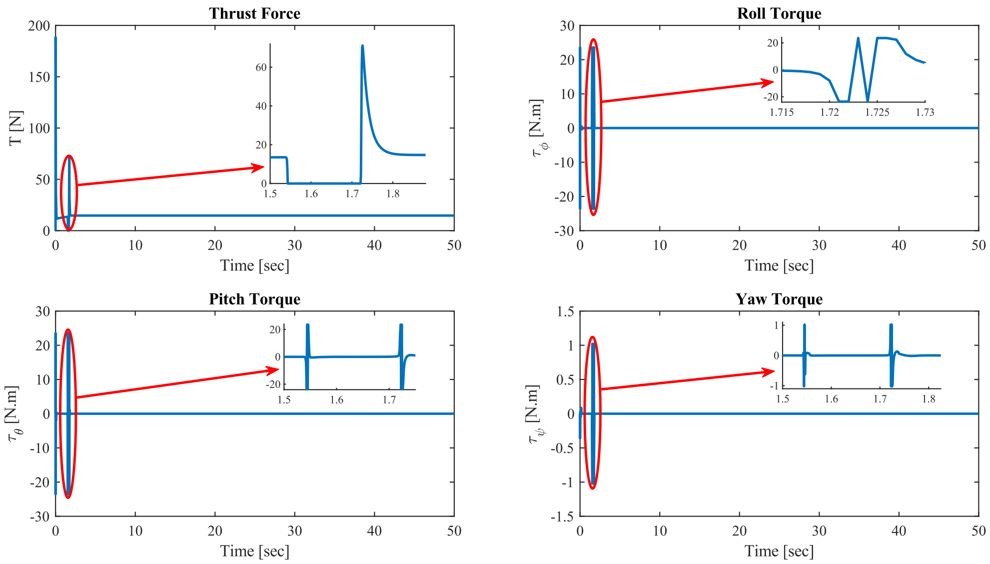


Figure 5.14 Control inputs to create circular trajectory tracking

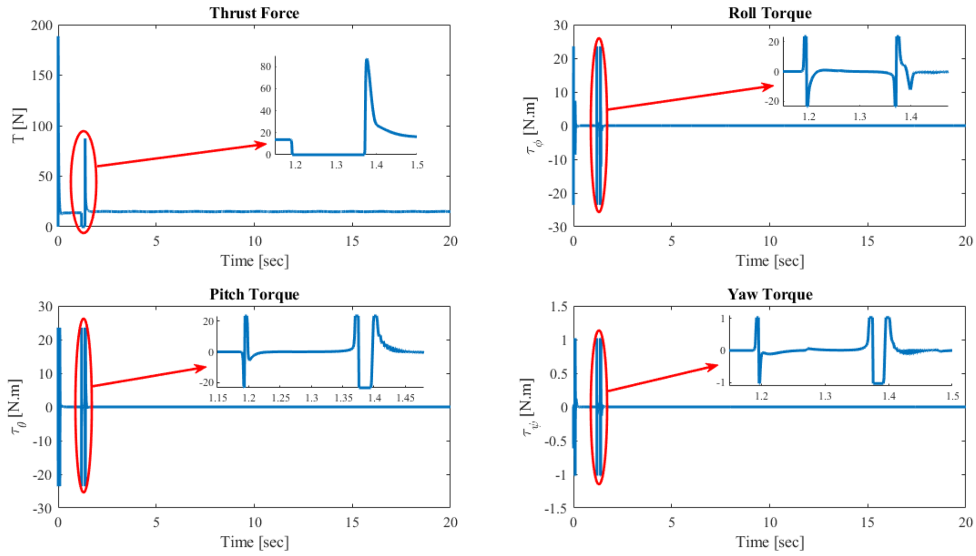


Figure 5.15 Control inputs to create an infinite eight shape trajectory tracking

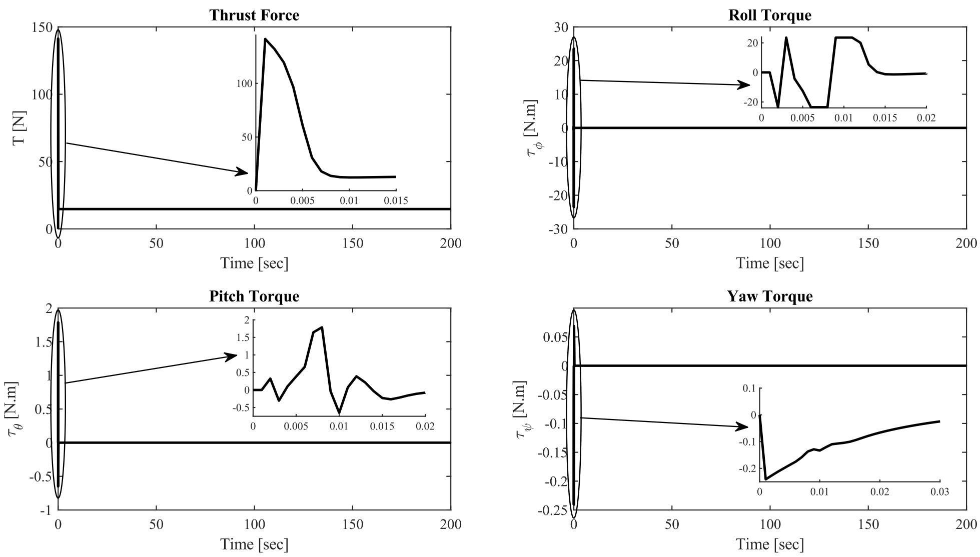


Figure 5.16 Control inputs to create a complex helix trajectory tracking

5.3.6 Output Speed of the Motors

Figures from 5.17 to 5.20 shows the desired speed founded by calculation and the real output speed of the actuators (motors) together. Thus, as shown from the figures the real output speed from the motors is very smooth which prevents the quadrotor from vibrating.

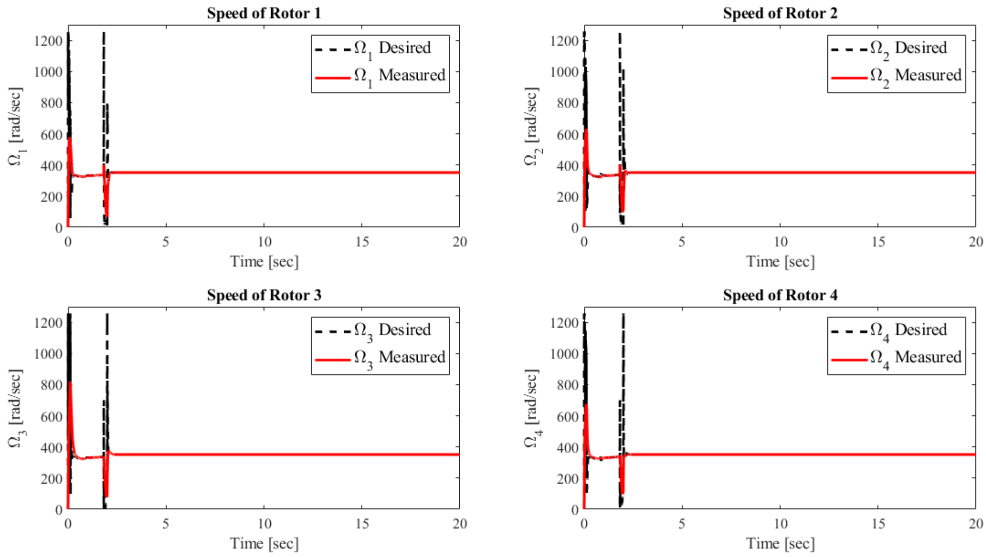


Figure 5.17 Speed of the motors for set point trajectory tracking

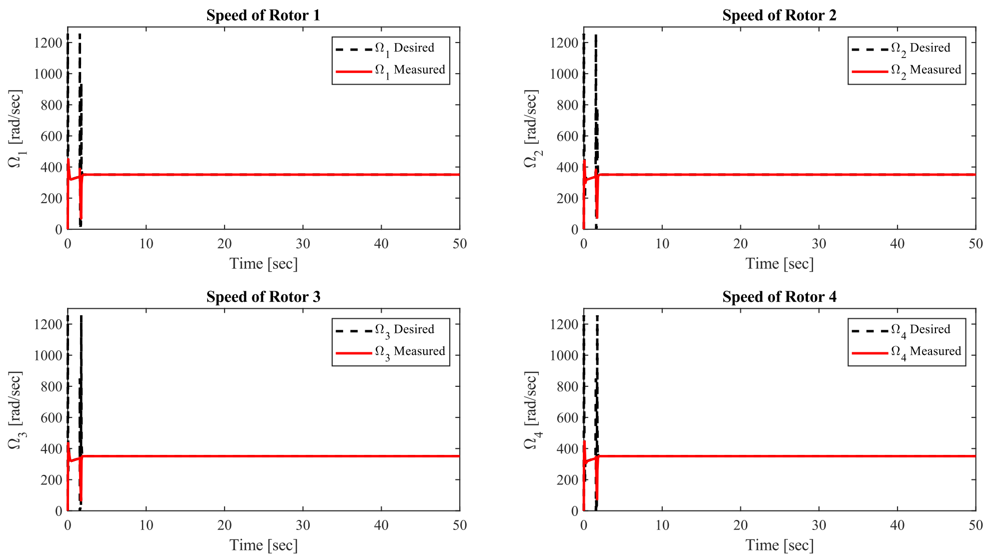


Figure 5.18 Speed of the motors for circular trajectory tracking

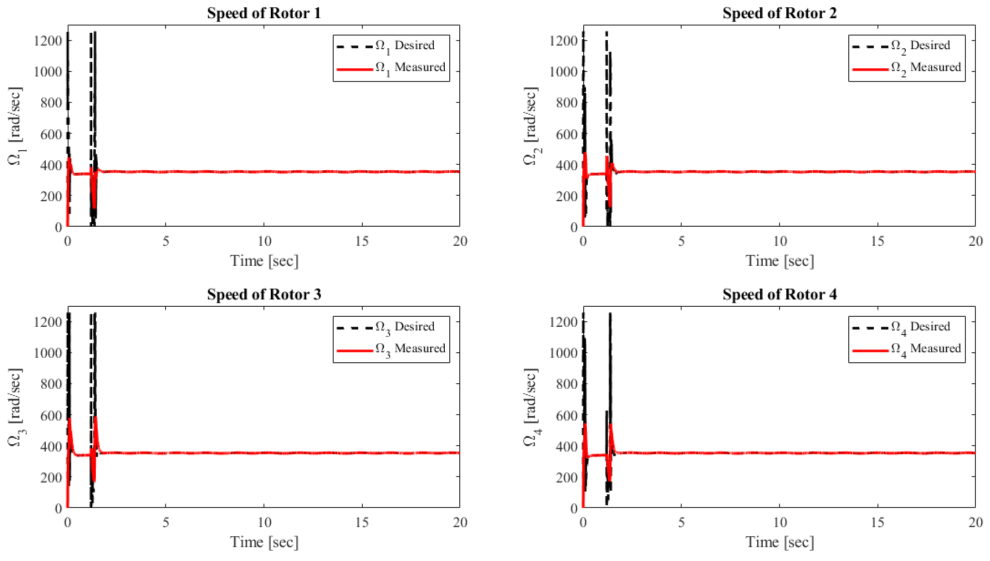


Figure 5.19 Speed of the motors for infinite eight shape trajectory tracking

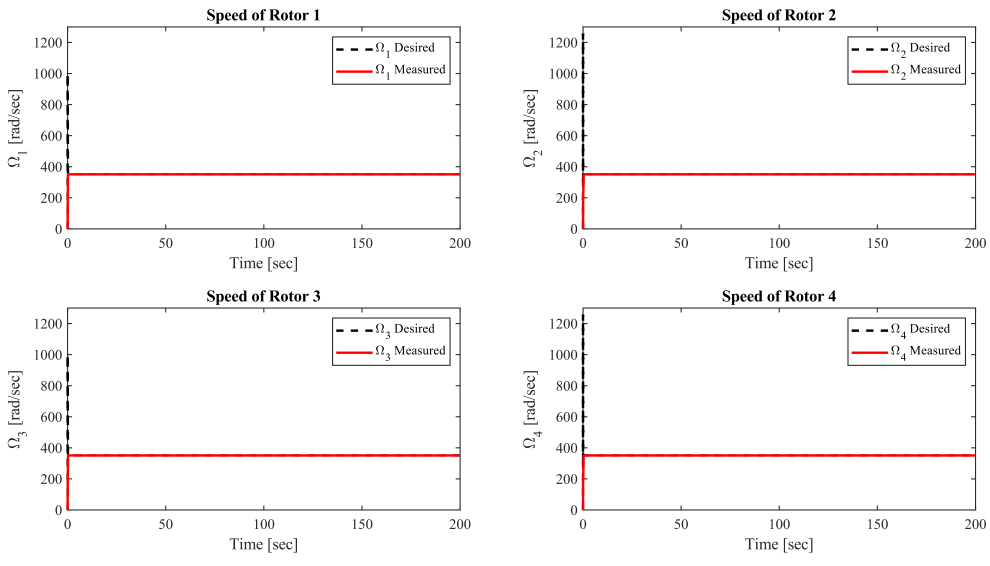


Figure 5.20 Speed of the motors for complex helix trajectory tracking

5.4 Result when the Quadrotor is Subjected to Time Varying External Disturbance and Parametric Uncertainties

In this section, the robustness and ability of the controller to track the given reference trajectory in the presence of parameter uncertainties and external disturbances are evaluated and analyzed using MATLAB.

The parametric uncertainties and external disturbances applied to the system are as follows

1. The parameter uncertainty is represented by 0.5*kg*-18*kg* mass variation under no external disturbances and 0.5*kg*-12*kg* when the system is subjected to a time varying external disturbance.

2. A time varying disturbance with a maximum magnitude varies from -2 to +2 in position *X*, from -3 to +3 in position *Y* and from -5 to +5 in *Z* direction is added to the system as a wind disturbance (Liu et al. 2022). By assuming the quadrotor system is under the pressure of time varying wind given in Figure 5.21.

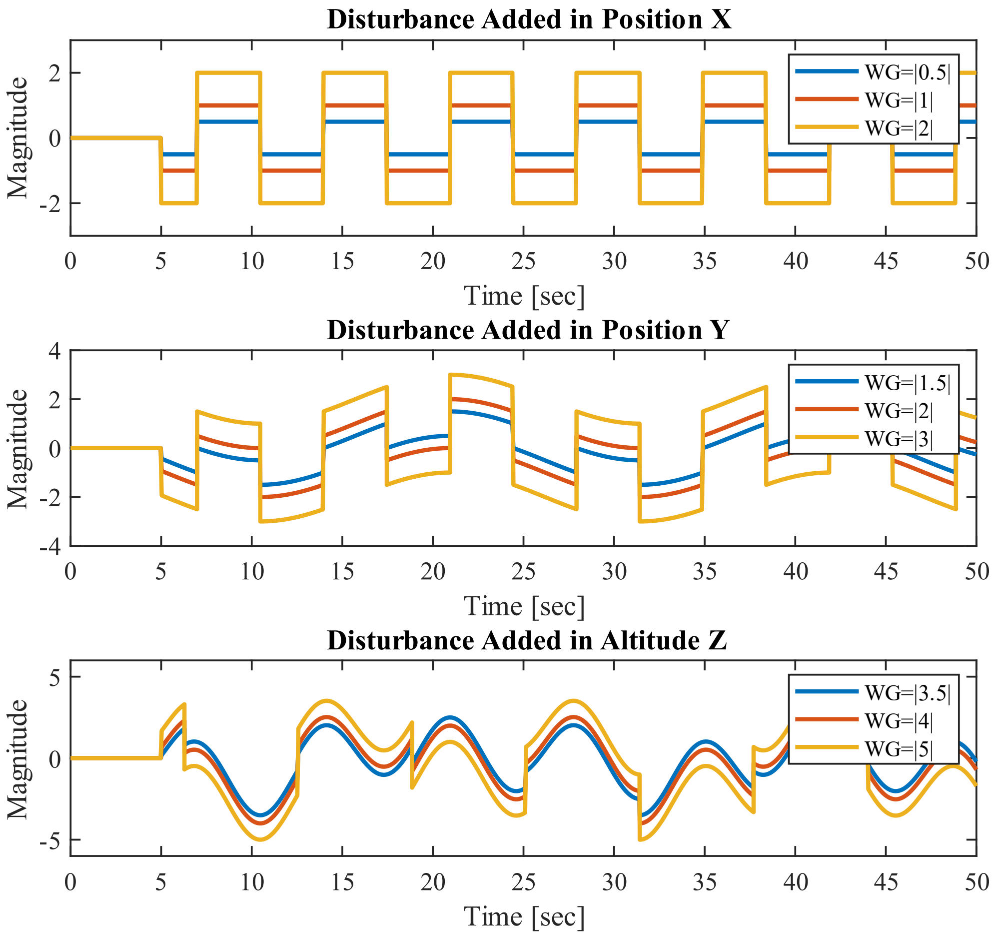


Figure 5.21 Disturbance added to the system as a wind gust

5.4.1 Circular Trajectory Tracking Performance for mass Varies from 0.5kg-18kg

The parametric uncertainty of the quadrotor system is mass variation and it is varied initially not during flight time. This means the mass of the quadrotor is changed before the quadrotor flies and all the signals are arranged for the changed mass of the quadrotor. The controller is designed once and applied to the system. The quadrotor does not change its mass during flights, the mass is changed initially. And this BSMC is used for different masses (0.5*kg*-18*kg*) of the quadrotor.

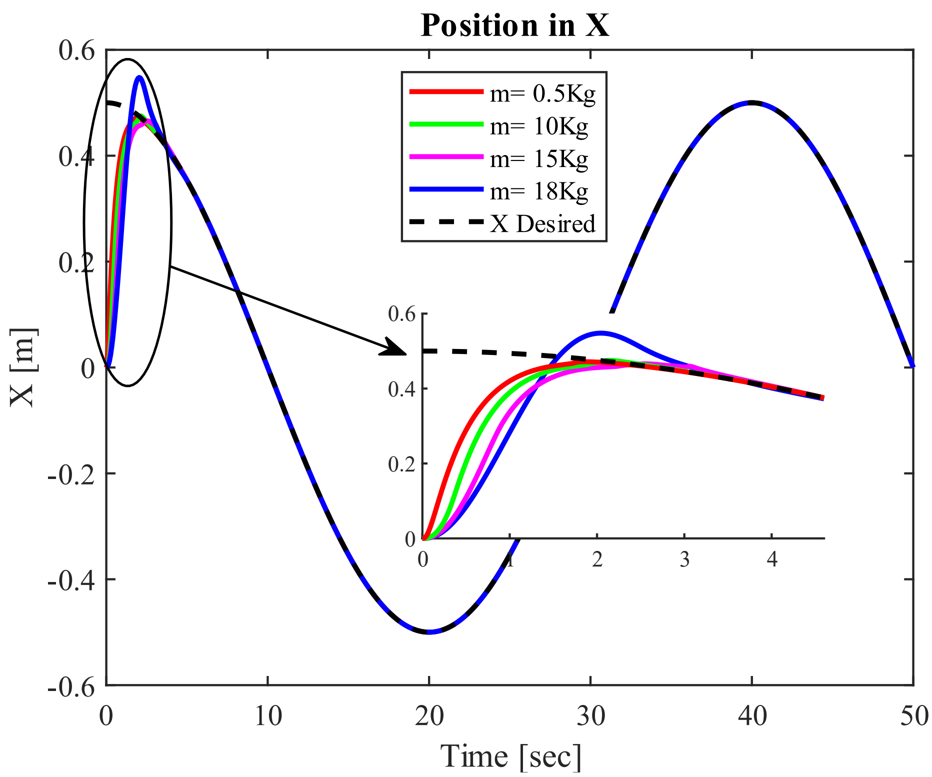


Figure 5.22 Response in position X for different masses

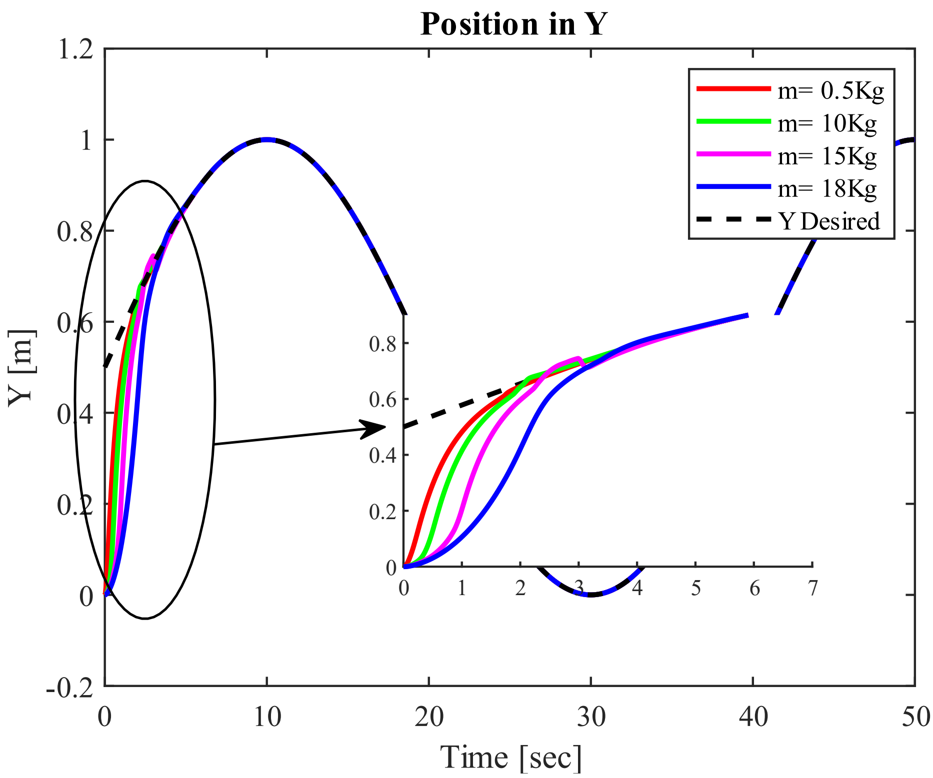


Figure 5.23 Response in position Y for different masses

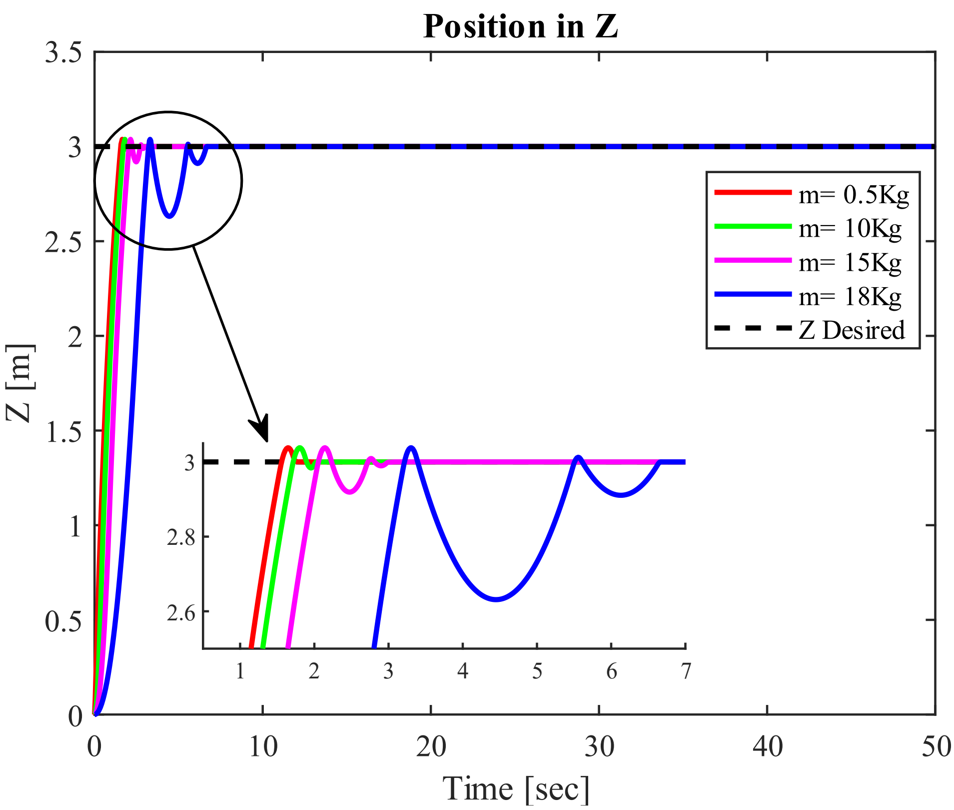


Figure 5.24 Altitude response of Z for different masses

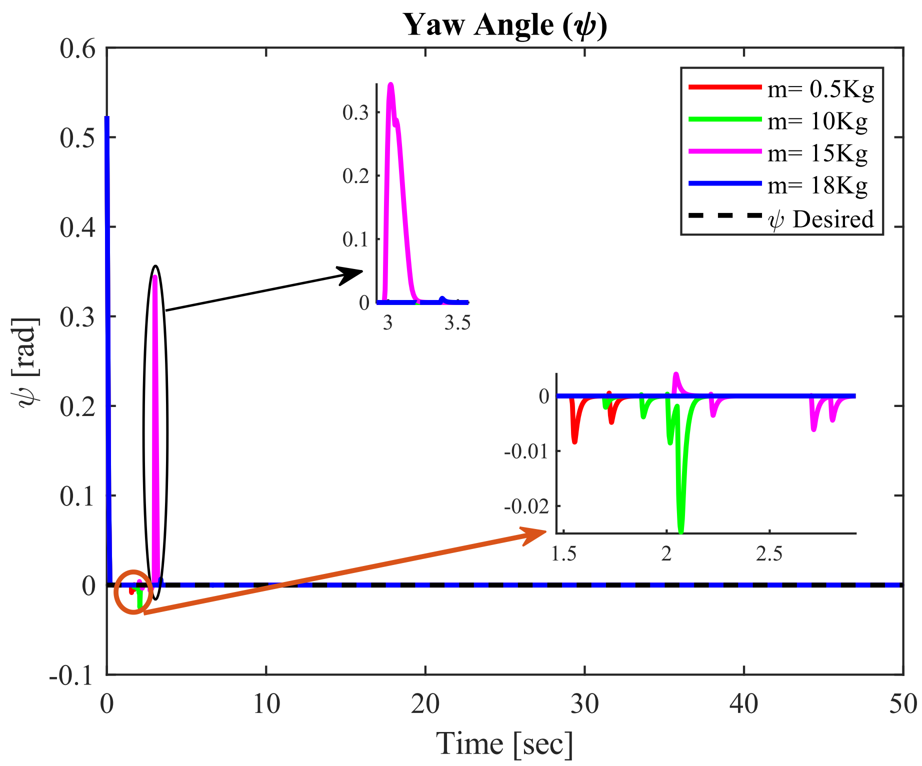


Figure 5.25 Response of Euler angle yaw for different masses

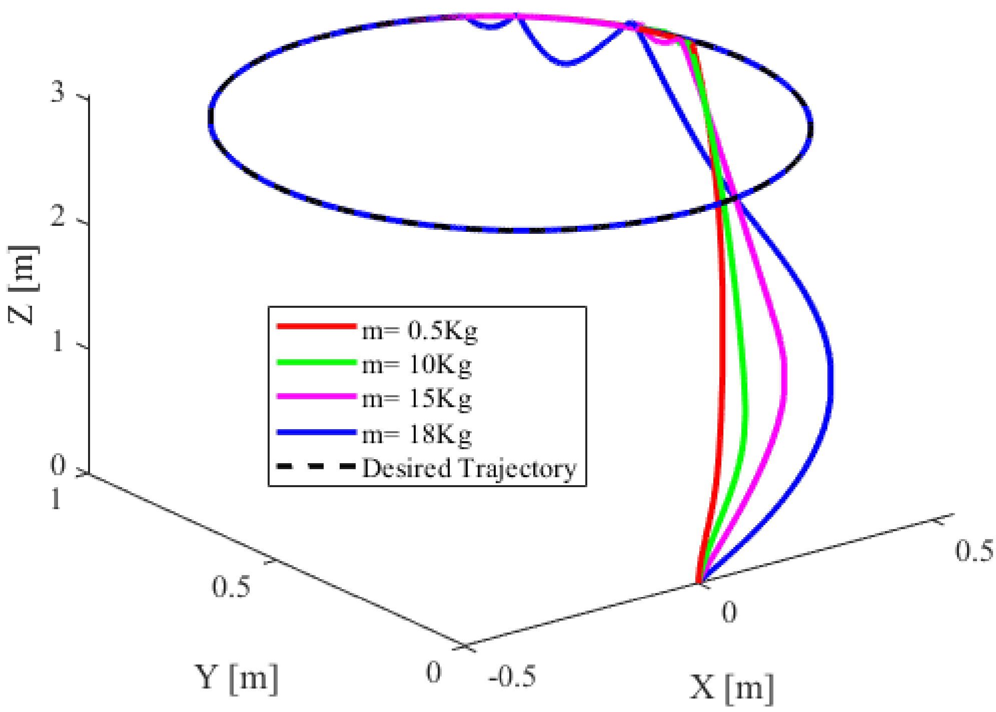


Figure 5.26 3*D* view of circular trajectory tracking for different masses

Responses of the quadrotor system for a varying load under normal conditions are shown in the figures from 5.22 to 5.26 thus as shown from the figures the designed controller can handle a mass varies from 0.5*kg* to 18*kg* under normal conditions with a total integral absolute error of 2.5846, 3.2951, 4.6648 and 8.2977 for *m*=0.5*kg*, *m*=10*kg*, *m*=15*kg* and m=18*kg* respectively. The IAE of the above results are given in Table 5-3 below:

Table 5-3 IAE of circular trajectory tracking for different masses of the quadrotor

Mass (*m*)	IAE					
	X	Y	Z	ϕ	θ	ψ
m=0.5kg	0.2782	0.3216	1.9333	0.0076	0.0052	0.0387
m=10*kg*	0.3560	0.4588	2.3905	0.0357	0.0140	0.0401
m=15*kg*	0.4511	0.6938	3.3416	0.0802	0.0191	0.0789
m=18*kg*	0.5301	0.9982	6.7203	0.0062	0.0040	0.0390

5.4.2 Circular Trajectory Tracking Performance Under Time Varying External Disturbance having Different Magnitudes

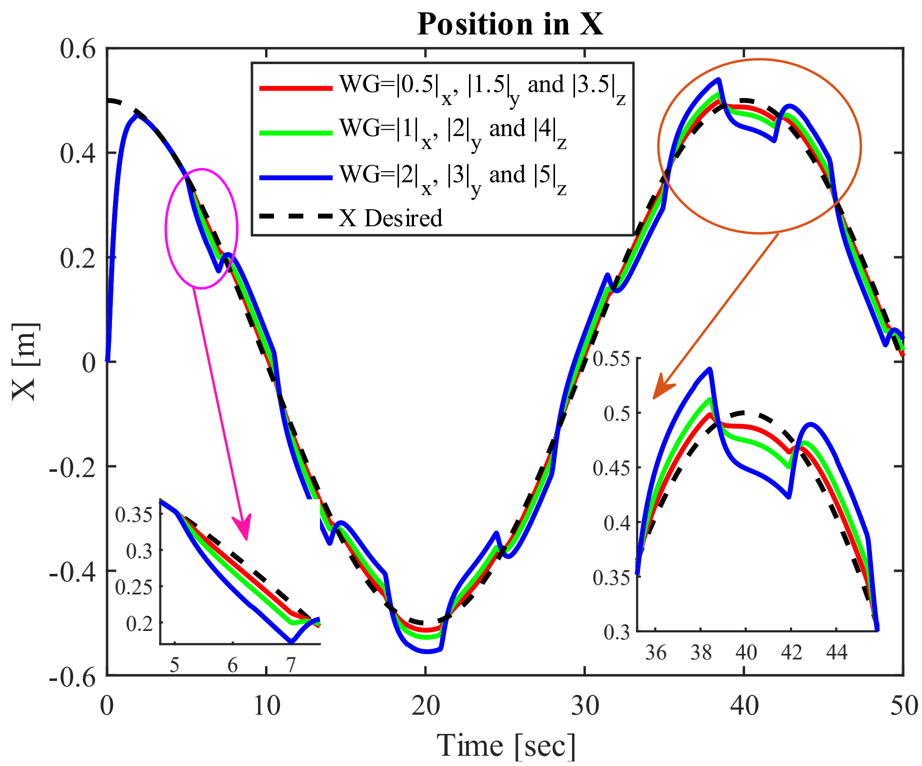


Figure 5.27 Response of position X for disturbances having different magnitudes

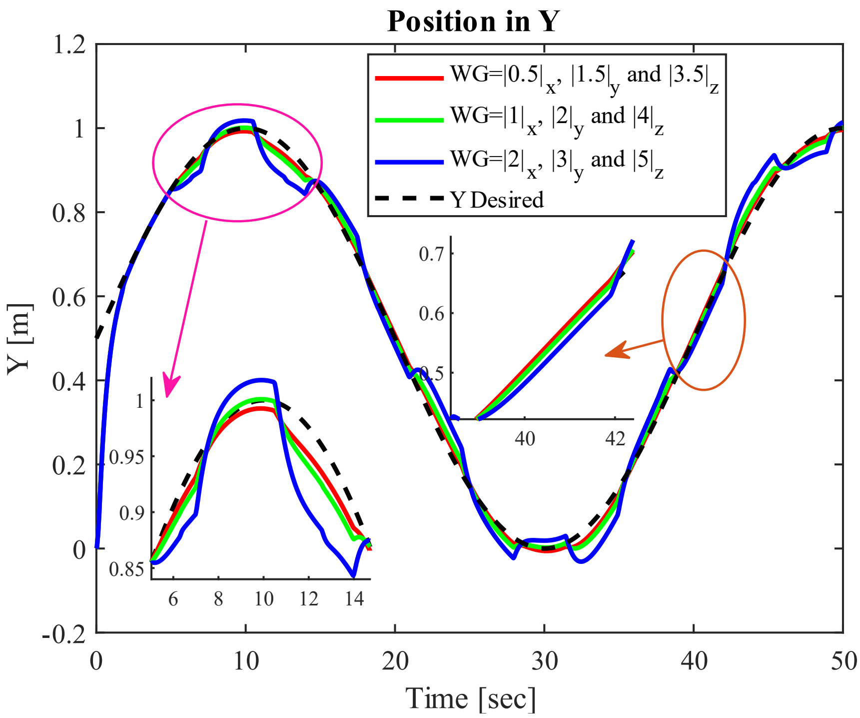


Figure 5.28 Response of position Y for disturbances having different magnitudes

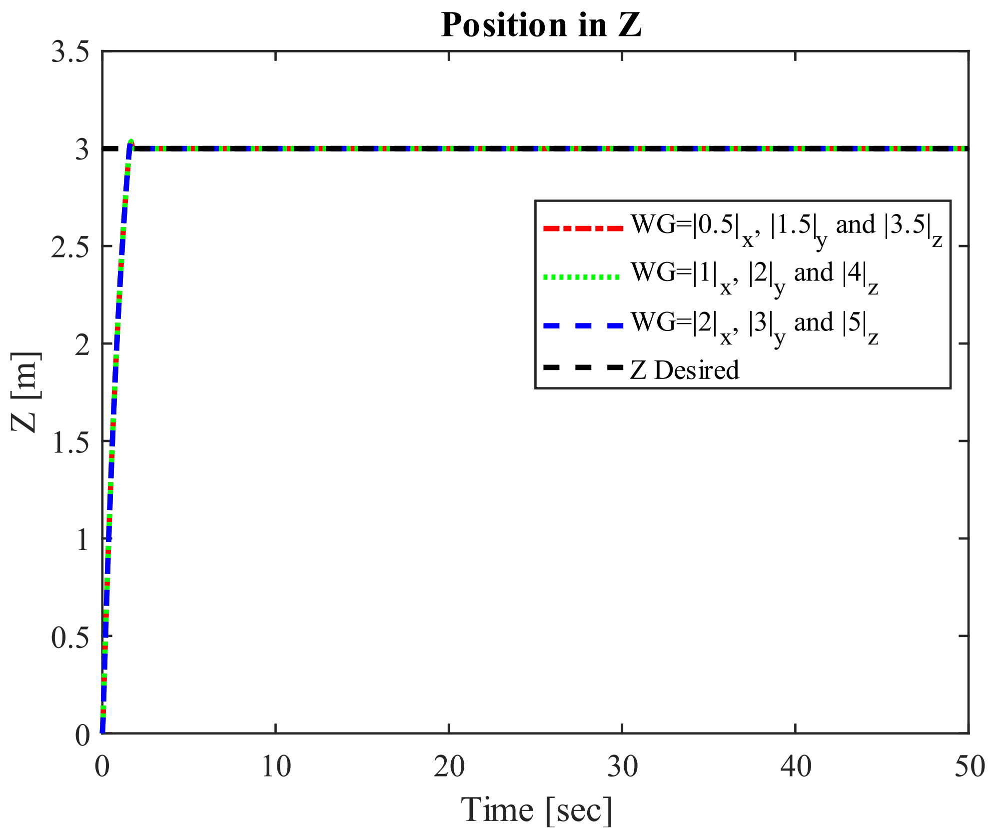


Figure 5.29 Response of Altitude *Z* for disturbances having different magnitudes

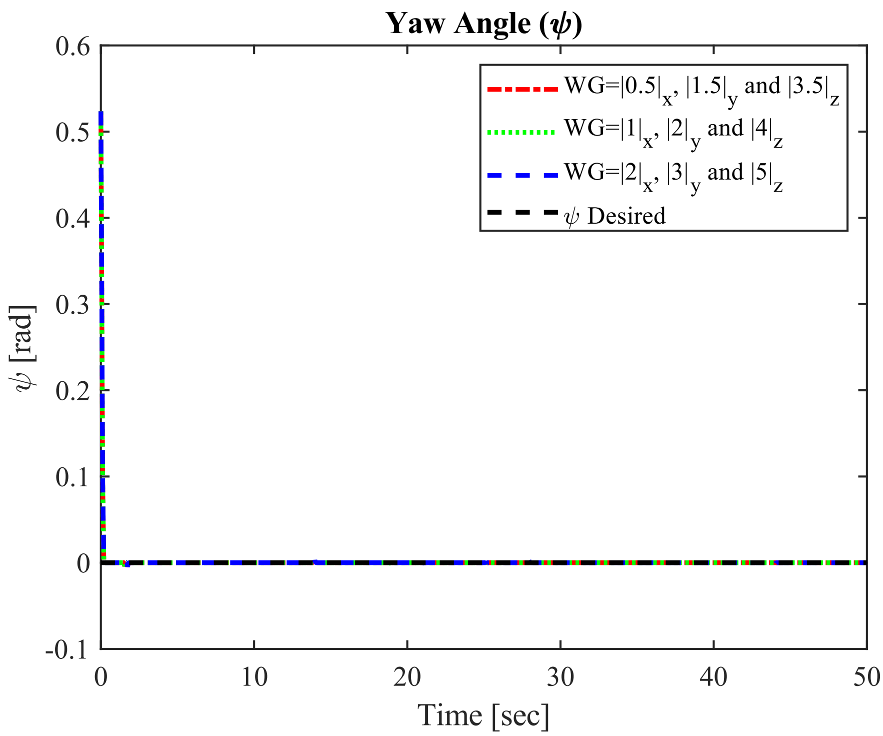


Figure 5.30 Response of yaw angle for disturbances having different magnitudes

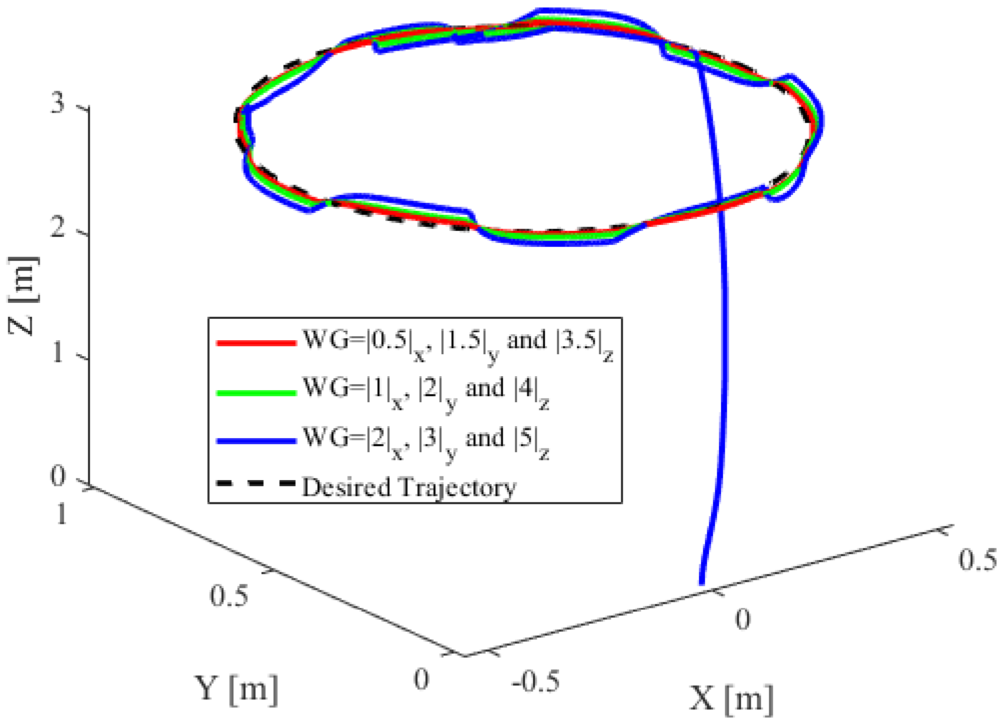


Figure 5.31 3*D* view of circular trajectory tracking under time varying external disturbance having different magnitudes

Responses of the quadrotor system when it is subjected to a time varying disturbances of Figure 5.21 are shown in figures from 5.27 to 5.31 thus as shown from the figures the designed controller can handle a time varying disturbance varies from -2 to +2 in *X*, -3 to +3 in *Y* and -5 to +5 in *Z* direction. IAE of the six outputs over the given simulation time of 50*secs* are given in Table 5-4.

Table 5-4 IAE of circular trajectory tracking when the quadrotor is subjected to a wind gust having different magnitudes

Lower and Upper bound of applied wind gust (WG)	IAE					
	X	Y	Z	ϕ	θ	ψ
$-0.5<\|WG_x\|<0.5$ $-1.5<\|WG_y\|<1.5$ $-3.5<\|WG_z\|<3.5$	0.7737	0.8445	1.9434	0.0113	0.0070	0.0386
$-1<\|WG_x\|<1$ $-2<\|WG_y\|<2$ $-4<\|WG_z\|<4$	1.2711	1.0141	1.9435	0.0198	0.0121	0.0387
$-2<\|WG_x\|<2$ $-3<\|WG_y\|<3$ $-5<\|WG_z\|<5$	2.2655	1.9306	1.9443	0.0502	0.0283	0.0390

5.4.3 Circular Trajectory Tracking Performance Under Time Varying External Disturbance and Mass Varies from 0.5*kg* to 12*kg*

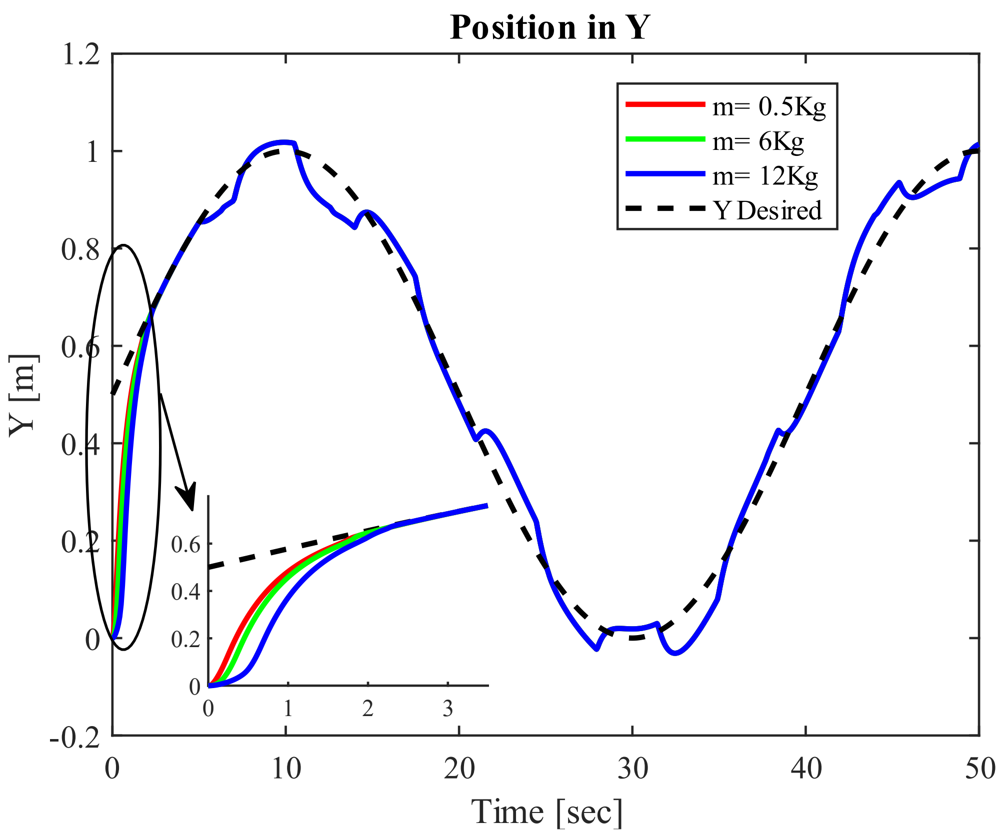


Figure 5.32 Response of position *X* under time varying external disturbance

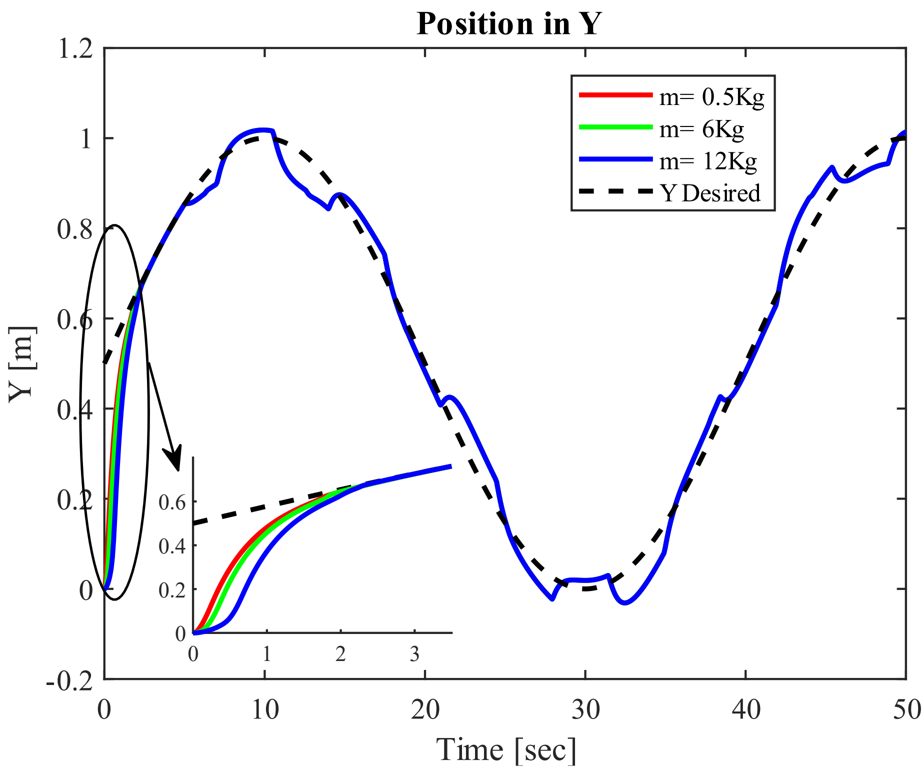


Figure 5.33 Response of position Y under time varying external disturbance

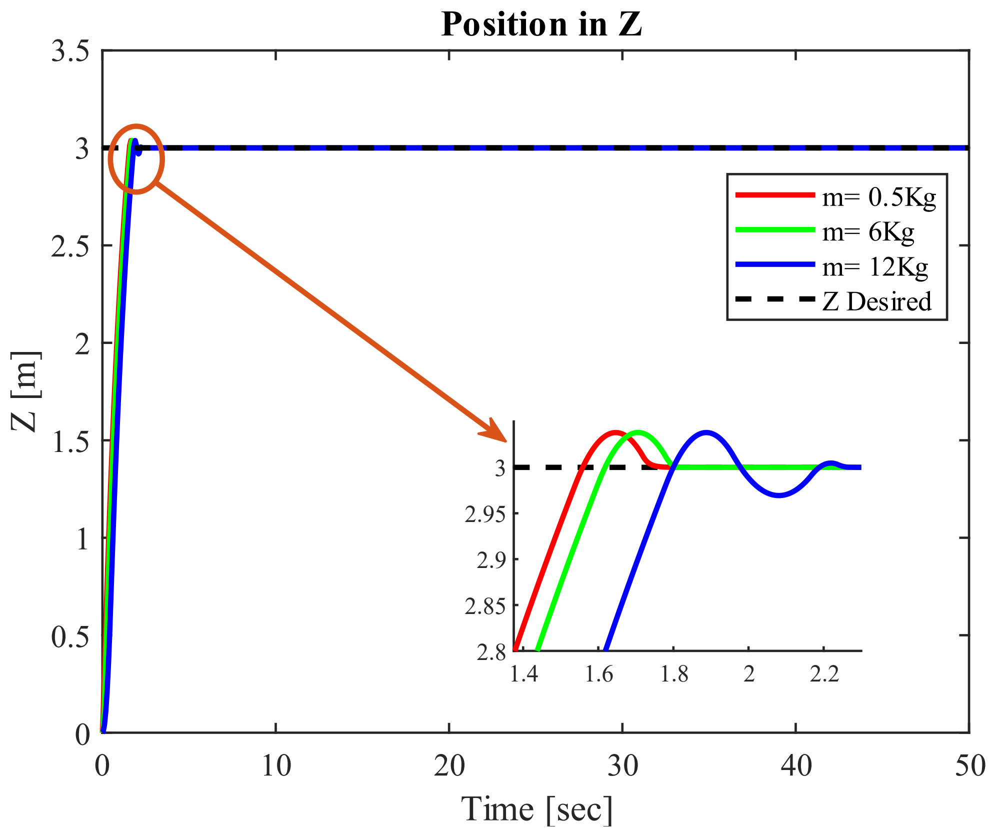


Figure 5.34 Response of height Z under time varying external disturbance

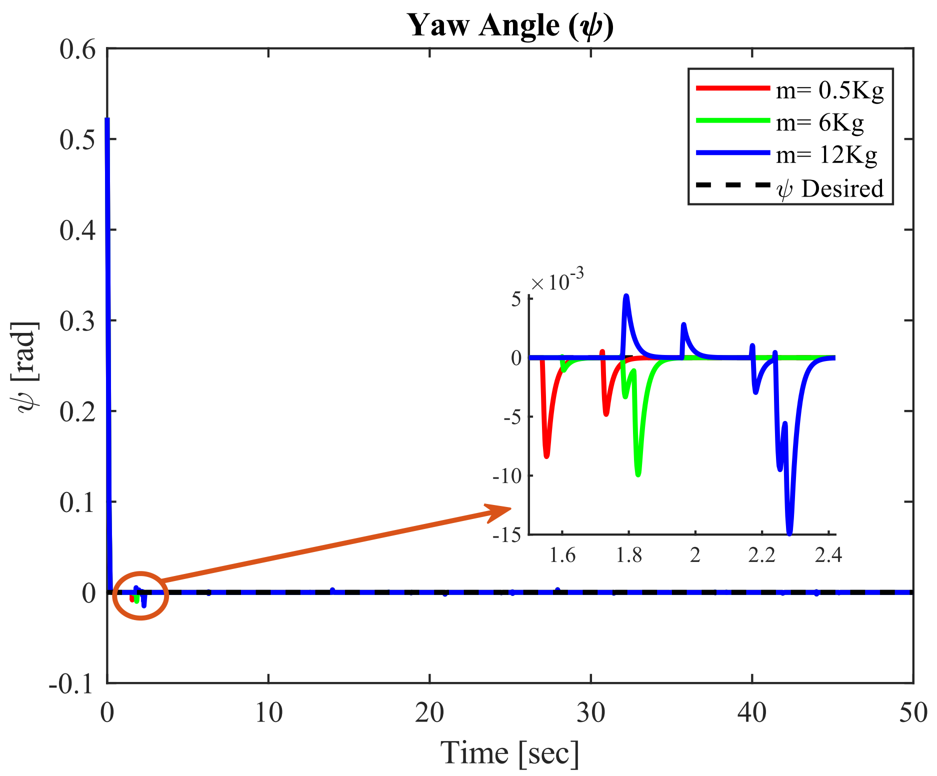


Figure 5.35 Response of Euler angle yaw under time varying external disturbance

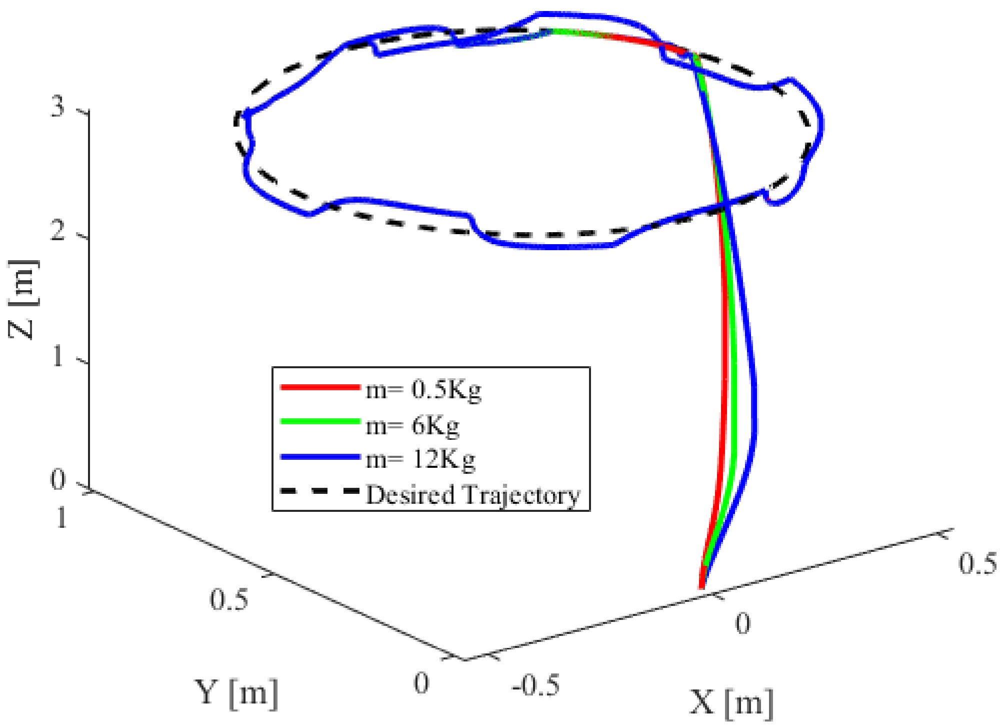


Figure 5.36 3D view of circular trajectory tracking under time varying external disturbance

Responses of the system when the quadrotor is subjected to a time varying external disturbance are shown in figures from 5.32 to 5.36 thus as shown from the figures the designed controller can handle a load varies from 0.5*kg* to 12*kg* when a time varying external disturbance having magnitude varies from -2 to +2 in position *X*, from -3 to +3 in position *Y* and from -5 to +5 in *Z* direction is added to the system as a wind disturbance.

5.5 Comparison of the Proposed Controller with SMC and BSC

Figure 5.37 to 5.39 shows the comparison of Backstepping Control (BSC), Sliding Mode Control (SMC) and the proposed Backstepping Sliding Mode Control (BSMC) for a circular trajectory tracking under normal conditions.

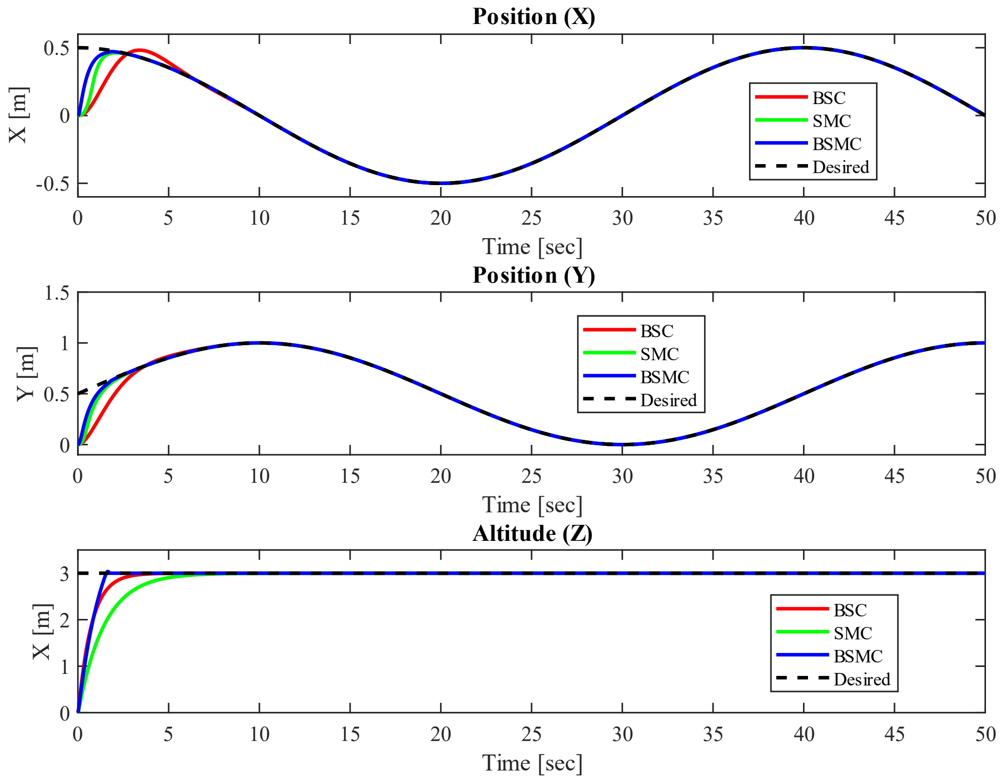


Figure 5.37 Comparison of translational subsystem output for circular trajectory tracking

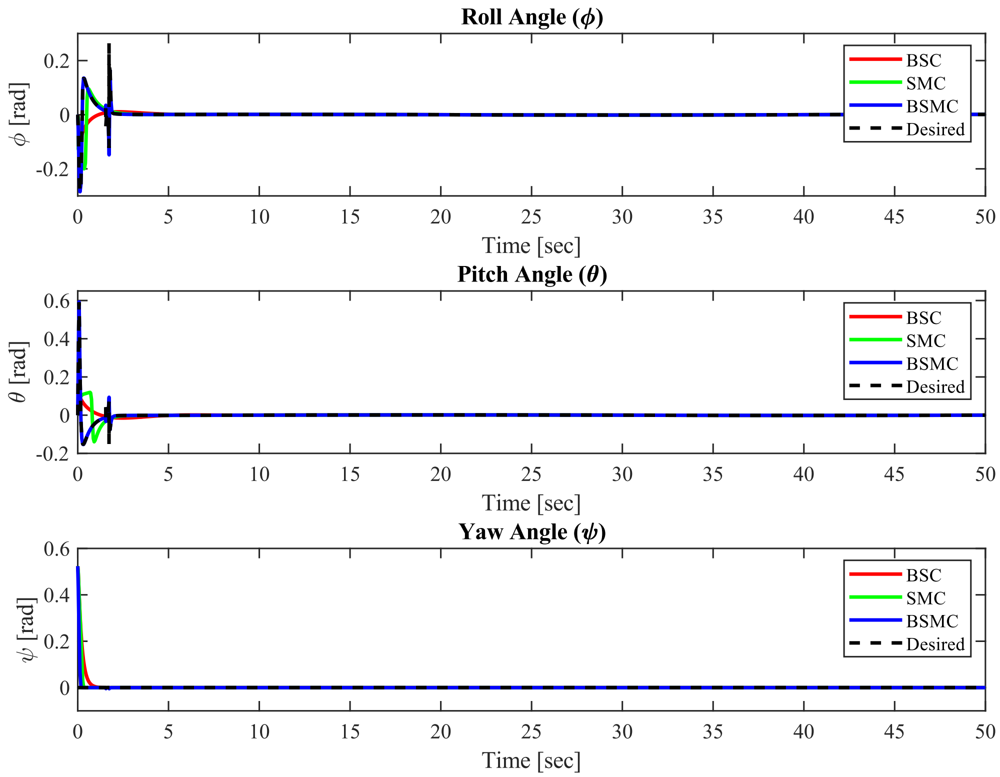


Figure 5.38 Comparison of rotational subsystem output for circular trajectory tracking

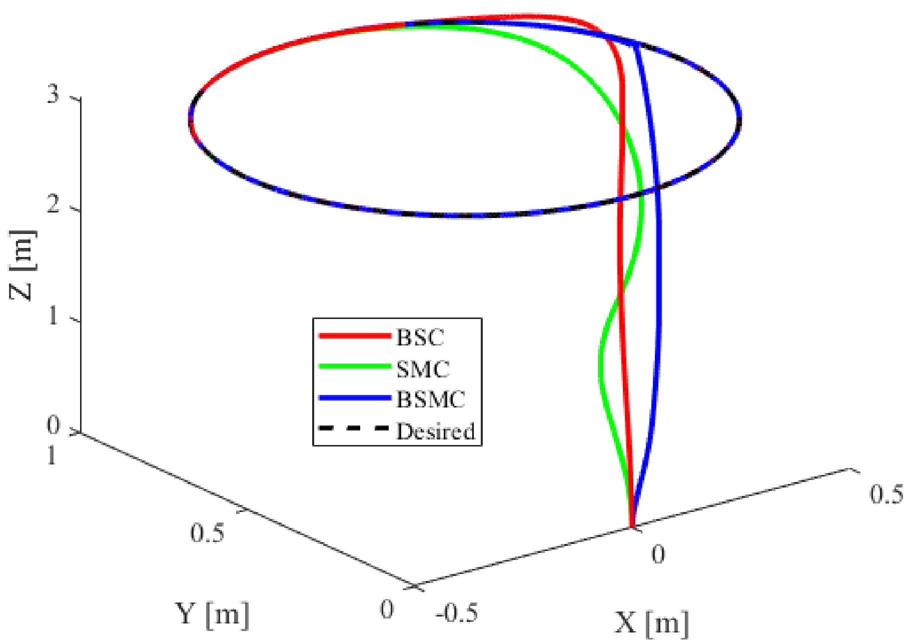


Figure 5.39 Comparison of circular trajectory tracking

As we have seen from the figures the proposed controller performs well when it is compared to the BSC and SMC. The total integral errors are 4.1287, 5.5410 and 2.5941 for BSC, SMC and BSMC respectively.

Table 5-5 Comparison of BSC, SMC and BSMC for a circular trajectory tracking based on IAE

Controllers	IAE					
	X	Y	Z	ϕ	θ	ψ
BSC	0.8836	0.8651	2.2646	0.0002	0.0002	0.1150
SMC	0.4619	0.4508	4.3914	0.0789	0.0783	0.0797
BSMC	0.2785	0.3222	1.9418	0.0078	0.0052	0.0386

5.5.1 Performance of SMC for Varying Load

In Figure 5.40 the performance of SMC for a mass of 1.5*kg* and 12*kg* is shown and the result shows that SMC cannot handle a mass of 12*kg* under normal condition. Also, Figure 5.41 shows a control input for SMC controller which shows unwanted phenomena called chattering which causes a vibration and heating to the system.

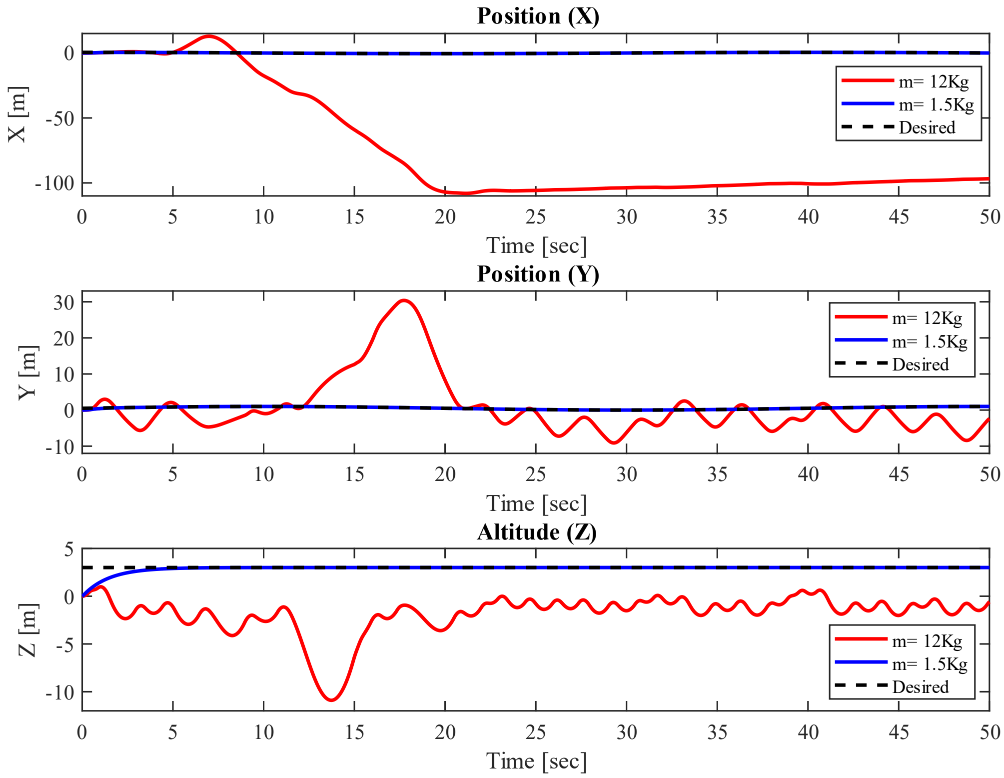


Figure 5.40 Response of translational subsystem for different mass using SMC

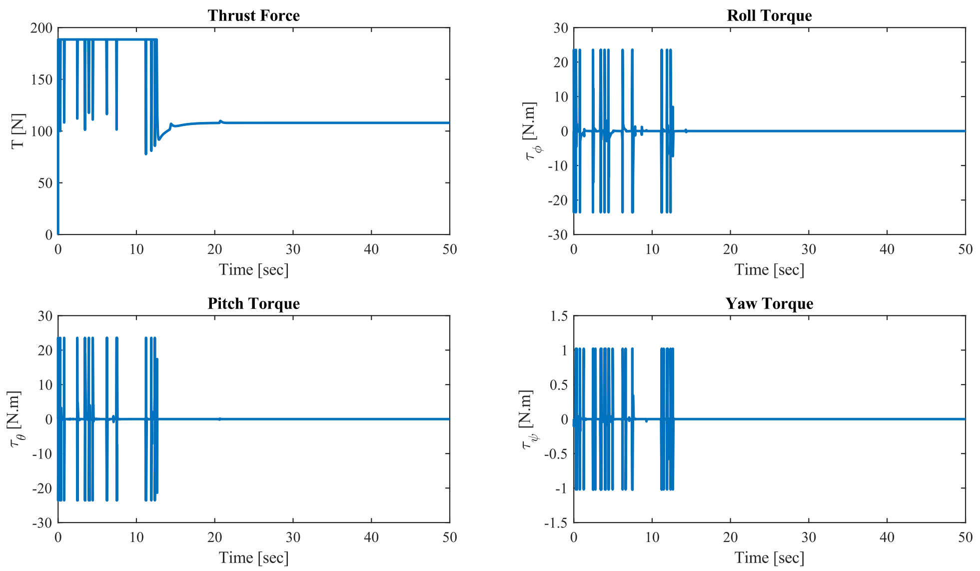


Figure 5.41 Control inputs from SMC

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

This research work introduced a backstepping sliding mode control (BSMC) technique to solve the trajectory control problem of quadrotor UAV which is uncommon in most of the literatures. The nonlinear dynamic equation of the quadrotor UAV is obtained based on Newton-Euler method. In addition to the system the actuator dynamics also considered by taking a BLDC motor as an actuator that drive the system. After derivation of the system model and actuator dynamics BSMC, SMC and BSC controllers are designed which makes the quadrotor follow the given desired trajectories.

The performance of the designed controllers is tested for different types of desired trajectories in MATLAB software environment by considering parametric uncertainties in mass and a time varying external disturbances with known upper and lower bound.

Finally, the simulation results conducted shows that the designed controller (i.e., BSMC) tracks the given trajectories with a very small tracking error. And it can handle a mass varies from 0.5*kg* to 18*kg* efficiently under normal conditions which means without external disturbance. Additionally, when the quadrotor is subjected to a time varying external disturbance with a magnitude varies from -2 to +2 in the position X direction, -3 to +3 in the position Y direction and -5 to +5 in the altitude Z direction it can handle a mass varies from 0.5*kg* to 12*kg*. But when it comes to SMC and BSC they can handle a mass varies from 0.5*kg* to 10*kg* and 0.5*kg* to 13*kg* respectively under normal conditions which means without external disturbance and also regarding control input SMC experiences a large chattering problem.

After observing the conducted simulations, it can be concluded that it is possible to control and stabilize the position, height, and attitude of a quadrotor system with a BSMC technique. Regarding the control input in this BSMC control technique a very small chattering effect is noticed.

6.2 RECOMMENDATIONS

For future researchers, this research work recommends the following:

1. Implementing of path planning algorithms which gives the quadrotor to choose an optimal waypoint based on the initial and the final destination given.

2. Analysis of battery performance.

3. Hardware implementation of the modelled quadrotor.

Acknowledgements

The research for this paper was financially supported by Adama Science and Technology University under the grant number of **ASTU/SM-R/804/23**, Adama, Ethiopia.

REFERENCES

Ahmad, Taher Azar' "Sundarapandian, Vaidyanathan." 2021. *Backstepping Control of Nonlinear Dynamical Systems*.

Alderete, Thomas S. 2023. *SIMULATOR AERO MODEL IMPLEMENTATION*.

Belsty, Derseh. 2021. *Particle Swarm Optimization Tuned Fractional Order Sliding Mode Controller for Altitude Stabilization and Trajectory Tracking of Agricultural Monitoring Quadcopter*.

Bui, Duy Nam, Thi Thanh Van Nguyen, and Manh Duong Phung. 2022. "Lyapunov-Based Nonlinear Model Predictive Control for Attitude Trajectory Tracking of Unmanned Aerial Vehicles." *International Journal of Aeronautical and Space Sciences*.

Chandra, Anoop, and Priya P.S. Lal. 2022. "Higher Order Sliding Mode Controller for a Quadrotor UAV with a Suspended Load." In *IFAC-PapersOnLine*, Elsevier B.V., 610–15.

Chovancová, Anežka, Tomáš Fico, Lubos Chovanec, and Peter Hubinský. 2014. "Mathematical Modelling and Parameter Identification of Quadrotor (a Survey)." In *Procedia Engineering*, Elsevier Ltd, 172–81.

Derafa1, L, T Madani2, and A Benallegue2. 2006. *Dynarnic Modelling and Experirnental Identification of Four Rotors Helicopter Pararneters*.

Dev, Meta, and Prasad Murthy. 2015. *DESIGN OF A QUADCOPTER FOR SEARCH AND RESCUE OPERATION IN NATURAL CALAMITIES Industrial Design*.

Gad, Ahmed G. 2022. "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review." *Archives of Computational Methods in Engineering* 29(5): 2531–61.

Hawi Aboma. 2022. "Grey Wolf Optimization and Integral Based Nonlinear Backstepping Controller for Unmanned Aerial Vehicles." Adama Science and Technology University.

Hou, Zhiwei, Xiang Yu, and Peng Lu. 2022. "Terminal Sliding Mode Control for Quadrotors with Chattering Reduction and Disturbances Estimator: Theory and Application." *Journal of Intelligent and Robotic Systems: Theory and Applications* 105(4).

Idrissi, Moad, Mohammad Salami, and Fawaz Annaz. 2022a. "A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms." https://doi.org/10.1007/s10846-021-01527-7.

Jinay S. Gadda. 2013. "QUADCOPTER (UAVS) FOR BORDER SECURITY WITH GUI SYSTEM." *International Journal of Research in Engineering and Technology* 02(12): 620–24. https://www.academia.edu/21216830/QUADCOPTER_UAVS_FOR_BORDER_SECURITY_WITH_GUI_SYSTEM (December 24, 2022).

Jinkun Liu. 2017. "Sliding Mode Control Using MATLAB." In *Sliding Mode Control Using MATLAB*, Elsevier, i–iii.

Li, Yang Rui, Chih Chia Chen, and Chao Chung Peng. 2023. "Integral Backstepping Control Algorithm for a Quadrotor Positioning Flight Task: A Design Issue Discussion." *Algorithms* 16(2).

Lidiya Girma. 2021. "Minimum Jerk Trajectory Tracking and Control of Quadcopter Using Adaptive Sliding Mode Controller." Adama Science and Technology University.

Liu, Lei et al. 2022. "Fault-Tolerant Control for Quadrotor Based on Fixed-Time ESO." *Mathematics* 10(22).

Norouzi Ghazbi, S., Y. Aghli, M. Alimohammadi, and A. A. Akbari. 2016. "Quadrotors Unmanned Aerial Vehicles: A Review." *International Journal on Smart Sensing and Intelligent Systems* 9(1): 309–33.

Patel, Parth N et al. 2013. "Quadcopter for Agricultural Surveillance." 3(4): 427–32. http://www.ripublication.com/aeee.htm (December 24, 2022).

Rinaldi, Marco, Stefano Primatesta, and Giorgio Guglieri. 2023. "A Comparative Study for Control of Quadrotor UAVs." *Applied Sciences* 13(6): 3464.

Saibi, Ali, Razika Boushaki, and Hadjira Belaidi. 2022. "Backstepping Control of Drone †." *Engineering Proceedings* 14(1).

Singh, Brajesh Kumar, and Awadhesh Kumar. 2023. "Model Predictive Control Using LPV Approach for Trajectory Tracking of Quadrotor UAV with External Disturbances." *Aircraft Engineering and Aerospace Technology* 95(4): 607–18.

Singhal, Gaurav, Babankumar Bansod, and Lini Mathew. 2018. "Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review Prediction Model for Moisture Determination of Indian Wheat Based on Electrical Properties Variations View Project Facile Synthesis of Nano Sized ZnO by Hydrothermal Method View Project Babankumar Shyam Bansod Central Scientific Instruments Organization Unmanned Aerial Vehicle Classification, Applications and Challenges: A Review." http://revistapesquisa.fapesp.br.

Teixeira, João Marcelo, Ronaldo Ferreira, Matheus Santos, and Veronica Teichrieb. 2014. "Teleoperation Using Google Glass and AR, Drone for Structural Inspection." *2014 XVI Symposium on Virtual and Augmented Reality*: 28–36.

Teppo Luukkonen. 2014. *Modelling and Control of Quadcopter*.

Thu, Kyaw Myat, and A. I. Gavrilov. 2017. "Designing and Modeling of Quadcopter Control System Using L1 Adaptive Control." In *Procedia Computer Science*, Elsevier B.V., 528–35.

Wang, Dong, Quan Pan, Jinwen Hu, and Chunhui Zhao. 2022. "Nonlinear Model Predictive Control for Trajectory Tracking of Quadrotors Using Lyapunov Techniques." *Science China Information Sciences* 65(6).

Xu, Lin Xing et al. 2020. “Backstepping Sliding-Mode and Cascade Active Disturbance Rejection Control for a Quadrotor UAV.” *IEEE/ASME Transactions on Mechatronics* 25(6): 2743–53.

Zaidi, Adeel, Muhammad Kazim, and Hui Wang. 2022. “A Robust Backstepping Sliding Mode Controller with Chattering-Free Strategy for a Swarm of Drones.” In *Journal of Physics: Conference Series*, IOP Publishing Ltd.

Zhu, Zhiqiang, and Songyin Cao. 2019. “Back-stepping Sliding Mode Control Method for Quadrotor UAV with Actuator Failure.” *The Journal of Engineering* 2019(22): 8374–77.

REFERENCE LINKS

[L1] https://www.researchgate.net/publication/336815990/figure/fig1/AS:941820632633357@1601558945311/Quadrotor-UAV-architecture_Q640.jpg

[L2] https://th.bing.com/th/id/OIP.pyV6gt_AZwMqLM-_uju_qQHaF8?pid=ImgDet&rs=1

APPENDICES

A.1 MATLAB FUNCTION CODES

% MODEL PARAMETRES

```
Ix    = 3.8278e-3; Iy= 3.8278e-3; Iz= 7.1345e-4; %Nmperradpersec^2
m     = 1.5; %Kg
L     = 0.5; %metre
g     = 9.81; %mpersec^2
Jr    = 2.8385e-5; %Kgm^2
Kt    = 2.9842e-5; %Nperradpersec
Kd    = 3.232e-7; %Nmperradpersec
om_max= 400*pi; %radpersec 1000Kv Motor by 12V Lipo Battery
u     = [0;0;0;0;0;0]; %Initial Control Input
um1= 0; um2= 0; um3= 0; um4= 0;

a1 = (Iy-Iz)/Ix;
a2 = Jr/Ix;
a3 = (Iz-Ix)/Iy;
a4 = Jr/Iy;
a5 = (Ix-Iy)/Iz;
b1 = 1/Ix;
b2 = 1/Iy;
b3 = 1/Iz;
```

function xdot= RealModel(t,x,u)

```
global m g u om_bar a1 a2 a3 a4 a5 b1 b2 b3 Ix Iy Iz Jr U1 U2 U3 U4 Ux Uy Fx Fy Fz
a1 = (Iy-Iz)/Ix;
a2 = Jr/Ix;
a3 = (Iz-Ix)/Iy;
a4 = Jr/Iy;
```

```
a5 = (Ix-Iy)/Iz;
b1 = 1/Ix;
b2 = 1/Iy;
b3 = 1/Iz;
u = [U1; U2; U3; U4; Ux; Uy];
xdot    = zeros (12,1);

  % (X, Y, Z)-> (x1, x3, x5) position wrt INERTIAL frame

xdot(1) = x(2);
xdot(2) = (u(1)/m)*(cos(x(10))*cos(x(12))*sin(x(11)) + sin(x(12))*sin(x(10))) + Fx;
xdot(3) = x(4);
xdot(4) = (u(1)/m)*(cos(x(10))*sin(x(12))*sin(x(11)) - cos(x(12))*sin(x(10))) + Fy;
xdot(5) = x(6);
xdot(6) = (u(1)/m)*cos(x(10))*cos(x(11)) - g + Fz;

 % (p,q,r)->(x7,x8,x9) angular velocity wrt body fixed frame

xdot(7) = a1*x(8)*x(9) - a2*om_bar*x(8) + b1*u(2);
xdot(8) = a3*x(7)*x(9) + a4*om_bar*x(7) + b2*u(3);
xdot(9) = a5*x(7)*x(8) + b3*u(4);

 % (phiROLL,thetaPITCH,psiYAW)->(x10,x11,x12) angular position wrt inertial frame

xdot(10)= x(7) + sin(x(10))*tan(x(11))*x(8) + x(9)*cos(x(10))*tan(x(11));
xdot(11)= x(8)*cos(x(10)) - x(9)*sin(x(10));
xdot(12)= (x(8)*sin(x(10)))/cos(x(11)) + (x(9)*cos(x(10)))/cos(x(11));
end
```

```
% ACTUATOR DYNAMICS
%%MOTOR DYNAMICS
function xdot= Motor1(t,xm1,um1)
global am0 am1 am2 bm
bm= 280.19; am0= 189.63; am1= 6.0612; am2= 0.0122;
xdot= -(am0 + am1*xm1 + am2*xm1^2) + bm*um1;
 end

function xdot= Motor2(t,xm2,um2)
global am0 am1 am2 bm
bm= 280.19; am0= 189.63; am1= 6.0612; am2= 0.0122;
xdot= -(am0 + am1*xm2 + am2*xm2^2) + bm*um2;
 end

function xdot= Motor3(t,xm3,um3)
global am0 am1 am2 bm
bm= 280.19; am0= 189.63; am1= 6.0612; am2= 0.0122;
xdot= -(am0 + am1*xm3 + am2*xm3^2) + bm*um3;
end

function xdot= Motor4(t,xm4,um4)
global am0 am1 am2 bm
bm= 280.19; am0= 189.63; am1= 6.0612; am2= 0.0122;
xdot= -(am0 + am1*xm4 + am2*xm4^2) + bm*um4;
end
```