Optimal Sliding Mode Control with PID Surface based Trajectory Control of Quadcopter for Wind Turbine Surface Damage Detection

Birkie Chanie Agegnehu

A Thesis Submitted to Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the of Degree Master's in Electrical Engineering (control engineering)

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama, Ethiopia

Optimal Sliding Mode Control with PID Surface based Trajectory Control of Quadcopter for Wind Turbine Surface Damage Detection

Birkie Chanie Agegnehu

Advisor: Dr. Shimeles Demissie

Co-Advisor: Dr. Selewondim Eshetu

A Thesis Submitted to Department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the of Degree Master's in Electrical Engineering (control engineering)

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama, Ethiopia

DECLARATION

I so certify that this master's thesis," **Optimal Sliding Mode Controller with PID Surface based trajectory control of Quadcopter for Wind Turbine Surface Damage Detection**" is my original creation. In other words, no application for a degree, diploma, or certificate has been made to any other university. All of the information sources utilized to create this thesis have been correctly cited and acknowledged.

Birkie Chanie Agegnehu	________________	________________
Name of student	Signature	Date

RECOMMENDATION OF ADVISOR

As the thesis advisor, I hereby attest that I have read the updated thesis, "**Optimal Sliding Mode Controller with PID Surface Based Trajectory Control of Quadcopter for Wind Turbine Surface Damage Detection**" which was prepared under my guidance by **Birkie chanie Agegnehu** submitted as part of the requirements for a Master of Science in Electrical Power and Control Engineering (Control Engineering). As a result, I advise submitting the thesis to the department in amended form in accordance with the relevant protocols.

Dr. Shimeles Demissie	____________	____________
Advisor	Signature	Date

Dr. Selewondim Eshetu	____________	____________
Co-Advisor	Signature	Date

APPROVAL OF PAGE

I, the advisor of this thesis entitled ‘‘**Optimal SMC with PID Surface Based Trajectory Control of Quadcopter for Wind Turbine Surface Damage Detection**’’ and developed by Birkie Chanine Agegnehu, hereby certify that the recommendation and suggestion made by the board of examiners are appropriately incorporate into the

Dr. Shimeles Demissie

Advisor Signature Date

Dr. Selewondim Eshetu

Co-Advisor Signature Date

We, undersigned member of the Bord Examiners of the thesis by Birkie Chanie Agegnehu have red and evaluated the thesis entitled **Optimal SMC with PID Surface Based Trajectory Control of Quadcopter for Wind Turbine Surface Damage Detection** and examined 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

Dr Beteley Teka Sept 24, 2024

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) via the Department Graduate Council (DGC) and School Graduate Committee (SGC) in order for it to be approved and accepted.

Department Head Signature Date

School Dean Signature Date

Office of Postgraduate Study Dean Signature Date

ACKNOWLEDEMENT

First and foremost, I am grateful to the Almighty God for giving me the capacity, opportunity, and fortitude to start and finish my thesis. I would want to thank Adama Science and Technology University for giving me the chance to take part in this Master's degree. I would especially want to thank the Department of Power and Control Engineering and the School of Electrical Engineering and Computing.

I am overflowing with joy as I thank **Dr. Shimelse Demissie**, my mentor and adviser, for his steadfast support and direction. I am so appreciative of his services since they have been invaluable in helping me finish this thesis. He has provided me with constant inspiration, wise counsel, and steadfast support. My appreciation also goes to the instructors at my university for creating a stimulating learning environment, my friends and colleagues for their spirit of cooperation, and my husband and families for their unwavering support. This research is the result of many people's combined efforts, and I sincerely thank them for their contributions to its accomplishment.

I would also like to extend my sincere appreciation to my co-advisor, **Dr. Selewondim Eshetu**. His expertise and constructive criticism have been immensely helpful in refining my research and ensuring its clarity. **Dr. Selewondim Eshetu** dedication to his students is truly remarkable. He has invested countless hours in helping me to develop my skills and refine my ideas, and his commitment to excellence has inspired me to strive for the highest standards in my work.

Thank you.

Birkie Chanie Agegnehu.

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

Abbreviations	Description
SMC	Sliding Mode Control
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization
COG	Center of Gravity
DOF	Degree of Freedom
ITAE	Integral Time Absolute Error
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
mAP	mean Average Precision
YOLO	You Only Look Once
YOLOv8	You Only Look Once version 8
GPS	Global Positioning System

ASMC	Adaptive Sliding Mode Control
CW	Clockwise
CCW	Counterclockwise
IoU	Intersection over Union
MPC	Model Predictive Control
UAV	Unmanned Aerial Vehicle
ADRC	Active Disturbance Rejection Control
VTOL	Vertical Takeoff and Landing

NOMENCLATURE

kt	Aerodynamics thrust coefficient
d	Aerodynamics drug coefficient
Ω	Angular velocity of quadrotor
ϕ, θ, ψ	Euler angle
ϕ', θ', ψ'	Euler rate
g	Gravity
l	Length of propeller
m	Quad copter mass
Ωr	Quad copter relative speed
Jtp	Inertia of rotor
p, q, r	Angular velocity in body frame
Vx, Vy, Vz	Linear velocity in body frame
Ixx, Iyy, Izz	Moment of inertia around propeller

ABSRACT

Unmanned aerial vehicles (UAVs) are aircraft that are able to fly without the need for a pilot, thanks to financial and technological breakthroughs. these UAVs have made a number of things possible, including mapping, aerial photography, video recording, pollution and land monitoring, wind turbine inspection, firefighting, agricultural applications, military activities, and more. The nonlinear, underactuated nature, instability, and uncertainty around the parameters make it difficult to manage the quadrotor's flight. In this research work, a quadrotor dynamic model is developed, and Optimal sliding mode control using the proportional integral derivative (PID) surface technique is applied to control the trajectory of quadrotor including its trajectory during wind turbine surface damage detection. The designed controller incorporates SMC with PID surface control method and quasi function to adequately reduce the chattering effect of sliding mode control. Additionally, image acquired from the quadrotor are train using Roboflow in YOLOv8. Particle swarm optimization (PSO) is used to optimize the controller parameters. Finally, the performance of designed controller tested under parameter variation and wind wind disturbance with designed specific trajectory of quadrotor during wind turbine surface damage detection. The simulation result shows that quadrotor can handle parameter variation and wind disturbance and able to follow the designed trajectory effectively. Also chattering effect due to conventional SMC is reduced effectively. In ream of image processing, the system able to identify damaged part of wind turbine with best performance of training and validation. Additionaly, the performance of the SMC with PID surface control is compared with SMC to the proposed controller shows robust performance.

Keywords: - *Optimal SMC, Lyapunov stability, Quadcopter, UAV, PSO, YOLOv8, PID surface, Wind turbine surface damage detection.*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

Unmanned aerial vehicles, sometimes known as UAVs, are aircraft that are piloted remotely from the ground or by commands that have been preprogrammed, rather than by a human pilot on board. They come in a variety of sizes, from compact portable devices to massive drones fit for the military. UAVs are becoming more and more common because of their adaptability and capacity to carry out activities that could be hazardous or challenging for human pilots (Amiri & Hosseinzadeh, 2024). Flight control, navigation, and a communication link between the aircraft and the ground station are some of the essential parts of a UAV. In order to keep the UAV stable and regulate its movement, the flight control systems use sensors and software, while the navigation system uses GPS and additional sensors to direct the aircraft to its goal. The pilot can operate the UAV and get data and video in real time from the aircraft over the communication link.

Unmanned aerial vehicles with rotating wings that can take off and land vertically are known as quadrotors. This aerial aircraft features four motors that deliver thrust, or propulsion, to propellers. To maintain control and balance, the engine speed is adjusted. Due of its numerous potential applications in both the military and the civilian world, quadrotors have garnered the interest of researchers and engineers in recent times. Although the quadrotor is strongly linked, MIMO, intensely nonlinear, and underactuated, it possesses a wide range of capabilities. Although it appears to be an easy vehicle to operate, controlling it requires an accurate and useful model. A quadrotor's dynamics as a rotorcraft are mostly determined by the intricate aerodynamic effects of its rotors. They are capable of hovering, VTOL, and daring acrobatics, among other things. Moreover, its lightweight, compact design, and ability to carry weights significantly heavier than its own weight. The military (Kannan & Student, 2016), search and rescue (“Search and Rescue,” 1995), building inspections (Amaral & Henriques, 2013), surveillance (Tiwari et al., 2021), and additional use cases for quadrotors are numerous. UAVs with quadrotor types have been the focus of in-depth study and development in a number of fields. These vehicles, with their distinct capabilities and maneuverability, have drawn a lot of interest from the academic and industrial research communities. As the number of quadrotor applications rises, a powerful enough controller is necessary to enable the quadrotor to carry out the assigned tasks.

Quadcopters can be divided into two primary groups. aircraft equipped with remote and autonomous pilots. The commonality between the two groups is the absence of human presence or involvement in the flight environment. Humans use wireless connection to operate and guide an unmanned aerial vehicle (UAV) in a remotely piloted aircraft (Belsty, 2021). However, the car falls into the autonomous aircraft category because it is sophisticated enough to fly and finish the designated duty on its own. While pressure beneath the wing produces more air lift, fixed-wing unmanned aerial vehicles need an engine to provide thrust force in order to fly along the ground (Lungu, 2019). The capacity of unmanned aerial vehicles (UAVs) to perform (VTOL) and (CTOL), is the basis for another important classification or take off in some places.

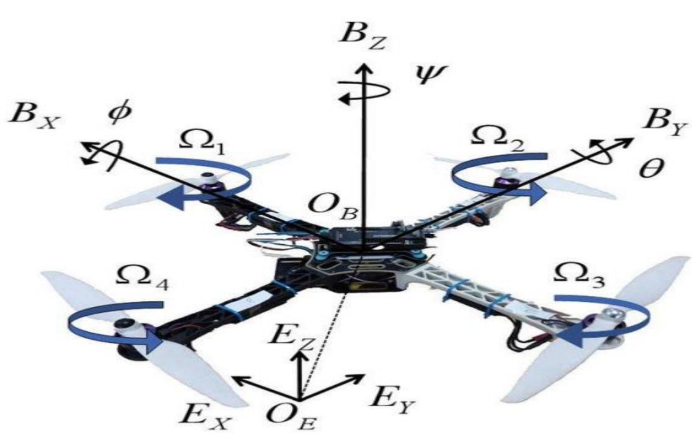


Figure 1.1 Quadrotor system(Haile G, 2023)

Today, VTOL drones make up the majority of UAVs deployed for commercial purposes. Unmanned aerial vehicles (UAVs) of the CTOL category are popular because of their high speed and ability to transport huge loads, even while conventional vehicles cannot land. Two of the diagonal rotors revolve in a clockwise manner, and the other two rotate counterclockwise. This setup is crucial because it maintains the quadcopter's equilibrium by producing torque in the opposite direction and in a comparable quantity.

Interestingly, the optimal sliding mode controller uses fixed parameters, which is why this study's automated parameter change is quadrotor via a PSO optimization method. This study investigates the use of image processing algorithms for wind turbine surface damage detection in conjunction with an optimal sliding mode controller combined with PID controls a quadcopter.

1.2 Statement of the Problem

Being able to fly in any direction, take off and land vertically, and hover at the proper altitude are the main advantages of quad rotor. Controlling a quad rotor UAV can be difficult because of its unique characteristics, which include under actuation, instability, no minimum phase, presence of parameter uncertainty, multivariable nature, tightly coupled non-linear difficulties, and external disturbances during structural inspection such as wind turbine. However, the challenge is in ensuring precise quadrotor trajectory control while maintaining stability under a range of environmental conditions.

Sliding mode controller: a reliable control technique, is mostly utilized for quadrotor trajectory control systems due to its rapid response to disturbances and parameter ambiguity. SMC does, however, have several disadvantages, including the chattering effect, decreased control precision, and heat losses in electrical components.

Therefore, trajectory control system based on SMC with PID surface is designed which takes into account of specific trajectory designed for quadrotor during wind tubine surface damage detection. Additionally, the image processing technique using YOLOv8 for training image captured by quadrotor and able to detect damage on wind turbine surface quadrotor-captured images and detecting damage. The recommended controller combines SMC and PID for the surface to reduce chattering. Finally, a comparison between SMC and SMC with PID control shows how effective the proposed controller.

1.3 Objective

1.3.1. General Objective

The main objective of this thesis is to design and simulate Optimal Sliding Mode Controller with PID surface Based trajectory control of quadcopter for wind turbine surface damage detection.

1.3.2 Specific Objective

- To define a mode for quadcopter.
- To design Optimal SMC with PID sliding surface for quadcopter system.
- To simulate the suggested system using MATLAB and analyzing the performance and robustness and draw relevant conclusion.
- To detect damage on wind turbine surface using YOLOv8.

1.4. Scope of Thesis

This thesis addresses several significant research topics an optimal sliding mode controller is used to control the quadcopter's movements using a full mathematical model, especially when it comes to following a predetermined reference trajectory. Furthermore, the research utilizes image processing methodologies to identify wind turbine surface impairments, specifically focusing on detecting wind turbine impairments in images captured by quadcopters. Finally, in order to assess the controller's performance, each of these components is simulated and analyzed using MATLAB/Simulink and YOLOv8 for wind turbine surface damage detection.

1.5. Limitation of the Study

The system is more expensive and time-consuming to implementation in real world. Because of this, the controller and system design of this thesis will only be Implemented using MATLAB/Simulink environment despite covering a lot of ground, it does not involve creating a working quadcopter prototype.

1.6. Motivation of the Study

Quadrotor drone technology's advancement and widespread use have opened up a wide range of options for several businesses. Numerous industries, including infrastructure inspection, photography, search and rescue, and agriculture, have demonstrated the effectiveness of these versatile vehicles. To fully utilize the capabilities of these sophisticated systems, complex control algorithm design such as SMC and SMC control with PID sliding surface to reduce chattering effect and for accurate control, effective operation, and job execution that is safe.

1.7. Significance of the Study

Enormous interest surrounds the study of using control systems to steer, tilt, and position unmanned aerial vehicles (UAVs). The diversity of benefits and uses that come with multirotor UAVs makes research efforts in this field fascinating. Currently, similar research projects are being carried out in numerous nations.

This thesis is groundwork for future wind turbine surface damage detection research, which makes it highly significant in the academic community. It serves as a strong platform for further research, enabling the discovery of new information and advancements in this area. The thesis also includes an integral term in the sliding surface for the purpose of reducing steady-state inaccuracy in the quadcopter's control. Additionally, it makes use of PSO to automatically modify the Sliding Mode Controller (SMC) using PID surface settings. This control technique improves the system's overall performance by reducing chattering, increasing resilience, and requiring less control effort.

1.8 Thesis Organization

The thesis is organized as follows:

Chapter 2: This section gives a review of UAVs, quadrotors, and provides a thorough analysis of pertinent literature, addressing earlier studies and conclusions about quadcopters, control systems, and image processing.

Chapter 3: The nonlinear dynamic model of the quadrotor system is built in this part using the Euler-Newton formalism, The quadrotor system is then modeled and expressed in state space form, and the open-loop model is simulated.

Chapter 4: Describes the SMC with PID controller design for the quadcopter, emphasizing the control approach used to improve performance and stability.

Chapter 5: examines the use of image processing methods—more especially, PyCharm for the detection and wind turbine surface damage identification in quadcopter-captured photos.

Chapter 6: presents the findings from the simulations and has a long discussion about these findings.

Chapter 7: presents a summary of the thesis's results, conclusions, and suggestions based on the findings of the research. It also suggests possible directions for further study and advancement in the area.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Summary of the Chapter

An outline of the different kinds, uses, background, and synopsis of UAV quadcopters is given in this chapter. Next, the concept and architecture of the UAVs belonging to the quadcopter category are discussed. Finally, a review and analysis of related studies are presented, along with pros and cons of each.

2.2. Unmanned Aerial Vehicle

Unmanned aerial vehicles (UAVs) are tiny aircraft that have been built or modified to run entirely on electronic controls that are triggered by a flight controller, eliminating the need for a human pilot. They can operate independently or under human control via a remote control. Onboard computers with pre-programming capabilities to do a variety of tasks or only a few, UAVs, serve a civilian purpose in addition to their primary military one. Worldwide Similar too many other technologies, unmanned aerial vehicles (UAVs) were developed during armed conflicts and used for a variety of military purposes, including targeted killing, reconnaissance, and surveillance. However, because of cost-cutting measures and technical advancements, they can now be utilized for non-military as well as military objectives (Belsty, 2021).

2.2.1. History of UAV

The term "UAV" was initially used to explain the initial production of aerial torpedoes by American firms Lawrence and Sperry in 1916, or unmanned aerial vehicles. These UAVs, shown in Figure 2.1, showed a 30-mile flight range. It's thought that Lawrence and Sperry used a gyroscope to keep the airplane stable (Lidiya Girma, 2021).

Figure 2.1 lawerence (Lidiya Girma, 2021)

2.2.2. Classification of UAVs

Numerous factors can be used to classify unmanned aerial vehicles (UAVs), such as their aerodynamic design, size, payload, range of operation, and level of autonomy (Lidiya Girma, 2021);

- Size and payload configuration
- Maximum altitude and endurance range of actions
- Configurations that are aerodynamic.
- Fixed-wing, Blimps, and Flapping-wing aircraft
- Helicopter, single-rotor, coaxial, and multirotor unmanned aerial vehicle models are among the UAVs that have rotary wings.

2.3. Quadcopters

Unmanned aerial vehicles (UAVs) with four rotors placed in a plus or "X" configuration are called quad rotors, or quadcopters. They have become remarkably more popular in recent decades because to their stability, agility, and maneuverability..

2.3.1 The Quadcopter History

Although quadrotors have existed since the early 20th century, major developments took place in the 1990s and early 2000s. One of the earliest known instances of a quadrotor is the "Gyroplane No. 1," created in 1990 by Étienne Oehmichen. Quadrotors, however, did not start to get interest for their possible uses until the 2000s (Norouzi Ghazbi et al. 2016). Ever since, quadrotors have gained significant attention in the scientific community, resulting in the creation of numerous varieties and uses.

2.3.2 Setup for a Quadcopter

Depending on how their rotors are arranged and oriented, quadrotors can have a variety of configurations or variations (Thu and Gavrilov 2017). The following are a few popular quadrotor configurations:

Plus (+) Configuration: This arrangement places the four rotors in the form of a plus sign. A cross-like configuration is formed by the placement of two rotors in the front and two in the rear. Stability and mobility are offered in every direction by this design.

Cross (X) Configuration: The rotors are arranged diagonally in the X configuration, which is similar to the + configuration in setup. The front left and rear right rotors make two diagonals, and the front right and rear left rotors form one. Additionally, this arrangement provides stability and agility and other many configuration like H shape, Y shape, V shape etc.

2.3.3. Components of Quadcopter

Four rotors power a quadrotor. Each rotor is made up of a number of parts that come together to allow the quadrotor to fly and be controlled (Dutta & Banerjee, 2019). A quadrotor's principal parts include:

- **Frame:** The quadrotor's structure and support are provided by the frame. For the purpose of minimizing weight without sacrificing durability, lightweight materials like aluminum or carbon fiber are usually used.
- **Motors:** Four brushless DC electric motors, one for each rotor, make up a quadrotor. The thrust required to raise and control the quadrotor in the air is produced by these motors.

- **Propellers:** Each engine has a propeller attached to it. Propellers rotate quickly to produce the airflow required for propulsion and lift. Typically, the propellers are composed of lightweight materials like carbon fiber or plastic.
- **Sensors:** To collect information about their orientation, altitude, and speed, quadrotors are outfitted with a variety of sensors such as Gyroscopes, magnetometers, barometers, accelerometers, and occasionally GPS receivers are among the most widely used.
- **The battery**: of the quadrotor is a rechargeable lithium polymer (LiPo) battery that powers the motors, flight controller, and other electronic parts of the device. Typically, the battery is attached to the ESCs, flight controller, and frame.
- **The flight controller:** is considered the "brain" of the quadrotor. It is a circuit board with an integrated microcontroller that interprets information from many sensors and applies algorithms to regulate the flying modes, steady the quadrotor, and carry out user directions

2.3.4 Applications of Quadcopter

There are several applications for quadrotors, here is a list of a few applications:

- **Filmmaking and Photography**: Camera-equipped quadrotors provide a distinctive viewpoint for filmmaking and photography from above.
- **Examining and Monitoring**: A practical platform for examining infrastructure, including wind turbines, pipelines, electricity lines, and buildings, is offered by quadrotors.
- **Operations for Search and Rescue:** Quadrotors equipped with heat cameras and other sensors can be useful in search and rescue operations rescue personnel can get up-to-date situational information also used for accurate farming, provision of services etc.

2.4. Related Works

Multinational field researchers have developed a range of UAV controllers. While several traditional quadcopter techniques have been used, each has pros and cons. Examples of these include proportional integral derivative (PID), linear quadratic regulator (LQR), back steeping controller, and sliding mode control (SMC), ASMC, BSSMC, NNSMC.

(Labbadi et al., 2020) This paper presents an FO control method guarantees rapid convergence, excellent precision, and resilience to stochastic disturbances and uncertainties. Lastly, an assessment of the FOSTPIDSMC's performance under various conditions is carried out. The simulation findings, in contrast to the nonlinear internal model control (NLIMC) and methods for FO backstepping sliding mode control (FOBSMC), clearly demonstrate enhanced control performance, efficiency, and severe disruption. This indicates that the controller used in this work's rejection capacity is effective. Nevertheless, this thesis does not automatically apply any optimization approach to change the value.

(Mokhtari, 2022) This work focuses on using FOPID controllers to regulate the nonlinear dynamics of a quadrotor; neural network tuning techniques were also applied. The best course that complies with speed restrictions while preserving the trajectory's safety and viability is examined in the following section. By employing a machine learning technique based on neural networks to simulate a controlled plant, path planning can be made more computationally efficient. One algorithm that is considered for path optimization is swarm optimization. The results show how well the previously mentioned algorithm worked in terms of accuracy and suitability. Nevertheless, because quadrotor dynamics are nonlinear, they are challenging to manage with these kinds of linear controllers, and changing a quadrotor's parameters does not measure performance.

(Nguyen et al., 2021) An system of attitude and altitude control for a quadcopter that can adjust to outside disturbances is presented in this paper using a neural network. Through numerical simulation, the suggested sliding mode control's effectiveness has been assessed. The findings demonstrate that the attitude and altitude controller that adheres to the recommended process has superior tracking performance. Regrettably, chattering effect is an issue with conventional SMC controllers, and chattering reduction strategies are ignored.

(Lidiya Girma, 2021) to control an unmanned aerial vehicle's attitude, position, altitude, and direction in the face of outside disturbances and unknown features, this project aims to develop an Adaptive Sliding Mode Controller (ASMC) and a minimum jerk trajectory. To lessen chattering problems, Sliding Mode Control (SMC) uses continuous adaptive dynamic vectors, and to attain asymptotic stability, and Lyapunov stability analysis is employed. Nevertheless, the suggested controller's resilience to outside disruptions and errors in the model were not assessed.

(Usman, 2020) The vehicle was operated and tested with Arduino hardware to collect data and evaluate the control system. For the project, the internet was the main source of information and study materials. Relevant books, articles, and online training courses were also reviewed in addition to the internet. It was feasible for the controller to lift the quadcopter off the ground while it was moving erratically on the hardware as it was not intended to maintain the vehicle in its take-off position. Nevertheless, processing signals scaling and restricting time for PIDs to modify the quadcopter's take-off behavior did not work as anticipated from simulated studies of the model due to hardware restrictions. But because PID is a linear controller, control is difficult since quadrotor dynamics is highly nonlinear in nature.

(Hoell & Omenzetter, 2014) Numerical models of a solitary WT blade under a disbanding damage scenario are conducted using this technology. The technique for determining damage based on response acceleration autocorrelations is described in this study. Transient simulations with a reduced aerodynamic loading provide time series of acceleration. Because the findings of damage identification are sensitive to the chosen damage scenario, they hold promise for the advancement of damage detection methods in WTs in the future. But most need dense sensor arrays or are not appropriate for in-service measurements.

(A, 2018) This work provides a case study on the diagnosis and detection of damage in an actual, albeit tiny, wind turbine blade using ultrasonic NDE/SHM techniques. We investigate two approaches: guided-wave "pitch-catch" SHM and nonlinear acoustics. The guided-wave method employed a "network of novelty detectors" mechanism to identify and diagnose the damage. However, when experimental damage was introduced into the blade, the nonlinear acoustics approach demonstrated a dismal lack of response. An first set of poorly designed guided wave experiments provided important insights into wave attenuation inside the desired configuration and validated the concept of actuator-sensor networks at a practical density for wind turbine blades. SHM. Nevertheless, the current results are inconclusive.

(Reizenstein, 2017) The development and application of quadcopter position and trajectory control is covered in this thesis. Originally, the quadcopter was outfitted with sensors and software for orientation estimation and control, rather than being designed to ascertain its present location. They then optimized it using an ingenious fuzzy-genetic method. Strong stability, robustness, and tracking performance were demonstrated by the recommended approach in both transient and steady states with external disturbances, and chattering issues were successfully reduced.

(Demir & Demir, 2023) An efficient Sliding Mode Control (SMC) technique for quadcopters, that utilizes the (ANFIS) in trajectory tracking, the ANFIS-SMC controller outperformed the traditional SMC controller, with a maximum mean error of in terms of convergence speed, robustness, precision, and stability. The research results presented in this study may be used to create reliable and flexible control schemes for tracking UAV trajectories. These results provide valuable information about how to design more effective and efficient UAV control systems, particularly when adjusting to changing environmental circumstances. As of right now, they have nothing to work on.

The research discussed in this section primarily examines and evaluates the effectiveness of various control system strategies that are employed to manage the position, orientation, and altitude of a quadrotor. The quadrotors X and Y are positioned in an odd way. Because it is based on attitude altitude control and immediately connects to model parametric errors, previous research has primarily concentrated on time-varying disturbances and payload change. Moreover, most wind turbine damage identification techniques use conventional approaches that ignore the YOLO model, the influence of the transformation matrix and trajectory during wind turbine surface damage detection by drone. This study, which draws inspiration from earlier research, suggests creating a PSO-based PID surface-based optimal SMC for managing quadcopter trajectory tracking during wind turbine surface damage detection. The paper includes MATLAB/Simulink simulation tests to demonstrate the effectiveness and success of this strategy.

2.5. Quad rotor drone for Wind Turbine surface damage detection

Figure 2.2 quad rotor for wind turbine inspection(Drone Inspection Guide Benefits & Industries Served, n.d.)

Not only are drones useful for inspecting blades, but they are also being utilized for inspecting other turbine parts, such as nacelles and towers, and for specialized construction-phase tasks like geometrics, site characterization, and figuring out the roughness or forest cover. Among these applications is (Solution & Farm, 2018):

- The process of resource evaluation involves characterizing the forest cover.
- Completion of the topography and mapping of a site occurs before building.

- End-of-warranty checks are carried out in advance of the warranty period's expiration and enable the operator to submit claims.
- Ultimately, met masts and the outside of buildings located at wind farm locations are examined.

2.6. Image Processing

There is still much to learn about the integration of image-based analytics and applications pertinent to wind turbine damage detection, even if deep learning algorithms have demonstrated promise in recent inspection research.

In order to support further research in this field and to motivate other researchers to take on the task of wind turbine damage identification, the whole publically accessible dataset with annotations was made available in. Bounding boxes and the YOLOv5 object identification framework were used in (Foster et al., 2022a) to localize wind turbine blades using this open-source dataset. Their method yielded good findings, but it lacked a comprehensive end-to-end framework that could be adopted by large numbers of workers, particularly in the area of high-throughput phenotyping in wind turbine identification and damage detection. YOLOv8 is used in this thesis' automated wind turbine damage detection system to meet this demand**.**

CHAPTER THREE

3. MODELING OF QUADCOPTER SYSTEM

This chapter covers the research study's techniques, as well as how the quadrotor system was created, how dynamic mathematical modeling using the Newton- Euler- method was done, and how the quadrotor model was finally verified.

3.1. Materials

Software like MATLAB, Microsoft Office, Draw.io, Pycharm and Math Type are used in this thesis. Technical toolboxes and Simulink are components of a computer program known as MATLAB. Draw.io is used to create a wide variety of additional diagram kinds. Math type is used to write mathematical calculations and equations, as well as to annotate images of wind turbines. The online computer vision program Roboflow is utilized.

3.2. Methods

To achieve the goals of this thesis, the following approaches and strategies are used.

- **Literature Review:** A thorough analysis of pertinent literature is carried out to build baseline knowledge in the field of study.
- **Mathematical Modeling and Verification:** It is carried out using the Newton-Euler method to explain quadrotor dynamics. In order to accomplish desired locations through translational dynamics and desired Euler angles through rotational dynamics, mathematical relationships must be formulated and MATLAB software is used to verify the model.
- **Controller Design:** The research then moves on to controller design, where optimal sliding mode controller with a PID surface is created to regulate the quadcopter's movement and allow for accurate trajectory tracking.

- **Image processing and Wind turbine surface damage detection:** The next step is using YOLOv8 in Pycharm to check wind turbine objects through the application of image processing techniques. Automated surface damage detection of wind turbines is made feasible by this feature, which allows surface damage detection based on images taken by quadcopters.
- **Result and Discussion:** The visual data produced by MATLAB simulations makes up the findings. These graphic depictions show how well the system performs in various settings. The focus shifts to understanding the results when these graphical results are shown.
- **Conclusion and Recommendation:** Making thorough inferences from the research findings is the last step. Furthermore, suggestions are furnished for possible improvements or additional inquiries inside the research field, directing subsequent research undertakings

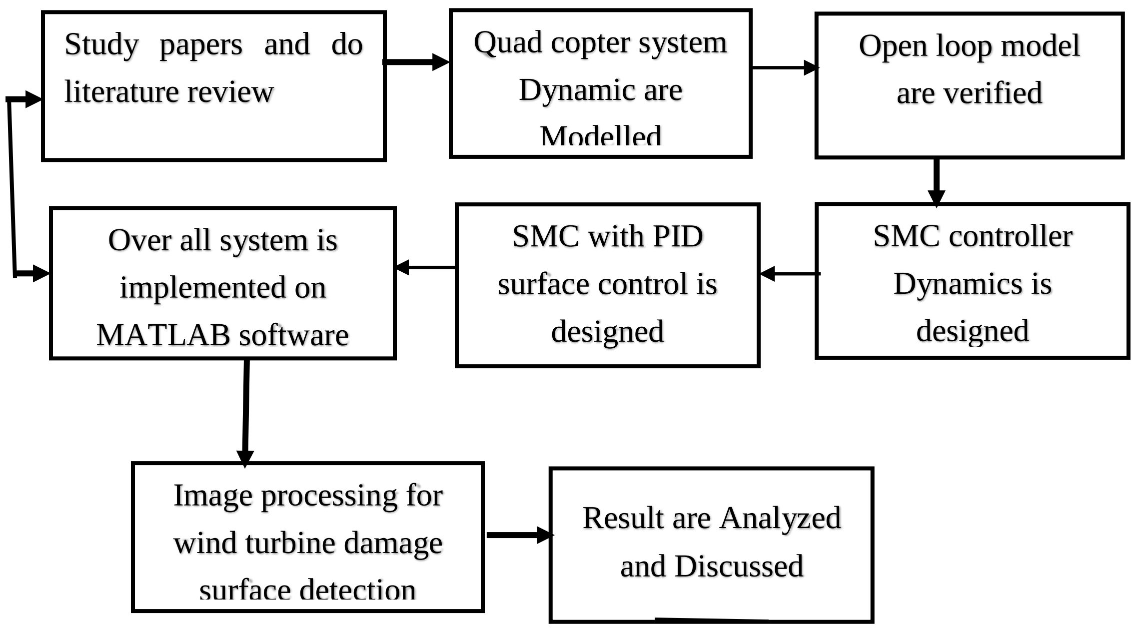


Figure 3.1 Methods follow throughout the thesis

3.3. Modeling Approach for Quadcopter System

Depending on the specific requirements and level of complexity, a quadrotor system can be modeled using several different methods. A typical quadrotor system modeling approach is as follows (Chovancová et al. 2014):

- **Rigid Body Dynamics:** The core of a quadrotor model is its stiff body dynamics. The quadrotor cannot be classified as a rigid body if Newton's equations of motion can be used to explain its motion.
- **Kinematics:** Finding the correlations between the quadrotor's position, orientation, linear velocity, and angular velocity is essential to characterizing its motion.
- **Aerodynamics:** We are able to consider the effects of aerodynamics on the quadrotor by utilizing elementary models. Forces and torques are produced by the propellers' interaction with the surrounding air, which these models then use to drive the quadrotor. Propulsion, lift, and drag are all considered.
- **Control System**: To control a quadrotor, a feedback control system is typically utilized. The control system evaluates the input from sensors (such as accelerometers and gyroscopes) and the desired quadrotor behavior to determine the appropriate control input.
- **Environmental impacts:** The necessity to consider environmental factors like wind, ground impacts, or other external disturbances that can affect the quadrotor's movements may depend on the application and degree of precision required.

Research is conducted on kinematics, rigid body dynamics, environmental impacts, and the control system component using the modeling approaches mentioned above.

3.3.1 Quadcopter Configuration Types

Referred to as the "Plus" and "Cross" configurationsThere are two main configurations that are frequently utilized in quadcopter designs with respect to a vehicle's body coordinates, as shown in Figure 3.2. Whereas the Cross configuration employs two rotors to maneuver the airplane, the Plus layout uses just one.

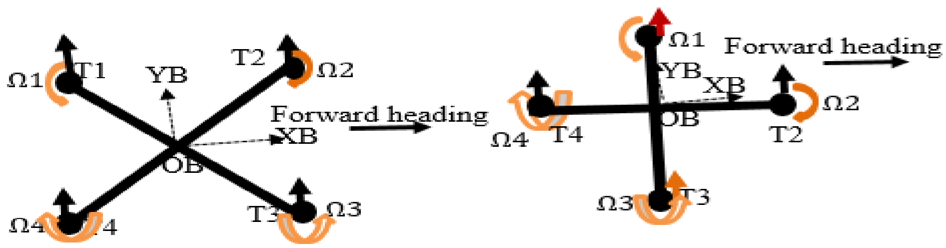


Figure 3.2 Plus and cross configuration of quad rotor

Cross configuration: In terms of the body coordinates of a vehicle, quadcopter designs often employ one of two primary layouts, known as the "Plus" or "Cross" arrangements see figure 3.2. A single rotor powers the aircraft in the. $f_i = b\Omega_i^2$ Two adjacent front rotors (Ω_2 and Ω_3) and two adjacent rear rotors (Ω_1 and Ω_4) slow down and rotate in CW and CCW directions, respectively, which results in nose-down pitching. They produce a small amount of torque, nevertheless. Conversely, as the front rotor speed rises, pitch is regulated and net torque balances, preventing a net yaw moment.

The cross design will be employed for the following investigation due to these two primary benefits:

- A quad copter running in a cross configuration provides more stability and smooth motion in the case of a rotor failure than one operating in a plus Configuration.
- The cross ensures that, when mounted on a quad copter, the camera arc remains unhindered.

3.3.2. Quadcopter Working Principles

Throttle: This is accomplished by raising each of the four rotors' separate speeds by the same amount in synchronization. Motors 2 and 4 spin against each other in a clockwise direction. The rotor delivers the same lift force ($T_1 = T_2 = T_3 = T_4$) since the four motors have similar rotational speeds. Since the torque generated by each arm is the same but directed in the opposite direction, the net torque simply equalizes and then levels out vertically. Now, if the rotation rates of all four motors rise simultaneously, the lift force of the body will also increase.

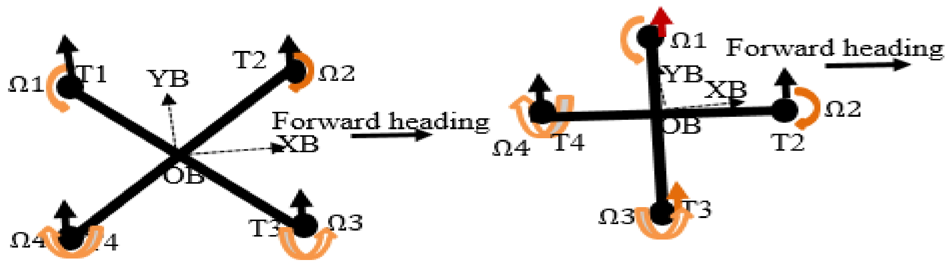


Figure 3.3 Throttle motion of quad-rotor

Roll: By modifying the rotor speeds on the left Ω_1 and the right Ω_3, which only require two rotors to roll, the plus configuration is put into practice. However, the cross-configuration rolls in the following manner by using all of its rotors. This allows the quadcopter body to flip left or right based on the left and right ends of the unequal lift by changing the rotors' Ω_1 and Ω_2 and Ω_3 and Ω_4 speeds, as shown in Figure 3.4

Figure 3.4 Roll motion of quad-rotor for cross and plus configuration

Pitch: As opposed to the cross configuration, which makes use of all four rotors, the plus configuration only employs two of its rotors, one must change the speed of the rotor on the front Ω_2 and the backΩ_4. When the rotor speeds are adjusted by Ω_1 and Ω_4, as well as Ω_2 and Ω_3, the four-wing aircraft's front and rear ends will experience uneven lift, which will cause the quadcopter body to flip to the front and rear.

Figure 3.5 pitch movement of quad-rotor

Yaw: adjusting the rotor's speed in the left, right, and rear directions. Circling the aircraft's central axis of rotation, this motion can be either clockwise or counterclockwise.

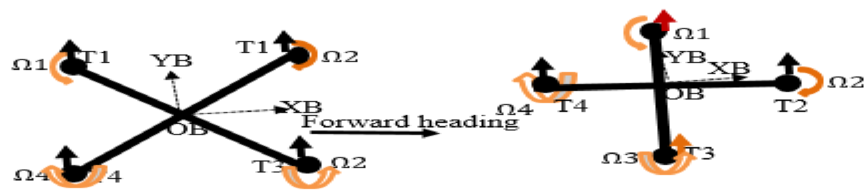


Figure 3.6 yaw movement of quad-rotor

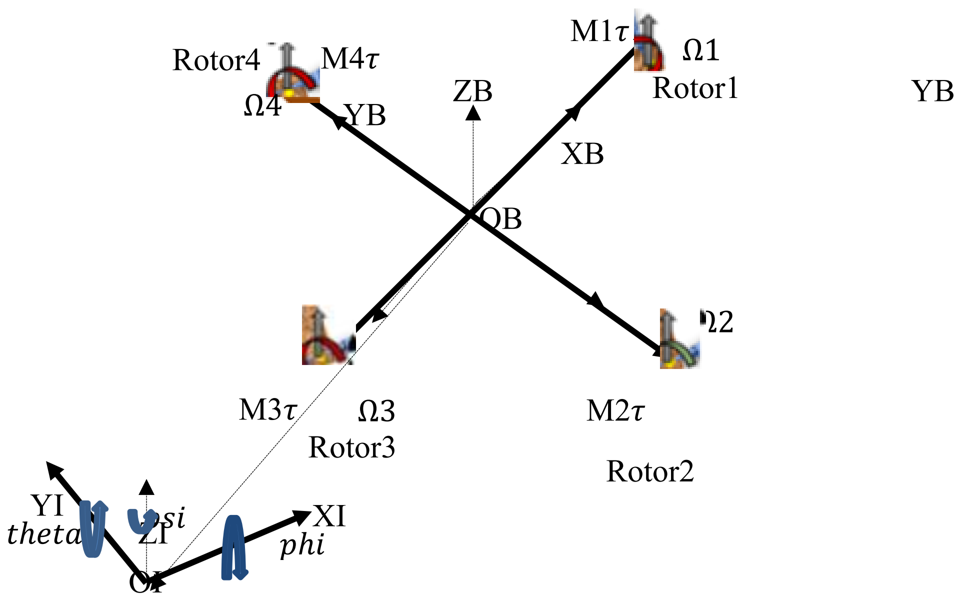


Figure 3.7 Cross quadcopter schematic diagram

Table 3.1 Motor combination for movement of quadcopter

Rotation	Motor1	Motor2	Motor3	Motor4
Up	Boost	Boost	Boost	Boost
Down	Diminish	Diminish	Diminish	Diminish
Hovering	Equal	Equal	Equal	Equal
Left	Diminish	Boost	Boost	Diminish
Right	Boost	Diminish	Diminish	Boost
Forward	Diminish	Diminish	Boost	Boost
Backward	Boost	Boost	Diminish	Diminish
CCC	Boost	Diminish	Boost	Diminish
CC	Diminish	Boost	Diminish	Boost

Table 3.2 Twelve states are practically suitable for a quad copter in 6DOF.

Description	Symbol	Description	Symbol
Placement on x axis	$\boldsymbol{x}$	Yaw rate	$\dot{\boldsymbol{\psi}}$
Placement on y axis	$\boldsymbol{y}$	Roll angle	$\boldsymbol{\phi}$
Placement on z axis	$\boldsymbol{z}$	Pitch angle	$\boldsymbol{\theta}$
Speed along x	$\dot{\boldsymbol{x}}$	Yaw angle	$\boldsymbol{\psi}$
Speed along y	$\dot{\boldsymbol{y}}$	Roll rate	$\dot{\boldsymbol{\phi}}$
Speed along z	$\dot{\boldsymbol{z}}$	Pitch rate	$\dot{\boldsymbol{\theta}}$

3.4 Quadcopter Kinematics

Kinematics is a field of mechanics that examines an item or motion of the system without regard to the forces and torques at play within the framework of six degrees of freedom. It is helpful in simulating the movement of an immobile body.two frameworks of reference must be specified.

3.4.1 Aligning the Reference Frame

Two crucial body and inertial reference frames must be established as points of reference in order to completely understand the behavior of a quadcopter. These two frames are necessary because the quadcopter has numerous sensors installed. Sensors like the accelerometer and gyroscope, which aid in determining both linear and angular velocity, provide measurements with regard to the body frame. However, sensors that offer readings in relation to the inertial frame, like a magnetometer and GPS, make it easier to determine linear and angular position. We choose an anchorage on the surface of Earth to serve as the inertial frame's origin, taking into consideration all torques and forces occurring on the quadcopter.

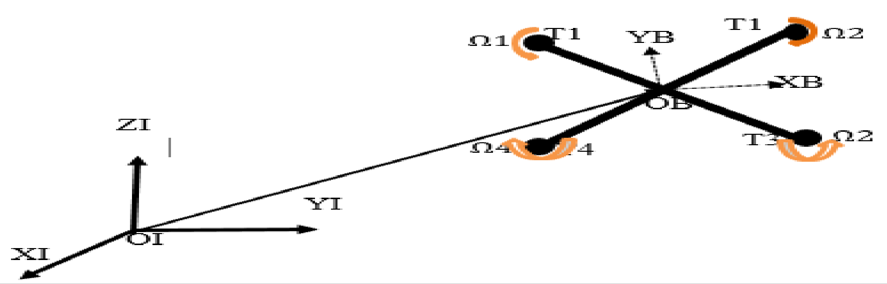


Figure 3.8 Inertial and Body Frames of References

1. Reference frame fixed to the body $(\mathbf{O_B} \quad \mathbf{X_B} \quad \mathbf{Y_B} \quad \mathbf{Z_B})$

- O_B Aligns with the quadcopter's center.
- Linear Velocity: $[u, v, w]^T = v^B$

- $[p, q, r]^T = w^B$Angular Velocity.
- Torque (τ^B) and Force (F^B).

2. **Inertial reference frame of Earth**$(\boldsymbol{O_I} \quad \boldsymbol{X_I} \quad \boldsymbol{Y_I} \quad \boldsymbol{Z_I})$**.**
    - The angular position and the linear position $[\xi^I] = [X, Y, Z]^T$are defined in this frame. Represented by coordinates that are East-North-up.
    - The Earth's fixed point serves as the inertial frame's genesis.
    - Where the vectors work together between O_B and O_I figure out in relation to the inertial frame. The body frame's orientation with relation to the inertial frame is denoted by η^I.

Assumption taken

- As a result of its axis-symmetrical properties, the quadcopter is regarded as a strong and unwavering structure.
- It has features that are axis-symmetrical.
- The analysis fails to take into consideration aerodynamic effects like blade-flapping.
- The torque and thrust, denoted as $\tau i = d\Omega^2$,$Fi = kt\Omega^2$ respectively are exactly Proportionate to the rotor speed squared.
- The quadcopter's center of mass and the body frame originate from the same place.

3.4.2.1 Matrix of Rotations

Rotation matrices are 3 × 3 orthogonal matrices with a determinant of 1. Euler angles, however are the most widely used tool for displaying rotations, quaternions are another useful tool in this thesis, Euler angle representation is used is used since it may be applied without the need for acrobatic motions. These rotations, which start at the Z axis (ϕ, Yaw), circle the Y axis (θ, Pitch), and end at the X axis (ϕ, Roll), are the next steps in the Z-Y-X standard. The rotation matrix (ϕ, θ, ϕ) ∈ R^3 can be broken down to get three matrices that represent rotations along axes (Gedefaw, 2023). An established rotation convention is the intrinsic rotations convention z, x. Pitch, yaw, and roll angles are ψ, ϕ, and θ, respectively. $(\phi, \theta, \psi) = (\phi)Ry(\theta)Rx(\psi)$ (3.1)

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \text{ Is the roll angle rotate with x-axis} \quad (3.2)$$

$$R_y(\theta) = \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix} \text{ Is the pitch angle rotate with y-axis} \quad (3.3)$$

$$R_Z(\psi) = \begin{bmatrix} C_\psi & S_\psi & 0 \\ -S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ Is the yaw angle rotate with z-axis} \quad (3.4)$$

Stated otherwise, the one transformation matrix found in equation 3.5 is the product of the three matrices that represent the trigonometric equations. the drone's orientation and rotation can thus be represented by the matrix, given that the roll, pitch, and yaw rotations are carried out consistently in the previously established order (ϕ, θ, ψ).

$$R_I^B = \begin{bmatrix} C_\psi C_\theta & S_\psi C_\theta & -S_\theta \\ -S_\psi C_\phi + C_\psi S_\theta S_\phi & C_\psi C_\phi + S_\psi S_\theta S_\phi & C_\theta S_\phi \\ S_\psi S_\phi + C_\psi S_\theta C_\phi & C_\psi S_\phi + S_\psi S_\theta C_\phi & C_\theta C_\phi \end{bmatrix} \quad (3.5)$$

These are the rotation matrices that convert the coordinate system of Earth (I) to the body frame (B). In order to solve trigonometric equations, multiply three matrices in the Z-Y-X order.

$$R = R_{z,\psi} x R_{y,\theta} x R_{x,\phi} = \begin{bmatrix} C_\theta C_\phi & -S_\psi C_\phi + C_\psi S_\theta S_\phi & -S_\psi S_\phi + C_\psi S_\theta C_\phi \\ -S_\psi C_\theta & C_\psi C_\phi + S_\psi S_\theta S_\phi & -C_\psi S_\phi + S_\psi S_\theta C_\phi \\ -S_\theta & S_\phi C_\theta & C_\psi C_\theta \end{bmatrix} \quad (3.6)$$

The earth (I) and body frame (B) coordinate systems are transformed via the rotation matrices Analyze how the quadcopter's roll, pitch, and yaw angles differ from the earth fixed frames. A linear quantity with Earthly coordinates is converted to body values using a rotation matrix. By ordering combining these three matrices, the matrix for the inertial to body frame transition can be found Appendix A.1. The order in which things are finished is crucial.

3.4.3 The Angular Velocity Conversion (Transfer matrix)

A gyroscope sensor detects the rotational speed (p, q, r), a time-varying variable, inside an airplane. Such that the angular velocity and Euler angle change rates with respect to the body-fixed frame [ϕ, θ, ψ] differ from one another. The rotating sequence shown in the following diagram governs their relationship.

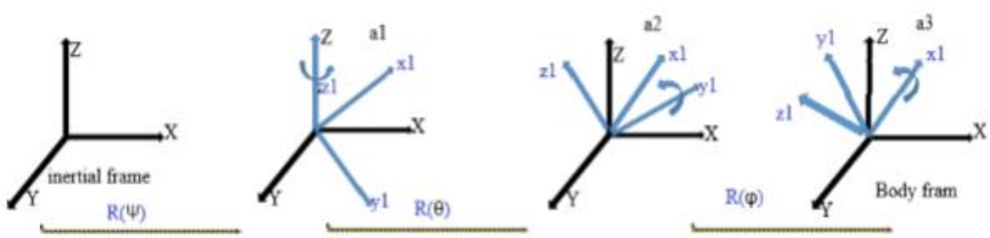


Figure 3.9: Euler Angle Rotations in Sequence (Belsty, 2021)

The Euler angle rate multiplied by a scalar determines the derived gyro rate; this relationship can be expressed in scalar form using the right-hand rule as follows:

$$px_1 + qy_1 + rz_1 = \dot{\psi}a_1 + \dot{\theta}a_2 + \dot{\phi}a_3 \tag{3.7}$$

To write in matrix form. $$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_\phi R_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix}^{a_1} R_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}^{a_2} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}^{a_3} \tag{3.8}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos_\phi & -sin_\psi \\ 0 & sin_\psi & cos_\psi \end{bmatrix} \begin{bmatrix} cos_\theta & 0 & -sin_\theta \\ 0 & 1 & 0 \\ sin_\theta & 0 & cos_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos_\psi & -sin_\psi \\ 0 & sin_\psi & cos_\psi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{3.9}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T = \begin{bmatrix} 1 & 0 & -sin_{\psi} \\ 0 & sin_{\phi} & sin_{\phi}cos_{\theta} \\ 0 & -sin_{\phi} & cos_{\phi}cos_{\theta} \end{bmatrix} \quad (3.10)$$

Euler rate find from body angular velocity using inverse matrix rule

$$T = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{inertia} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & sin_{\phi}T_{\theta} & cos_{\phi}T_{\theta} \\ 0 & cos_{\phi} & -sin_{\phi} \\ 0 & -sin_{\phi}sec_{\theta} & cos_{\phi}sec_{\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}_{bodyframe} \quad (3.11)$$

In this case, the rates of roll, pitch, and yaw are denoted as (p, q, r) and $s = sin, c = cosine\ angle.$

3.5. Quadcopter Dynamics

3.5.1. Newton Euler Model Approach

used to generate equations of motion for the six-degree-of-freedom system's parameters for translation and rotation. One cannot accomplish translational and rotational representation without the vectors of location, orientation, angular velocity, and linear velocity. Next, we simulate the quadcopter's flight under various circumstances to see how the mathematical model performs.

3.5.2 Analysis of Force and Momentum

Let the unit vectors in the inertial set along the direction $b = [b_1 \quad b_2 \quad b_3]$ It has an angular velocity instantaneous $\omega = [p \quad q \quad r]$. With regard to body frame, we wish to compute the derivative of the linear velocity vector v.

$$\begin{gathered} \sum F = ma \\ F = m\frac{d_v}{d_t} \\ F = m\left(\frac{d_u}{d_t}b_1 + \frac{db_1}{d_t}u + \frac{d_v}{d_t}b_2 + \frac{db_2}{d_t}v + \frac{dw}{d_t}b_3 + \frac{db_3}{d_t}w\right) \\ F = m\left(\frac{d_u}{d_t}b_1 + \frac{d_v}{d_t}b_2 + \frac{d_w}{d_t}b_3 + \right) + m\left(\frac{db_1}{d_t}u + \frac{db_2}{dt}v + \frac{db_3}{d_t}w\right) \end{gathered} \quad (3.12)$$

$$
\begin{gathered}
F = ma = m\frac{d_v}{d_t} \\
Fx = m(\dot{u} - v_r + w_q) \\
Fy = m(\dot{v} + w_p - v_r) \\
Fz = m(\dot{w} + u_q - v_p)
\end{gathered}
\tag{3.13}
$$

Rewrite as follows in matrix form. The appendix A.2 has the detailed derivation**.**

$$
F = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & m_w & -m_v \\ -m_w & 0 & m_u \\ m_v & -m_u & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.14}
$$

The second shows how coordinate frame rotation and velocity relate to one another. This formula can be used to obtain Coriolis' Theorem (see appendix A.2)

Frame of Inertial Reference:

Thrust Force: The rotation matrix, it is possible to characterize the thrust force (FT) in the Inertial frame using this initial specification from the Body frame. The quadcopter's thrust force is a representation of the total push produced by its propellers $Fi = kt\Omega^2$

$$
F_T^B = F_{T1} + F_{T2} + F_{T3} + F_{T4} = kt({\Omega_1}^2 + {\Omega_2}^2 + {\Omega_3}^2 + {\Omega_4}^2) \tag{3.15}
$$

Where kt, the propeller thrust coefficient, represents the quadratic relationship between the force and angular velocity (Ω).

Gravity in Body Frame: The following formulas in the Newton-Euler formalism express the low leads. A system's total force is equal to the product of its mass and acceleration, as per Newton's law.

$$F_T^I = {}_b^iR(F_T^b)F_T^I \;=\; \begin{bmatrix} C\phi S\theta S\psi + S\phi S\psi \\ C\phi S\theta S\psi - C\psi S\phi \\ C\phi C\theta \end{bmatrix} \begin{bmatrix} F_T^b \\ F_T^b \\ F_T^b \end{bmatrix} \quad F_g^n = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.16}$$

$$F_g^b = C_n^b F_g^n = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{c}\theta\mathrm{s}\psi & -\mathrm{s}(\theta) \\ -\mathrm{c}\phi\mathrm{s}\psi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\theta\mathrm{s}\phi\mathrm{s}\psi & \mathrm{c}\theta\mathrm{s}\phi \\ \mathrm{s}(\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta & -\mathrm{s}\phi\mathrm{c}\psi + \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.17}$$

After multiplication $F_g^b = \begin{bmatrix} -mg\sin\theta \\ mg\sin\phi\cos\theta \\ mg\cos(mg\cos\phi\cos\theta) \end{bmatrix}$ (3.18)

$applying\ Newton\ second\ law\ which\ is\ F = ma\ with\ in\ the\ body\ frame$

$$\begin{gathered} \sum F = ma \\ \frac{dv}{dt} = m\frac{d_{vi}}{d_t}, v\ is\ the\ sum\ of\ velocity\ in\ x, y, z \\ \frac{dv}{dt} = m\left(\frac{d_u}{d_t}b_1 + \frac{db_1}{d_t}u + \frac{d_v}{d_t}b_2 + \frac{db_2}{d_t}v + \frac{dw}{d_t}b_3 + \frac{db_3}{d_t}w\right) \\ \frac{dv}{dt} = m\left(\frac{d_u}{d_t}b_1 + \frac{d_v}{d_t}b_2 + \frac{d_w}{d_t}b_3 +\right) + m\left(\frac{db_1}{d_t}u + \frac{db_2}{dt}v + \frac{db_3}{d_t}w\right) \end{gathered} \tag{3.19}$$

$Coordinate\ axis\ constance\ in\ the\ inertaial\ frame$

$$\frac{db1}{dt} = \frac{db2}{dt} = \frac{db3}{dt} = 0, \frac{dv}{dt} = \frac{dub1}{dt} + \frac{dvb2}{dt} + \frac{dwb3}{dt} \tag{3.20}$$

Applying the chain rule is important when working with spinning bodies; Appendix A.3 provides more information on this.

$$\frac{dv}{dt} = \frac{dvxb1}{dt} + \frac{dvyb2}{dt} + \frac{dvzb3}{dt} + (w_x b_1)v_x + (w_x b_2)v_y + (w_x b_3)v_z \tag{2.21}$$

The following force operates within the body frame:

$$\sum^B F = m(\frac{dvxb1}{dt} + \frac{dvyb2}{dt} + \frac{dvzb3}{dt}) + m((w_x b_1)v_x + (w_x b_2)v_y + (w_x b_3)v_z \tag{2.22}$$

A moving item inside a rotating reference frame that with respect to an inertial frame experiences a pseudo force known as the Carioles force, is represented by the formula $m\,(w_x b_1)v_x + m(w_x b_2)v_y + m(w_x b_3)v_z$ the following mathematical expression describes it:$Fcoroloies = \ m(w\big(v_x b_1 + v_y b_2 + v_z b_3\big)\ after\ cross\ multiplication$

$$\sum FT^B + \sum Fg^B = ma, \tag{3.23}$$

$$\sum F^B = \begin{bmatrix} F_x^B \\ F_y^B \\ F_z^b \end{bmatrix} \begin{bmatrix} m(a_x + (\Omega_\theta v_z - \Omega_\psi v_y)) \\ m(a_y + (\Omega_\psi v_x - \Omega_\phi v_y)) \\ m(a_z + (\Omega_\phi v_y - \Omega_\theta v_x)) \end{bmatrix} \quad (3.24$$

$Which\ FT^B and\ Fg^B\ represente\ thrust\ force\ and\ gravitational\ force\ in\ body\ frame$

$$\begin{bmatrix} 0 \\ 0 \\ F_{T1} + F_{T2} + F_{T3} + F_{T4} \end{bmatrix} + \begin{bmatrix} -mg\sin\theta \\ mg\sin\phi\cos\theta \\ mg\cos(mg\cos\phi\cos\theta) \end{bmatrix} = \begin{bmatrix} m(a_x + (\Omega_\theta v_z - \Omega_\psi v_y)) \\ m(a_y + (\Omega_\psi v_x - \Omega_\phi v_y)) \\ m(a_z + (\Omega_\phi v_y - \Omega_\theta v_x)) \end{bmatrix} \tag{3.25}$$

Divide both sides by mass explains translational motion within the frame of the body

$$dvx/dt = g sin\theta + \Omega_\psi v_y - \Omega_\theta v_z \tag{3.26}$$

$$\mathrm{dvy/dt} = -\mathrm{gcos\theta sins\theta} - \Omega_\psi v_x + \Omega_\phi v_y \tag{3.27}$$

$$\mathrm{dvz/dt} = -\mathrm{cos\theta cos}\phi - \Omega_\phi v_y + \Omega_\theta v_x \tag{3.28}$$

Thrust Force: To represent the thrust force in the inertial frame, utilize the rotation matrix. It is at the body frame that the thrust force first becomes noticeable.

$$F_T^I = R_B^I(F_T^I, \sum F^I = F_{T+}^I F_{gma}^I = \left[\begin{bmatrix} C\phi S\theta S\psi + S\phi S\psi \\ C\phi S\theta S\psi - C\psi S\phi \\ C\phi C\theta \end{bmatrix} \begin{bmatrix} F_T^b \\ F_T^b \\ F_T^b \end{bmatrix} \right] \tag{3.29}$$

Gravitational force: In terms of the inertial frame, the gravitational force is described. Using the second law of Newton (Gedefaw, 2023), $F = ma$, inside the framework of the inertial reference frame yields the following equation: $\sum F^I = \quad F_T^I + \quad F_g^I = ma$ (3.32)

$$\sum F^I = \left[\begin{bmatrix} C\phi S\theta S\psi + S\phi S\psi \\ C\phi S\theta S\psi - C\psi S\phi \\ C\phi C\theta \end{bmatrix} \begin{bmatrix} F_T^b \\ F_T^b \\ F_T^b \end{bmatrix}\right] + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \tag{3.31}$$

Dividing by mass both sides $U1 = \quad (F_T^B)\ the\ equation\ become$ (3.32)

$$\ddot{x} = [cos\phi sin\theta sin\psi + sin\phi sin\psi]\frac{u1}{m} \tag{3.33}$$

$$\ddot{y} = [cos\phi sin\theta sin\psi - cin\psi sin\phi]\frac{u1}{m} \tag{3.34}$$

$$\ddot{z} = [cos\phi cosin\theta]\frac{u1}{m} - g \tag{3.35}$$

3.5.3 Examination of Torque

The combined gyroscopic and thrust torques apply the torque to the quadrotor.

Moments of the thrust: The torques and forces that are directly produced by the main motion inputs are considered in the third contribution. The squared propeller speed is negatively correlated with torques and forces, as shown by aerodynamic studies.

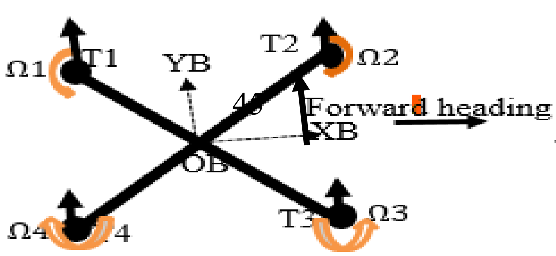


Figure 3.10 The UAV's x-configuration far away the next propellers

Determine the distance (x) between the quadrotor's center and the propeller's middle(Gedefaw, 2023). Using the law of sin $\sum \tau_x^B = F1lcos45 - F2lcos45 - F3lcos45 + F4lcos45$ (3.36)

$$=\sum \tau_x^B = 1\frac{(\sqrt{2})}{2}(F_1 - F_2 - F_3 + F_4) \tag{3.37}$$

$$\sum \tau_Y^B = -F1lsin45 - F2lsin45 + F3lsin45 + F4lsin45 \tag{3.38}$$

$$=\sum \tau_y^B = 1\frac{(\sqrt{2})}{2}(F_1 - F_2 - F_3 + F_4) \tag{3.39}$$

We consider $U = [U_1 \quad U_2 \quad U_3 \quad U_4]^2$ as the four rotorcraft's input for control. This produces a lift force, which is also known as the altitude control input, which is represented by the symbol. The force that modifies the vehicle's pitch angle and, in turn, the quadcopter's yaw angle is what follows in changing the roll angle of the aircraft. The input of the control U CW (clockwise) directions are perceived as negative, whereas CCW (counterclockwise) directions are perceived as positive, states this hypothesis.

$$\sum_x^b \tau = lkt/\sqrt{2}(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \tag{3.40}$$

$$\sum_y^b \tau = lkt/\sqrt{2}(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{3.41}$$

$$\sum_z^b \tau = ld/\sqrt{2}(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{3.42}$$

$$U2 = \frac{\mathrm{lkt}}{\sqrt{2}}(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \tag{3.43}$$

$$U3 = \frac{kt_l}{\sqrt{2}}(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{3.44}$$

$$U4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \tag{3.45}$$

$$\Omega_r = -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 \tag{3.46}$$

The propeller speeds are indicated by l, this is how far the propeller center is from the quadrotor. The thrust factor is represented by kt, the drift factor by d. Ω_i the propeller's speed. Here are some methods to increase the vertical force using the three axes of the body coordinate system:

$$F^B = \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}, F_i = b\Omega_i^2 \tag{3.47}$$

The rotational torque will cause the quadcopter to roll, pitch, and yaw while it is in the air, changing its attitude. The precise torques for roll, yaw, and pitch are expressed as

$$\Lambda = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix}_{propeller} = \begin{bmatrix} 0 \\ 0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{3.48}$$

$$\sum F = m\dot{\Gamma} + \omega^B X v^B = F_g + F_{gro} + F_{prope} \;, V = [u \quad v \quad \omega \quad p \quad q \quad r]^T \tag{3.49}$$

Gyroscopic Impacts: The propellers on two of the quadcopters, numbered 2 and 4, rotate in a clockwise direction, while numbers 1 and 3, which are located on the opposite arm, rotate in a counterclockwise direction. The roll and pitch speeds and the gyroscopic effect brought on by this imbalance are directly correlated. The generic gyroscopic torque equation has the following definition (Belsty, 2021):

$$F_{gro}^b = \mathrm{J} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ p & -p & p & -p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega_s = \mathrm{J} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -q \\ p \end{bmatrix} \Omega_r \tag{3.50}$$

$$\Omega_s = [\Omega_1 \quad \Omega_2 \quad \Omega_3 \quad \Omega_4]^T$$

$$\Omega_r = -\Omega_1 \quad \Omega_2 \quad -\Omega_3 \quad \Omega_4$$

$$\tau g = J\alpha g = J(\omega X\Omega), where\ \omega = [0\ 0\ \omega]^T, \Omega = [\Omega\phi\ \Omega\theta\ \Omega\psi]^T \tag{3.51}$$

$$\tau g = (\begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} X \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix}) = \mathrm{J} \begin{bmatrix} \Omega\phi\omega \\ \Omega\theta\omega \\ 0 \end{bmatrix} \tag{3.52}$$

$$\tau_{gtotal} = \tau_{g4} + \tau_{g3} + \tau_{g2} + \tau_{g1}, \omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{3.53}$$

Is a picture of the entire moment of inertia during rotation centered on the propeller axis J, Ω_r the propellers' overall speed Refer to appendix A.3 for the derivation.

3.5.4. Euler Angle Equation's Momentum (Inertia Tensor)

I stand for the moment of inertia and momentum $= lxw$. The momentum change rate at the inertial frame equals the spinning torque on the body frame. $\tau = dl/dinertia$ Making Use of Coriolis' Theorem Once more, to determine the rotational Mobility for the angular velocity time derivative, utilize the formula below (Belsty, 2021)**:**

$$\tau = \frac{d}{d_t}[I_{xp} \quad I_{yq} \quad I_{zr}]^T + [p \quad q \quad r]^T x [I_{xp} \quad Y_{yq} \quad I_{zr}]^T \tag{3.54}$$

Λ can be defined as the generalized force vector and can be expressed as

$$\wedge = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} = [F_x \quad F_y \quad F_z \quad \tau_x \quad \tau_y \quad \tau_z]^T \tag{3.55}$$

$$\wedge = \begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{\omega} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & m\omega & -mv \\ 0 & 0 & 0 & -m\omega & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_z r & -I_y q \\ 0 & 0 & 0 & -I_z r & 0 & I_x q \\ 0 & 0 & 0 & I_y q & I_y p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \\ p \\ q \\ r \end{bmatrix} \tag{3.56}$$

Three distinct components (marked by Λ) can be identified based on the type of quadrotor force contributions. $\sum F = ma = F_g + F_{gro}$ (3.57)

When every propeller rotates, the first force is produced; when the primary movement inputs directly produce the force and torque, the second and third forces are generated. First among forces is gravity. The inertia tensor formulation for a rigid body is,

$$I = \begin{bmatrix} Ixx & -Ixy & -Ixz \\ -Iyx & Iyy & -Iyz \\ -Izx & -Izy & Izz \end{bmatrix} \tag{3.58}$$

Ixx: The moment of inertia an object experiences as it is rotated around the X axis is represented by this symbol. Ixy denotes the moment of inertia on the Y-axis for objects orbiting the X-axis. We get $yz = Izy$, $Ixz = Izx$. The values of Ixy, Ixz, and Iyz for geometrically symmetric objects, are all equal to zero, just like in a conventional multicopter (Gedefaw, 2023). Following that, equation 3.35 becomes $\sum \tau = \tau g + \tau T$.

$$\tau^b = \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix} x \begin{bmatrix} Ixx & 0 & 0 \\ 0 & Iyy & 0 \\ 0 & 0 & Izz \end{bmatrix} + \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix} \begin{bmatrix} Ixx(\dot{\Omega\phi}) + \Omega\phi\Omega\theta(\mathrm{Izz} - \mathrm{Iyy}) \\ Iyy(\dot{\Omega}\theta) + \Omega\phi\Omega\psi(Ixx - Izz) \\ Izz(\Omega\psi) + \Omega\psi\Omega\theta(\mathrm{Iyy} - \mathrm{Ixx}) \end{bmatrix} + \begin{bmatrix} Ixx & 0 & 0 \\ 0 & Iyy & 0 \\ 0 & 0 & Izz \end{bmatrix} \begin{bmatrix} \dot{\Omega\phi} \\ \dot{\Omega}\theta \\ \dot{\Omega}\psi \end{bmatrix} \tag{3.59}$$

$$\begin{bmatrix} U1 \\ U2 \\ U3 \end{bmatrix} + JT \begin{bmatrix} \Omega\theta(\Omega r) \\ \Omega\phi(-\Omega r) \\ 0 \end{bmatrix} = \begin{matrix} Ixx(\dot{\Omega\phi}) + \Omega\phi\Omega\theta(\mathrm{Izz} - \mathrm{Iyy}) \\ [Iyy(\dot{\Omega}\theta) + \Omega\phi\Omega\psi(Ixx - Izz)] \\ Izz(\Omega\psi) + \Omega\psi\Omega\theta(\mathrm{Iyy} - \mathrm{Ixx}) \end{matrix} \tag{3.60}$$

Then , $\dot{\Omega\phi} = 9u2 + Jt\Omega\theta(\Omega r) - \Omega\psi\Omega\theta(Izz - Iyy))/Ixx$ (3.61)

$$\dot{\Omega\theta} = (u3 + Jt\Omega\phi(-\Omega r) - \Omega\phi\Omega\psi(Ixx - Izz))/Iyy \tag{3.62}$$

$$\Omega\dot{\psi} = (u4 - \Omega\phi\Omega\theta(Iyy - Ixx))/Izz \tag{3.63}$$

Inertial Reference Frame: Dynamics in the inertial reference frame are described by means of the transfer matrix.

$$\begin{bmatrix} \Omega\dot{\phi} \\ \Omega\dot{\theta} \\ \Omega\dot{\psi} \end{bmatrix} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & tan\theta & cos\phi/cos\theta \end{bmatrix} \begin{bmatrix} \Omega\dot{\phi} \\ \Omega\dot{\theta} \\ \Omega\dot{\psi} \end{bmatrix} \tag{3.64}$$

Transpose matrix $T = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & tan\theta & cos\phi/cos\theta \end{bmatrix}$ (3.65)

The small angle approximation states that cosθ is about equal to one, but $sin\theta$ and $tan\theta$ become almost zero at tiny angles θ. For ϕ and Four-person helicopter Features: Four control inputs, symbolized by the vector $[U1, U2, U3, U4]^T$ have an impact on the quad copter. $U1$ regulates altitude specifically, which establishes the lift force. $U2$ influences the roll angle of the quadcopter, $U3$ impacts the pitch angle of it, and $U4$ modifies its yaw angle. We can see translational and rotational equation below

$$\ddot{x} = (\sin\theta\cos\phi\cos\psi + \sin\phi\sin\psi)\frac{\mathrm{U1}}{m} \tag{3.66}$$

$$\ddot{y} = (\sin\theta\cos\phi\cos\psi - \sin\phi\cos\psi)\frac{\mathrm{U1}}{m}$$

$$\ddot{z} = \cos\phi\cos\theta\frac{\mathrm{U1}}{m} - g$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T^{-1}\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \qquad \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{(I_{yy}-I_{zz})}{I_{xx}}\dot{\theta}\dot{\psi} - \frac{J_T}{I_{xx}}\dot{\theta}\Omega_r + \frac{U_2}{I_{xx}} \\ \frac{(I_{zz}-I_{xx})}{I_{yy}}\dot{\phi}\dot{\psi} - \frac{J_T}{I_{yy}}\dot{\phi}\Omega_r + \frac{U_3}{I_{yy}} \\ \frac{(I_{xx}-I_{yy})}{I_{zz}}\dot{\theta}\dot{\phi} + \frac{U_4}{I_{zz}} \end{bmatrix} \tag{3.67}$$

The relationship between the rotor's angular velocity and the control inputs is inverse and can be stated as follows. Based on the control inputs, the propeller speeds can be determined using the following inverse relationship.

$$\begin{aligned}
\Omega_1 &= 0.5\sqrt{\frac{\mathrm{U1}}{kt} + \frac{\sqrt{2}}{ktl}\mathrm{U2} + \frac{\sqrt{2}}{ktl}U_3 + \frac{\mathrm{U4}}{d}} \\
\Omega_2 &= 0.5\sqrt{\frac{\mathrm{U1}}{kt} + \frac{\sqrt{2}}{ktl}\mathrm{U2} - \frac{\sqrt{2}}{ktl}U_3 - \frac{\mathrm{U4}}{d}} \\
\Omega_3 &= 0.5\sqrt{\frac{\mathrm{U1}}{kt} - \frac{\sqrt{2}}{ktl}\mathrm{U2} - \frac{\sqrt{2}}{ktl}U_3 + \frac{\mathrm{U4}}{d}} \\
\Omega_4 &= 0.5\sqrt{\frac{\mathrm{U1}}{kt} - \frac{\sqrt{2}}{ktl}U_2 + \frac{\sqrt{2}}{ktl}\mathrm{U3} - \frac{\mathrm{U4}}{d}}
\end{aligned} \tag{3.68}$$

The quadcopter, an underactuated system, has six degrees of freedom, denoted by $x, y, z, \phi, \theta \; and \; \psi$. It is controlled by four input variables, $U1, U2, U3 \; and \; U4$. Converting the underactuated system to a fully-actuated one is necessary to achieve autonomous control. U1 represents the thrust force in the body frame. it is translated into the inertial frame by means of a rotating matrix.

$$\begin{bmatrix} Ux \\ Uy \\ Uz \end{bmatrix} = \begin{bmatrix} C_\theta C_\phi & -S_\psi C_\phi + C_\psi S_\theta S_\phi & -S_\psi S_\phi + C_\psi S_\theta C_\phi \\ -S_\psi C_\theta & C_\psi C_\phi + S_\psi S_\theta S_\phi & -C_\psi S_\phi + S_\psi S_\theta C_\phi \\ -S_\theta & S_\phi C_\theta & C_\psi C_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.69}$$

$$\text{Vector's magnitude} \begin{bmatrix} Ux \\ Uy \\ Uz \end{bmatrix} = \begin{bmatrix} (\sin\theta\cos\phi\cos\psi + \sin\phi\sin\psi)\mathrm{U1} \\ (\sin\theta\cos\phi\cos\psi - \sin\phi\cos\psi)\mathrm{U1} \\ \cos\phi\cos\theta U1 - mg \end{bmatrix} \tag{3.70}$$

$$U1 = \sqrt{ux^2 + uy^2 + (uz + mg)^2} \quad \ddot{x} = \frac{Ux}{m}, \ddot{y} = uy/m, \frac{Uz}{m} - g = \ddot{z} \tag{3.71}$$

Desired roll and pich angle: See the derivation on Appendix A.1

$$\psi_d = tan^{-1}(c\theta d\,\frac{Uxs\psi - Uyc\psi}{Vz+g}) \text{ and } \theta_d = tan^{-1}(\frac{Uxc\psi + Uys\psi}{Vz+g}) \tag{3.72}$$

3.5.6. State Space Representation Equation of Motion

Placing the resulting representation of mathematics in a state space form facilitates the application of control methods. These variables consist of position, velocity, direction, and angular velocity. The position's x, y, and z coordinate the roll, pitch, and yaw degrees of orientation, as well as their corresponding the state variables that comprise the quadcopter's state are called derivatives. (Engineering, 2022). As opposed to X, which is the state vector, u is the control input vector.

$$\dot{X} = f(x,u) = \begin{array}{c} \mathrm{x1} = x \\ \mathrm{x2} = \dot{x} \\ \mathrm{x3} = y \\ \mathrm{x4} = \dot{y} \\ \mathrm{x5} = z \\ \mathrm{x6} = \dot{z} \\ \mathrm{x7} = \phi \\ \mathrm{x8} = \dot{\phi} \\ \mathrm{x9} = \theta \\ \mathrm{x10} = \dot{\theta} \\ \mathrm{x11} = \psi \\ \mathrm{x12} = \dot{\psi} \end{array} \tag{3.73}$$

State vector $x = (\mathrm{x1}, \mathrm{x2}, \mathrm{x3}, \mathrm{x4}, \mathrm{x5}, \mathrm{x6}, \mathrm{x7}, \mathrm{x8}, \mathrm{x9}, \mathrm{x10}, \mathrm{x11}, \mathrm{x12})^T \in R^{12}$ (3.74)

The entire mathematical model represented in state space. (3.75)

$$\dot{X}_1 = X_2 = \frac{Ux}{m}, \dot{X}_3 = X_4 = \frac{U_y}{m}, \dot{X}_5 = X_6 = U_z/m$$

$$\dot{X}_7 = X_8 = b_1 U_2 + c_2 X_{10} wr + c_1 X_{10} X_{12}$$

$$\dot{X}_9 = X_{10} = b_2 U_3 - c_4 X_8 wr + c_1 X_8 X_{12}$$

$$\dot{X}_{11} = X_{12} = \quad b_3 U_4 + c_5 X_8 X_{12}$$

$$c1 = \frac{Iyy - Izz}{Ixx}, c3 = \frac{Izz - Ixx}{Iyy}, c5 = \frac{Ixx - Iyy}{Izz}, c4 = \frac{J}{Ixx}, c2 = \frac{J}{Iyy}, b1 = \frac{J1}{Ixx}, b_2 = \frac{1}{Iyy}, b_3 = \frac{1}{Izz}$$

The virtual loop (x and y dynamics) supplies the phi and theta reference or projected trajectory using Ux and Uy . The developed controllers' anticipated roll and pitch trajectories are calculated by applying equation (3.68). Ux and Uy.

3.6. Model Verification

Nonlinear equations were used in the development of the Simulink model, which makes that the quadcopter abides by physical limitations. It is possible to validate the quadrotor model using the control signals from the altitude and attitude inputs$U1, U2, U3, U4$ using the specifications given in the table below. In the visualization, the Simulink scope block is utilized. Determine the propeller speeds using the control inputs. Next, use the speed data to investigate the quadcopter's actual dynamics. The plot's scope is the same. The quadcopter tracks and compares changes in its state's behavior using altitude x, y, and z and attitude phi, theta and psi.

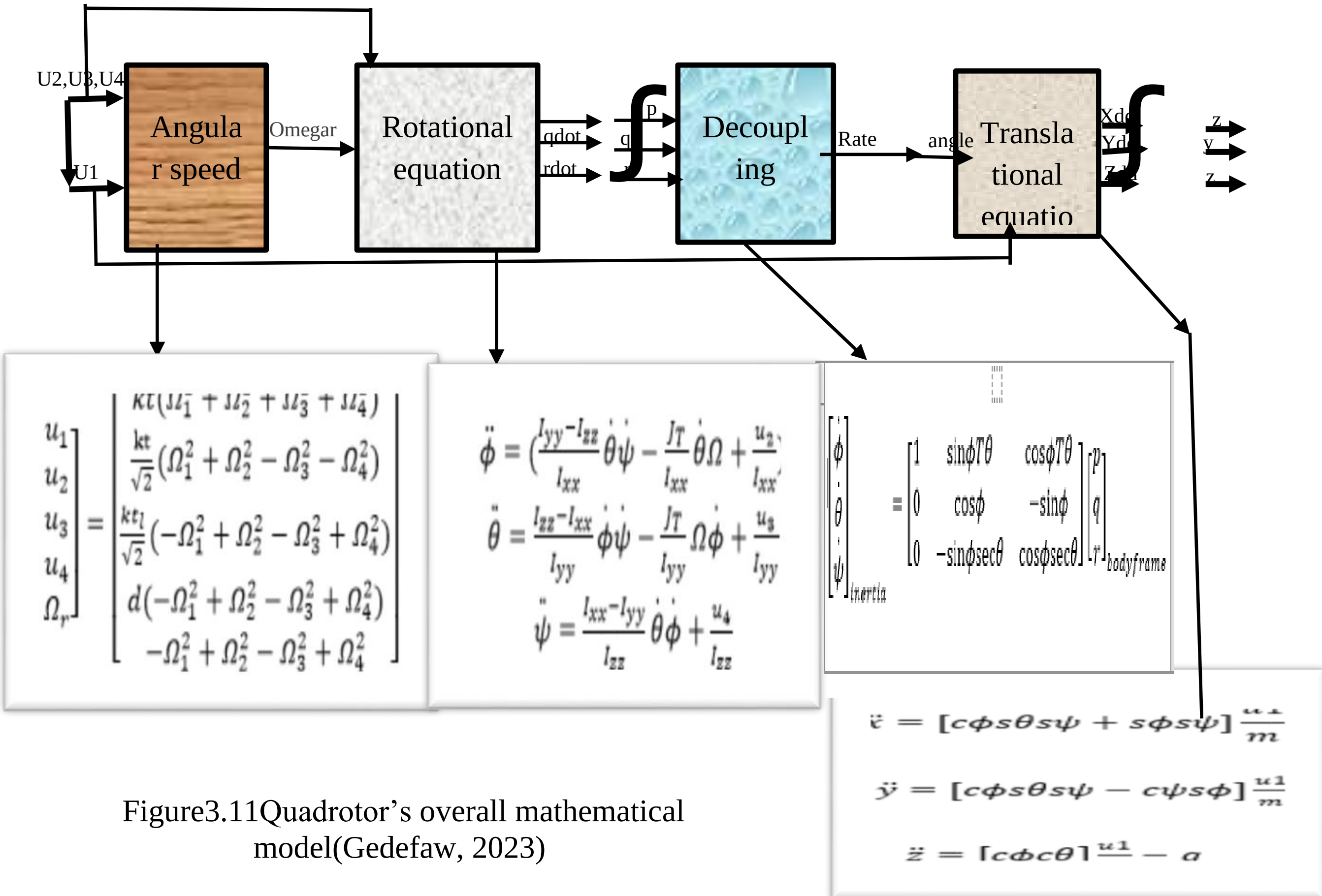


Figure3.11Quadrotor's overall mathematical model(Gedefaw, 2023)

Table 3.3 Characteristics of the quadcopter (Belsty, 2021)

Parameters	Symbol	Value
Gravity	g	9.81 m/sec^2
Drug coefficient	d	0.0000542Nm(rad/s)
Thrust coefficient	kt	0.0000011 N(rad/sec)
Propeller inertia	J	0.000104 Kg/m^2
Propeller length	l	0.24m
Mass	m	1kg
Inertia along x	Ixx	0.0081 Nm(rad/sec^2)
Inertia along y	Iyy	0.0081 Nm(rad/sec^2)
Inertia along z	Izz	0.024 Nm(rad/sec^2)

Hovering condition: The quadcopter hangs there if the thrust value (T=mg) is kept Constant at that altitude. The quadcopter is currently six meters above the ground in a hover. $U_2 = U_3 = U_4 = 0Nm$. In the example, and U_1= mg=9.81 N. The calculated values for the angular velocities are $\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4 = 212.7rad/s$. Every angle stays in its zero position.

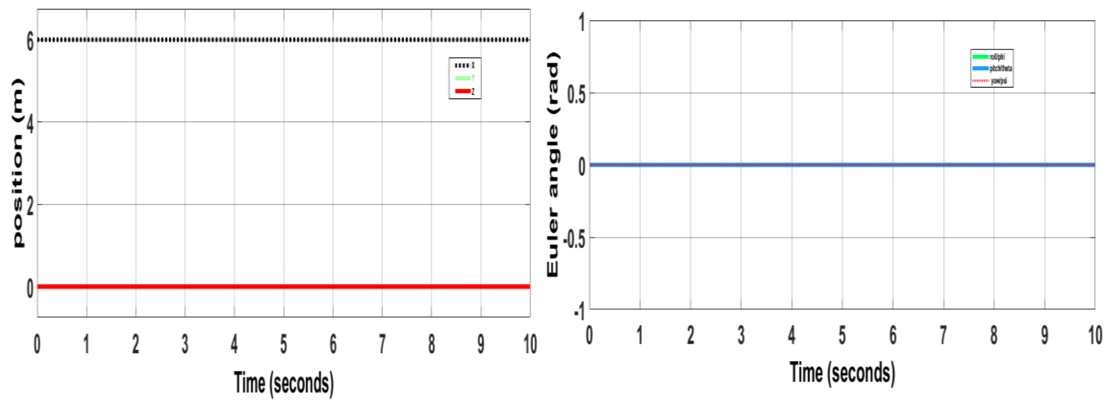


Figure 3.12 position and angle at hovering state

Ascending Action: The vehicle moves in an upward direction along the positive Z-axis when the thrust (T) is higher than the gravitational force (mg). For example, if the quadcopter starts at zero meters above the ground, and we have $U_1 = 15N, U_2 = U_3 = U_4 = 0Nm$, then the observed angular velocities $\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4 = 263rad/s$. Every single angle remains at zero degrees.

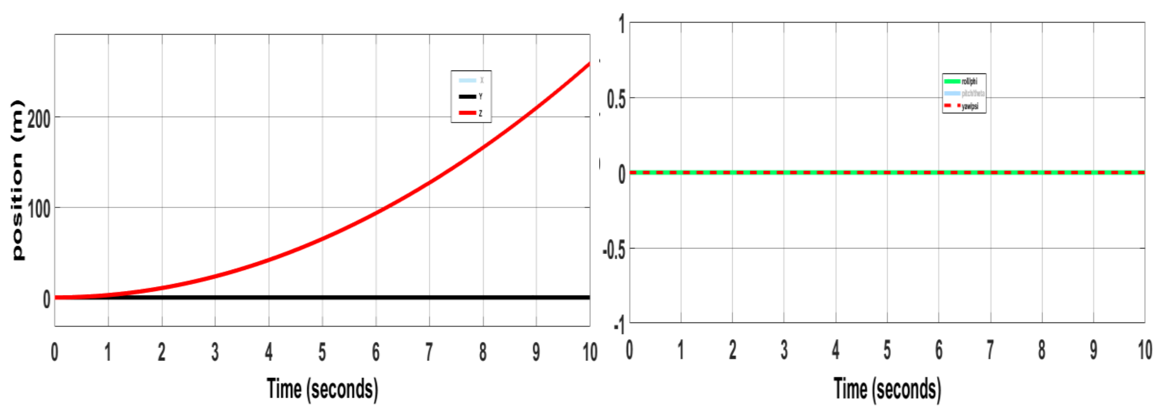


Figure 3.13 position and angle for ascending action

Descending Motion: Subtracting the gravitational force (mg) from the thrust quantity (T) causes the vehicle to sink along the negative Z-axis. Using $U_1 = 7N, U_2 = U_3 = U_4 = 0Nm$, and the quadcopter's initial altitude set at 5 meters above the ground, for example, the recorded angular velocities are $\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4 = 151.9 rad/s$. Every angle stays in its zero-degree.

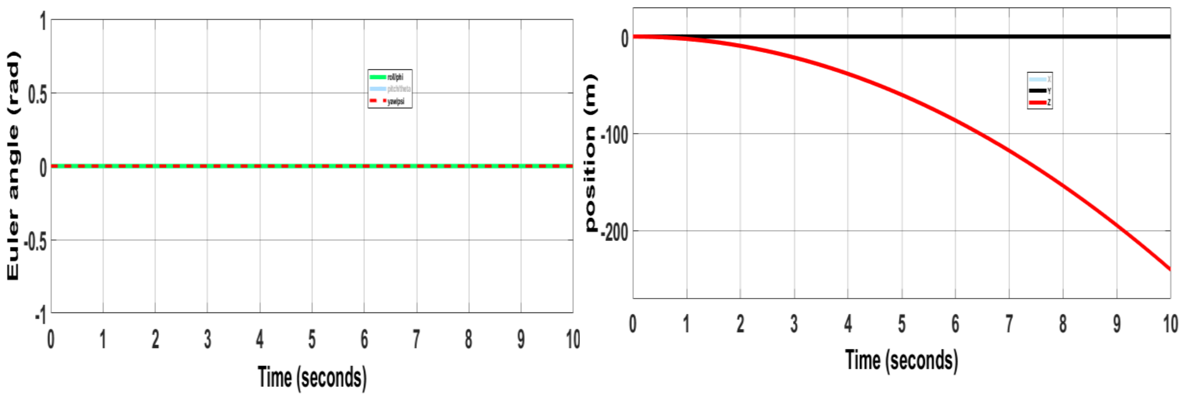


Figure 3.14 position and angle at descending motion

Roll Movement: This refers to the Y and Z-axis movements of the quadcopter. The rotational speeds of ($\Omega_1 = \Omega_4 = 212.7248 rad/s$) and $\Omega_2 = \Omega_3$ become 212.712 rad/s when specific control signals, such as $U_1 = 9.81Nm, U_2 = 0.0001Nm, and\ U_3 = U_4 = 0NM$, are applied. The quadcopter's roll angle changes as a result of these adjustments. The X-axis stays stationary as the quadcopter advances in the direction of the negative Y-axis and lowers along the negative Z-axis as a result.

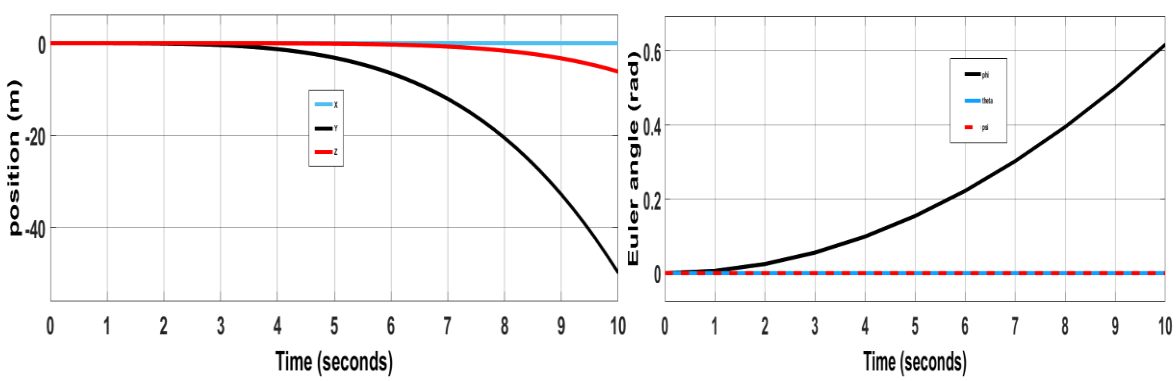


Figure 3.15 position and angle at roll movement

Pitch Movement: This explains the Z and X axes of the quadcopter's movement. Upon applying specific control signals, namely $U_1 = 9.81N, U_3 = 0.0001Nm, and\ U_2 = U_4 = 0\ Nm$, while the quadcopter is initially five meters above the ground, it experiences acceleration ($\Omega_1 = \Omega_4 = 212.7248 rad/s\ and\ \Omega_2 = \Omega_3 = 212.712 rad/sec$ and deceleration, therefore altering its pitch angle. The quad copter moves in the direction of the positive X-axis and falls along the negative Z-axis, changing the pitch angle while the Y-axis stays the same

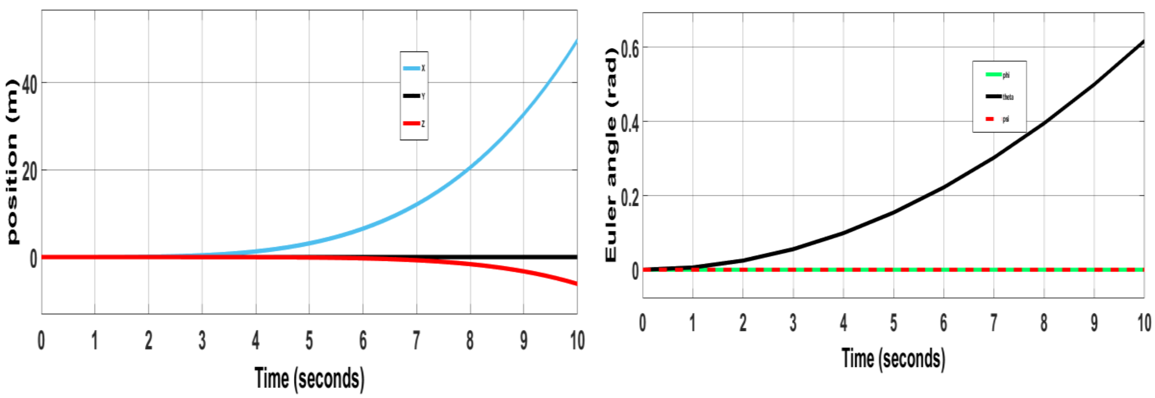


Figure 3.16 position and angle at pitch movement

Yaw movement: When$U_4 = 0.0001\ and\ U_2 = U_3 = U_1 = 0Nm$, the quad rotor's rotational speed drops to ($\Omega_1 = \Omega_4 = 212{,}662 rad/s$) and rises to ($\Omega_2 = \Omega_3 = 212.766 rad/s$). The quad rotor is initially positioned six meters above the ground. Instead of shifting positions, this speed differential led the quadrotor to revolve counterclockwise.

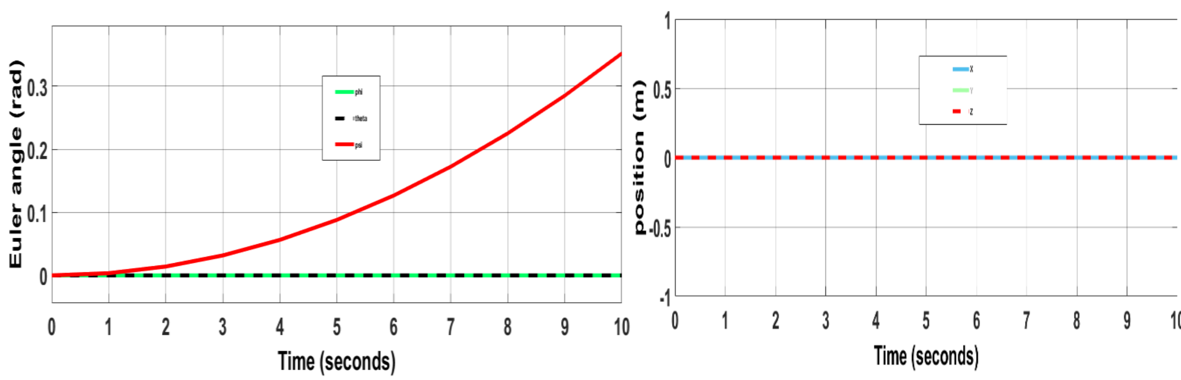


Figure 3.17 position and angle for yaw movement

CHAPTER FOUR

4. CONTROLLER DESIGN FOR THE QUADROTOR

4.1. Design of the Sliding Mode Controller

Sliding mode controllers, are a reliable control technique for dynamic systems that must withstand perturbations from the outside world and changes in parameters. For dynamically nonlinear systems, it is quite beneficial. Keeping the system's state variables directed and maintained about a predefined sliding surface is the aim of the control technique. With this skill, the oscillations and other disturbances to the system dynamics can be effectively eliminated by the controller. The controller employs discontinuous control signals, which are generated from the system malfunction, to do this (Rao, 1995).

The challenges of complex, high-order systems functioning in unpredictable contexts can be effectively addressed by sliding mode control. Reducing system order, effectively rejecting disruptions, and maintaining resilience in the face of parameter fluctuations are some of its positive attributes. It does, however, have a chattering problem. Fast, high-frequency switching between two control rules that takes place in close proximity to the sliding surface is what is known as chattering (Gedefaw, 2023). The switching functions maintain the state trajectory on this surface by using a high switching frequency.

4.1.1. Adaptable Surface

Sliding surfaces are mathematical constructs that can be effectively utilized to precisely characterize a portion of a system's state space. The control action moves and functions inside this defined zone in a predetermined way in order to accomplish the intended system performance. As such, selecting the sliding surface is an essential first step when utilizing the sliding mode to obtain particular system characteristics. To characterize this sliding surface, we apply the subsequent equation: The time-varying function of the n-dimensional state space S(t) (Gedefaw, 2023).

The following is a definition of the tracking mistake. Consider the following: $e = Xr - x = (e, e^{\cdot}, e^{\cdot\cdot} \dots e^{n-1})T.$ The tracking error vector is represented by t. Furthermore, the definition of a time-varying surface is as follows.

$$s(t) = (\frac{d}{dt} + c)^{n-1} e \tag{4.1}$$

Equation (4.1) states that the coefficient c controls the sliding surface's slope by maintaining a constant, positive value. any perturbations or changes in the plant parameters have no bearing on how the system behaves. proportional integral derivative (PID) is approaches used to find the sliding surface. You may represent these three categories using the following formulas (Piltan, 2011).

$$S_{PID} = kde(t) + k\imath \int \dot{e}(t) + kpe(t) \tag{4.2}$$

An integral term is purposefully added to the error formulations to account for disturbances and oscillations in parameters because static error cannot be completely avoided. The PID surface was chosen for this thesis because it reduces steady-state error and enhances tracking.

4.1.2. Regulating Law: When constructing sliding mode controllers, two separate control rules must be applied.

Equivalent Control Legislation ($\boldsymbol{Ueuq(t)}$): The corresponding control rule for the sliding mode control framework indicates the necessary maintain the sliding surface moving in the intended direction by controlling movement. (Fiori et al., 2024). Recall that the discontinuous control input rather than the actual control input is what is causing this cumulative impact.

Getting to the Law: The conditions under which the system's state approaches and reaches are indicated by the reaching condition. $\mathrm{Udis(t)} = \mathrm{ksign(s(t))}$ (4.3)

By employing the quasi function below, where σ is a tiny positive value, the chattering problem can be reduced. $\text{Udis} = \text{ksign}(s(t)) = \frac{ks(t)}{|s(t)|+\sigma}$ (4.4)

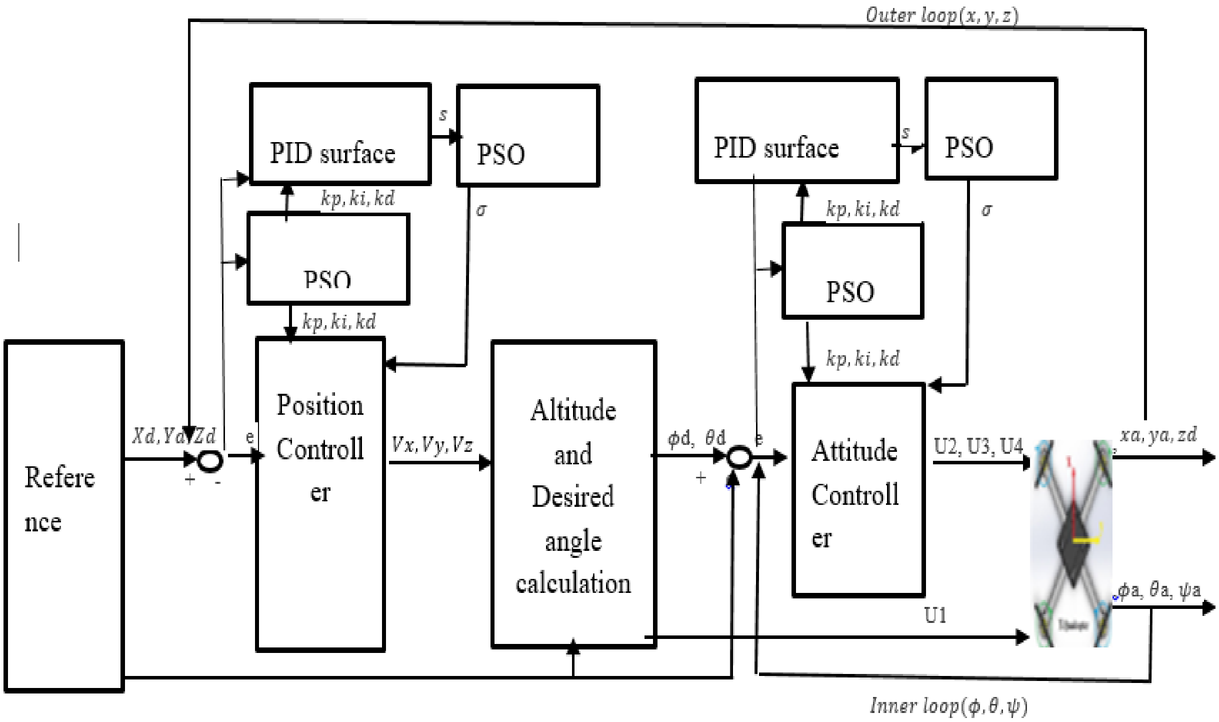


Figure 4.1 Block schematic of the entire control mechanism

The rolling surfaces at the yaw, pitch, and roll angles, as well as their derivatives, in the same outer loop control technique architecture, are expressed as follows (Belsty, 2021):

$$\begin{bmatrix} e_\phi \\ e_\theta \\ e_\psi \end{bmatrix} = \begin{bmatrix} \phi - \phi_d \\ \theta - \theta_d \\ \psi - \psi_d \end{bmatrix}, \begin{bmatrix} \dot{e}_\phi \\ \dot{e}_\theta \\ \dot{e}_\psi \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\phi}_d \\ \dot{\theta} - \dot{\theta}_d \\ \dot{\psi} - \dot{\psi}_d \end{bmatrix}, \begin{bmatrix} \ddot{e}_\phi \\ \ddot{e}_\theta \\ \ddot{e}_\psi \end{bmatrix} = \begin{bmatrix} \ddot{\phi} - \ddot{\phi}_d \\ \ddot{\theta} - \ddot{\theta}_d \\ \ddot{\psi} - \ddot{\psi}_d \end{bmatrix} \quad (4.5)$$

$$\begin{aligned} s_\phi &= c_\phi e_\phi + \dot{e}_\phi \\ s_\theta &= c_\theta e_\theta + \dot{e}_\theta \\ s_\psi &= c_\psi e_\psi + \dot{e}_\psi \end{aligned} \quad (4.6)$$

The equivalent and discontinuous controllers: $Utotal = Udis + Uequ$ (4.5)

Design for x $\dot{x}_1 = x_2$, $\dot{x}2 = \frac{Ux}{m}$ Let the error $e_x = x_d - x$ (4.6)

$$\dot{e}_x = x_2 - \dot{x}_d \tag{4.7}$$

$$\ddot{e}_x = \ddot{x} - \ddot{x}_d = \frac{Ux}{m} - \ddot{x}_d \tag{4.8}$$

$$\dot{s}_x = \dot{x} - \dot{xd}, \ddot{s}_x = \ddot{x} - \ddot{x}_d = \frac{Ux}{m} - \ddot{x}_d \tag{4.9}$$

$$U_x = U_{equ} = m(\ddot{x}_d - cx(\ddot{x}_d - \ddot{x}) \tag{4.10}$$

$$U_{xdis} = k_x sign(s_x),\ U_{total} = U_{xequ} + U_{dis} \tag{4.11}$$

$$U_x = m(\ddot{x}_d - cx(\ddot{x}_d - \ddot{x}) + k_x sign(s_x) \tag{4.12}$$

Design for y $\dot{x}_3 = x_4, \dot{x}_4 = U_y/m$ let error $y_d - y$ (4.13)

$$\dot{x}_3 = x_4, \dot{x}_4 = U_y/m \tag{4.14}$$

$$e_y = x_3 - y_d, \dot{e}_y = \dot{x}_4 - \dot{y}_d \tag{4.15}$$

$$\ddot{e}_y = \dot{x}_4 - \ddot{y}_d = \frac{Uy}{m} - \ddot{y}_d \tag{4.16}$$

$$\dot{s}_x = \dot{y} - \dot{yd}, \ddot{s}_y = \ddot{y} - \ddot{y}_d = \frac{Uy}{m} - \ddot{y}_d \tag{4.17}$$

$$U_y = U_y equ = m(\ddot{y}_d - cy(\ddot{y}_d - \ddot{y}) \tag{4.18}$$

$$U_{dis} = k_y sign(y) \quad ,U_{total} = U_{yequ} + U_{dis} \tag{4.19}$$

$$U_y = m(\ddot{x}_d - cy(\ddot{y}_d - \ddot{y}) + \ k_y sign(s_y) \quad (4.20)$$

Design for z $\dot{x}_5 = x_6, \dot{x}_6 = U_z/m$ let error $zd - z$ (4.21)

$$e_y = x_5 - z_d, \dot{e}_y = \dot{x}_5 - \dot{z}_d \qquad \ddot{e}_y = \dot{x}_5 - \ddot{z}_d = \frac{Uz}{m} - \ddot{z}_d \quad (4.22)$$

$$\dot{s}z = \dot{z} - \dot{zd}, \ddot{s}_z = \ddot{z} - \ddot{z}_d = \frac{Uz}{m} - \ddot{z}_d, Uz = Uzequ = m(\ddot{z}_d - cz(\ddot{z}_d - \ddot{z}) \quad (4.23)$$

$$U_{dis} = k_z sign(z) \qquad , Utotal = Uzequ + Udis \quad (4.24)$$

$$U_z = m(\ddot{z}_d - cz(\ddot{z}_d - \ddot{z}) + \ k_z sign(s_z) \quad (4.25)$$

Design for phi

$$\dot{X}_7 = X_8, \dot{X}_8 = X_8 c_1 + 1/m(cos\varphi cos\psi \sin\theta - sin\phi cos\psi)U1 \quad (4.26)$$

Using the error variable as $e_\phi = X_7 - \phi d, \dot{e}_\phi = \dot{X}_7 - \dot{\phi}_d$ (4.27)

$$\ddot{e}_\phi = \dot{X}_7 - \ddot{\phi}_d = X_8 - \ddot{\phi}_d = b_1 U_2 + c_1 X_{10} X_{12} - c_2 X_{10} - \ddot{\phi}_d \quad (4.28)$$

Select Sliding Surface $s_\phi = c_\phi e_\phi + \dot{e}_\phi, \dot{s}\phi = c_\phi e_\phi + \dot{e}_\phi \dot{=} c_\phi e_\phi + \phi - \dot{\phi d}$ (4.29)

$$\ddot{s}_\phi = c_\phi e_\phi + \ddot{\phi} - \ddot{\phi}_d = c_\phi e_\phi + b_1 U_2 + c_1 X_{10} X_{12} - c_2 X_{10} - \ddot{\phi}_d \quad (4.30)$$

$$U_2 = U_{equ} = 1/b1(\ddot{\phi}_d - c1 X_{10} X_{12} + c2 X_{10} + c_\phi e_\phi) \quad (4.31)$$

$$U_{dis} = k_\phi sign(\phi) \qquad , U_{total} = U_{zequ} + U_{dis} \quad (4.32)$$

$$U_2 = U_{equ} = 1/b1(\ddot{\phi}_d - c_1 X_{10} X_{12} + c_2 X_{10} + c_\phi(\dot{\phi d} - \phi) + k_\phi sign(\phi) \quad (4.33)$$

SMC controller design for theta

$$\dot{X}_9 = X_{10}, \dot{X}_{10} = b_2U_3 + c_3X_8X_{12} - c_4X_8 \tag{4.34}$$

Using the error variable as $e_\theta = \theta - \theta_d, \dot{e}_\theta = \dot{\theta} - \dot{\theta}_d$ (4.35)

$$\ddot{e}\theta = \ddot{\theta} - \ddot{\theta}_d = b_2U_3 + c_3X_8X_{12} - c_4X_8 - \ddot{\theta}_d \tag{4.36}$$

Select for sliding surface $s_\theta = c_\theta e_\theta + \dot{e}_\theta$

$$\ddot{s}\theta = c_\theta\dot{e}_\theta + \ddot{\theta} - \ddot{\theta}_d = c_\theta\dot{e}_\theta + b_2U_3 + c_3X_8X_{12} - c_4X_8 - \ddot{\theta}_d \tag{4.37}$$

$$U3 = U3equ = 1/b2(\ddot{\theta}_d - c_3X_8X_{12} + c_4X_8 - c_\theta(\dot{\theta} - \dot{\theta}_d) \tag{4.38}$$

$$Udis = k\theta sign(s\theta), U_3 = U_{3equ} + U_{dis} \tag{4.39}$$

$$U3 = 1/b2(\ddot{\theta}_d - c_3X_8X_{12} + c_4X_8 + c_\theta(\dot{\theta} - \dot{\theta}_d) + k\theta sign(s\theta) \tag{4.40}$$

SMC controller design for psi $\dot{X}11 = X12, \quad \dot{X}12 = b_3U_4 + c5X8X10$ (4.41)

Using the error variable as $e\psi = \psi - \psi d, \dot{e}\psi = \dot{\psi} - \dot{\psi}d$ (4.42)

$$\ddot{e}\psi = \dot{\psi} - \ddot{\psi}d = b_3U_4 + c_5X_8X_{10} - \ddot{\psi}d \tag{4.43}$$

Select sliding surface $s_\psi = c_\psi e_\psi + \dot{e}_\psi$

$$\ddot{s}\psi = c_\psi\dot{e}_\psi + \ddot{\psi} - \ddot{\psi}_d = c_\psi\dot{e}_\psi + b_3U_4 + c_5X_8X_{10} - \ddot{\psi}d \tag{4.44}$$

$$U_4 = U4equ = 1/b3(\ddot{\psi}d - c_\psi\dot{e}_\psi - b_3U_4 - c_5X_8X_{10}) \tag{4.45}$$

$$U_{dis} = k\psi sign(s\psi),\ U_{total} = U_{equ} + U_{dis} \tag{4.46}$$

$$U_4 = 1/b3(\ddot{\psi}d - c_\psi \dot{e}_\psi - b3u4 - c5X10X8) + k\psi sign(s\psi) \tag{4.47}$$

Representation of Overall SMC controller equation

$$U_{x=}\ddot{x}d - c_x(\dot{x}_d - \dot{x}) + k_x sign(s_x)$$

$$U_{y=}\ddot{y}d - c_y(\dot{y}_d - \dot{y}) + k_y sign(s_y)$$

$$U_{z=}\ddot{z}d - c_z(\dot{z}_d - \dot{z}) + k_z sign(s_z)$$

$$U_{2=}1/b_1(\ddot{\phi}_d - c_1X_{10}X_{12} + c_2X_{10} + c_\phi(\dot{\phi}_d - \dot{\phi}) + b_1U_2) + k_\phi sign(s_\phi)$$

$$U3 = 1/b_2(\ddot{\theta}_d - c_3X_8X_{12} + c_4X_8 + c_\theta(\dot{\theta} - \dot{\theta}_d) + k_\theta sign(s_\theta)$$

$$U_4 = 1/b_3(\ddot{\psi}d - c_\psi \dot{e}_\psi + c_5X_8X_{10}) + k_\psi sign(s_\psi)$$

4.1.3 Designing surface controllers using SMC and PID

Summation of the equivalent and discontinuous controller yields the sliding mode controller, which is expressed as follows: $U_{total} = U_{xequ} + U_{dis}$ (4.48)

PID sliding surface design for x $\dot{x}_1 = x_2$, $\dot{x}_2 = \frac{Ux}{m}$, Let the error $e_x = x_d - x$

$$\dot{e}_x = x \dot{-} \dot{x}_d = x_2 - \dot{x}_d \ddot{e}_x = \ddot{x} - \ddot{x}_d = \frac{Ux}{m} - \ddot{x}_d \tag{4.49}$$

Sliding surface $s_x = K_{px}e_{x(t)} + K_{dx}\dot{e}_{x(t)} + K_{ix}e_{x(t)}$ (4.50)

$$\dot{s}_x = K_{px}\dot{e}_x + K_{dx}\ddot{e}_x + K_{ix}e_x \quad (4.51)$$

Substitute equation (4.6) in to equation (4.9)

$$\dot{s}_x = K_{px}\dot{e}_x + K_{ix}e_x + K_{dx}(-\frac{Ux}{m} + \ddot{x}_d) \quad (4.52)$$

Realizing that $\dot{s}_x$at $U_{equ} = 0$ yields the analogous control law. Assume that $\dot{s}x$ = 0.

Rearrange equation (4.52) $U_{equ(x)} = m/K_{dx}(K_{dx}\ddot{x}_d + K_{px}\dot{e}_x + K_{ix}e_x$ (4.53)

Selecting steady-state rate reaching: $\dot{s}_x = U_{dis} = \frac{k_x(s_x)}{|s_x|+\sigma}$ (4.54)

above equation $U_x = U_{equ} + U_{dis} = m/K_{dx}(K_{dx}\ddot{x}_d + K_{px}\dot{e}_x + K_{ix}e_x) + \frac{k_x(s_x)}{|s_x|+\sigma}$ (4.55

$$Udis = m/K_{dx}(\frac{k_x(s_x)}{|s_x|+\sigma}) \quad (4.56)$$

Stability for X $define\ lyaponov\ candidate\ V_x = \frac{1}{2}s_x 2, \mathrm{v}(0) = 0, \mathrm{v(sx)} > 0$ is posstive function $\dot{V}_x = \dot{s}_x s_x < 0$ (4.57)

$$\dot{v}(sx) = S_x(K_{dx}\ddot{x}_d - K_{px}\dot{e}_x - K_{ix}e_x + m/K_{dx}(K_{dx}\ddot{x}_d - K_{px}\dot{e}_x - K_{ix}e_x\ ) \quad (4.58)$$

$$\dot{V}(S_x) = S_x((\frac{k_x(s_x)}{|s_x|+\sigma}) \quad (4.59)$$

$if(S_x) > 0, K_x > 0, if(S_x) < 0, K_x < 0, so\ kx\ must\ be\ positve\ to\ be\ stable$

Plan for Y: Examining the second translational subsystem will help us:

$$\dot{x}_3 = x_4, \dot{x}_4 = U_y/m \quad (4.60)$$

Using the error variable $e_y = x_3 - y_d, \dot{e}_y = \dot{x}_4 - \dot{y}_d, \ddot{e}_y = \dot{x}_4 - \ddot{y}_d = \frac{Uy}{m} - \ddot{y}_d$ (4.61)

Choose sliding surface $sy = k.pyey + kiy \int ey + kd\dot{y}ey$ (4.62)

$$\dot{s}_y = K_p\dot{e}_y + K_ie_y + K_d\ddot{e}_y = K_p\dot{e}_y + K_ie_y + K_d(-\frac{Uy}{m} + \ddot{y}_d) \quad (4.63)$$

$$\frac{KdyUequ}{m} = K_{dy}\ddot{y}_d - K_{iy}e_y - K_{py}\dot{e}_y \quad (4.64)$$

$$U_{equ} = m/K_{dy}(K_{dy}\ddot{y}_d + K_{iy}e_y + K_{py}\dot{e}_y) \quad (4.65)$$

$$selecting\ steady\ state\ error\ reaching\ S_y = (\frac{k_y(s_y)}{|s_y|+\sigma}) \quad (4.66)$$

Realizing that will lead to the comparable control law. S_y is 0 atU_{equ} Equation is rearranged, and the resultant control law is: $U_y = U_{dis} + U_{equ}$ (4.67)

$$U_y = m/K_{dy}(K_{dy}\ddot{y}_d + K_{iy}e_y + K_{py}\dot{e}_y + \frac{k_y(s_y)}{|s_y|+\sigma}) \quad (4.68)$$

Stability for position Y: $define\ lyaponov\ candidate\ V_y = \frac{1}{2}s_y2, \mathrm{v}(0) = 0, \mathrm{v}(\mathrm{sy}) > 0$ it is posstive function, $\dot{V}_y = \dot{S}_yS_y < 0$ (4.69)

$$\dot{V}_{(sy)} = S_y(K_{dy}\ddot{y}_d + K_{iy}e_y + K_{py}\dot{e}_y + m/K_{dy}(K_{dy}\ddot{y}_d + K_{iy}e_y + K_{py}\dot{e}_y\ ) \quad (4.70)$$

$$\dot{V}_{(sy)} = S_y(-K_ysign(S_y)) = S_y(\frac{k_y(s_y)}{|s_y|+\sigma} \quad if(sy) > 0, ky > 0 \quad (4.71)$$

$if(S_y) < 0, K_y < 0, so\ ky\ must\ be\ positve\ to\ be\ stable.$

Plan for Z: Examining the second translational subsystem will help us:

$$\dot{x}_5 = x_6, \dot{x}_6 = u_z/m \tag{4.72}$$

Using the error variable as $e_z = x_5 - z_d, \dot{e}_z = \dot{x}_5 - \dot{z}_d$ (4.73)

$$\ddot{e}_z = \dot{x}_6 - \ddot{z}_d = \frac{Uz}{m} - \ddot{z}_d \tag{4.74}$$

Choose sliding surface $S_z = kpzez + kiz \int ez + kd\dot{z}ez$ (4.75)

$$\dot{s}_z = K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z = K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z(-\frac{U_z}{m} + \ddot{z}_d) \tag{4.76}$$

$$\frac{K_{dz}Uequ}{m} = K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z \tag{4.77}$$

$$Ueuq = m/K_{dz}(K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z) \tag{4.78}$$

$$selecting\ steady\ state\ error\ reaching\ s_z((\frac{K_z(s_z)}{|s_z|+\sigma}))) \tag{4.79}$$

Finding the corresponding control law requires realizing that $z\ at$ uequ is zero . The control law as a whole is obtained by rearranging equation (4.35).

$$U_z = U_{equ} + U_{dis} \tag{4.80}$$

$$U_z = m/K_{dz}(K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z)\left(\frac{K_z(s_z)}{|s_z|+\sigma})\right)) \tag{4.81}$$

Stability for position Z $define\ lyaponov\ candidate\ V_z = \frac{1}{2}s_z 2, \mathrm{v}(0) = 0, \mathrm{v(sz)} > 0$ it is posstive function . $\dot{V}z = \dot{S}_z S_z < 0$ (4.82)

$$\dot{V}(S_z) = S_z(K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z + m/K_{dz}(K_{dz}\ddot{z}_d + K_{iz}e_z + K_{pz}\dot{e}_z) \tag{4.83}$$

$$\dot{V}(S_z) = S_z(-K_z sign(S_z) = S_z\left(\frac{K_z(s_z)}{|s_z|+\sigma})\right) \quad (4.84)$$

$If(S_z) > 0, kz > 0, if(S_z) < 0, kz < 0, so\ kz\ must\ be\ positve\ to\ be\ stable$

plan for U2: Examining the second translational subsystem will help us:

$$\dot{X}_7 = X_8, \dot{X}_8 = X_8 c_9 + 1/m(cos\varphi cos\psi \sin\theta - sin\phi cos\psi)U1 \quad (4.85)$$

Using the error variable as $e_\phi = X_7 - \phi d, \dot{e}_\phi = \dot{X}_7 - \dot{\phi}_d$ (4.86)

$$\ddot{e}_\phi = \dot{X}_7 - \ddot{\phi}_d = X_8 - \ddot{\phi}_d = b_1 U_2 + c_1 X_{10} X_{12} - c_2 X_{10} = \ddot{\phi}_d \quad (4.87)$$

Choose sliding surface $S_\phi = Kp_\phi e_\phi + Ki_\phi \int e_\phi + Kd_\phi \dot{e}_\phi$ (4.88)

$$\dot{S}_\phi = Kp_\phi \dot{e}_\phi + Ki_\phi e_\phi + Kp_\phi \ddot{e}_\phi = kp\dot{e}_\phi + Ki_\phi e_\phi + Kd_\phi(-X_8 + \ddot{\phi}_d) \quad (4.89)$$

Finding the corresponding control law requires realizing that. Understanding that $\dot{S}_\phi\ at\ Uequ(2) = 0$ is necessary in order to find the matching control law. Let us assume that $\dot{S}_\phi = 0$

$$Kd_\phi b1Uequ(2) = Kd_\phi \ddot{\phi}_d + Ki_\phi e_\phi + Kp_\phi \dot{e}_\phi + Kd_\phi c_1 X_{10} X_{12} + Kd_\phi c_2 X_{10} \quad (4.90)$$

Rearranging the formula

$$U_{2equ} = 1/Kd_\phi b1\left(Kd_\phi \ddot{\phi}_d - Ki_\phi e_\phi - Kp_\phi \dot{e}_\phi - Kd_\phi c_1 X_{10} X_{12} - Kd_\phi c_2 X_{10}\right) \quad (4.46)$$

$$selecting\ steady\ state\ error\ reaching\ S_\phi = \left(\frac{K_\phi(S_\phi)}{|S_\phi|+\sigma}\right)) \quad (4.91)$$

Finding the analogous control law requires realizing that S_{ϕ} at $Uequ_{equ} = 0$. Equation (4.47) is rearranged, and the resultant control law is: $U_2 = U_{dis} + U_{equ}$ (4.92)

$$U_2 = 1/Kd_{\phi}b1\left(Kd_{\phi}\ddot{\phi}_d - Ki_{\phi}e_{\phi} - Kp_{\phi}\dot{e}_{\phi} - Kd_{\phi}c1X_{10}X_{12} - Kd_{\phi}c2X_{10}\right) + \left(\frac{K_{\phi}(S_{\phi})}{|S_{\phi}|+\sigma}\right) 4.93$$

$$U_{dis} = 1/kd\phi\left(\frac{K_{\phi}(S_{\phi})}{|S_{\phi}|+\sigma}\right) \quad (4.94)$$

$$U_2 = U_{dis} + U_{equ} \quad (4.95)$$

Stability for phi

The roll angle controller parameters are K_{ϕ} . Applying the Lyapunov stability theorem to the ranges of the discontinuous controller gain are obtained similarly for position control. $define\ lyaponov\ candidate\ V_{\phi} = \frac{1}{2}s_{\phi}2, \mathrm{v}(0) = 0, \mathrm{v}(\mathrm{sz}) > 0$ (4.96)

$$\dot{V}\phi = \dot{s}\phi S_{\phi} < 0 \quad (4.97)$$

$$\dot{V}(s_{\phi}) = s_{\phi}(Kd_{\phi}\ddot{\phi}_d - Ki_{\phi}e_{\phi} - Kp_{\phi}\dot{e}_{\phi} + m/Kd_{\phi}(Kd_{\phi}\ddot{\phi}_d - Ki_{\phi}e_{\phi} - Kp_{\phi}\dot{e}_{\phi} +$$

$$1/Kd_{\phi}b1\left(Kd_{\phi}\ddot{\phi}_d - Ki_{\phi}e_{\phi} - Kp_{\phi}\dot{e}_{\phi} - Kd_{\phi}c1X_{10}X_{12} - Kd_{\phi}c2X_{10}\right) - K_{\phi}sign(s_{\phi})$$

$$\dot{V}\left(s_{\phi}\right) = s_{\phi}(\left(\frac{K_{\phi}(S_{\phi})}{|S_{\phi}|+\sigma}\right)) \quad (4.98)$$

$$if\left(s_{\phi}\right) > 0, k\phi > if\left(s_{\phi}\right) < 0, K_{\phi} < 0, so\ ky\ must\ be\ positve\ to\ be\ stable$$

Design U3: The second rotational subsystem will be discussed now.

$$\dot{X}_9 = X_{10}, \dot{X}_{10} = b_2 U_3 + c_3 X_8 X_{12} - c_4 X_8 \quad (4.99)$$

Using the error variable as $e_\theta = \theta - \theta_d, \dot{e}_\theta = \dot{\theta} - \dot{\theta}_d$ (4.100)

$$\ddot{e}\theta = \ddot{\theta} - \ddot{\theta}_d = b_2 U_3 + c_3 X_8 X_{12} - c_4 X_8 = \ddot{\theta}_d \quad (4.101)$$

Choose sliding surface $S_\theta = Kd_\theta e_\theta + Ki_\theta \int e\theta + Kp_\theta \, \dot{e}_\theta$ (4.102)

$$\dot{S}\theta = Kp_\theta \dot{e}\theta + kie\theta + kd\ddot{e}\theta = Ki_\theta \int e_\theta + Kp_\theta \, \dot{e}_\theta + Kd_\theta(-\ddot{\theta} + \ddot{\theta}d) \quad (4.103)$$

Finding the corresponding control law requires realizing that $s$at $Uequ$ is 0. Understanding that $\dot{S}\theta$ at $Uequ(2) = 0$ is necessary in order to find the matching control law. Let us assume that S_θ= 0.

$$Kd_\theta b_2 U_{3equ} = Kd_\theta \ddot{\theta}d + Ki_\theta e_\theta + kp\theta\dot{e}\theta + Kd_\theta c3X8X12 - Kd_\theta c4X8 \quad (4.104)$$

Rearranging the formula (4.62)

$$U_{3equ} = 1/Kd_\theta b2(Kd_\theta \ddot{\theta}d + Ki_\theta e_\theta + kp\theta\dot{e}\theta + Kd_\theta c3X8X12 - Kd_\theta c4X8\ ) \quad (4.105)$$

$$Selecting\ steady\ state\ error\ reaching\ S_\theta = \left(\frac{K_\theta(S_{\theta)}}{|S_\theta| + \sigma}\right) \quad (4.106)$$

Finding the analogous control law requires realizing that sv at $Uequ$ the control law as a whole is obtained by rearranging equation (4.47): $U_3 = U_{equ} + U_{dis}$ (4.107)

$$U_3 = 1/Kd_\theta b2(Kd_\theta \ddot{\theta}d + Ki_\theta e_\theta + kp\theta\dot{e}\theta + Kd_\theta c3X8X12 - Kd_\theta c4X8) + \left(\frac{K_\theta(S_{\theta)}}{|S_\theta| + \sigma}\right)(10)$$

$$1/Kd_{\theta}b2\left(\frac{K_{\theta}(S_{\theta)}}{|S_{\theta}|+\sigma}\right) = Udis \tag{4.108}$$

Stability for theta: Just like with position control, the nonlinear gain ranges of the controller can be determined by applying the Lyapunov stability theorem. We can check the stability for each theta angle as follow

$define\ lyaponov\ candidate\ V_{\theta} = \frac{1}{2}s_{\theta}2, \mathrm{v}(0) = 0, \mathrm{v(sz)} > 0$ it is posstive function

$$\dot{V}\theta = \dot{s}\theta s_{\theta} < 0 \tag{4.109}$$

$\dot{V}(s_{\theta}) = s_{\theta}(Kd_{\theta}\ddot{\theta}d + Ki_{\theta}e_{\theta} + kp\theta\dot{e}\theta + m/Kd_{\theta}(Kd_{\theta}\ddot{\theta}d + Ki_{\theta}e_{\theta} + kp\theta\dot{e}\theta + 1/$
$Kd_{\theta}b2(Kd_{\theta}\ddot{\theta}d + Ki_{\theta}e_{\theta} + kp\theta\dot{e}\theta + Kd_{\theta}c3X8X12 - Kd_{\theta}c4X8) + \left(\frac{K_{\theta}(S_{\theta)}}{|S_{\theta}|+\sigma}\right)$ 4.110

$$\dot{V}(s_{\theta}) = s_{\theta}(-k_{\theta}sign(s_{\theta}) = s_{\theta}\left(\frac{K_{\theta}(S_{\theta)}}{|S_{\theta}|+\sigma}\right) \tag{4.111}$$

$if(s_{\theta}) > 0, k_{\theta} > 0, if(s_{\theta}) < 0, k_{\theta} < 0, so\ ky\ must\ be\ positve\ to\ be\ stable$

Design control law for U4 : The third rotational subsystem will be discussed now.

$$\dot{X}_{11} = X_{12}, \quad \dot{X}_{12} = U_4b_3 + X_{10}X_8c_5 \tag{4.112}$$

Using the error variable as , $e_{\psi} = \psi - \psi_d, \dot{e}_{\psi} = \dot{\psi} - \dot{\psi}_d$ (4.113)

$$\ddot{e}\psi = \dot{\psi} - \ddot{\psi}d = U_4b_3 + X_{10}X_8c_5 = \ddot{\psi}d \tag{4.114}$$

Choose sliding surface $s_{\psi} = kp_{\psi}e_{\psi} + ki_{\psi}\int e_{\psi} + kd\dot{\psi}e\psi$ (4.115)

$$\dot{s}\psi = kp_{\psi}\dot{e}\psi + kd_{\psi}e_{\psi} + kd_{\psi}\ddot{e}\psi = kp_{\psi}\dot{e}\psi + ki\psi e_{\psi} + kd\psi(-\psi + \ddot{\psi}d) \tag{4.116}$$

Finding the corresponding control law requires realizing that S at $Uequ$ is 0. Acquiring the knowledge that $\dot{S}\psi$ at $Uequ(4) = 0$ is necessary to find the matching control law. Assume Sψ = 0,

$$Kd_{\psi}b_3U_{4equ} = kd_{\psi}\ddot{\psi}d + ki_{\psi}e\psi + kp_{\psi}\dot{e}\psi + kd_{\psi}X_{10}X_8c_5 \quad (4.117)$$

Rearranging the formula (4.76)

$$U_{4equ} = 1/Kd_{\psi}b_3(Kd_{\psi}\ddot{\psi}d + Ki_{\psi}e\psi + KP_{\psi}\dot{e\psi} + Kd_{\psi}c5X8X10) \quad (4.118)$$

$$selecting\ steady\ state\ error\ reaching\ s\psi = \left(\frac{S\psi(S\psi)}{|S\psi|+\sigma}\right) \quad (4.119)$$

Finding the corresponding control law requires realizing that Sψ at Uequ is zero. Equation (4.60) is rearranged, and the resultant control law is: $U_4 = U_{equ} + U_{dis}$ (4.120)

$$U_{4equ} = 1/Kd_{\psi}b3(Kd_{\psi}\ddot{\psi}d + KI_{\psi}e\psi + KP_{\psi}\dot{e\psi} + Kd_{\psi}c5X8X10) + \left(\frac{S\psi(S\psi)}{|S\psi|+\sigma}\right) \quad 4.121$$

Stability for psi: Using the previously determined Lyapunov function to verify for psi stability yielded the switching gain.

$define\ lyaponov\ candidate\ V_{\psi} = \frac{1}{2}s_{\psi}2, \mathrm{v}(0) = 0, \mathrm{v}(\mathrm{s}\psi) > 0$ it is posstive function ,

$\dot{V}\psi = \dot{s}\psi s_{\psi} < 0$ (4.122)

$$\dot{V}(s_{\psi}) = s_{\psi}(Kd_{\psi}\ddot{\psi}d + KI_{\psi}e\psi + KP_{\psi}\dot{e\psi} + m/kd\phi(Kd_{\psi}\ddot{\psi}d + KI_{\psi}e\psi + KP_{\psi}\dot{e\psi}\ ) + 1/Kd_{\psi}b3(Kd_{\psi}\ddot{\psi}d + KI_{\psi}e\psi + KP_{\psi}\dot{e\psi} + Kd_{\psi}c5X8X10) + \left(\frac{S\psi(S\psi)}{|S\psi|+\sigma}\right) \quad (4.123)$$

$$\dot{V}(s\psi) = s_{\psi}(-k_{\psi}sign(s_{\psi}) = s_{\psi}\left(\frac{S\psi(S\psi)}{|S\psi|+\sigma}\right) \quad (4.124)$$

$if(s_{\psi}) > 0, k_{\psi} > 0\ so\ k\psi\ must\ be\ positve\ to\ be\ stable.$

Over all equations Summary for SMC with PID sliding surface

$$U_x = \; m/K_{dx}(K_{dx}\ddot{x}_d + K_{px}\dot{e}_x + K_{ix}e_x) \; + \frac{k_x(s_x)}{|s_x|+\sigma}$$

$$U_y = m/K_{dy}(K_{dy}\ddot{y}_d + K_{iy}e_y + K_{py}\dot{e}_y + \frac{k_y(s_y)}{|s_y|+\sigma})$$

$$Uz = m/k_{dz}(k_{dz}\ddot{Z}d + k_{iz}(Zd - z) + k_{pz}(\dot{Z}d - \dot{z}) + \left(\frac{k_z(s_z)}{\lfloor s_z\rfloor+\sigma}\right))$$

$$U_2 = 1/Kd_\phi b1\left(Kd_\phi\ddot{\phi}_d - Ki_\phi e_\phi - Kp_\phi\dot{e}_\phi - Kd_\phi c_1X_{10}X_{12} - Kd_\phi c_2X_{10}\right) + \left(\frac{K_\phi(S_\phi)}{|S_\phi|+\sigma}\right)$$

$$U_3 = 1/Kd_\theta b2(Kd_\theta\ddot{\theta}d + Ki_\theta e_\theta + kp\theta\dot{e}\theta + Kd_\theta c_3X_{10}X_8 - Kd_\theta c4X8) + \left(\frac{K_\theta(S_{\theta)}}{|S_\theta|+\sigma}\right)$$

$$U_4 = 1/Kd_\psi b3(Kd_\psi\ddot{\psi}d + KI_\psi e\psi + KP_\psi e\dot{\psi} + Kd_\psi c_5X_{10}X_8) \; + \left(\frac{S\psi(S\psi)}{|S\psi|+\sigma}\right)$$

4.2.2 Gain Tuning of SMC and SMC and PID controller

PSO is used to optimize position control gains and the orientation control gains $k_{p\phi}, k_{p\theta}, k_{p\psi}, k_{i\phi}, k_{i\theta}, k_{i\psi}, k_{d\phi}, k_{d\theta}, k_{d\psi}, \sigma, k_{px}, k_{py}, k_{pz}, k_{ix}, k_{iy}, k_{ix}, k_{dy}, k_{dz}, k_{dx,}k_y, k_{x,}k_z, \sigma$ in the quad copter control laws of overall equation above. Drawing inspiration from the movements of fish schools and bird flocks, a method for optimizing non-linear systems through the particle swarm approach is presented and developed in It has shown to be especially successful at solving challenging optimization issues (Belsty, 2021). PSO compares numerous samples of the optimization variables in a series of steps in an effort to minimize the objective function.

PSO is a clever method for choosing the best parameters among N particles.

- N particles are scattered throughout a D-dimensional search space in the initialization matrix.

- A fitness function that is pertinent to the particle's position is needed for PSO.
- Every particle i stores its optimal location $pb(t+1)$ as well as the optimal solution nearby $gb(t+1)$, which is the particle with the lowest fitness value in the swarm.

4.2.3 Algorithms for PSO

1. Using D dimensions as the search space, build a population array using accelerated and scattered particles at random.

2. Ascertain which optimization for each particle is required to minimize the D variables of the swarm's fitness function.

3. Evaluate the particle's fitness in relation to its personal pest ($pbest$). The personal pest is the particle's current two locatin to its current value if it exceeds it.

4. Particle pbests are compared to one another, the global best position of the swarm with the highest fitness is updated, and update the particle $gbest$.

5. Update the particle's position and velocity using the corresponding equations below.

In the dimension D search space, the particle $i-1,2\ldots N$ with $i-1,2\ldots D$ has a new velocity matrix vij and position matrixxij.

$$Vij(t+1)=wvij(t)+R1C1(pvij(t)-xij(t)+R2C2(gbij(t)-xij(t)))$$

$$Xij(t+1)=xij(t)+vij(t+1),pbest=xi\ ,\ fbest=fi \tag{4.125}$$

$$if\ fi<f(gbest)\qquad \{fgbest=f\big(p(best)\big),if\ f(pbest<f(gbest))$$

For example, pbest indicates the ideal location as determined by particle i, and gbest indicates the ideal location Two random variables, R1 and R2, were constructed in [0, 1] using a uniform distribution.

There exist three components to the update velocity of equation 4.125.

- Section 1: The momentum part $(wvxij(t)$ keeps the particle from quickly altering its course by serving as a fly by memory of earlier flights. W, a variable parameter that controls the particle's displacement
- Section 2: Cognitive portion $R1C1(pvij(t) - xij(t)$ requires calculating the particle's performance in relation to its historical performance.
- Section 3: Social component $R2C2(gbij(t) - xij(t)$measures each particle's performance in relation to its neighbors. Two acceleration coefficients, C1 and social C2, respectively, describe the strength of attraction of each particle to the global optimum position.
- If $C1$ is equal to $C2$and $C1$ is greater than zero, particles explore locally; if $C1$ is equal to $C2$, in the interval [0, 1] are the coefficients $R1$ and $R2$, which ensure good particle exploration in the search space.

4.2.4 ITAE Performance Index

A performance index measures the optimization performance of the quadrotor's responses using the proposed method. Reduced positional and directional errors in quadcopter trajectory tracking are the goal of these theses.

$$ITAE = \int_0^{\infty} t(|e(t)|)\, dt \tag{4.126}$$

Little damped oscillations and overshoots are characteristic of systems constructed with iterative adaptive control (ITAE While any significant initial error is highly penalized, Later-in response errors are severely penalized during ITAE, which prolongs the settling time and causes control issues. In this study, the ITAE performance index was applied.

CHAPTER FIVE

5. WINDTURBINE SURFACE DAMEAGE DETACTION

5.1. Introduction

One of the most difficult tasks in this industry is minimizing maintenance costs and identifying wind blade defects early on. For this reason, early blade inspection is crucial, and emerging technology like computer vision and object detection offer promising avenues for reducing risks associated with human use and maintenance expenses. UAV and machine learning applications to identify and classify damage to wind turbine blades and trajectory around wind turbine during inspection is investigated in this work. As a result, YOLOv8, a cutting-edge object detection model, is trained with a unique dataset made up of pictures showing different variations across several damage severity categories. Therefore, in our study early detection and preventive measures, exhibits room for our improvement.

5.2 Image Recognition

In computer vision, image recognition refers to the process by which a computer examines and recognizes objects, patterns, or other features in images or, more generally, extracts information from images. There has been a notable progress in the last few years. At the vanguard of this development is YOLOv8 (You Only Look Once version8), which provides cutting-edge capabilities in real-time object identification (Dijkstra et al., n.d.). For this thesis, YOLOv8 training is carried out with the PyCharm software. Software designed specifically with Python programming in mind is called PyCharm. Python developers use it extensively for coding 68 and it is created by JetBrains.

5.2.1 YOLOv8 Training process

Training YOLOv8 involves:

Gathering and annotating data: Tough model training requires high-quality annotations. Which contains 6043 photos, is the source of the data used in this thesis. 28 are used for testing, 496 for validation, and 5518 for training out of these photos (Foster et al., 2022b). Rob flow is an online annotation tool that is used to label photos for object recognition, classification, and segmentation. It is used to annotate data in the YOLOv8 format. An online annotation tool called MakeSense.ai was developed for computer vision applications, such as object identification. It makes the process of creating training data easier by offering an interface for identifying and annotating them (Gedefaw, 2023).

Loss Function: Throughout training, YOLOv8 makes use of a variety of loss functions. The accuracy of bounding box predictions (localization loss) and item existence (confidence loss) are the main metrics evaluated by these functions. The model uses these losses as guidance to produce precise forecasts.

Training: One important tactic that improves object detection tasks in the context of YOLOv8 is transfer learning. Large picture datasets have been used to extensively train these pretrained models for image recognition applications.

Evaluation: A separate test dataset is utilized to assess the model's item detection ability following training using metrics like (mAP). A trustworthy indicator for assessing the efficacy of an object detection model since it considers both the quantity of successfully recognized items and the caliber of the detections.

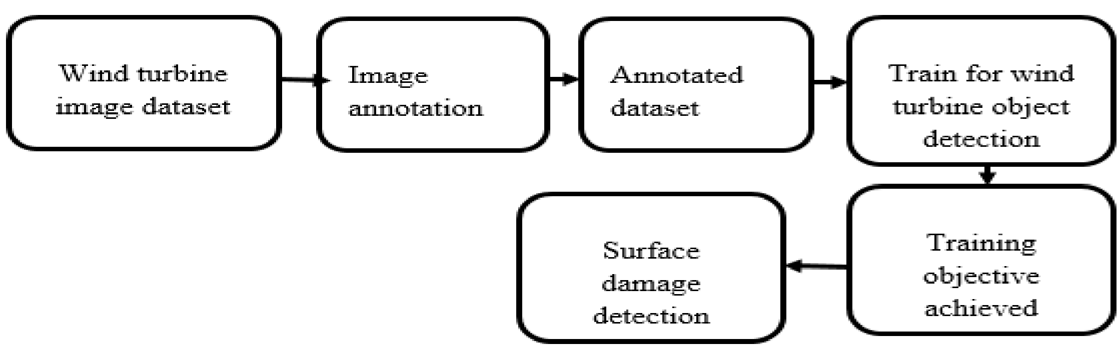


Figure 5.1 Steps for wind turbine surface damage detection

YOLO's Working Steps:

Input Image: YOLO divides an input image into a grid format.

Grid division: Divided into grids the image is used to build a grid of cells whose task is to predict what will be in each given region.

Feature extraction: A deep convolutional neural network is used by YOLO to extract features from the full image.

A form of advanced deep learning model called convolutional neural networks (CNNs) was developed specifically for the interpretation of visual input, including photographs. Using convolutional layers to extract features from the input data, pooling layers to minimize spatial dimensions while maintaining important information, and fully connected layers to integrate derived features and make high-level choices, it uses a layered structure. CNNs are effective at tasks like object detection and picture recognition because these fully connected layers use the retrieved characteristics to make predictions.

Prediction of Bounding Boxes: YOLO makes bounding box predictions for every grid cell.

- Every bounding box is uniquely recognized by its x and y center coordinates. The bounded box's height (h) and width (w).
- Confidence score: a measure of the likelihood that an object is inside the bounding box.

Class Prediction: YOLO forecasts the class probabilities for the objects found in each cell.

Output: The remaining bounding boxes, together with the matching class labels and confidence scores, are the final product of YOLO.

Figure 5.2 Sampled data before annotating (Foster et al., 2022b)

5.2.2 Description Quadrotor for Wind Turbine Inspection

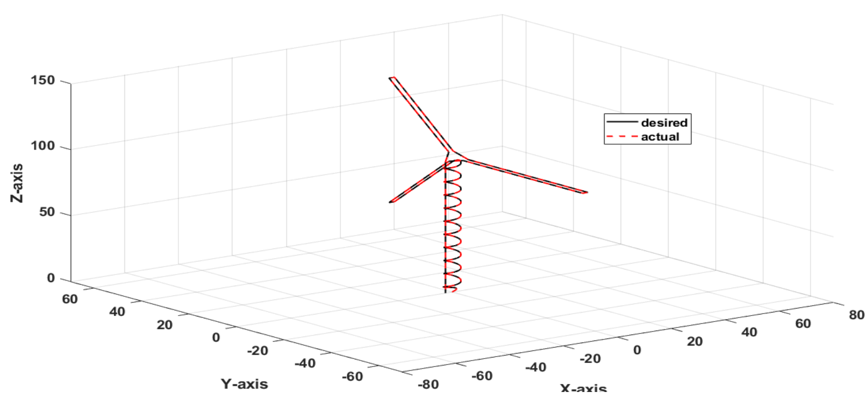


Figure 5.3 quadrotor trajectory around wind turbine

Assumption taken during trajectory generation:

- The trajectory is generating for single wind turbine we can use for many quadrotors which is found in specific area step by step to inspect the turbine and capture photos.
- The number of turbine inspection depend on the quadcopter battery and power capability however it's not considered in this work.
- Quadrotor is small distance apart from tower and blades i.e. not contact during

Drone specification for wind turbine inspection

DJI Mavic 3 Enterprise is a good alternative if you're searching for a reasonably priced drone inspection solution for wind farms. The Mavic 3 has a little camera, but it has a powerful wide-angle micro-four-thirds CMOS camera that can capture clear images quickly, even in low light. A dual system with a DFOV: 61° thermal camera is another option if you wish to visually check for overheating in the mechanical and electrical parts of the turbine.

Principal attributes for this quadrotor:

- With a LiPo 4S 5000 mAh battery, you can fly for up to 45 minutes.
- Wide angle 4/3 CMOS camera with 56X hybrid zoom
- There is a 40mm option for a thermal camera with 640 x 512 pixels.
- Three-axis camera gimbal with a tilt range of 90°–35°
- Automatically rerouting around obstructions and preventing collisions

CHAPTER SIX

6. SIMULATION RESULT AND DISCUSSION

6.1 Introduction

There are two main components to this chapter. The test findings from using the STSMC with PID surface to control the quadcopter in a variety of trajectory circumstances are examined in the first section. The results of image processing, the trained system's method to identify surface degradation on wind turbines, and the ensuing conclusions are all thoroughly explained in the second section.

Tunned parameter setting

PSO tuned parameters of the SMC with PID surface controller using 0 as a lower bound and 5 as an upper bound are given below.

Table 6.1 PSO tuned parameters of controller

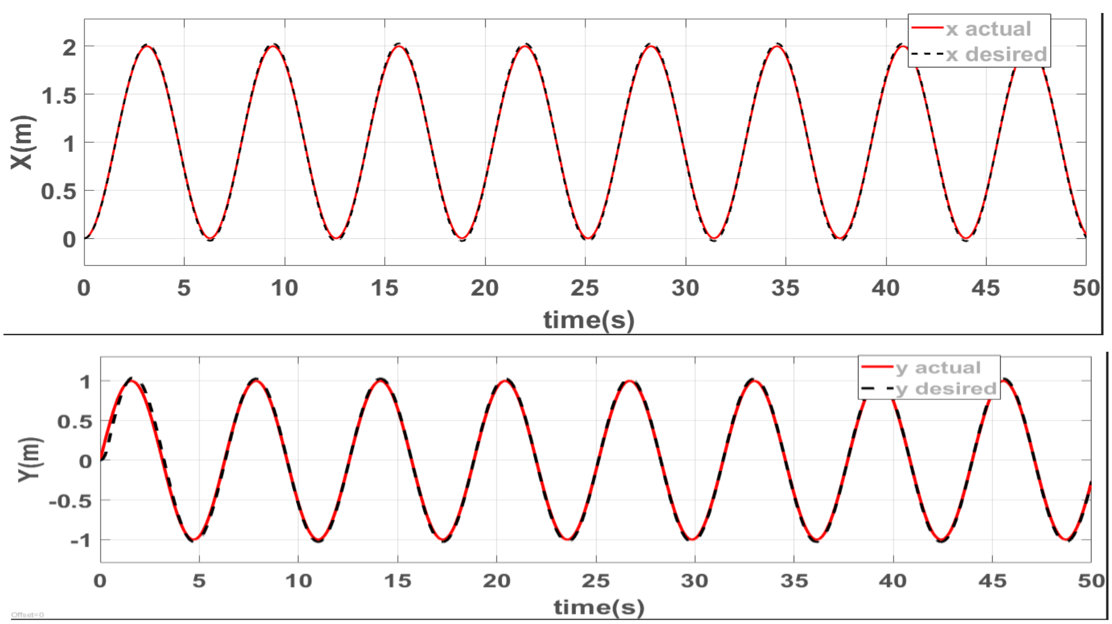
parameters	values	parameters	Values
$k_{px} = k_{py} = k_{pz}$	3.1694	$k_{p\phi} = k_{p\phi} = k_{p\phi}$	2.6119
$k_{ix} = k_{ix} = k_{ix}$	4.9704	$k_{i\theta} = k_{i\theta} = k_{i\theta}$	4.2532
$k_{dx} = k_{dx} = k_{dx}$	4.6508	$k_{d\psi} = k_{d\psi} = k_{d\psi}$	1.3372
$k_x = k_y = k_z$	3.9758	$k_\phi = k_\theta = k_\psi$	4.2753
σ	0.2	σ	0.2

6.2 SMC and Combined SMC and PID Controller Result

MATLAB/SIMULINK-created graphical representations of simulation results shed light on the functionality of the Optimal SMC with a surface PID. The robustness of Optimal

SMC and the adaptability of the PSO algorithm are combined in this controller. Testing involves circular, rectangular, and spiral infinity trajectories both with and without disruption. Furthermore, a comparative study is conducted between spiral cylindrical trajectories that have parameter adjustments and those that do not.

6.2.1 Spiraling Cylinder Trajectory: The following represents the tracking reference: $Xr = 1 - \cos(t)\,, Yr = \sin(t)\ and\ Psi\_r = \sin(t)\,,\ Zr = t.$ Applying the PSO technique to acquire optimal gains of the quasi SMC with PID controller to track spiral cylindrical motion by iteratively running and picking the optimal result is illustrated in figure 6.4, which displays the resulting 3D helical trajectory tracking response. X, Y, and Z position trajectories are displayed in Figure 6.1, the above figure demonstrating the quadrotor's effective ability to follow the provided X , Y and Z position command. Integral time absolute value errors (ITAE) for the X and Y position trajectories were 0.0804 and 0.136, respectively, for the specified simulation time. Likewise, the quadrotor tracks the specified ramp height with an integral time absolute value inaccuracy of 0.017 and 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 6.2.



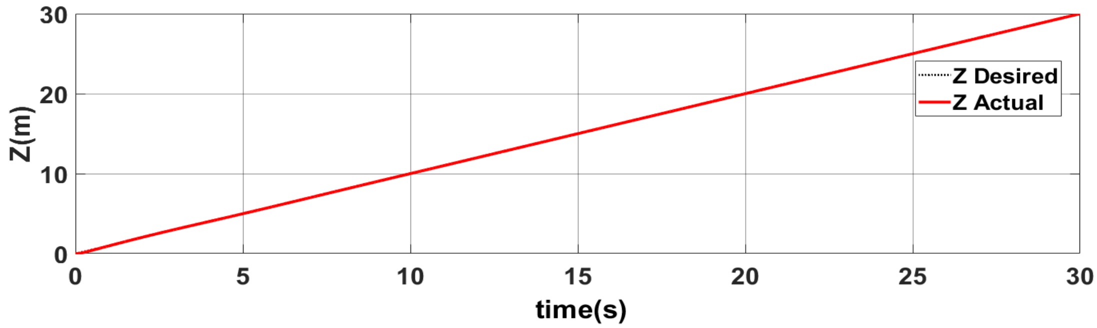


Figure 6.1 the translational response when it follows the spiral cylindrical trajectory

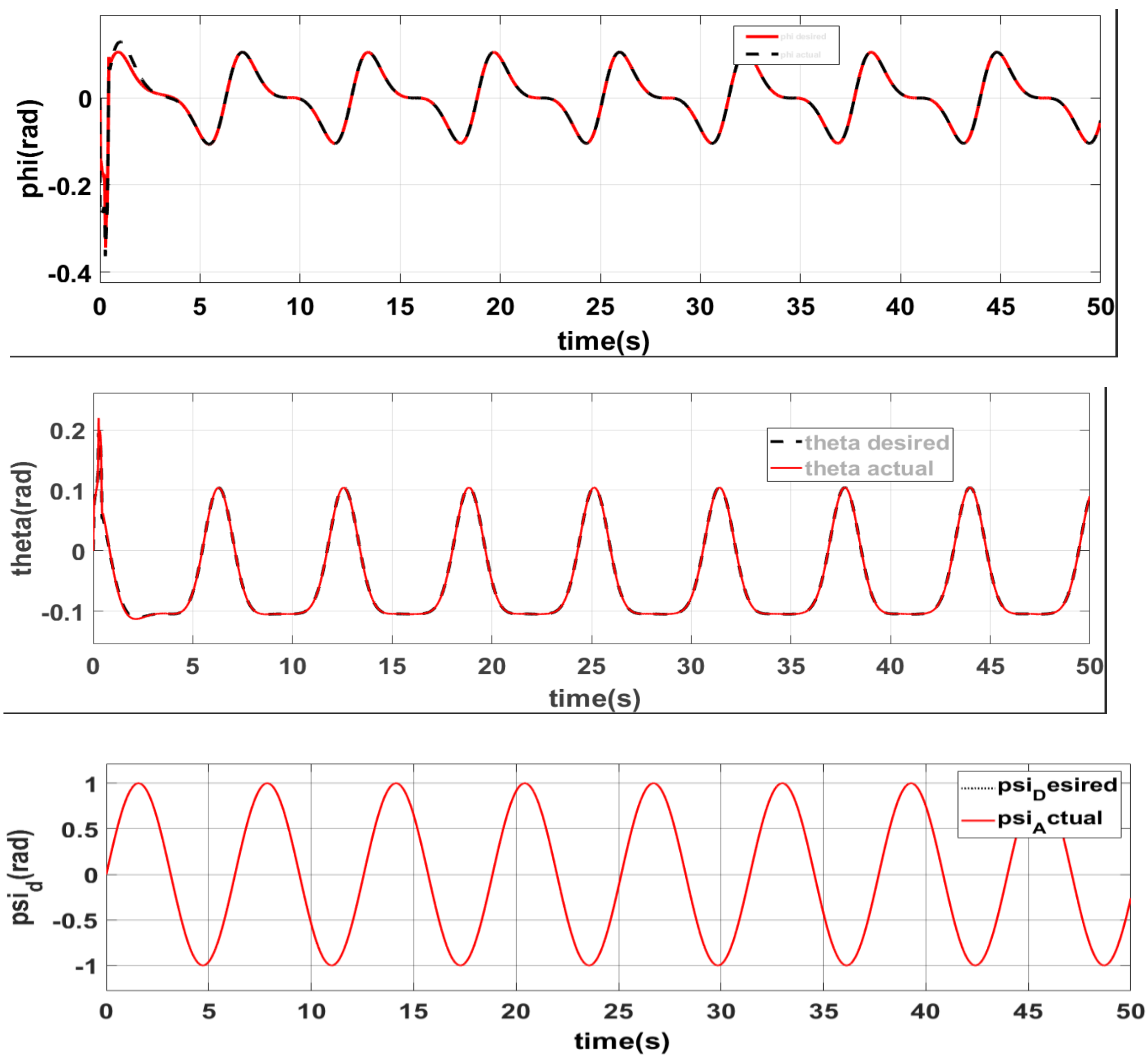


Figure 6.2 the translational response when it follows the spiral cylindrical trajectory.

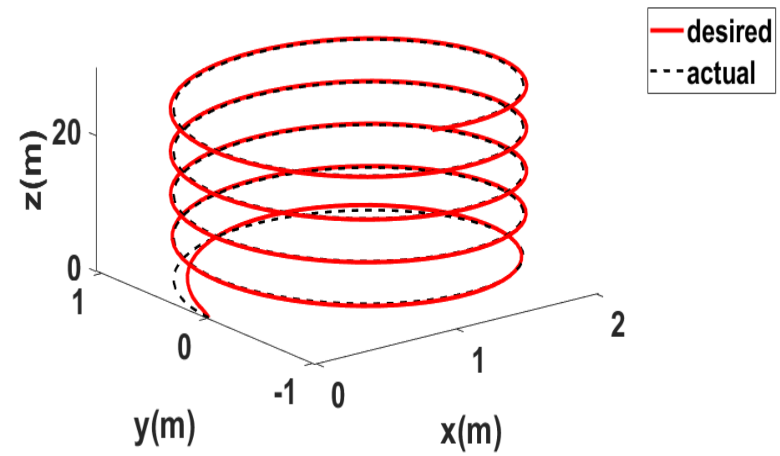


Figure 6.3 3D spherical cylindrical trajectory

Control effort: From Figure 6.4 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.

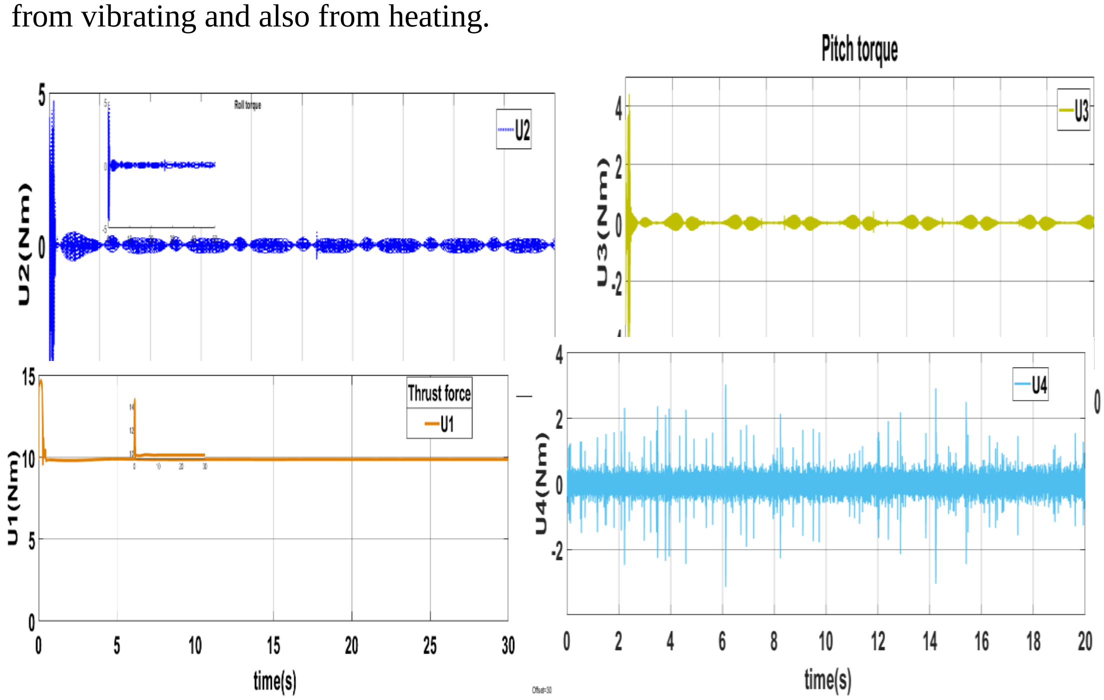


Figure 6.4 control input to create spherical cylindrical trajectory trucking

6.2.2 Tracking of Infinite Eight Shape Trajectory

This tracking issue commands the quadrotor to follow the trajectory of an infinite eight shape at a height of two meters, generated by $Xr = \cos(t)\,, Yr = 0.5 * \sin(2t), Zr = 3)$ $and\ \psi r = 0,$ The resulting X, Y, and Z position trajectories are displayed in Figure 6.5, which demonstrates how well the quadrotor can follow the given X and Y position command. Similarly, the quadrotor hovers at three meters above the ground, the commanded height.Figure 6.6 displays the roll and pitch angles that can produce the specified psi angle as well as the commanded X and Y positions, respectively. The quadrotor produces the roll and pitch angles required to precisely establish the Y and X positions, as seen in the picture, and it successfully executes the specified yaw angle command.

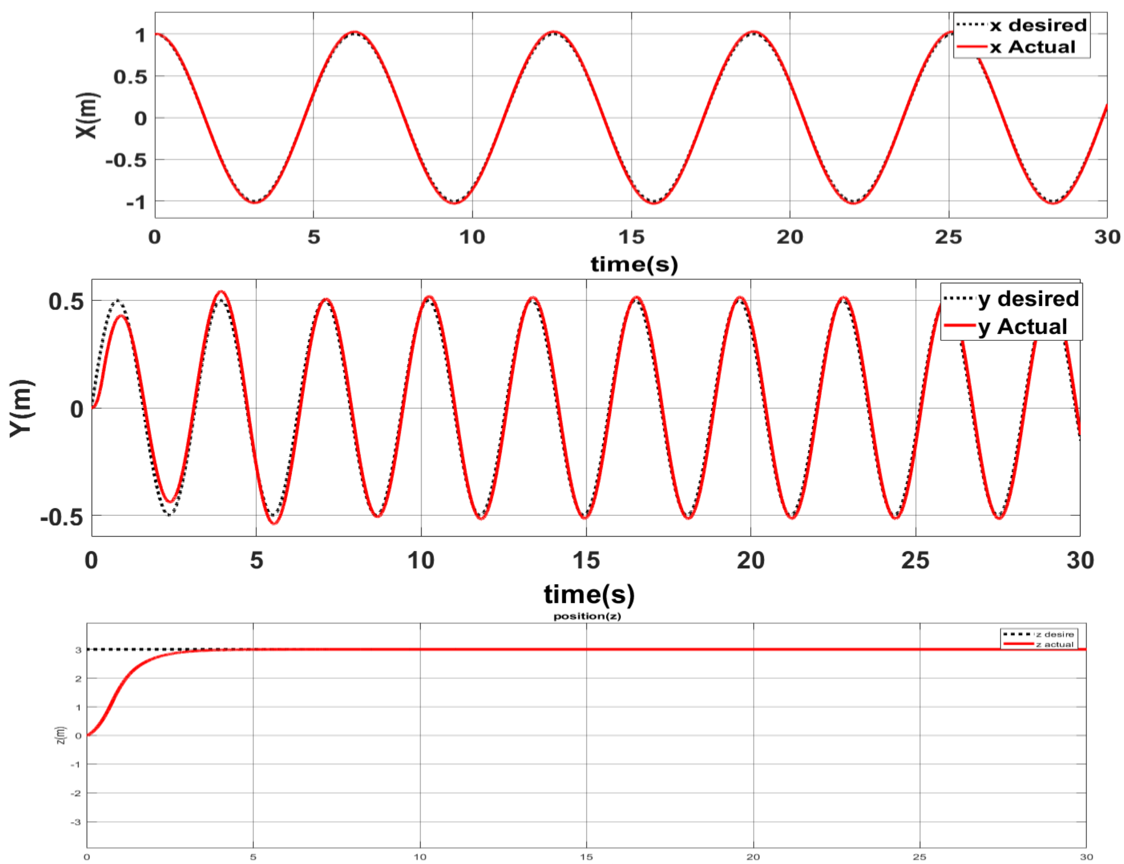


Figure6.4 Translational response for eight shape trajectories

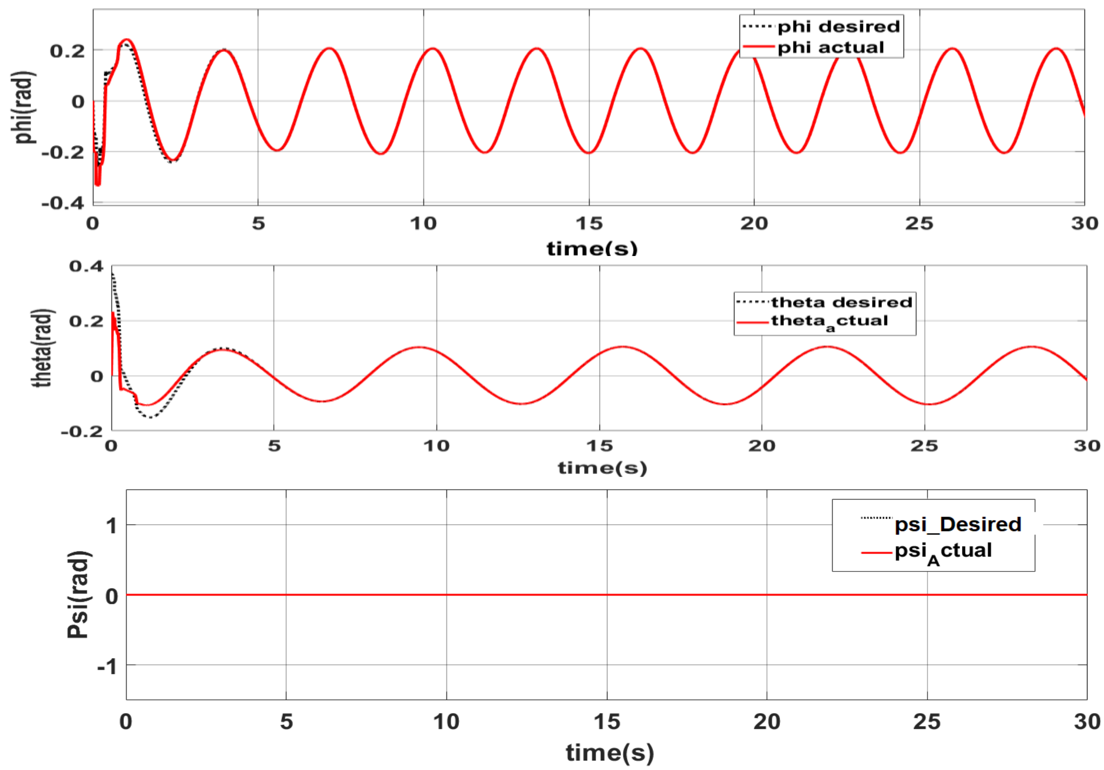


Figure 6.6 Rotational subsystem's response for tracking an end eight-shape trajectory

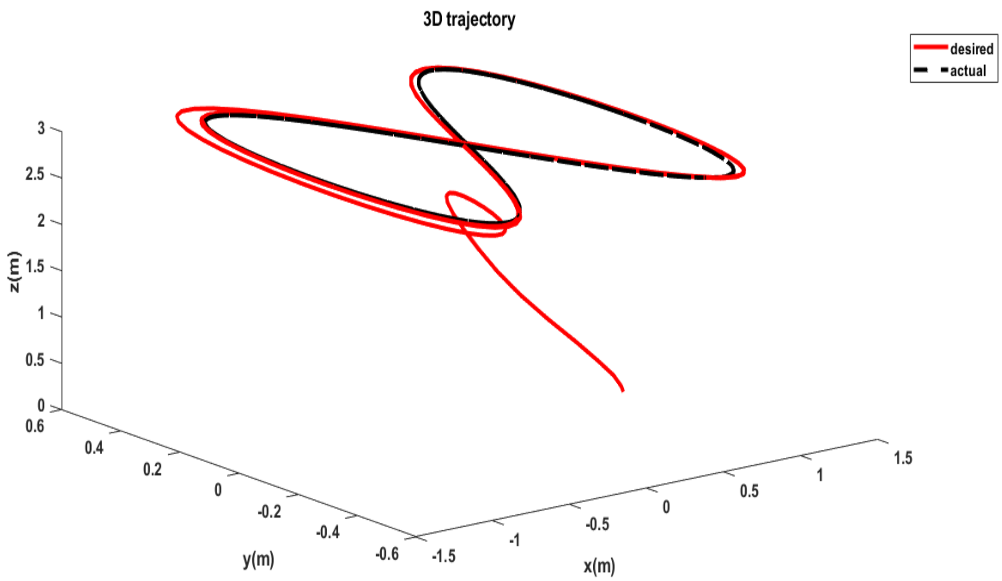


Figure 6.7 An eight-shaped trajectory in 3D

6.2.3 Spiral Infinity Trajectories: The tracking reference in space navigation was an infinite spiral trajectory. The X and Y axes received sinusoidal input trajectories, whereas the Z axes received ramp input. The tracking reference is represented by the following: $Xr = 1 - \cos(t)\,, Yr = 0.5\sin(2t)\ and\ psi_r = 0,, Zr = t.$ The gain are tuned by Particle swarm optimization algorithm. The X, Y, and Z position trajectories have integral absolute value errors of 0.08188, 0.1602, and 0.0185 during the specified simulation time, respectively. Figure 6.9 displays the commanded psi angle. This picture makes it evident that the degrees of roll and pitch that can to precisely generate the positions of X and Y are produced by the quadrotor. The integral absolute value errors for the roll, pitch, and yaw Euler angles are for the specified simulation duration 0.02342, 0.0000006825, and 0.0000000000687 respectively.

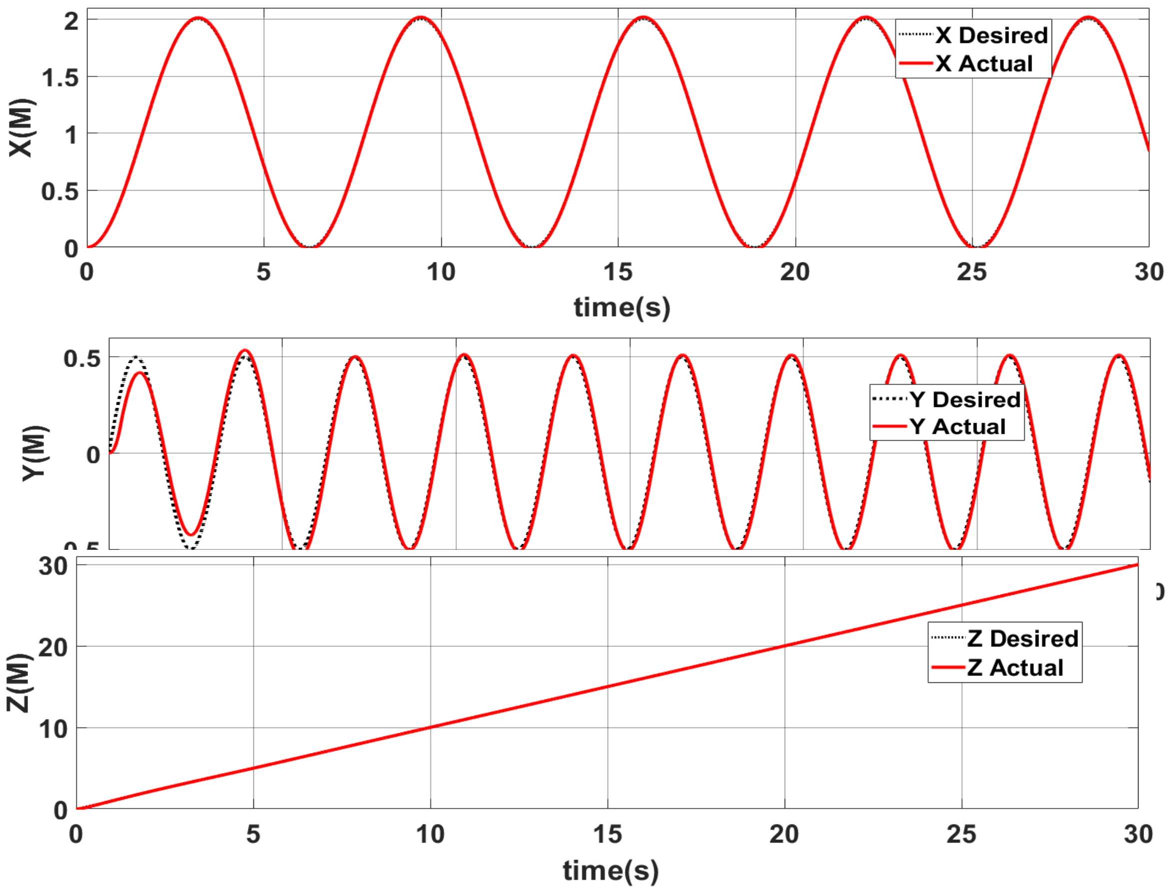
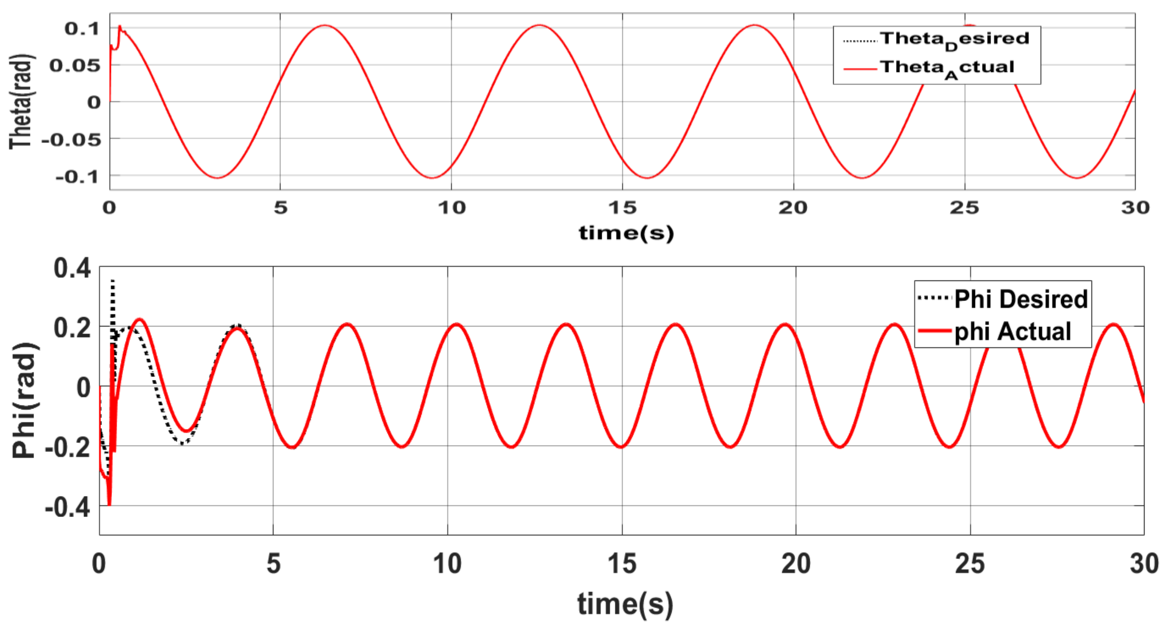


Figure 6.7 Response of the translational for following the spiral infinity trajectory



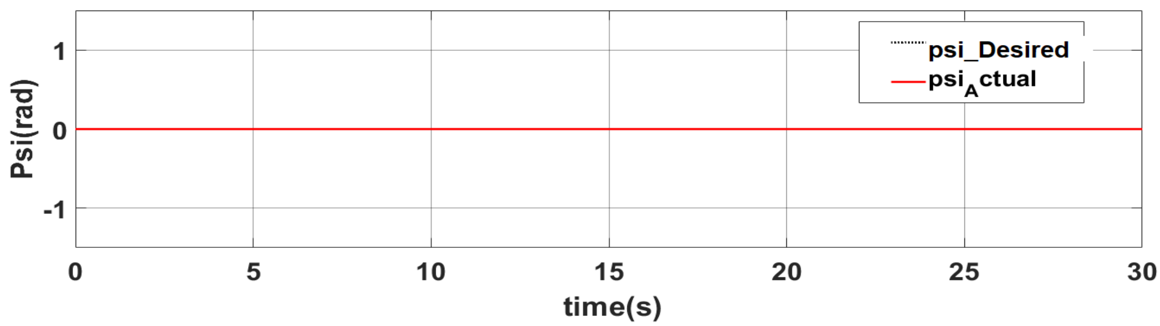


Figure 6.9 Rotational subsystem response for spiral infinity trajectory

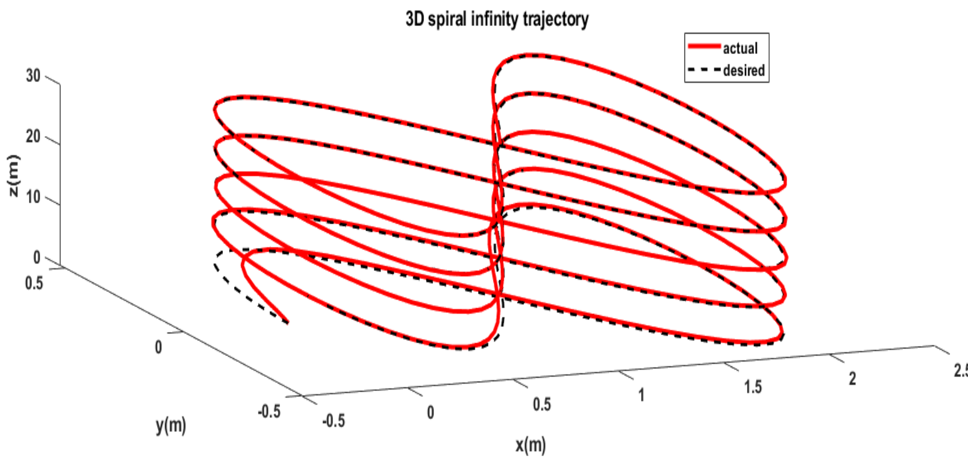
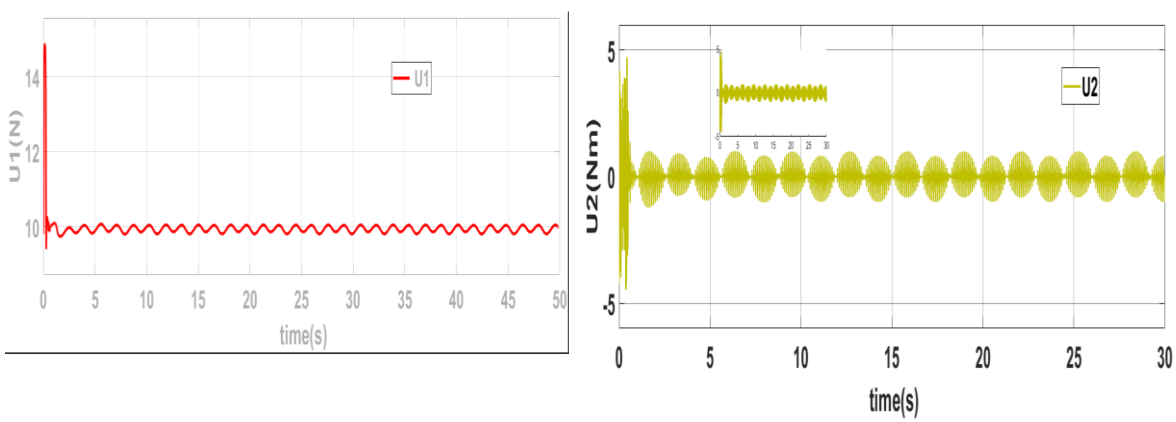


Figure 6.10 3D Spiral Infinity Trajectory Tracking

The control actions needed to generate the intended trajectories are displayed in Figures 6.10, where a very slight chattering is shown that keeps the quadrotor from heating up and vibrating.



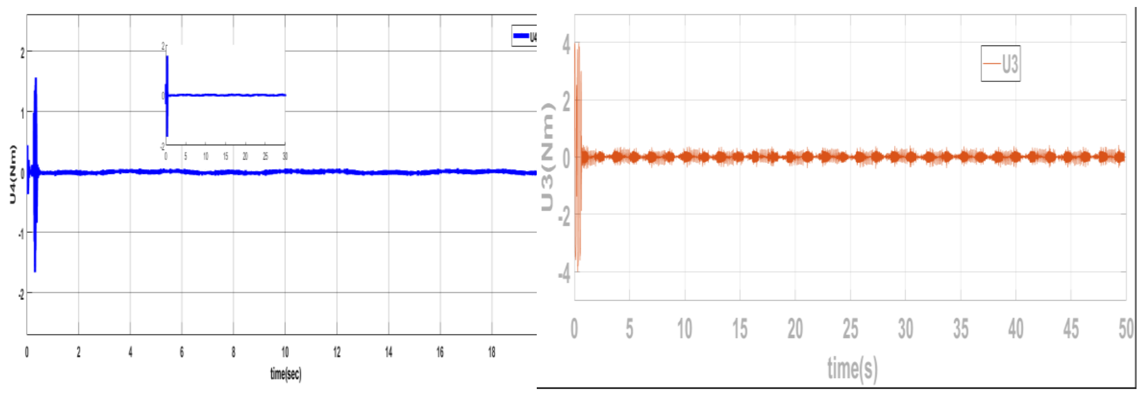


Figure 6.11 control effort for spiral infinity trajectory

6.3 Application-based trajectory (helical around tower and Rectangular (constant velocity) around wind turbine blade

The following are typical dimensions for a 1.5 MW wind turbine.

Wind Turbine: Vertical distance: about 80–100 meters - Base diameter: about 3–4 meters

- Depth: Depends on the layout of the foundation, usually 2 to 3 meters.
- Trajectory is generated for single wind turbine.
- **Hub** - Length: roughly two to three meters.
- Breadth: roughly two to three meters.
- Depth: Between one and two meters.
- **Blade:** Approximately 40 to 50 meters in length, Width: Varies from 3 to 5 meters depending on the blade and Depth: Varies from 0.5 to 1 meter along the blade.

To design_trajectory$(\mathrm{t,T}, \mathrm{a_x}, \mathrm{a_y}, \mathrm{a_z}, \mathrm{b_x}, \mathrm{b_y}, \mathrm{b_z})$ be defined such that it computes the trajectory coordinates (x, y, z) based on time t and the intervals T. The overall design trajectory can be summarized as follows:

1. for t in the first interval (helical path for wind turbine tower):

$$(x, y, z) = \left(2.5 \cdot \cos\left(\frac{20\pi}{T(1)} t\right), 2.5 \cdot \sin\left(\frac{20\pi}{T(1)} t\right), (b_z(1) - a_z(1)) \cdot \left(\frac{t}{T(1)}\right)\right)$$

2. for subsequent intervals (for constant velocity trajectory for blades):

$$(x, y, z) = \left(a_x(i) + \frac{b_x(i) - a_x(i)}{\text{duration}} \cdot (t - T_0), a_y(i) + \frac{b_y(i) - a_y(i)}{\text{duration}} \cdot (t - T_0), a_z(i) + \frac{b_z(i) - a_z(i)}{\text{duration}} \cdot (t - T_0)\right)$$

Where i indicates the current interval based on the total time T. These measurements may change depending on the particular turbine types and producers. See trajectory generation code and mathmatical discription in Appendix B1.

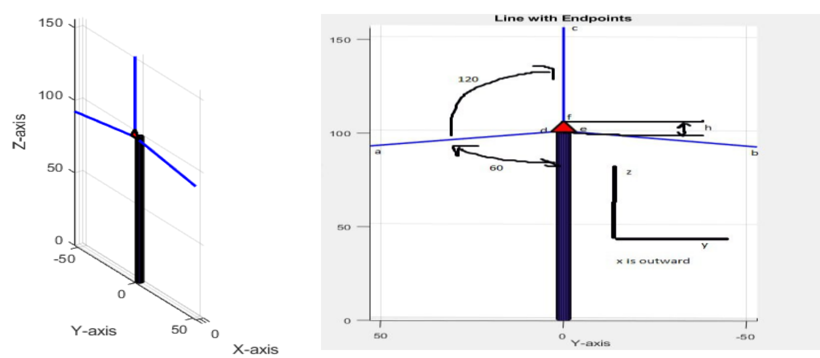


Figure6.12 3D and 2D simulation view of wind turbine

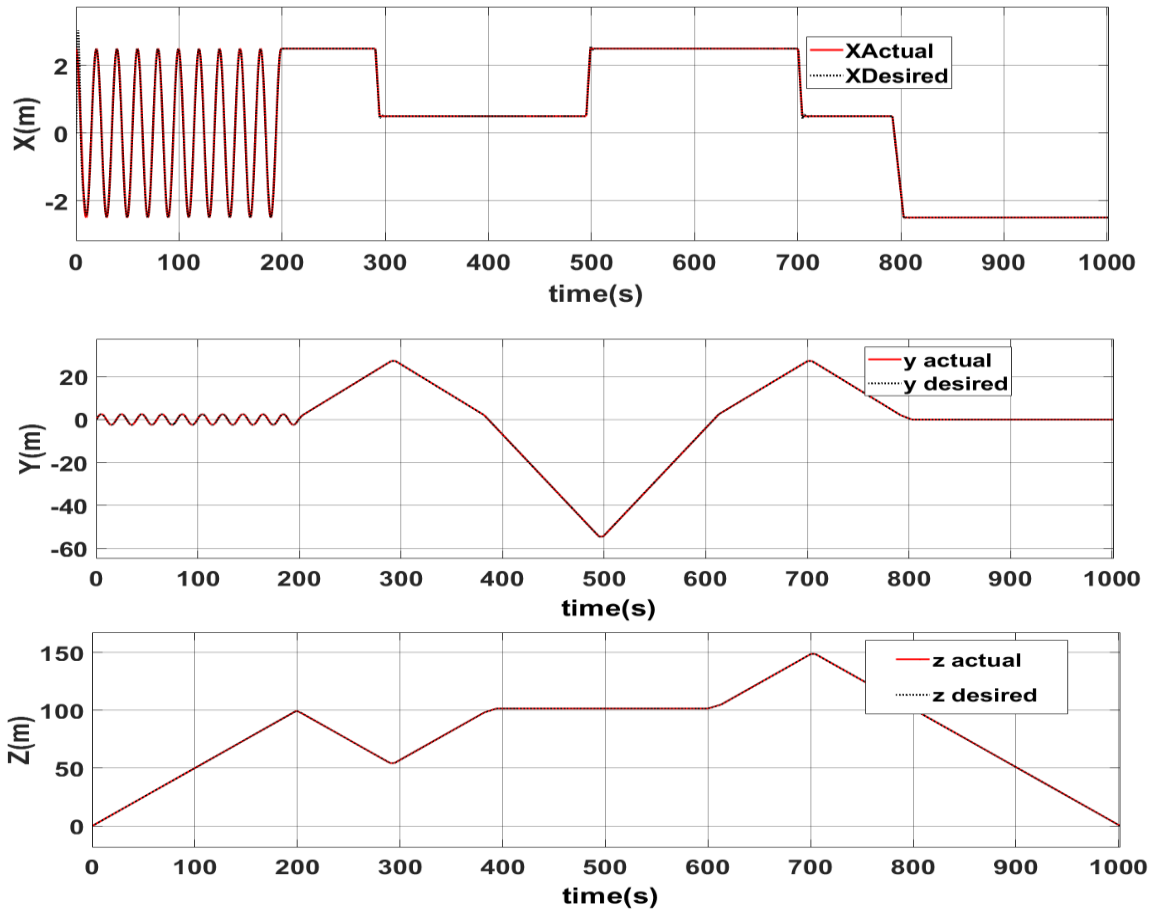


Figure 6.13Translational response helical for tower and constant velocity for blades trajectory

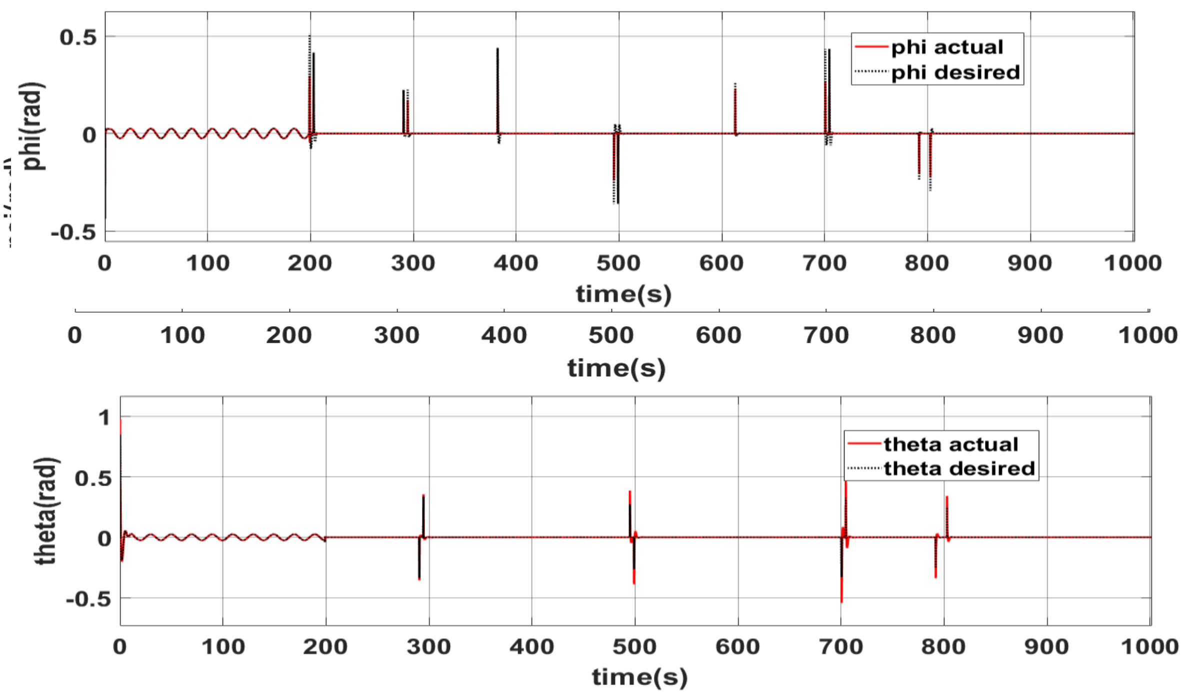


Figure 6.14 Rotational response helical for tower and constant velocity for blades

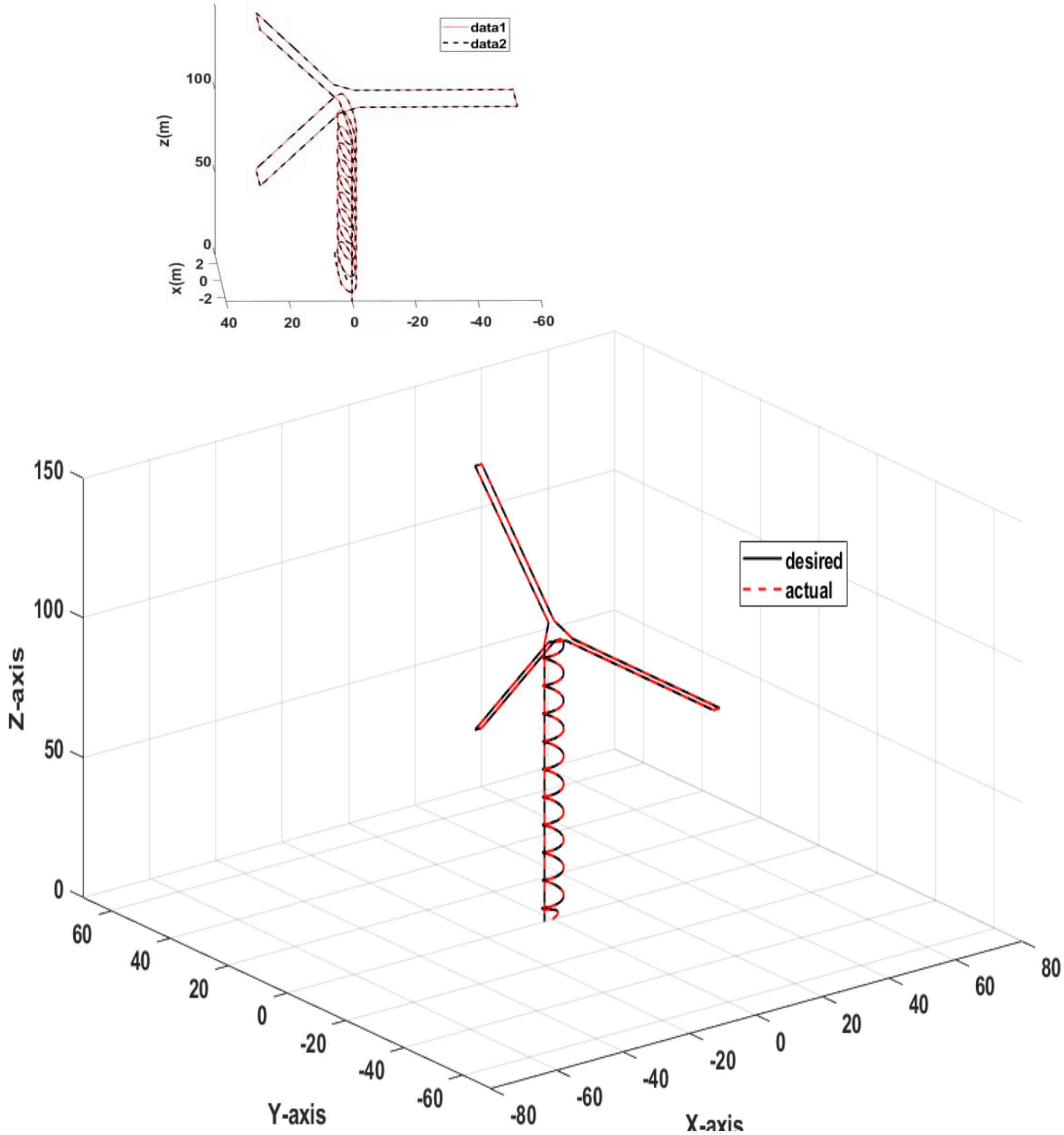


Figure 6.15 3D Response for helical for tower and constant velocity trajectory for blades

Control effort: The control efforts required to produce the desired trajectories are shown in Figures 6.16 and 6.17, where a very slight chattering is shown that keeps the quadrotor from heating up and vibrating

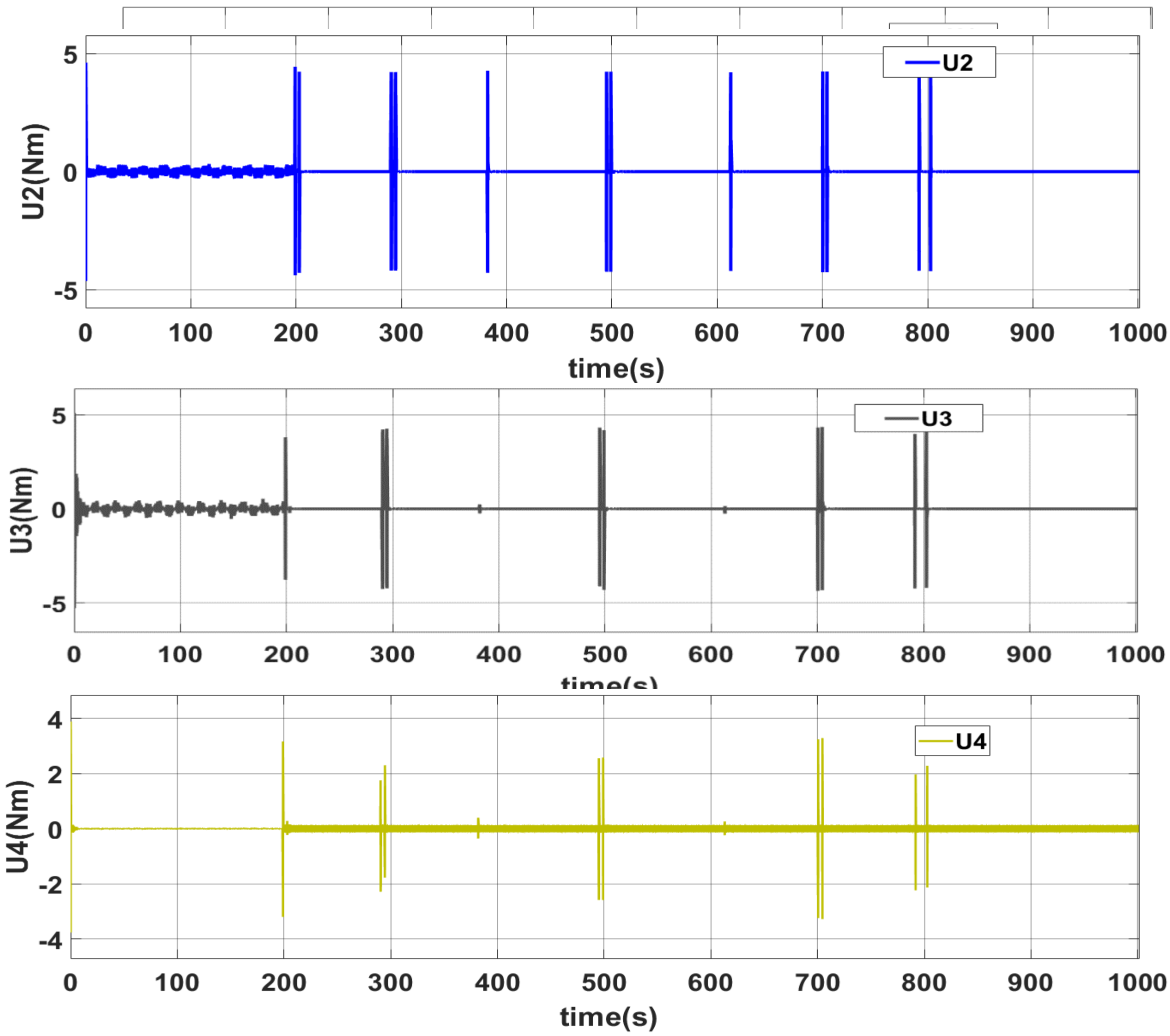


Figure 6.16 Altitude and attitude controller effort response of quadrotor

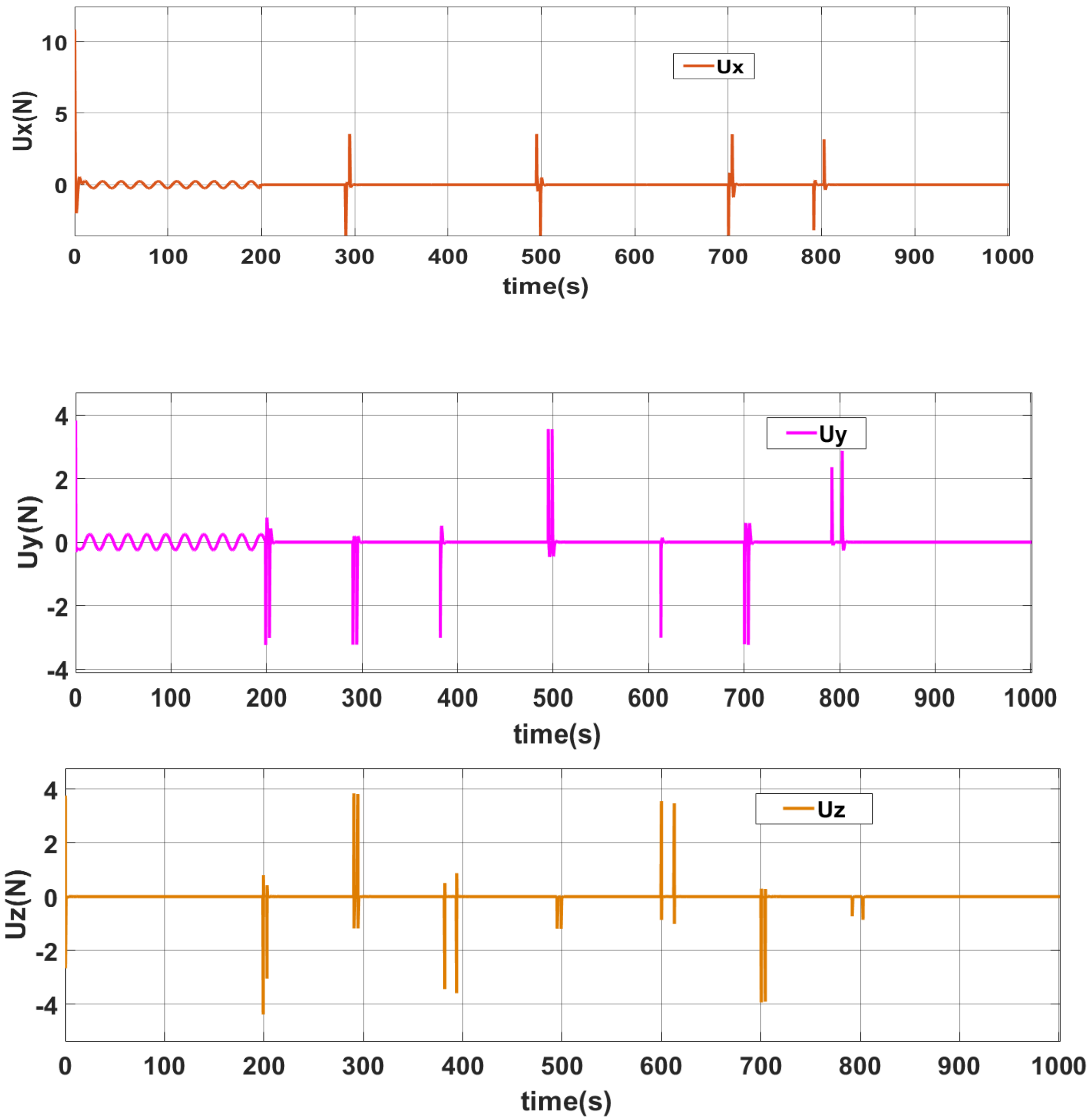


Figure 6.17 Virtual controller reaction for trajectory of location

Figure 6.13 clearly shows that the suggested controllers perform well (have less mistakes) for controlling positions X, Y, and Z. Furthermore, the control plan makes sure the quadrotor keeps its attitude angles stable so it can track the intended configuration. The system's 3D response, as obtained by the suggested way, is displayed Figure 6.17. The planned roll (phi), pitch (theta), and attitude controller U_1 are calculated using these signals. The task manager sets the necessary yaw angle and altitude, which are managed by the yaw angle controller U4 and the altitude controllerU_1, respectively. On the other hand, the reference trajectory produced by the outer loop controller is managed by the pitch angle controller $U_3$and the roll angle controllerU_4. The total time taken by quadrotor to inspect wind turbine and come back to the ground is 1000sec or 16.66 min.

6.4 Disturbance Rejection Capacity

The effectiveness and resilience of the Optimal SMC with PID surface scheme in tracking trajectory are demonstrated by applying bounded disturbance, which is used in real-world applications to monitor response for helical for tower and constant velocity trajectory for wind turbine blades. An external wind disturbance is incorporated into the system. This disturbance has a time-varying maximum magnitude that varies from -2 to +2 in position X, -4 to +4 in position Z, and -5 to +5 in direction y. Assuming the time-varying wind pressure shown in Figure 5.21 is applied to the quadrotor system.ITAE value for x, y and z respectively are 23.93, 6.50 and 4.559 when the quadrotor is exposed to varying magnitude wind gusts, to trace a required trajectory. And for roll, pitch and yaw angle ITAE value become 39.19, 51.09 and 0.0007898 respectively.

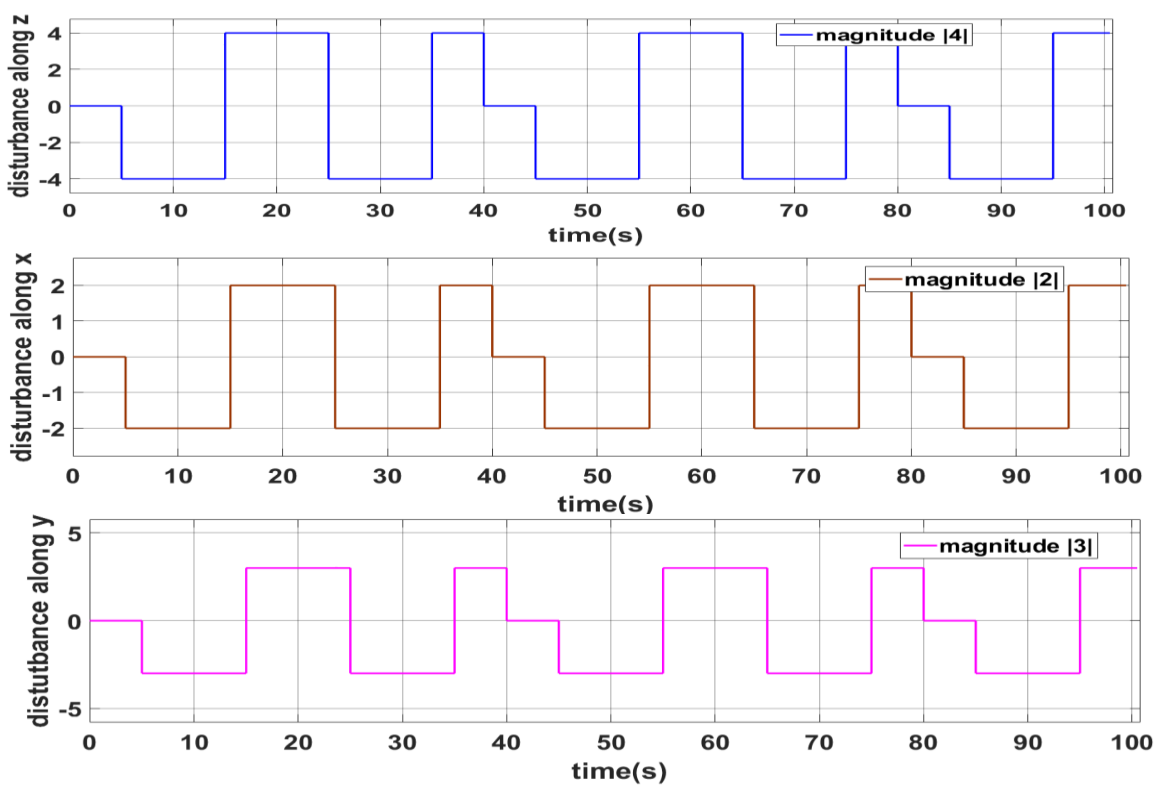


Figure 6.18 Disturbance applied along X, Y and Z

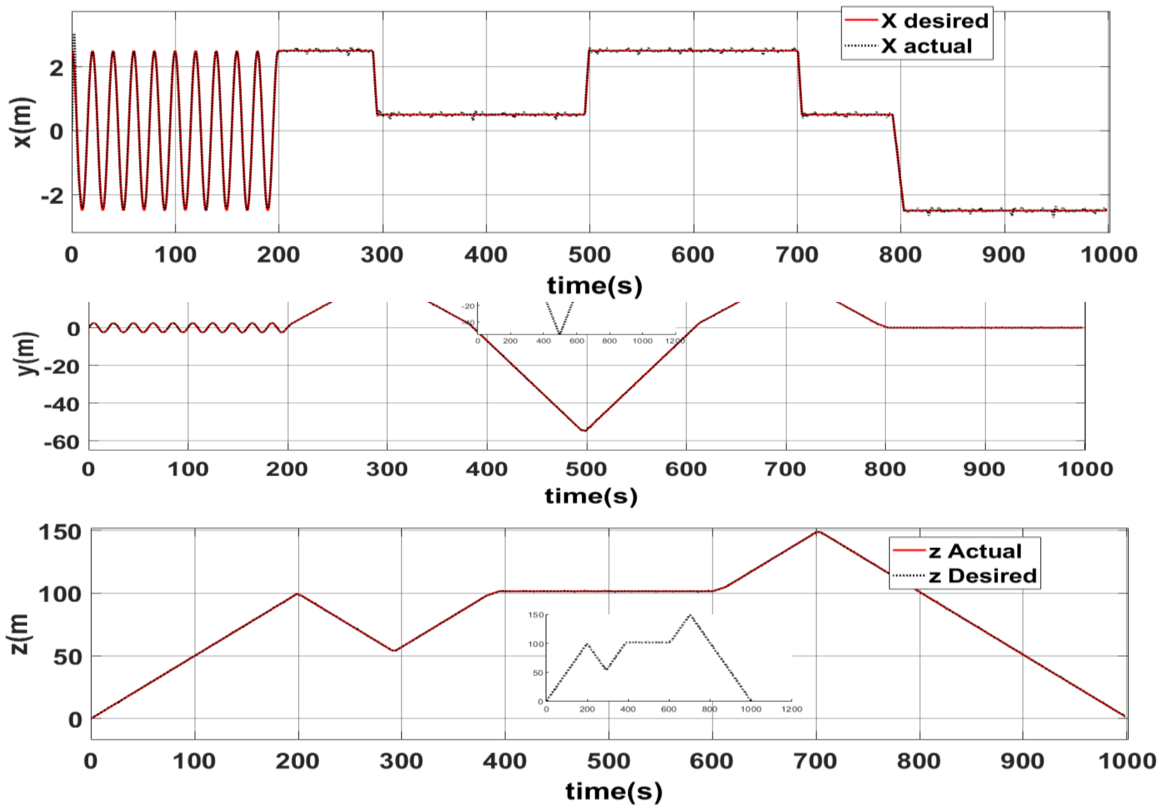
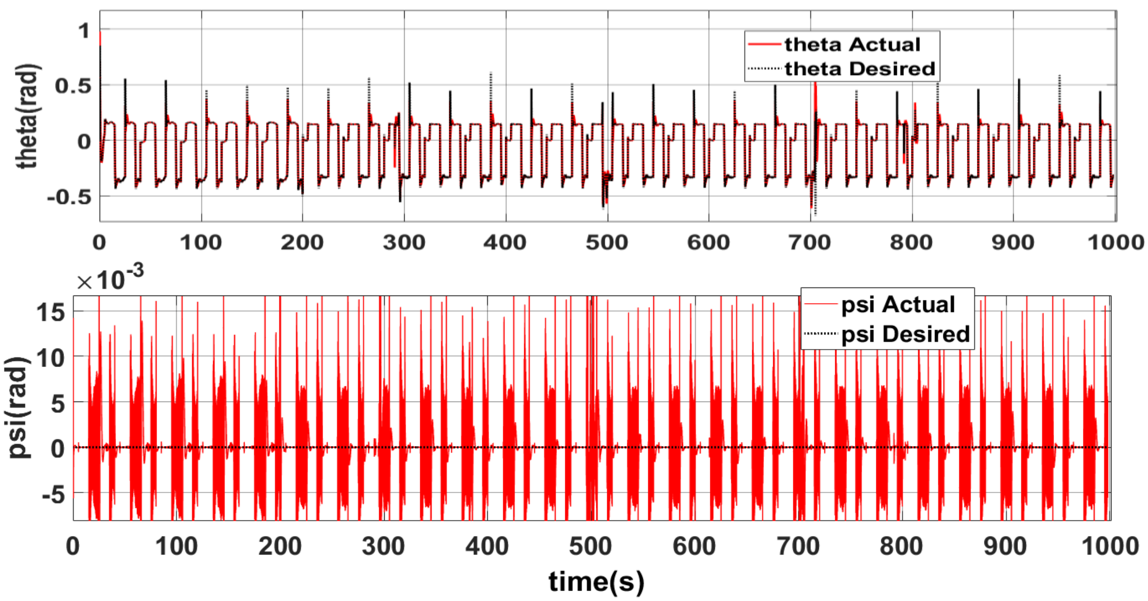


Figure 6.19 Translational response with disturbance



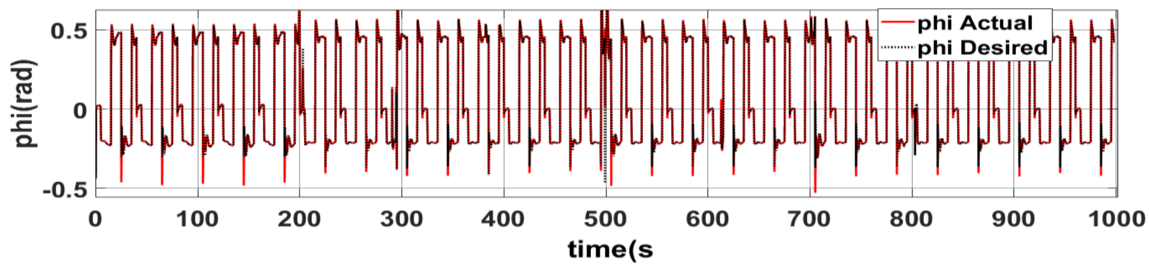


Figure 6.20 Rotational response with disturbance

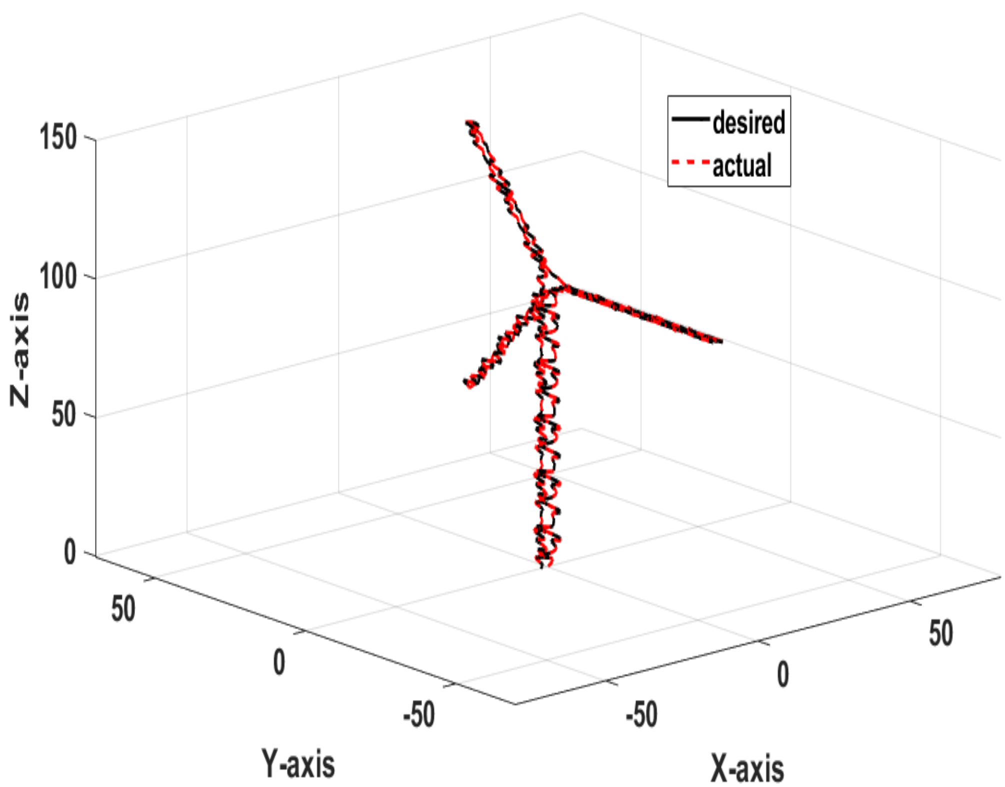
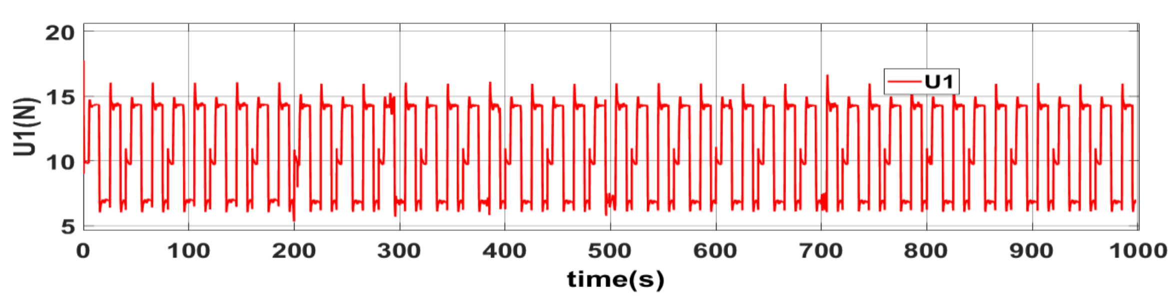


Figure 6.21 3D trajectory response with disturbance



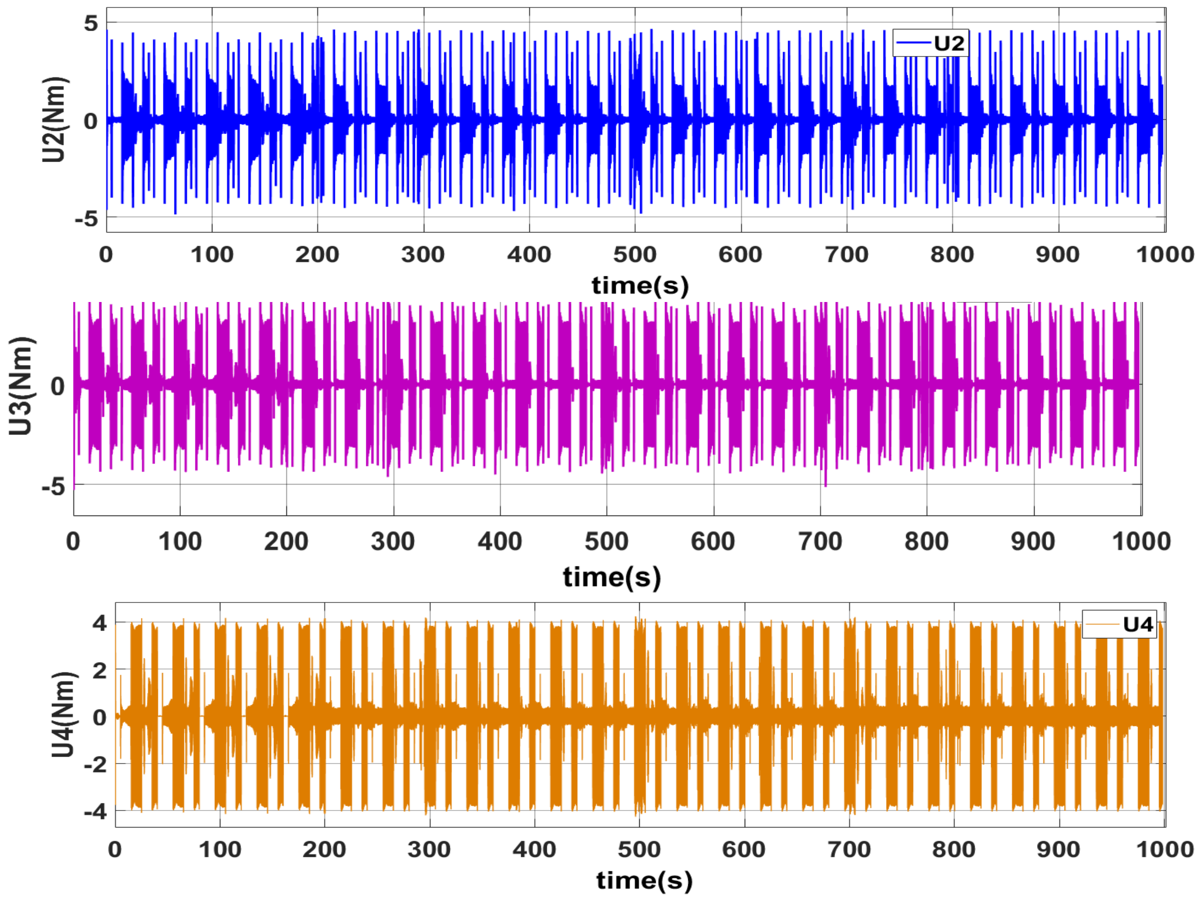


Figure 6.22 Controller effort response with disturbance

As shown from the above figure there is some vibration occurred the controller tray to stabilize the system with high disturbance rejection capability.

6.4.2 Comparison the Designed Optimal SMC with PID and Conventional SMC

A spiral inifinity trajectory is used to show how well optimal SMC with PID surface performs in comparison to SMC. Because of the complexity of this trajectory, optimizing controller performance is crucial. The input is $Xr = 1 - \cos(t)\,, Yr = 0.5 * \sin(2 * t)\,, Zr = t\ and\ psi_r = 0$. The Optimal SMC with PID sliding surface simulation result demonstrates fast state convergence to the intended reference signals and good tracking performance. However, as Figures 6.23 and 6.24 show, the sliding mode controller performs poorly and converges slowly. The recommended controller reduces chattering in the SMC control inputs, as seen in Figures 6.25.

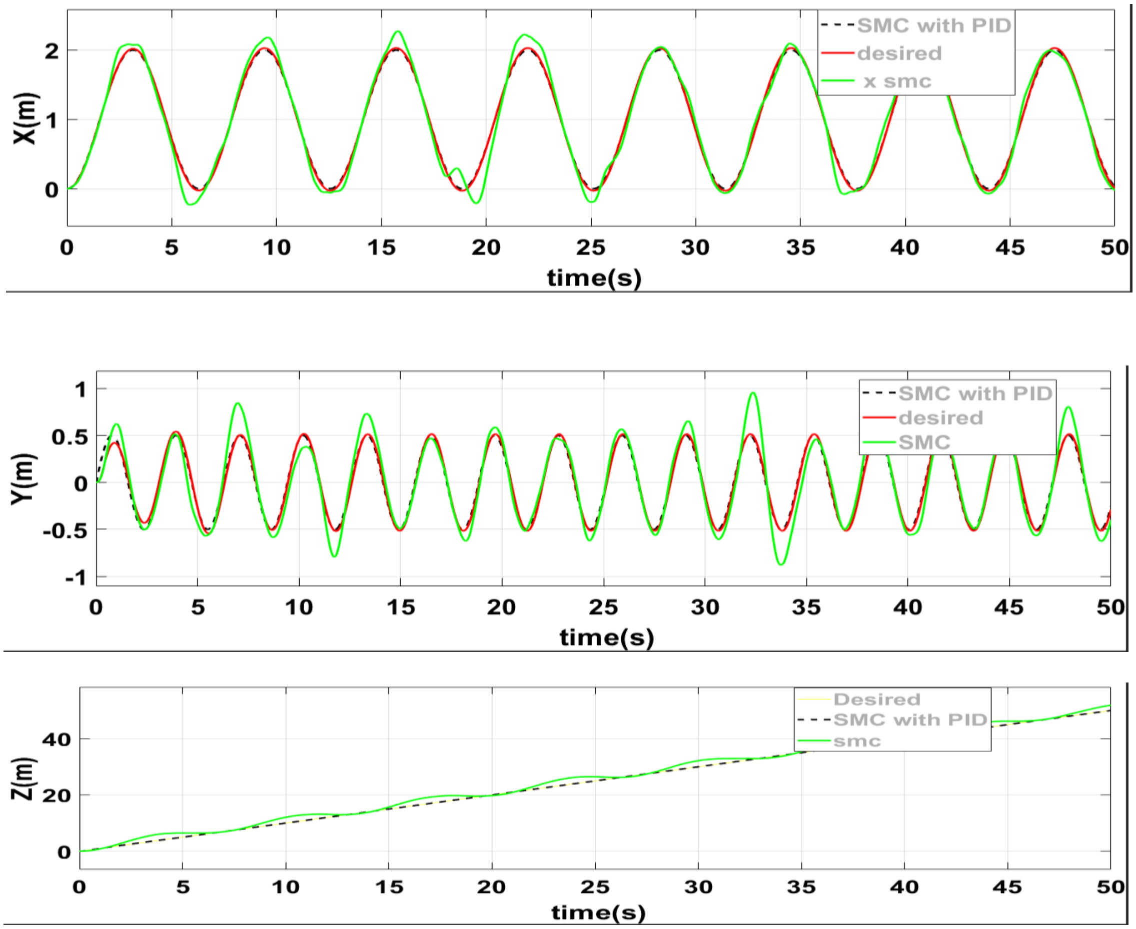
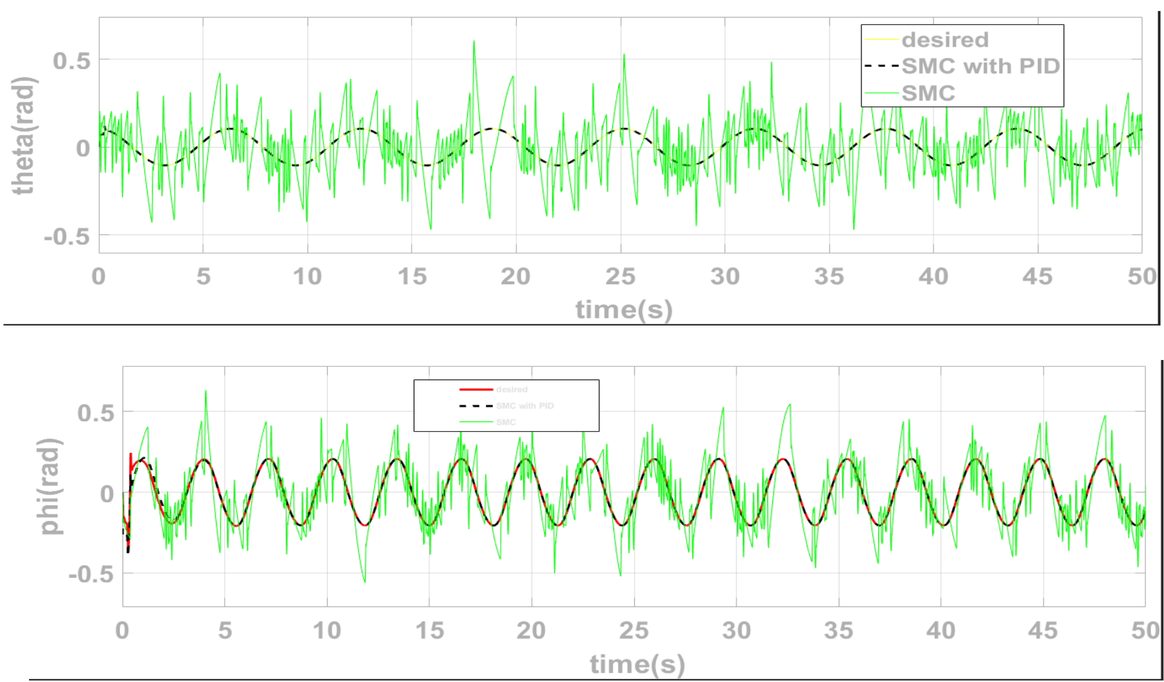


Figure 6.23 Translational response for spiral infinity trajectory



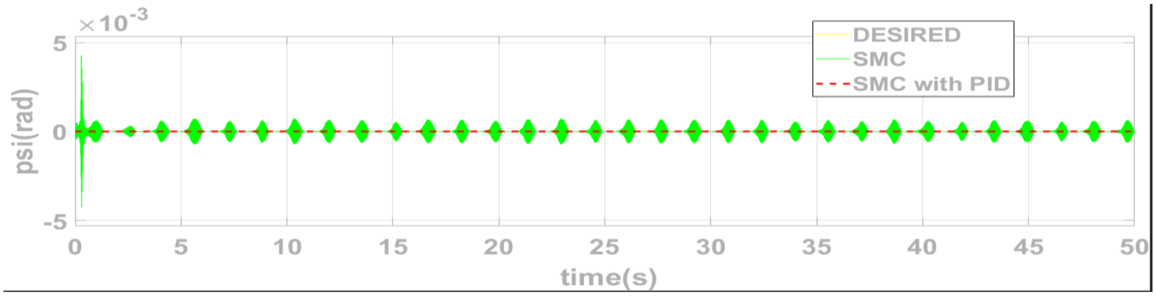


Figure 6.24 Rotational response for spiral infinity trajectory

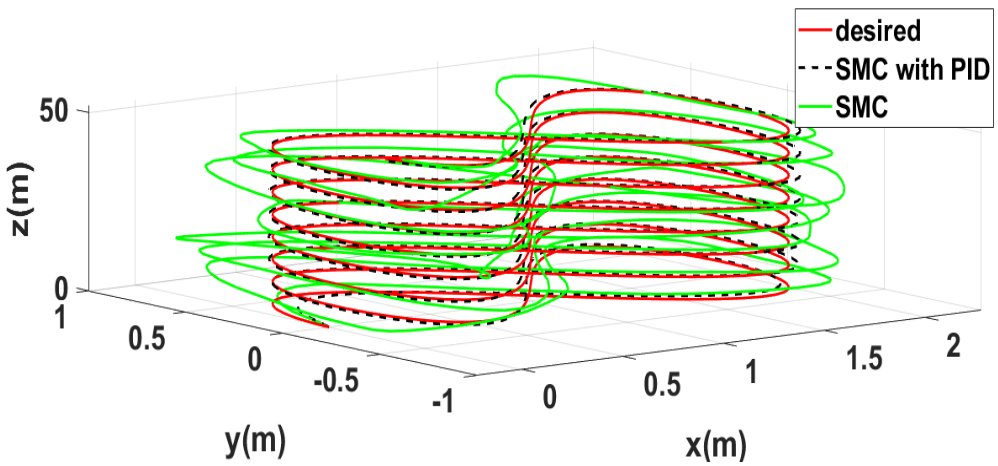
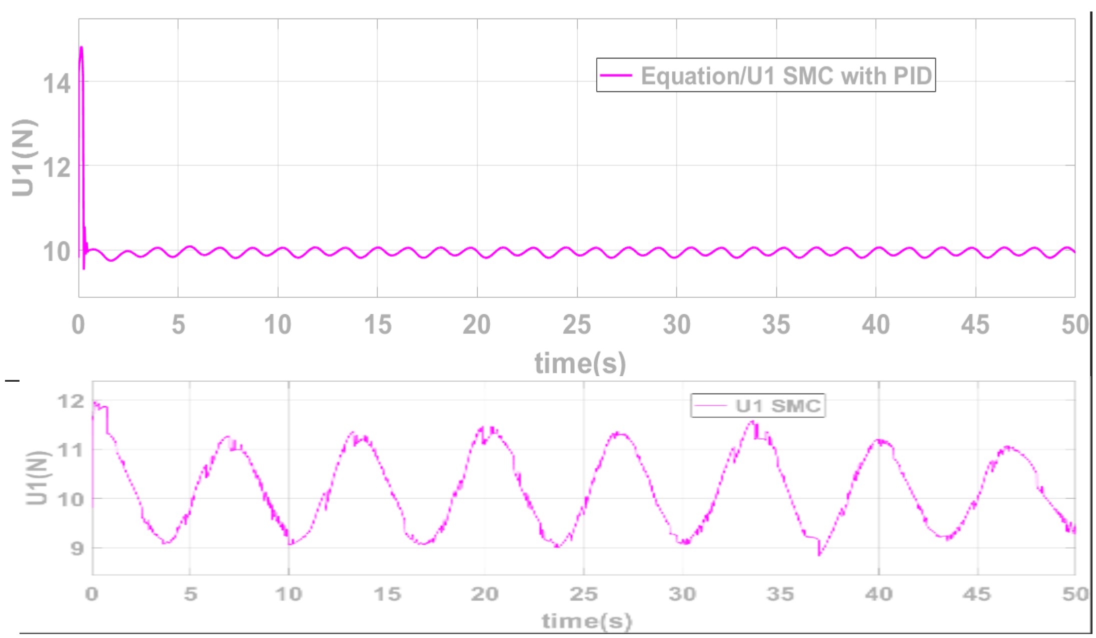


Figure 6.25 3D Response for spiral infinity trajectory



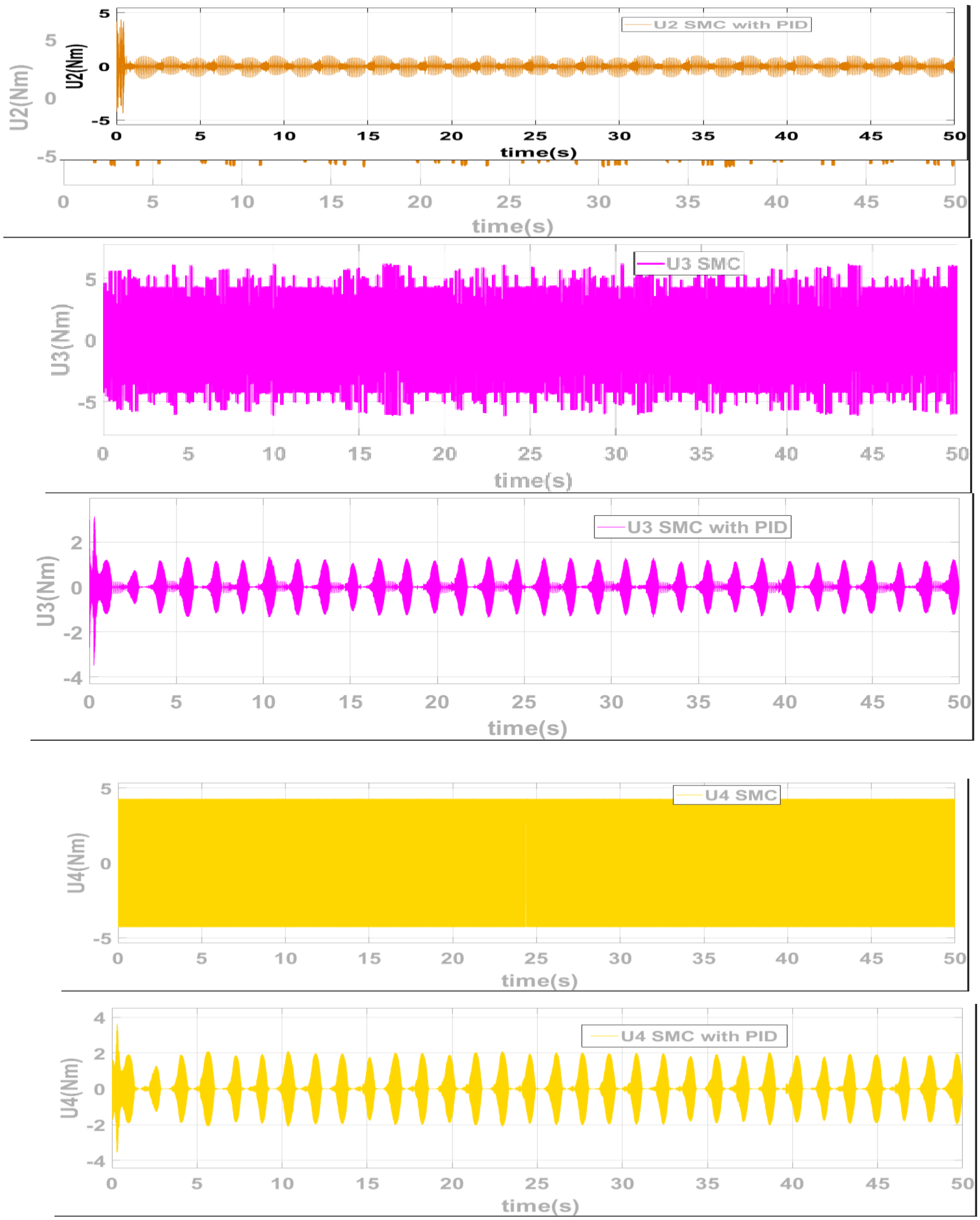


Figure 6.26 Control effort response for spiral infinity trajectory

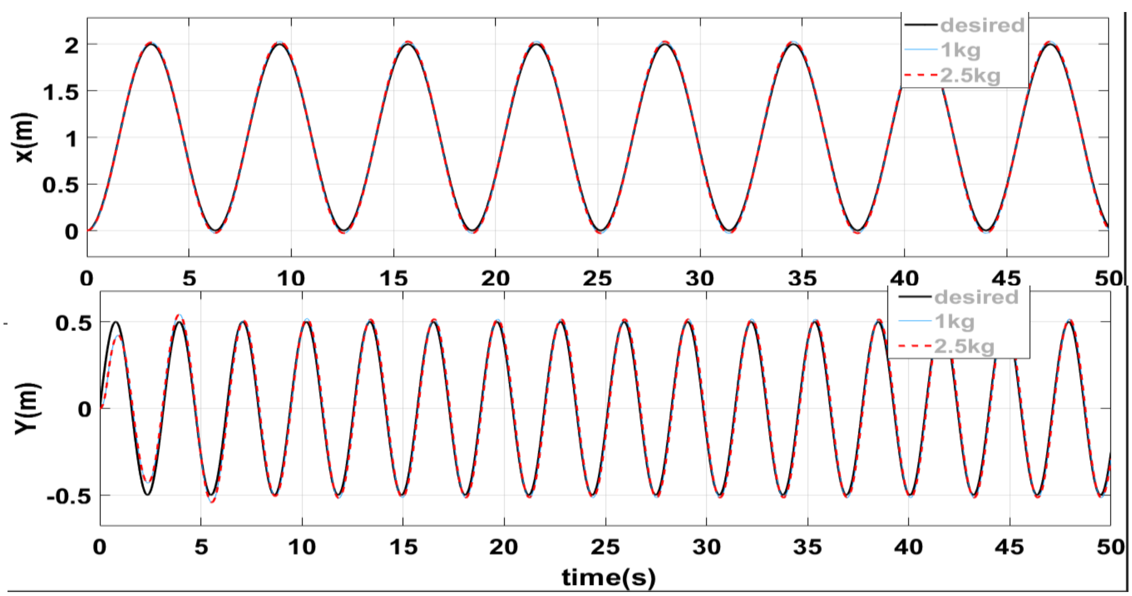
Table 6.2 ITAE value for $t = 50sec$

	SMC with PID	SMC
X	0.5273	17.02
Y	0.8452	21.055
Z	0.01604	212.055
Phi	0.006285	192.1
Theta	0.00005034	238.05
Psi	0.00004783	0.02093

The performance difference between the suggested controller and the traditional controller is demonstrated above.

6.4.3 Spiral Infinity Trajectory Tracking Based on Parameter Variation

Using 1.5 kg, of the quadcopter's mass, as an example, we can demonstrate the effectiveness and resilience of the Optimal SMC scheme in tracking trajectory when the quadcopter's mass varies. This makes the quadcopter's mass become 2.5 kilogram. As we can see from the figure 6.29 the quadrotor track perfact trajectory even if we add 1.5kg of mass on it.



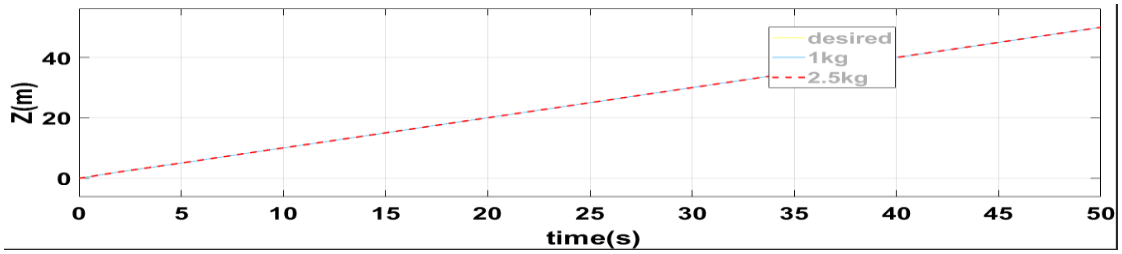


Figure 6.27 Translational trajectory response for mass added to the system

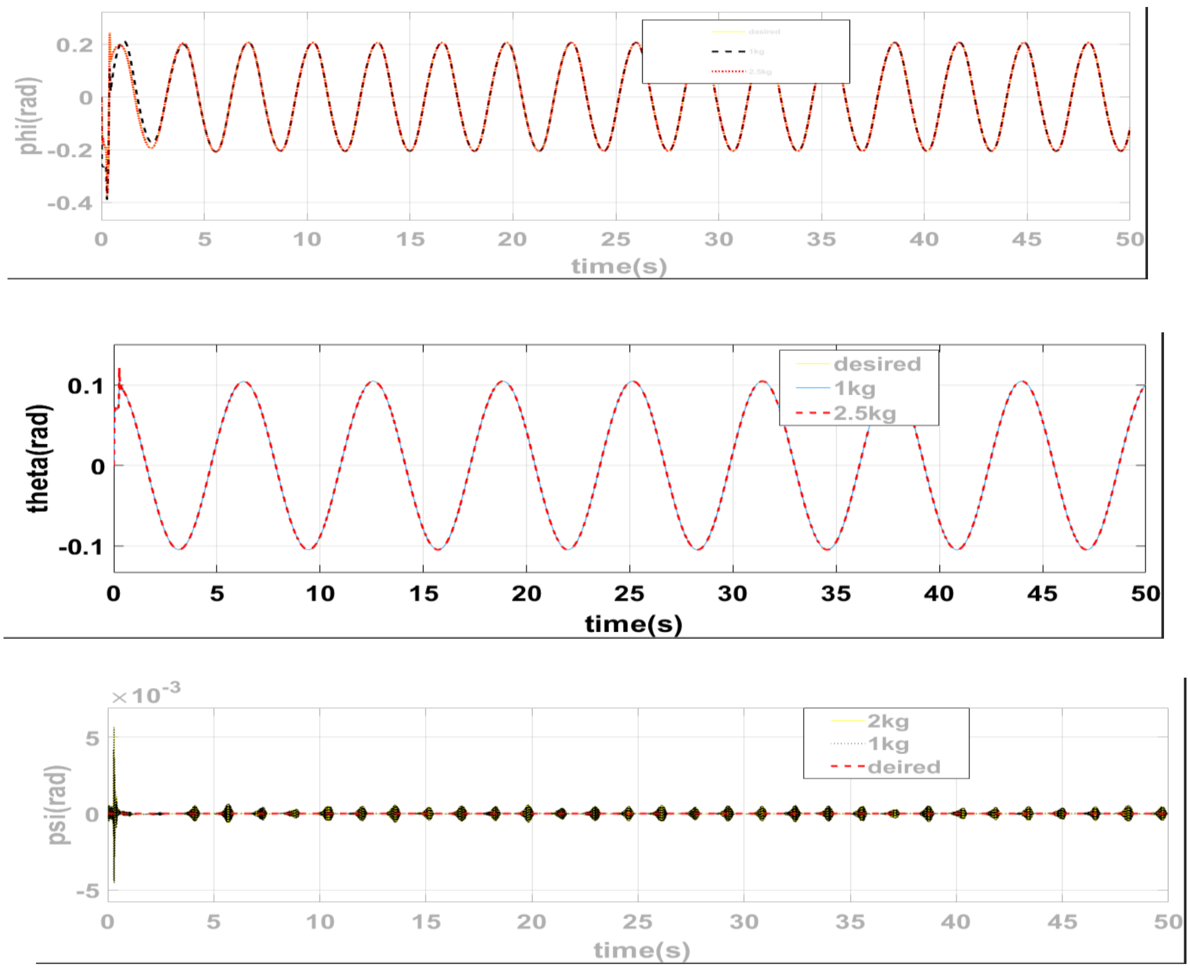


Figure 6.28 Rotational trajectory tracking response for mass added to the system

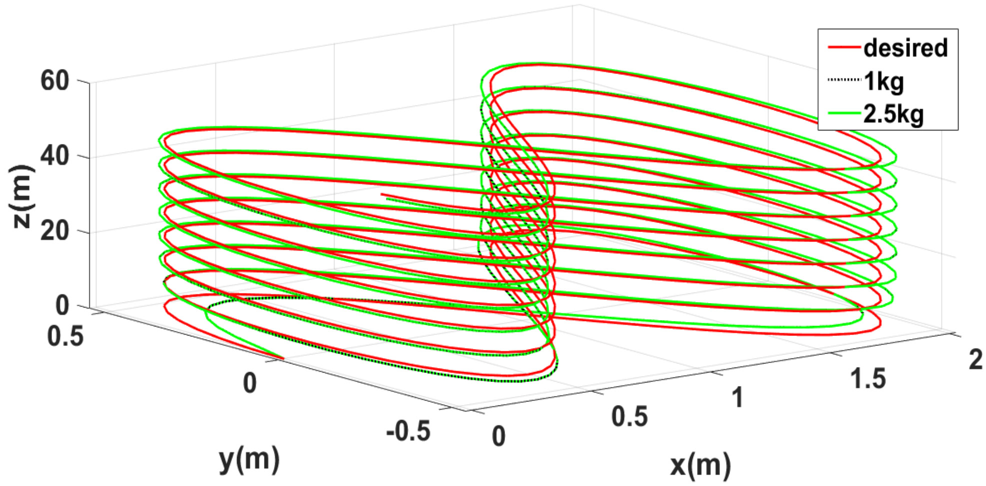


Figure 6.29 3D trajectory response for mass variation

6.5 Wind Turbine Blade Detection Outcome

The validation and training of a wind turbine blade detection are the two main sections of this section. The outcomes of these efforts are outlined in this part, emphasizing the usefulness and practical implications of the YOLOv8-based wind blade recognition system.

6.5.1 Model Performance

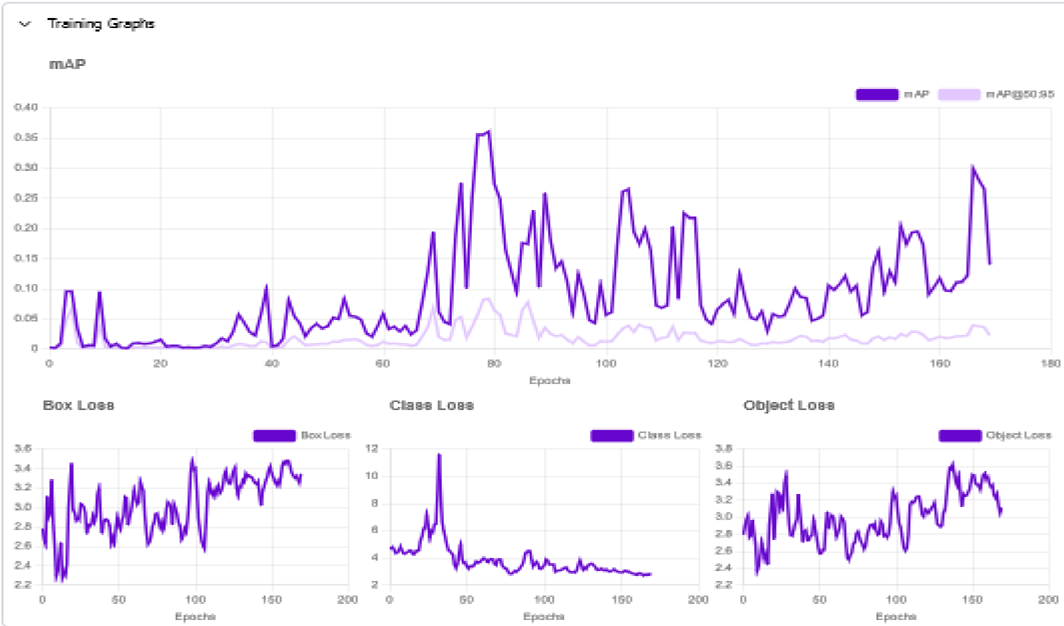


Figure 6.30 Model performance graphical response

This model yields 36.7% for the training mean average precision and 93.8% for the validation mean average precision.

A) Train Box Loss: This measure monitors the difference between the The training data contains bounding boxes that are both predicted and actual for the objects. The box loss is lower when the bounding boxes predicted by the model are nearer to the real boxes with boundaries.

B) Train Class Loss: is a metric that measures how much the actual class labels of the items in the training data are not in line with the predicted class probabilities. A stronger correspondence between the actual class labels and the models' anticipated class probability is observed when there is little class loss.

C Train DFL Loss: Train DFL (Dynamic Feature Learning) loss indicator measures the discrepancy between projected

D) Metrics Precision (B): Out of all the projected bounding boxes the percentage of actual positive detections is determined by the metrics precision (B) statistic. A greater precision means that more genuine positives will be detected by the model, and fewer false positives will be generated.

E) Metrics Recall (B): Out of all the actual bounding boxes, the metric calculates the percentage of genuine positive identification. A higher recall indicates that all true positive detections are accurately identified by the model, while false negatives are minimized.

F) Metrics mAP50 (B): The mAP50 (B) statistic evaluates using a 50% intersection-over-union (IoU) criterion. Increased mAP50 signifies enhanced item detection and localization precision in numerous areas.

G) Metrics mAP50-95 (B): The IoU criteria for this metric range from 50% to 95%, and it evaluates the model's mean average precision over a variety of object categories. Greater accuracy in recognizing and localizing items across several categories with a broader range of IoU thresholds is indicated by a higher mAP50-95.

H) Metrics like train/box_loss: are used to assess how well the model recognizes things in training images. These measure the model's accuracy in identifying the presence of objects, which is important for jobs involving object detection.

I) The (validation classification loss, or Val/clsloss): measures how well the model can categorize items. A reduced value during the specified epoch signifies the improved capacity of the model to precisely categorize items within the validation dataset.

6.2 Wind Turbine Object Detection Using YOLOv8

When we give various types of data such as wind turbine object, tower and grass etc to yolov8 able to train and process and select the wind turbine only as you can see from the figure blow.

Figure 6.31 Wind turbine object after detection

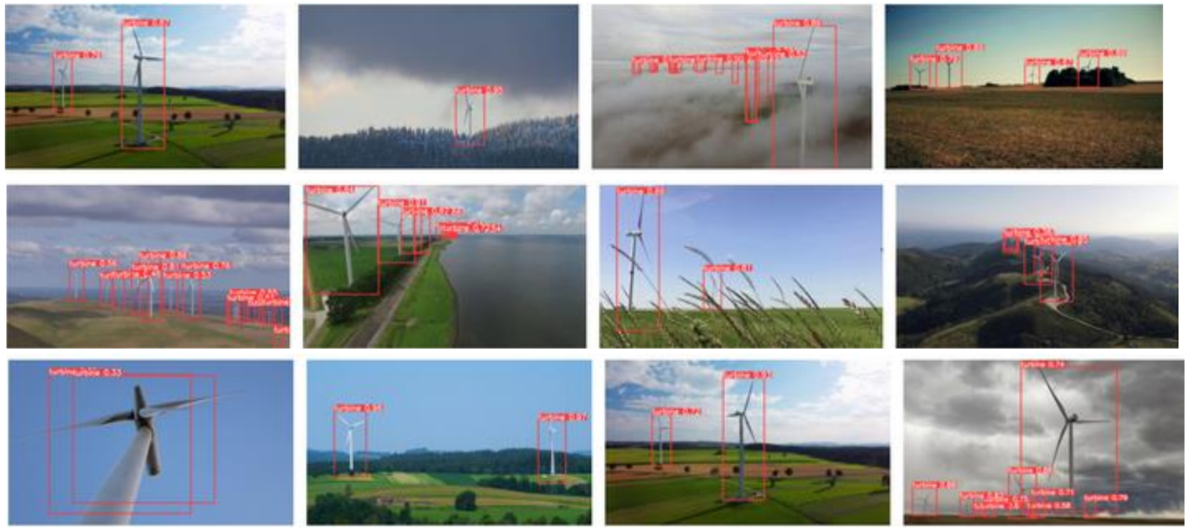


Figure 6.32 wind turbine count and label response

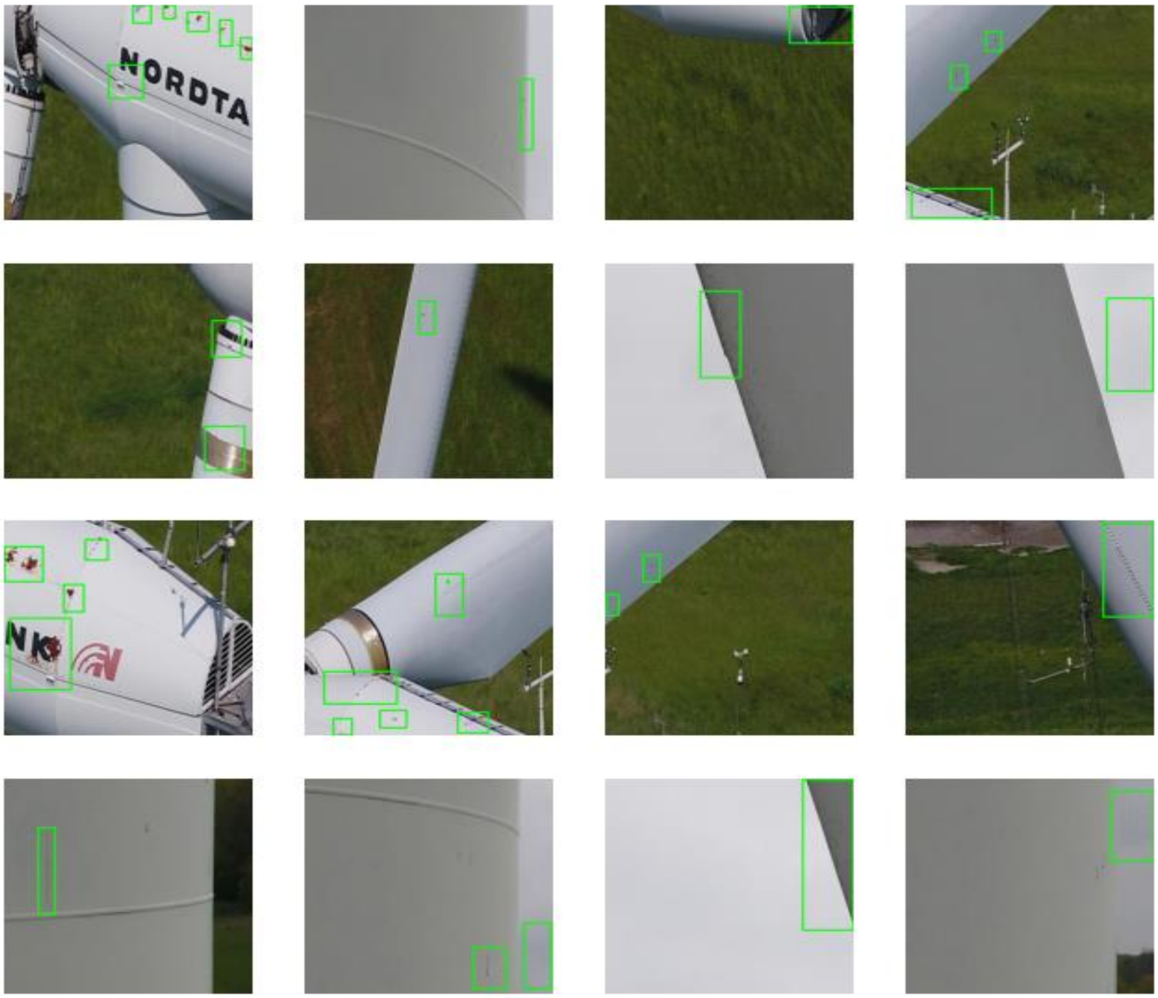


Figure 6.33 damaged wind turbine part after detection

As you can see from figure 6.32 YOLOv8 able to detect the damaged parts of wind turbine form its surface 36.7% for the training mean average precision and 93.8% for the validation mean average precision.

CHAPTER SEVEN

7. CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

This research work introduced an optimal SMC with PID sliding surface to solve the trajectory control problem of quadrotor including specific trajectory around the wind turbine during inspection and image processing algorithm which is uncommon in most of the literatures. The nonlinear dynamics model of the quadrotor is based on Newton-Euler method. After the derivation of the system model SMC and SMC with PID surface controllers are which makes quadrotor able to follow a given desired trajectory.

In the MATLAB software environment, Performance evaluation is done for different kinds of expected trajectories using the designed controllers, considering parametric uncertainties in mass and external disturbances. Additionally wind turbine image acquired from quadrotor are trained using Roboflow in YOLOv8.

Ultimately, the simulation results demonstrate that the developed controller (SMC with PID surface control), tracks the provided trajectories and specifically trajectory around wind turbine with extremely low tracking error when quadrotor is subjected to wind disturbance with different altitude and it is capable of handling parameter uncertainty effectively and, it can be concluded that it is able to control and stabilize altitude, attitude and position. Additionally, YOLOv8 training yielded excellent outcomes in automating wind turbine object detection and then identify damage on the surface in both validation and training.

Generally, the domains of wind turbine inspection technology and control systems have benefited greatly from this thesis. In order to preserve stability, the combination of PSO and SMC with PID sliding surfaces has improved resilience, decreased chattering, and faster state convergence.

7.2 Recommendations

The main emphasis of the thesis is the analysis and assessment of a quadcopter made especially for wind turbine surface damage detection. With the main objective of reducing tracking errors for intended trajectories, emphasis is given on evaluating the resilience and efficacy of the controller. The thesis also investigates automatic controller gain tuning strategies experimenting with helical path for tower and constant velocity trajectory for three blades of wind turbine, and various flight trajectory for quadrotor.

- Preemptively implementing the quadcopter model and control system on actual hardware is highly encouraged for fellow researchers. Furthermore, significant advancements in the relevance and improvement of wind turbine technology can be made by employing the quadcopter to take pictures and the trained system to detect damage on wind turbine surface in real world.
- Train YOLOv8 algorithm which is able to classify the types of damage on wind turbine surface.
- Consider motor dynamics when modeling and controlling the quadcopters
- Analysis battery and power performance.

REFERENCES

A, J. (2018). *techniques of ultrasonic NDT and SHM . Version : Accepted Version Article : This is the peer reviewed version of the following article : Yang K , Rongong JA , Worden K . Damage detection in a laboratory wind turbine blade using techniques of ultrasonic ND*.

Amaral, S., & Henriques, D. F. (2013). Inspection and diagnosis: a contribution to modern buildings sustainability. *Portugal SB13*, 75–82.

Amiri, M., & Hosseinzadeh, M. (2024). *Closed-Loop Model Identification and MPC-based Navigation of Quadcopters: A Case Study of Parrot Bebop 2*. http://arxiv.org/abs/2404.07267

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

Demir, B. E., & Demir, F. (2023). Comparison of Metaheuristic Optimization Algorithms for Quadrotor PID Controllers. *Tehnicki Vjesnik*, *30*(4), 1096–1103. https://doi.org/10.17559/TV-20221108150435

Dijkstra, W., Kakra, V., & Dijkstra, M. (n.d.). *[004]Damage Detection on Wind Turbine Blades*. https://github.

Drone Inspection Guide Benefits & Industries Served. (n.d.).

Dutta, P. K., & Banerjee, S. (2019). Monitoring of aerosol and other particulate matter in air using aerial monitored sensors and real time data monitoring and processing. *Journal of System and Management Sciences*, *9*(2), 104–113. https://doi.org/10.33168/JSMS.2019.0205

Foster, A., Best, O., Gianni, M., Khan, A., Collins, K., & Sharma, S. (2022a). Drone Footage Wind Turbine Surface Damage Detection. *IVMSP 2022 - 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop*. https://doi.org/10.1109/IVMSP54334.2022.9816220

Foster, A., Best, O., Gianni, M., Khan, A., Collins, K., & Sharma, S. (2022b). Drone Footage Wind Turbine Surface Damage Detection. *IVMSP 2022 - 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop*, 1–5. https://doi.org/10.1109/IVMSP54334.2022.9816220

Gedefaw, E. A. (2023). *Trajectory Tracking Control of Quadcopter using Fuzzy Super Twisting SMC with PID Surface for Wheat Yield Estimation Addis Ababa Institute of Technology School of Electrical and Computer Engineering Trajectory Tracking Control of Quadcopter using Fuzzy Su*.

Haile G, A. M. and E. A. (2023). No Title**การบริหารจัดการการบริการที่มีคุณภาพในโรงพยาบาลสังกัดกระทรวงสาธารณสุข**. *วารสารวิชาการมหาวิทยาลัยอีสเทิร์นเอเชีย*, *4*(1), 88–100.

Hoell, S., & Omenzetter, P. (2014). Damage detection in a wind turbine blade based on time series methods. *7th European Workshop on Structural Health Monitoring, EWSHM 2014 - 2nd European Conference of the Prognostics and Health Management (PHM) Society*, 237–244.

Kannan, M., & Student, B. E. (2016). *Quadcopter (UAVs) for Effective Human Resource Management with Full-Duplex Wireless A / V Communication for Academic Utility*. *2*(10), 764–768.

Labbadi, M., Boukal, Y., & Cherkaoui, M. (2020). Path Following Control of Quadrotor UAV With Continuous Fractional-Order Super Twisting Sliding Mode. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *100*(3–4), 1429–1451.

https://doi.org/10.1007/s10846-020-01256-3

Lidiya Girma. (2021). *Minimum Jerk Trajectory Tracking and Control of Quadcopter Using Adaptive Sliding Mode Controller*. *September*, 6–6.

Mokhtari, S. A. (2022). FOPID Control of Quadrotor Based on Neural Networks Optimization and Path Planning Through Machine Learning and PSO Algorithm. *International Journal of Aeronautical and Space Sciences*, *23*(3), 567–582. https://doi.org/10.1007/s42405-022-00461-8

Nguyen, N. P., Mung, N. X., Thanh, H. L. N. N., Huynh, T. T., Lam, N. T., & Hong, S. K. (2021). Adaptive Sliding Mode Control for Attitude and Altitude System of a Quadcopter UAV via Neural Network. *IEEE Access*, *9*, 40076–40085. https://doi.org/10.1109/ACCESS.2021.3064883

Rao, S. (1995). *cor ~ TROL OF RIGID ROBOTIC MA . I ~ IPULATORS*.

Reizenstein, A. (2017). Position and Trajectory Control of a Quadcopter Using PID and LQ Controllers. *Linköping University*, 81. http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Aliu%3Adiva-139498

Search and rescue. (1995). In *Air Med* (Vol. 1, Issue 2, pp. 21–24).

Tiwari, S., Gupta, S., Aggarwal, S., & Kaushik, S. (2021). Aerial Surveillance Quadcopter. *CEUR Workshop Proceedings*, *3283*, 86–95.

Usman, M. (2020). Quadcopter modelling and control with MATLAB/Simulink implementation. *Bachelor's Thesis 2019 LAB University of Applied Sciences Technology Lappenranta*, 68.

APPENDICES

Appendix A.1 Derivation of Rotation Matrix

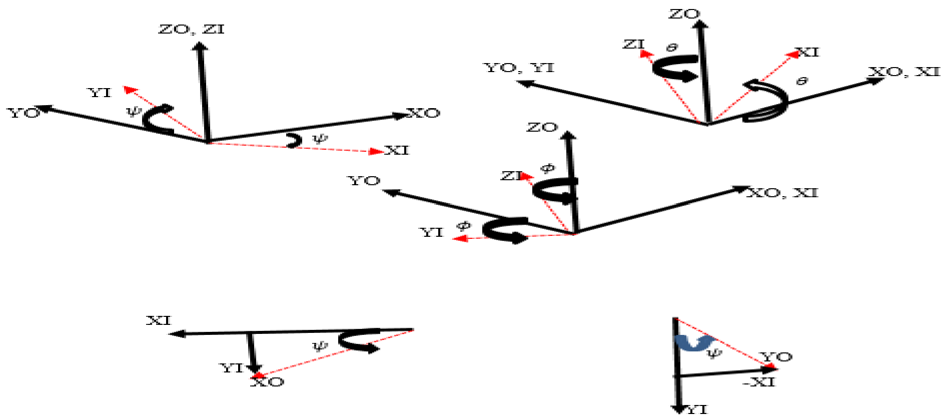


Figure A.1 Euler angle

(c) Roll motion, or rotation about the X-axis respectively

$XOYI = Sin\psi$

$XOZI = 0$

$YOXI = -sin\psi$

$YOYI = COS\psi$

$YOZI = 0$

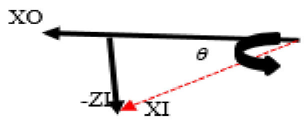
$ZOXI = 0 \quad ,ZOYI = 0 \quad ZOZI = 1$

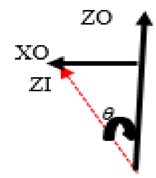
ZO Is perpendicular to ZI andYI, hence ZI's projection onto XI and YI is zero. ZO Is perfectly aligned with ZI and has a length of one. Z0 projects a single image onto ZI.

$$R_Z(\psi) = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \textit{Is the yaw angle rotate with Z-axis}$$

$$[X_I \; Y_I \;\; Z_I]^T = R_Z(\psi)[\, X_O \; Y_O \;\; Z_O]^T$$

Derivation of rotation about Y-axis





$XIXO = COS\theta$ $\qquad$ $ZIXO = sin\theta$

$XIYO = 0$ $\qquad$ $ZIZO = COS\theta$

$YIXO = YIZO = 0 \; and \; YIYO = 1$

$$R_Y(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}$$ Is the pitch angle rotate with Y-axis

$[\,XO\ YO\ ZO]^T$=Y$(\theta)[\,XI\ YI\ ZI]^T$

Derivation of rotation about X-axis similar to above z and y excepet variable x

$$R_I^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_I^B = \begin{bmatrix} C_\psi C_\theta & S_\psi C_\theta & -S_\theta \\ -S_\psi C_\phi + C_\psi S_\theta S_\phi & C_\psi C_\phi + S_\psi S_\theta S_\phi & C_\theta S_\phi \\ S_\psi S_\phi + C_\psi S_\theta C_\phi & C_\psi S_\phi + S_\psi S_\theta C_\phi & C_\theta C_\phi \end{bmatrix}$$

$$(R_I^B)^T = R_B^I = \begin{bmatrix} C_\theta C_\phi & -S_\psi C_\phi + C_\psi S_\theta S_\phi & -S_\psi S_\phi + C_\psi S_\theta C_\phi \\ -S_\psi C_\theta & C_\psi C_\phi + S_\psi S_\theta S_\phi & -C_\psi S_\phi + S_\psi S_\theta C_\phi \\ -S_\theta & S_\phi C_\theta & C_\psi C_\theta \end{bmatrix}$$

A.2 Finding the Required Angles

Drive the necessary angles and altitude controller U1 with this equation:$\ddot{x} = [cos\phi sin\theta sin\psi + sin\phi sin\psi]\frac{u1}{m}$

$$\ddot{y} = [cos\phi sin\theta sin\psi - cos\psi sin\phi]\frac{u1}{m}$$

$$\ddot{z} = [cos\phi cos\theta]\frac{u1}{m} - g$$

$$\frac{U1}{m} = V\text{z} + \text{g}/\cos\theta cos\phi$$

$$Ux = V\text{z} + \text{g}/\cos\theta cos\phi(cos\phi sin\theta sin\psi + sin\phi sin\psi)$$

$$cos\phi sin\theta = (\frac{Uxcos\phi cos\theta}{Vz+g})1/cos\psi \text{ - } sin\phi sin\psi$$

Take the y-axis extract of cosϕ sinθ from the virtual controller as

$$cos\phi sin\theta = (\frac{Uycos\phi cos\theta}{Vz+g})1/cos\psi \text{ +}sin\phi sin\psi$$

Multiply these equations by cϕcsψ after equating them.

$$s\phi c\psi^2 + s\phi s\psi^2 \ = (\frac{Uxc\phi c\theta - Uyc\phi c\theta}{Vz+g})1/c\psi \text{ - } s\phi s\psi$$

$$tan\theta = (c\theta\frac{Uxs\psi - Uyc\psi}{Vz+g}) \text{ ,}\psi d = tan^{-1}(c\theta d\frac{Uxs\psi - Uyc\psi}{Vz+g})$$

Using the same procedure and rearranging the equation above to obtain θd as

$$\theta d = tan^{-1}(\frac{Uxc\psi+Uys\psi}{Vz+g})$$

Appendix A.3 Coriol's force formula

We wish to determine the derivative of the linear velocity vector v with respect to body frame, given the inertial set of unit vectors along the direction b = b1, b2, b3 which has an instantaneous angular velocity ω = p, q, r (Belsty, 2021).

$$\frac{d_v}{d_{in}} = \frac{d(_{ub1+bv2+wb3}}{d_t} \qquad \text{Use chine rule}$$

$$\frac{d_v}{d_{in}} = \left(\frac{d_u}{d_t}b_1 + \frac{db_1}{d_t}u + \frac{d_v}{d_t}b_2 + \frac{db_2}{d_t}v + \frac{dw}{d_t}b_3 + \frac{db_3}{d_t}w\right)$$

The issue at hand is how to calculate d(.)/. The following represents the relationship between those angles and angular velocity:

$$\frac{\mathrm{d}\theta}{d_t} = \omega, \theta = \omega idt \qquad where\ i = b1, b2, b3$$

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} c(rdt) & s(rdt) & 0 \\ -s(rdt) & c(rdt) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vector$$

Using the identical process for the b2(y) and b1(x) axes, we obtained

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} c(qdt) & s(qdt) & 0 \\ 0 & 1 & 0 \\ -s(qdt) & 0 & c(qdt) \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vector$$

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(pdt) & -s(pdt) \\ 0 & s(pdt) & c(pdt) \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vecotr$$

The unit vectors' instantaneous change as dt approaches zero is of interest to us since it results in cos(θ) = 1 and sin(θ) = θ.in where s = sin(.) and c = cos(.) The rotations expressed as follows along the z, x, and x-axes:

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & -r(dt) & 0 \\ rdt & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vecotr$$

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -pdt \\ 0 & 1 & 0 \\ -pdt & 0 & 1 \quad 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vecotr$$

The following equation can be expressed in matrix form as follows by dividing both sides by dt: $\begin{bmatrix} b1/dt \\ b2/dt \\ b3/dt \end{bmatrix} = \begin{bmatrix} 1 & -r & 0 \\ r & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vector$

We may obtain the angular velocities along the y- and x-axes by following the same process. $\begin{bmatrix} b1/dt \\ b2/dt \\ b3/dt \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -p \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vector, \begin{bmatrix} b1/dt \\ b2/dt \\ b3/dt \end{bmatrix} = \begin{bmatrix} 1 & 0 & q \\ 0 & 0 & 0 \\ q & 0 & 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vector$

Integrate every rotation for every axis into a single matrix

$$\begin{bmatrix} b1/dt \\ b2/dt \\ b3/dt \end{bmatrix} = \begin{bmatrix} 1 & -r & q \\ r & 0 & -p \\ q & p & 0 \end{bmatrix} \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} vector, w \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} is\ coriols\ equatio$$

Next, return to the chain rule and change the values.

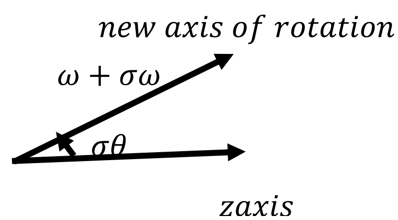
$$\frac{d_v}{d_{in}} = \left(\frac{d_u}{d_t} b_1 + \frac{db_1}{d_t} u + \frac{d_v}{d_t} b_2 + \frac{db_2}{d_t} v + \frac{dw}{d_t} b_3 + \frac{db_3}{d_t} w \right)$$

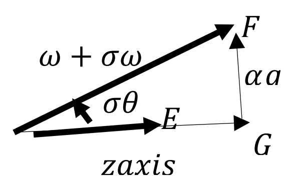
$$F = ma = m\frac{d_v}{d_t}$$

Change to Newton's Second Law, $\begin{aligned} Fx &= m(\dot{u} - v_r + w_q) \\ Fy &= m(\dot{v} + w_p - v_r) \\ Fz &= m(\dot{w} + u_q - v_p) \end{aligned}$ Rewrite in matrix form

$$F = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & m_w & -m_v \\ -m_w & 0 & m_u \\ m_v & -m_u & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

A.4 Gyroscopic impacts





$$\alpha g = lim\sigma t\ 0\left(\omega + \frac{\omega\sigma}{\sigma t}\right) = lim\sigma t\ 0\left(\frac{\omega sin\sigma\theta + \sigma w sin\sigma\theta}{\sigma t}\right) \mathrm{Zsin}\ is\ very\ small\ sin\sigma\theta = \sigma\theta$$ Ω1

$$\alpha g = lim\sigma t\ 0\frac{\omega\sigma\theta + \sigma w\sigma\theta}{\sigma t}), \sigma w\sigma\theta = 0\ is\ very\ small = lim\sigma t\ 0\left(\frac{w\sigma\theta}{\sigma t}\right) = \frac{\omega d\theta}{dt} = \omega\Omega$$

$where\ \omega\ is\ angular\ velocity\ of propeller$, Ω is angulare rate of angle

$$\tau g = J\alpha g = J(\omega X \Omega)$$

$$where\ \omega = [0\ 0\ \omega]^T, \Omega = [\Omega\phi\ \Omega\theta\ \Omega\psi]^T$$

$In\ horizontal\ plane\ of\ quadcopter\ four\ propeller\ and\ each\ contribut$

$$\tau g1 = \left(\begin{bmatrix} 0 \\ 0 \\ \omega 1 \end{bmatrix} X \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix}\right) = giroscope\ torque, Cross\ product\ of\ \omega\ and\ \Omega$$

$$\omega \mathrm{x} \Omega = -(\omega\Omega\phi)\mathrm{i} - (-\omega\Omega\theta)\mathrm{j} + 0\mathrm{k}$$

$$\tau g2 = (\begin{bmatrix} 0 \\ 0 \\ -\omega 2 \end{bmatrix} X \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix})\text{=J} \begin{bmatrix} \Omega\phi\omega 2 \\ \Omega\theta\omega 2 \\ 0 \end{bmatrix}, \tau g3 = (\begin{bmatrix} 0 \\ 0 \\ \omega 3 \end{bmatrix} X \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix})\text{=J} \begin{bmatrix} \Omega\phi\omega 3 \\ \Omega\theta\omega 3 \\ 0 \end{bmatrix},$$

$$\tau g4 = (\begin{bmatrix} 0 \\ 0 \\ -\omega 4 \end{bmatrix} X \begin{bmatrix} \Omega\phi \\ \Omega\theta \\ \Omega\psi \end{bmatrix})\text{=J} \begin{bmatrix} \Omega\phi\omega 4 \\ \Omega\theta\omega 4 \\ 0 \end{bmatrix},$$

$$\tau gtotal = \tau g4 + \tau g3 + \tau g2 + \tau g1, \omega r = \omega 1 - \omega 2 + \omega 3 - \omega 4$$

Appendix B: for wind turbine trajectory matlab code

```
clear;
% Parameters
clear;
close all;
ag = pi/2;
sd = 0.5;
rt = 2;
ht = 100 - sd;
lh = 3;
rh = 3;
lb = 50 + sd;
db = 5;
h = 2 * rh;
a = h / 3;
xa = rt + sd;
ya = -(2 * a + lb) * sin(ag + pi - pi / 3);
za = ht + a + (2 * a + lb) * cos(ag + pi - pi / 3);
xb = xa;
yb = -(2 * a + lb) * sin(ag + pi + pi / 3);
zb = ht + a + (2 * a + lb) * cos(ag + pi + pi / 3);
xc = xa;
yc = -(2 * a + lb) * sin(ag);
zc = ht + a + (2 * a + lb) * cos(ag);
xd = xa;
yd = -2 * a * sin(ag + pi - pi / 3);
zd = ht + a + (2 * a) * cos(ag + pi - pi / 3);
xe = xa;
ye = -2 * a * sin(ag + pi + pi / 3);
ze = ht + a + (2 * a) * cos(ag + pi + pi / 3);
xf = xa;
yf = -2 * a * sin(ag);
zf = ht + a + 2 * a * cos(ag);
va = xa - 2;

% Define points
if ag <= pi / 3
```

```
    points = [
        xa,  0, 0;
        xa,  0, ht;
        xa, yd, zd;
        xa, ya, za;
        va, ya, za;
        va, yd, zd;
        va, yf, zf
        va, yc, zc
        xa, yc, zc
        xa, yf, zf;
        xa, xe, ze;
        xa, yb, zb;
        va, yb, zb;
        va, ye, ze;
       -xa, 0, ht;
       -xa, 0, 0;
    ];
else
    points = [
        xb,  0, 0;
        xb,  0, ht;
        xc, yd, zd;
        xa, ya, za;
        va, ya, za;
        va, yd, zd;
        va, yf, zf;
        va, yc, zc;
        xb, yc, zc;
        xb, yf, zf;
        xb, xe, ze;
        xb, yb, zb;
        va, yb, zb;
        va, ye, ze;
       -xb, 0, ht;
       -xb, 0, 0;
    ];
end

% Extract x, y, and z coordinates
x = points(:, 1)';
y = points(:, 2)';
z = points(:, 3)';
% Determine initial and final points for each segment
a_x = x(1:end-1);
a_y = y(1:end-1);
a_z = z(1:end-1);
b_x = x(2:end);
b_y = y(2:end);
b_z = z(2:end);

[a_theta, a_phi, a_si] = deal(zeros(1, length(a_x)));
[b_theta, b_phi, b_si] = deal(zeros(1, length(a_x)));

% Determine required time for each segment and total required time
```

```
T_r = abs([b_x - a_x; b_y - a_y; b_z - a_z]);
T = 2 * max(T_r);
T_s = sum(T);
% Initialize T_r with the first element of T
T_r = zeros(1, length(T));
% Calculate cumulative sums for T_r (time segment)
for i = 1:length(T)
    T_r(i) = sum(T(1:i));
end
```

Appendix B1: Let $\mathrm{design\_trajectory}(\mathrm{t}, \mathrm{T}, \mathrm{a_x}, \mathrm{a_y}, \mathrm{a_z}, \mathrm{b_x}, \mathrm{b_y}, \mathrm{b_z})$ be defined such that it computes the trajectory coordinates (x, y, z) based on time t and the intervals T.

Summary of the Model

The overall design trajectory can be summarized as follows:

1. For t in the first interval (helical path):

$$(x, y, z) = \left(2.5 \cdot \cos\left(\frac{20\pi}{T(1)} t\right), 2.5 \cdot \sin\left(\frac{20\pi}{T(1)} t\right), (b_z(1) - a_z(1)) \cdot \left(\frac{t}{T(1)}\right)\right)$$

2. For subsequent intervals:

$$(x, y, z) = \left(a_x(i) + \frac{b_x(i) - a_x(i)}{\text{duration}} \cdot (t - T_0), a_y(i) + \frac{b_y(i) - a_y(i)}{\text{duration}} \cdot (t - T_0), a_z(i) + \frac{b_z(i) - a_z(i)}{\text{duration}} \cdot (t - T_0)\right)$$

Where i indicates the current interval based on the total time T.