DECLARATION

For the support and advice of the staff members of electrical and power engineering Mr. Addisu Safo Bosera

To: Power and Control Engineering Department

Subject: Project Submission

This is to certify that the project entitled "**Design and Simulation of Drone Flight Control Using PID Controller**", submitted in the fulfilment of the requirements for the BSC degree in Electrical and Computing Engineering, Department of Power and Control Engineering and has been carried out by:

1	SELAM BEZUALEM BIRHAN	UGR/19618/12
2	TSION TSEGAYE YOHANNES	UGR/19900/12
3	BELAY DEMEKE SHIFERAW	UGR/19613/12
4	WENDIMAGEGN ASHENAFI GEMECHU	UGR/19581/12
5	ABDULKREIM SEID ALI	UGR/16674/11

respectively, under my supervision.

Therefore, I recommend that the student has fulfilled the requirements and hence they can submit the project to the department.

Mr. Addisu Safo Bosera	____________________	________________
Name of Advisor/supervisor	Signature	Date

RECOMMENDATION

I, the advisor of this semester project, hereby certify that I have read the revised version of the semester project entitled **“Design and Simulation of Drone Flight Control Using PID Controller”** prepared under my guidance by Selam Bezualem Birhan, Tsion Tsegaye Yohannes, Belay Demeke Shiferaw, Wendimagegn Ashenafi Gemechu, and Abdulkreim Seid Ali submitted in partial fulfillment of the requirements for the BSc of Electrical Power and Control Engineering. Therefore, I recommend the submission of a revised version of the semester project to the department following the applicable procedures.

Advisor Name: Mr. Addisu Safo Bosera

Advisor Signature: _______________

Date: _______________

Advisor's Approval Sheet

To: Department of Electrical Power and Control Engineering

Subject: Semester Project Submission

This is to certify that the semester project entitled "**Design and Simulation of Drone Flight Control Using PID Controller**", submitted in the fulfillment of the requirements for the BSc degree in Electrical Engineering and Computing, Department of Electrical Power and Control Engineering, and has been carried out by Selam Bezualem Birhan, Tsion Tsegaye Yohannes, Belay Demeke Shiferaw, Wendimagegn Ashenafi Gemechu, and Abdulkreim Seid Ali Id. No (UGR/19618/12), (UGR/19900/12), (UGR/19613/12), (UGR/19581/12) and (UGR/16674/11), under my supervision. Therefore, I recommend that the student has fulfilled the requirements and hence he/she can submit the semester project to the department.

Mr. Addisu Safo Bosera	______________________	________________
Name of Advisor	Signature	Date

Approval of Examiner(s)

I/we, the undersigned, an examiner of the final open defense by have read and evaluated his/her semester project entitled "**Design and Simulation of Drone Flight Control Using PID Controller**" and examined the candidate. This is, therefore, to certify that the Semester Project has been accepted in the fulfillment of the requirements of the BSc Degree of Department of Electrical Power and Control Engineering.

Mr. Addisu Safo Bosera	______________________	____________________
Supervisor /Advisor	Signature	Date
______________________________	______________________	____________________
Examiner	Signature	Date

Acknowledgment

We would like to extend our heartfelt gratitude to our esteemed advisor, Mr. Addisu Safo Bosera, for his expert guidance, sincere support, and valuable encouragement throughout the duration of this project. His aspiring mentorship, constructively critical insights, and amicable advice have been instrumental in the successful completion of our work.

Further, we express our deep appreciation to the Department of Power and Control Engineering and the Space Science and Technology Institute (STI) at Adama Science and Technology University. Special thanks are due to Mr. Minyamer Gelawe Wase for his invaluable contributions and support.

We are also immensely grateful to Mr. Mesfin Megra Rorisa, Head of Power and Control Engineering, for his collaborative efforts and guidance that significantly enhanced our project. Moreover, our sincere thanks go to the dedicated staff members of the Electrical and Power Engineering Department, specifically Mr. Alemante and Mr. Getachew, for their supportive advice and assistance.

Additionally, we are profoundly thankful to Dr. Kevin C. Craig, Professor of Mechanical Engineering and Director of the Center for Innovation at the School of Engineering & Applied Science, Hofstra University. His generous provision of essential files and unwavering support has been a cornerstone of our project's success.

Each of these individuals has played a pivotal role in our academic journey, and their collective expertise and encouragement have been a beacon of inspiration and learning. We are truly honored to have had the opportunity to work under their guidance.

Abstract

The usage of Quadcopter in commercial fields has evolved significantly due to its phenomenal development. It has been an increasingly popular research topic in recent year due to their low cost, maneuverability, simplicity of structure, ability to hover, their vertical take-off and landing (VTOL) capacity and ability to perform variety of tasks. Besides, it is a great platform for control systems research, which is highly nonlinear and under-actuated system. Its motion can be divided into two subsystems; a rotational subsystem (attitude and heading) and a translational sub system (altitude, x and y motion). The main target of this project is to mathematically model the quadcopter nonlinear dynamics using Newton-Euler equations and design controllers for attitude, heading & altitude of quadcopter. However, controlling the movements of a Quadcopter is a demanding task due to its complex dynamics. The usage of the Proportional-Integral-Derivative (PID) controller for stability control is quite challenging in regards to the complexity of Quadcopter's nonlinear structure. Path planning is one of the important concepts for a Quadcopter to move from one point to another point effectively. In this project, we want to control our quadcopter using Newtons Euler equation.

Next, the PID controller is designed to stabilize the quadcopter's attitude and altitude by adjusting the motor speeds. The controller's gains are optimized through MATLAB tinnier to achieve desired performance, such as fast response, stability, and robustness against disturbances.

To validate the proposed control system, a simulation environment is developed using MATLAB Simulink. The quadcopter model, derived from the Newton-Euler equations, is implemented along with the PID controller. Various flight scenarios, including attitude control and altitude hold, are simulated to assess the system's performance.

Overall, this research provides insights into controlling a quadcopter using a PID controller, employing the Newton-Euler for accurate modeling, and implementing the system in MATLAB Simulink for comprehensive simulation studies. The findings contribute to the advancement of quadcopter control methodologies and serve as a foundation for further research and development in our field.

Table of Contents

List of Figures

List of Tables

Acronyms

X, Y, Z	Roll, Pitch, and Yaw Angles
MPC	Model Predictive Control
RPVS	Remotely Piloted Vehicles
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing
DC	Direct Current
PID	Proportional Integral-Derivative
k_p	Proportional Gain
k_i	Integral Gain
k_d	Derivative Gain
τ_φ	Pitch Torque
τ_θ	Roll Torque
τ_Ψ	Yaw Torque
Rad	Radians
T_p	Peak Time
T_r	Rise Time
T_s	Settling Time
B_{RI}	Rotational Matrix from The Inertial to The Body–Fixed Frame
RPV	Remotely Piloted Vehicle
SISO	Single Input Single Input

CHAPTER ONE

1. INTRODUCTION

Unmanned Aerial Vehicle (UAV) is an emerging field in aerospace industry that has a niche in performing dangerous and critical missions. It has been widely used in the military department over a last few decade and the success of these military applications is increasingly driving efforts to establish unmanned aerial in non-military roles in technologically advanced countries like USA, Japan, Italy etc. When we come to eastern African, especially in our country Ethiopia their cover is less since the demand of automated systems constantly increased while the technology is limited.

The trend in unmanned aerial vehicle (UAV) usage is not for a limited purpose but also for multi-purpose by modifying some parts, which is possible to make security-purposed system to aerial camera and videography. The share of UAV for military purpose decreased while the share for other technology transfer increased day to day.

In recent days, UAVs are becoming very popular among the researcher and control engineers. The rotorcraft UAVs has several merits over fixed wing UAVs, such as hovering at a particular place, vertical takeoff, landing and aggressive maneuvering. Due to this feature rotorcraft UAVs are used in several applications such as search and rescue operations, traffic monitoring, Wild fire suppression, Border surveillance, pipeline inspection and crop spraying in agriculture etc.

Under the category of rotorcraft UAVs, Quadcopter have acquired much attention among researcher. Quadcopter is a multi-copter that lifted and propelled by four rotors, each mounted in one end of a cross-like structure as shown in Figure 1. Each rotor consists of a propeller fitted to a separately powered Brushless DC motor. Quadcopter has 6 degrees of freedom (three translational and three rotational) and only four actuators. Hence, quadcopter is an under actuated system and highly nonlinear in nature. Unlike a conventional helicopter, a quadcopter's rotor blade pitch angle need not to be varied, which makes the quadcopter manufacturing and maintenance easier. Moreover, quadcopters have capacity to carry large payload due to the presence of four motors providing higher thrust.

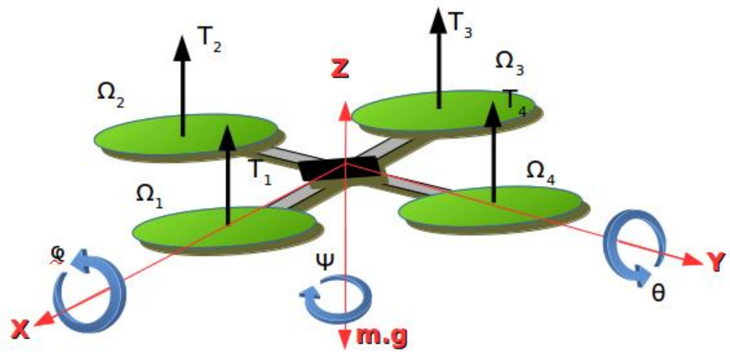


Figure 1.1: Quadcopter Motion Structure

1.1. Background Of the Research

An unmanned aerial vehicle (UAV) is a type of aircraft, fixed wing aircraft and quadcopters that has no on broad crew or passengers. UAVs include both autonomous drones and remotely piloted vehicles (RPVs). Started from World War I up to 19th century UAVs used only for aviation academy, which means they are designed and built only for military purpose but here in 21st century, technology reached a point of sophistication that UAV is given greatly expanded role in many areas of aviation.

1.2. Statement Of the Problem

Stability Controller is the important component for the Quadcopter as it helps to avoid external disturbances and collisions with the objects. It is considered as the brain of the Quadcopter System. At every point in time, the output from the sensors is taken and analyzed to make it better. The speed of the motors must be adjusted based on the output values generated from the controller. The inputs to the Quadcopter are the angular velocity of each rotor, thereby controlling the voltages.

Since we have four control motor inputs and four (roll, pitch, yaw, and thrust) outputs from the Quadcopter system, we need to distribute all these outputs to four motors equally to control the speed of the motors. So, it’s very important to maintain our stability and speed. Using PID Controller Design: provide stability by adjusting control signals based on error feedback. Without a PID controller, it becomes more challenging to maintain stable flight conditions. The quadcopter may exhibit oscillatory behavior, instability, or difficulty in recovering from disturbances.

PID controllers are known for their ability to provide responsive control by quickly adapting to changes in the system. Without a PID controller, the quadcopter's response to external inputs or desired setpoints may be sluggish or delayed, leading to suboptimal control performance.

1.3. Objective

1.3.1. General Objective

To develop a robust control system for a quadcopter using a PID controller, incorporating the Newton-Euler for accurate modeling and simulation in MATLAB Simulink. The aim is to achieve precise control of the quadcopter's attitude and altitude, ensuring stability, responsiveness, and robust performance under various flight conditions.

1.3.2. Specific Objective

Specific objective of our project:

- The ultimate goal is to provide a comprehensive understanding of the quadcopter's dynamics and control, enabling effective and efficient control system design for improved flight performance.
- Obtain the mathematical model of quadcopter dynamics
- Design Linear PID controllers for the quadcopter
- Implementing the controllers on MATLAB Simulink

1.3.3. Significant Of the Study

The utilization of the Newton-Eule for modeling the quadcopter's dynamics enhances the accuracy of the system representation. This precise modeling allows for a deeper understanding of the complex interactions between the quadcopter's motion, forces, and torques. The findings can contribute to more accurate control system design and performance evaluation by Precise Modeling of Quadcopter Dynamics.

The simulation environment developed using MATLAB Simulink allows for comprehensive evaluation of the control system's performance. Through simulations, you can assess the system's behavior under various flight scenarios, evaluate its responsiveness, stability, and robustness, and identify areas for further improvement. This provides a cost-effective and safe means of testing and optimizing the control system before real-world implementation.

1.3.4. Scope Of the Project

This project purview is utilizing MATLAB/Simulink software to derive mathematical modeling of quadcopter dynamics all the way up to the simulation of the finished system. It serves to track and stabilize the quadcopter the desired trajectory by modifying the input controls based on current state system. Scope of PID includes:

- Quadcopter stabilization in hover during flight
- Tracking trajectories for a given trajectories such as Helix, Oblique Helix and given path
- Robustness to disturbances and uncertainties in the system

1.4. Limitation of The Study

- **Simplification of Real-world Dynamics**: Linear models often oversimplify real-world phenomena, which are typically non-linear. This can lead to inaccuracies in predictions or controls, especially in complex systems.
- **Inadequacy in Handling Non-Linear Behaviors**: They struggle to accurately model systems with non-linear characteristics, such as those exhibiting high degrees of variability or subject to threshold effects.
- **Modeling Errors**: All models have inherent uncertainties due to approximations and assumptions, which can lead to errors in predictions.
- **Data Quality and Availability**: Limited or poor-quality data can introduce significant uncertainties into the model, affecting its reliability.
- **System Overload**: In control systems, saturation refers to the condition where a system component reaches its maximum capacity, leading to a performance decline or failure.
- **Inadequate Response to Extreme Conditions**: Saturation can prevent the system from adequately responding to extreme inputs or conditions, resulting in suboptimal performance or instability.
- **Computational Intensity**: Complex trajectories require sophisticated calculations, which can be computationally intensive and time-consuming.

- **Control Precision**: Maintaining precision in following complex trajectories can be challenging, especially under varying environmental conditions or system uncertainties.
- **Scalability Issues**: The model or control system might not scale well for larger or more complex systems.
- **Environmental Factors**: External factors such as weather, terrain, and obstacles can significantly affect the performance of the system.

Considering the limitations outlined earlier, the suitable controllers for the quadcopter, each addressing specific challenges within the scope of our project, are as follows:

1. Model Predictive Control (MPC): Ideal for handling multi-input, multi-output systems with constraints. MPC is excellent for managing complex dynamics and predicting future states, making it beneficial for maneuvering in variable environments.
2. Adaptive Control: This controller adjusts its parameters in real time to adapt to changing conditions. It's particularly useful for systems where the dynamics are uncertain or vary significantly.
3. Fuzzy Logic Controllers: These are based on fuzzy logic and can handle imprecision and non-linearity effectively. They're useful for systems where the mathematical model is not fully known or is too complex.

However, given that our quadcopter is designed with a linear model, we opted for a PID (Proportional-Integral-Derivative) controller for the following reasons:

- **Simplicity and Effectiveness**: PID controllers are straightforward to implement and understand. Their effectiveness in a wide range of applications, particularly in systems with linear dynamics, makes them a reliable choice.
- **Ease of Implementation**: Implementing a PID controller is generally less complex than advanced control strategies like MPC or adaptive controllers. PID algorithms are well-established and can be easily coded or even implemented using pre-built modules in many control systems
- **Ease of Tuning**: Tuning a PID controller, while it requires some trial and error, is generally simpler compared to more complex controllers. This is especially beneficial for linear systems where the dynamics are predictable.

- **Robustness and Stability**: PID controllers provide a good balance of responsiveness and stability, which is crucial for the steady operation of a quadcopter. They are particularly effective in maintaining stable flight and responding to disturbances in a predictable manner.
- **Proven Track Record**: PID controllers have a long history of successful applications in various fields, including aerial vehicles like quadcopters. Their reliability and performance are well-documented, providing confidence in their use for our project.

1.5. Organization Of the Project

Generally, this project reports divided in to six chapters, where it consists;

Chapter 1: Introduction: - In this chapter, introduces the historical background of UAV, Significant of the project It also includes statement of the problem, the detail objective of the project.

Chapter 2: Literature review: The Modeling and Control of quadcopter using linear quadratic optimal tracking Controller. Studies on literature review help in understanding the fundamental of the project.

Chapter 3: Mathematical Modeling: How to create a nonlinear dynamic quadcopter or quadcopter model in MATLAB or Simulink and how to validate it and methodology and block diagram of the system.

Chapter 4: PID Control Design: This chapter details the development and tuning of a PID control system for quadcopter stabilization, highlighting its implementation and effectiveness through testing.

Chapter 5: Simulation result and analysis: provided simulation result and detail analysis about the system.

Chapter 6: conclusion and recommendation: here final result will be summarized and the future extension will be forwarded under this chapter.

1.6. Methodology

The first step in conducting this project described objectives is doing a literature survey, which entails reviewing prior work. this project about quadrotor/quadcopter modeling and design of controllers for attitude, heading & position. Then drive the mathematical model of the quadrotor or quadcopter by using Newton's Euler equations. Next, design PID controller for the model, then implement the controllers and simulate them by MATLAB/Simulink .In this project, the different parameters can be controlled like altitude, *X* and *Y*, pitch, roll and yaw are some of control variables available for control system and torques at each rotor is always manipulating variables. The accomplishment of this project has been achieved using the following methodologies:

- **Literature survey**: A variety of literature that is relevant to this project work is examined, and several concepts are used.
- **Quadrotor modeling*:*** dynamic modeling of quadrotor will be obtained.
- **Controller design***:* linear quadratic optimal tracking has been designed for position, heading and attitude control of quadrotor.
- **Controller testing*:*** PID have been put to the test and exposed to various impacts to confirm their resilience. Specifically, against matching parameter uncertainty.
- **Documentation and presentation*;*** in this section we summarize the results and presentation.

If modeling and controller are not appropriately handled error arise lead to poor results so one can need to see his/her modeling and controller design again in order to achieve our objectives.

1.6.1. PID quadrotor block diagram controller

A block diagram is a visual representation of a system or process that demonstrates how information or signals move between various components. The block diagram would depict the PID controllers interact with the quadcopter system and with each other in the case of a quadcopter with its controllers. It modifies the system output in response to that. If the error does not reach zero, this process is repeated until the feedback variable's value is comparable to a fixed point.

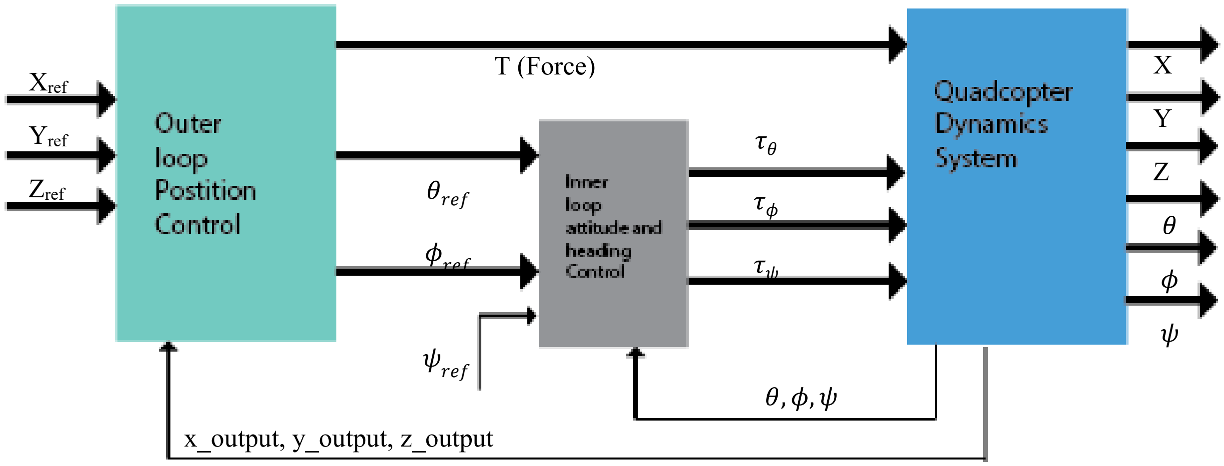


Figure 1.2 Block Diagram of The Quadcopter Control

1.6.2. Flow Chart representation of methodology

Therefore, the above methodologies can be illustrated briefly using the following flowchart shown in the figure 1.2 below. As it seen from the flowchart main things need to filled in this work that modeling and controller design. If modeling and controller are not appropriately handled error arise lead to poor results so one can need to see his/her modeling and controller design again in order to achieve our objectives.

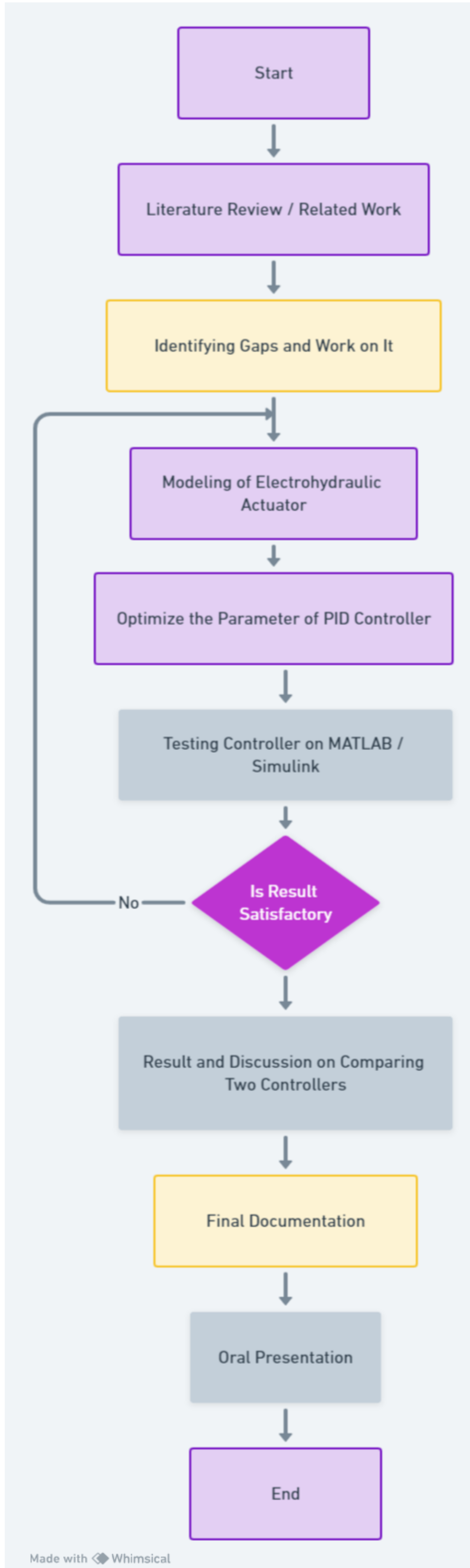


Figure 1.3: Flow chart of the trajectory tracking of quadrotor

CHAPTER TWO

2. LITERATURE REVIEW

Researchers have long studied quadcopter controller design and modeling, which is utilized in a variety of civil and military applications, security applications, and by filmmakers for aerial photography and videography. The research on control design and modeling for quadcopters has been approached in many literatures by various methods, some of which are given here.

1. in 1920 [2]. Among the six designs he tried, his second multicopper had four rotors and eight propellers, all driven by a single engine. The Oehmichen used a steel-tube frame, with two-bladed rotors at the ends of the four arms. The angle of these blades could be varied by warping. Five of the propellers, spinning in the horizontal plane, stabilized the machine laterally. Another propeller was mounted at the nose for steering. The remaining pair of propellers was for forward propulsion.
2. In 1924 it established the first-ever Federation Areonautique International (FAI) [3] distance record for helicopters of 360 m. Later, it completed the first 1-kilometer closed-circuit flight by a rotorcraft After Oehmichen, Dr. George de Bothezat and Ivan Jerome developed this aircraft [1], with six bladed rotors at the end of an X-shaped structure. Two small propellers with variable pitch were used for thrust and yaw control.
3. Convert wings Model a quadcopter was intended to be the prototype for a line of much larger civil and military quadcopter helicopters [2]. The design featured two engines driving four rotors with wings added for additional lift in forward flight. No tail rotor was needed and control was obtained by varying the thrust between rotors. Flown successfully many times in the mid-1950s, this helicopter proved the quadcopter design and it was also the first four-rotor helicopter to demonstrate successful forward flight. However, due to the lack of orders for commercial or military versions however, the project was terminated [3, 4].
4. When came to the recent paper works, numerous control methods have been proposed for quadcopters, for both regulation and tracking problems. The goal is to find a control strategy that allows the states of a quadcopter to converge to an arbitrary set of time-varying reference states. Many previous works [5, 6, 7, 8] have demonstrated that it is possible to control the quadcopter using linear control techniques by linearizing the

dynamics around an operating point, usually chosen to be the hover. However, a wider flight envelope and better performances can be achieved

2.1. Review Of Related Works

The review of related works on the topic of PID control for trajectory tracking of a quadcopter reveals several key findings and insights. It offers effective disturbance rejection capabilities and can manage system dynamics uncertainties. PID control has been successfully used in a number of experiments to track the trajectory of quadcopter aircraft. To further improve PID controllers' performances researchers have additionally adaptive-PID, and fractional-order PID controllers. PID controller is capable of achieving acceptable trajectory tracking performance In relation to accuracy, robustness, disturbance rejection, and handling system uncertainties, PID controllers. PID controllers are preferred in situations where ease of use, minimal computational demands, and real-time performance are important considerations. PID monitoring quadcopter trajectory control is still a few topics that need more research in terms of future approaches. Future research might concentrate on creating sophisticated control methods that combine the benefits of PID performance. Real-world Using, they would benefit applicants PID controllers' robustness against external disturbances, sensor noise, and model uncertainty

2.2. Summary Of Literature Review

From this literature review it has been observed that various techniques have been employed to improve the performance of the quadcopter position, heading and attitude control system, a simplified linearized model, PID. In this project work, dynamic model of quadcopter derived by considering the above modeling effects and optimal control designed under model uncertainty. Moreover, in this research works all state variables are controlled for quadcopter in three-dimensional space accurate control of system with fast and high precisions. Trajectory tracking control the capacity for with quadcopter. Reference trajectory is use to compare reference tracking based on the amount of the control input and to examine the impact of controller gains on control effort. In order to track trajectories and reduce power consumption, a quadcopter vehicle was equipped with a reliable PID controller. Better performance and resistance to disturbances and uncertainties are provided by the PID controller. The structure of a PID controller, however, requires the creation of a mathematical model of system, which may not always be available. In such circumstances, a PID controller is usable as an alternate control method for trajectory tracking. The tracking error, overshoot, settling time, and control effort of PID controllers' capacity to follow a predetermined trajectory will allow us to assess both controllers' performance. According to the simulation results, PID can monitor the preferred course of action using respectable precision.

CHAPTER THREE

3. MATHEMATICAL MODELING

Design and analysis of control systems are usually started by considering mathematical models of physical systems. In this section, a complete dynamical model of Quadcopter UAV is established using the Newton's-Euler formalism

Physical Model Simplifying Assumption

- The aircraft is assumed to be a rigid body
- The propellers are assumed to be rigid.
- The quadcopter frame is symmetrical: x-z and y-z planes are planes of symmetry.
- Mass center of the quadcopter coincides with the geometric center of the rigid frame.
- The mass moment of inertia of each motor rotor and attached propeller is assumed to be small and negligible.
- The Earth is assumed to be flat and is the inertial reference frame.
- Quadcopter interaction with the ground is neglected.
- Sensors (e.g., gyroscope, accelerometer) measure relative to the body-fixed-axes.

3.1. Reference Frame

A reference frame is a group of spatial points where the separation between any two locations is always constant. Despite the fact that, as we will see from this definition, any three or more non-collinear locations can be used as a frame of reference, the simplest method to analyze a reference frame is to consider a rigid body. In this context, we will employ reference frames to first identify a reference frame with a rigid body and then study the motion of this reference frame with regard to an inertial reference. As a result, two coordinate systems are needed: the earth frame system, which is fixed to the ground and sometimes referred to as an inertial coordinate system, and the body frame system, which is fastened to the quadcopter at its Centre of gravity. Newton's Euler formulation will be used for deriving dynamic model of quadcopter. The nonlinear dynamics are represented in Figure 3.1 in the body-fixed B and inertial I frame. The position unit vectors along the body-fixed frame's axis used to show direction are indicated by $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ and the unit vector along the I axis of the inertial frame is represented by

$\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$. The body-fixed frame B starting point is believed to be the quadcopter's center of mass.

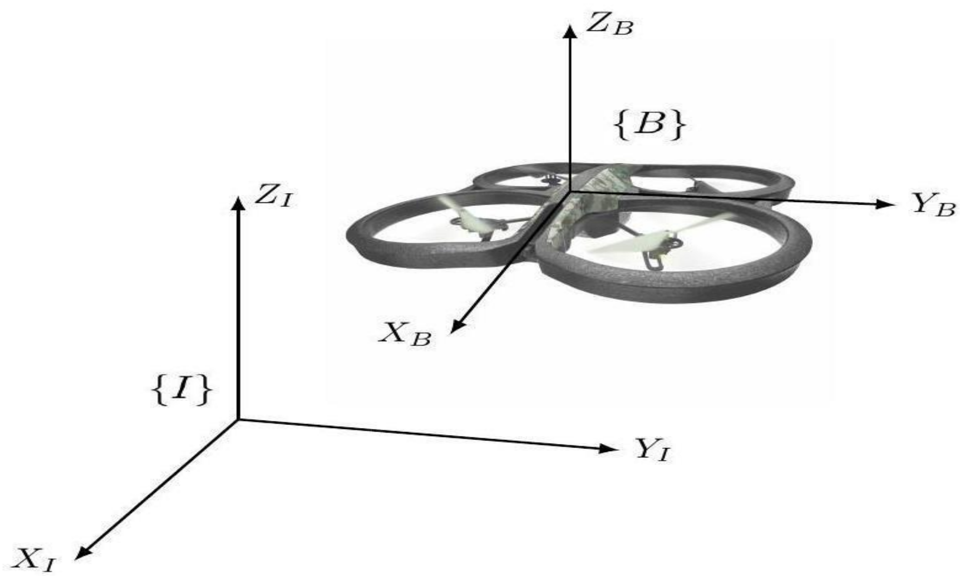


Figure 3.1 The Nonlinear Dynamics Body-Fixed B And Inertial I Frame

Addition Theorem for angular Velocities

- Consider multiple reference frames: R_1, R_2, …… R_N.
- The following relation applies, whether the angular velocities are simple or not:

$$R \underset{\boldsymbol{\omega}}{\rightarrow} R_N = R \underset{\boldsymbol{\omega}}{\rightarrow} R_1 + R_1 \underset{\boldsymbol{\omega}}{\rightarrow} R_2 + \cdots + R_{N-1} \underset{\boldsymbol{\omega}}{\rightarrow} R_N \quad \text{3-1}$$

- There exists at any one instant only one $R \underset{\boldsymbol{\omega}}{\rightarrow} R_N$
- Also

$$R \underset{\boldsymbol{\omega}}{\rightarrow} R_N = -R_N \underset{\boldsymbol{\omega}}{\rightarrow} R \quad (3\text{-}2)$$

- This addition theorem is very powerful as it allows one to develop an expression for a complicated angular velocity by using intermediate reference frames, real or fictitious, that have simple-angular-velocity relations between each of them.

Quadcopter Absolute Angular Velocity in body-fixed frame

R ground

$$R \rightarrow R_1 \; Yaw \; \widehat{K} = \widehat{K}_1 \;\; \psi \qquad \text{3-3}$$

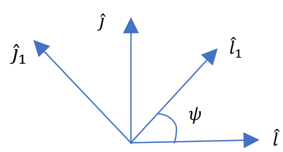
$$R_1 \rightarrow R_2 \; Pitch \; \hat{J} = \hat{j}_2 \;\; \theta \qquad \text{3-4}$$

$$R_2 \rightarrow R_3 \; Yaw \; \hat{I} = \hat{I}_3 \;\; \phi \qquad \text{3-5}$$

$$R \underset{\boldsymbol{\omega}}{\rightarrow} R_3 = \dot{\phi}\hat{I}_3 + \dot{\theta}\hat{j}_2 + \dot{\psi}\widehat{K}_1 \qquad \text{3-6}$$

$R \rightarrow R_1$

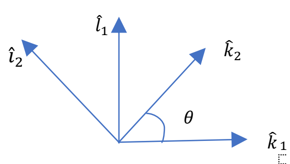
$\hat{k} = \hat{k}_1$



$$\begin{bmatrix} \hat{I} \\ \hat{J} \end{bmatrix} = \begin{bmatrix} cos\psi & -sin\psi \\ sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} \hat{I}_1 \\ \hat{J}_1 \end{bmatrix} \qquad \text{3-7}$$

$R_1 \rightarrow R_2$

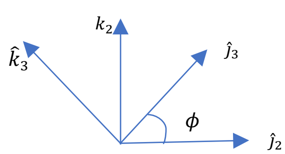
$\hat{j}_1 = \hat{j}_2$



$$\begin{bmatrix} \hat{k}_1 \\ \hat{I}_1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} \hat{k}_1 \\ \hat{I}_1 \end{bmatrix} \qquad \text{3-8}$$

$R_2 \rightarrow R_3$

$\hat{I}_3 = \hat{I}_2$



$$\begin{bmatrix} \hat{j}_2 \\ \hat{k}_2 \end{bmatrix} = \begin{bmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} \hat{j}_3 \\ \hat{k}_3 \end{bmatrix} \qquad \text{3-9}$$

$$R \underset{\boldsymbol{\omega}}{\rightarrow} R_3 = R \underset{\boldsymbol{\omega}}{\rightarrow} R_1 + R_1 \underset{\boldsymbol{\omega}}{\rightarrow} R_2 + R_2 \underset{\boldsymbol{\omega}}{\rightarrow} R_3 \qquad \text{3-10}$$

Addition theorem for angular velocity

All simple Angular Motion

$${}^{R}\vec{\omega}^{R_3} = \dot{\phi}\hat{I}_3 + \dot{\theta}\left(cos\phi\hat{j}_3 - sin\phi\hat{k}_3\right) + \dot{\psi}\left(cos\phi\hat{k}_2 - sin\theta\hat{I}_3\right) \qquad \text{3-11}$$

$${}^{R}\vec{\omega}^{R_3} = \dot{\phi}\hat{I}_3 + \dot{\theta}\left(cos\phi\hat{j}_3 - sin\phi\hat{k}_3\right) + \dot{\psi}cos\theta\left(sin\phi\hat{j}_3 + cos\phi\hat{k}_3\right) - \psi sin\theta\left(\hat{I}_3\right) \qquad \text{3-12}$$

$$\boxed{{}^{R}\vec{\omega}^{R_3} = \hat{\imath}_3\left(\dot{\phi} - \dot{\psi}sin\theta\right) + \hat{\jmath}_3\left(\dot{\theta}cos\phi + \dot{\psi}cos\theta sin\phi\right) + \hat{k}_3\left(-\dot{\theta}sin\phi + \dot{\psi}cos\theta cos\phi\right)} \quad 3\text{-}13$$

The translational position and rotational position of a reference frame associated with the rigid body relative to the inertial frame of a quadcopter can be used to express the location of a rigid body in space. The orientation or rotational position of the quadcopter is indicated by the Euler angle of rotation about center of quadcopter. The following rotations are believed to align the earths with the body's fixed frame transfer of an earth-fixed coordinate system to a body-fixed coordinate system is described by a rotating matrix The transformation can be consisting of the following rotation along each axis;

- Rotating the body frame around the z-axis of the earth frame by yaw angle ψ,
- Then followed by rotating around the y-axis by the pitch angle θ and
- Finally, by rotating around the x- axis by the roll angle ϕ.

Where all angles are bounded; in order to avoid to the system singularities, it is sufficient to assume

$$\frac{-\pi}{2} < \phi < \frac{\pi}{2}; \frac{-\pi}{2} < \theta < \frac{\pi}{2}; -\pi < \psi < \pi \quad 3\text{-}14$$

$$R_{(x,\phi)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}, \quad 3\text{-}15$$

$$R_{(y,\theta)} = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}, \quad 3\text{-}16$$

$$R_{(z,\psi)} = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad 3\text{-}17$$

The Euler rotation about Z-Y-X or R_{xyz} is given by,

$$R_{xyz} = {}^{I}R_B\, R_{(z,\psi)}\ R_{(y,\theta)}\, R_{(x,\phi)} \quad 3\text{-}18$$

$$R_{xyz} = \begin{bmatrix} cos\theta cos\psi & cos\theta sin\psi & -sin\theta \\ sin\phi sin\theta cos\psi - cos\phi sin\psi & sin\phi sin\theta sin\psi + cos\phi cos\psi & sin\phi cos\theta \\ cos\phi sin\theta cos\psi - sin\phi sin\psi & cos\phi sin\theta sin\psi - sin\phi cos\psi & cos\phi sin\theta \end{bmatrix} \quad 3\text{-}19$$

$$\begin{pmatrix} b1 \\ b2 \\ b3 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \times \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \times \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{3-20}$$

The rotating matrix's inverse is one of its characteristics is equal to the transpose (orthogonally).

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} cos\theta cos\psi & -cos\phi sin\psi + sin\phi sin\theta cos\psi & cos\phi sin\theta cos\psi + sin\phi sin\psi \\ cos\theta sin\psi & sin\phi sin\theta sin\psi + cos\phi cos\psi & cos\phi sin\theta sin\psi - sin\phi cos\psi \\ -sin\theta & sin\phi cos\theta & cos\phi cos\theta \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

3-21

Translational Kinematics: - The state variables $(\dot{x}, \dot{y}, \dot{z})$ are inertial frame parameters whereas, velocities (u, v, w) are body frame parameters. They can be related through the transformation, matrix as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = (R_b^I)^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad \text{3-22}$$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} cos\theta cos\psi & sin\phi sin\theta cos\psi - cos\phi sin\psi & cos\phi sin\theta cos\psi + sin\phi sin\psi \\ cos\theta sin\psi & sin\phi sin\theta sin\psi + cos\phi cos\psi & cos\phi sin\theta sin\psi - sin\phi cos\psi \\ -sin\theta & sin\phi cos\theta & cos\phi cos\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad \text{3-23}$$

Rotational Kinematics: - Since the yaw, pitch and roll are measured relative to different coordinate systems, the transformation for each is different. So, the angular velocities (p, q, r) are obtained as follows:

Body- Reference Frame

$$\begin{bmatrix} \omega_{roll} \\ \omega_{pitch} \\ \omega_{yaw} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{3-24}$$

OR

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -sin\theta \\ 0 & cos\theta & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad \text{3-25}$$

Reference Frame

$$^{R}\vec{\omega}^{R_3} = p\hat{\imath}_3 + q\hat{\jmath}_3 + r\hat{k}_3 \qquad \textit{3-26}$$

$$^{R}\vec{\omega}^{R_3} = \hat{\imath}_3(\dot{\phi} - \dot{\psi}sin\theta) + \hat{\jmath}_3(\dot{\theta}cos\phi + \dot{\psi}cos\theta sin\phi) + \hat{k}_3(-\dot{\theta}sin\phi + \dot{\psi}cos\theta cos\phi) \qquad \textit{3-27}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & tan\theta sin\phi & tan\theta cos\phi \\ 0 & cos\phi & -sin\phi \\ 0 & \frac{sin\phi}{cos\theta} & \frac{cos\phi}{cos\theta} \end{bmatrix} \begin{bmatrix} \omega_{roll} \\ \omega_{pitch} \\ \omega_{yaw} \end{bmatrix} \qquad \textit{3-28}$$

3.2. Dynamics Model

Translational Dynamics: By applying the Newton's second law to the translational motion, we get:

$$\boldsymbol{f} = m\frac{d\boldsymbol{v}}{dt} = m\left(\frac{d\boldsymbol{v}}{dt} + \boldsymbol{\omega_b} \times \boldsymbol{v}\right) \qquad \textit{3-29}$$

Where, $\frac{\mathbf{dv}}{\mathrm{dt}}$ is the time derivative of the velocity vector V, $\omega_b = (p, q, r)^T$ is the angular velocity vector of the rotating frame of reference, X denotes the cross-product operation.,$v = (u, v, w)^T$.

We have used Coriolis equation to evaluate time derivative of velocities in the non-inertial frame (body frame). Substituting the relevant equations, we get:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \qquad \textit{3-30}$$

Assuming absence of any external disturbances, we have,

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = R_v^b\begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ f \end{pmatrix} \qquad \textit{3-31}$$

Where, f is the total upward thrust and mg is the weight of the quadcopter.

Rotational Dynamics By applying Newton's laws to rotational motion, we get:

$$\tau = \frac{dL}{dt} = \left(\frac{dL}{dt} + \omega_b + L\right) \qquad \textit{3-32}$$

where, $L = J\omega_b$ is the angular momentum and τ is the applied torque. We have used Coriolis equation to evaluate time derivative of angular momentum in the non-inertial frame (body frame). Assuming quadcopter a symmetric body, J simplifies to a diagonal matrix. This equation can be simplified into:

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} qr\frac{J_y-J_z}{J_x} \\ pr\frac{J_z-J_x}{J_y} \\ pq\frac{J_x-J_y}{J_z} \end{pmatrix} + \begin{pmatrix} \frac{\tau_\phi}{J_x} \\ \frac{\tau_\theta}{J_y} \\ \frac{\tau_\psi}{J_z} \end{pmatrix} \qquad 3\text{-}33$$

Where, $\tau = \left(\tau_\phi, \tau_\theta, \tau_\psi\right)$ and J_x, J_y, J_z are the diagonal entries of the inertial matrix J.

These will be the various modules that will be used in our simulator. The Coriolis terms, time derivative of the transformation matrix is neglected to get these much simpler set of equations:

$$\ddot{\Phi} = \dot{\theta}\dot{\psi}\left(\frac{J_y-J_z}{J_x}\right) + \frac{U_2}{J_x} \qquad 3\text{-}34$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{J_z-J_x}{J_y}\right) + \frac{U_3}{J_y} \qquad 3\text{-}35$$

$$\ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{J_x-J_y}{J_z}\right) + \frac{U_4}{J_z} \qquad 3\text{-}36$$

$$\ddot{z} = \frac{U_1}{m}cos\phi cos\theta - g \qquad 3\text{-}37$$

$$\ddot{X} = \frac{U_1}{m}(cos\phi sin\theta cos\psi + sin\phi sin\psi) \qquad 3\text{-}38$$

$$\ddot{Y} = \frac{U_1}{m}(cos\phi sin\theta sin\psi - sin\phi cos\psi) \qquad 3\text{-}39$$

Where, the control inputs U_1-U_4 are given as:

$$U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) - mg \qquad 3\text{-}40$$

$$U_2 = b(\omega_4^2 - \omega_2^2) = \tau_\phi \qquad 3\text{-}41$$

$$U_3 = b(\omega_3^2 + \omega_1^2) = \tau_\theta \qquad 3\text{-}42$$

$$U_4 = b(\omega_4^2 + \omega_2^2 - \omega_3^2 - \omega_1^2 = \tau_\psi \qquad 3\text{-}43$$

State space reorientation of dynamic system

$$x_1 = x \ (Postion\ along\ x - axis) \qquad 3\text{-}44$$

$$x_2 = \dot{x} \ (Velocity\ along\ x - axis) \qquad 3\text{-}45$$

$$x_3 = \theta \ \ (Pitch\ angle) \qquad 3\text{-}46$$

$$x_4 = \dot{\theta} \ \ (Pitch\ Rate) \qquad 3\text{-}47$$

$$x_5 = y \ \ (Postion\ along\ y - axis) \qquad 3\text{-}48$$

$$x_6 = \dot{y} \ (Velocity\ along\ y - axis) \tag{3-49}$$

$$x_7 = \phi \ (Roll\ angle) \tag{3-50}$$

$$x_8 = \dot{\phi} \ (Roll\ Rate) \tag{3-51}$$

$$x_9 = z \ (Postion\ along\ z - axis) \tag{3-52}$$

$$x_{10} = \dot{z} \ (Velocity\ along\ z - axis) \tag{3-53}$$

$$x_{11} = \psi \ (Yaw\ angle) \tag{3-54}$$

$$x_{12} = \dot{\psi} \ (Yaw\ Rate) \tag{3-55}$$

State Differential Equation

$$\dot{x}_1 = \dot{x} = x_2 \tag{3-56}$$

$$\dot{x}_2 = \ddot{x} = \frac{U_1}{m}(cos\phi sin\theta cos\psi + sin\phi sin\psi) \tag{3-57}$$

$$\dot{x}_3 = \dot{\theta} = x_4 \tag{3-58}$$

$$\dot{x}_4 = \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{J_z - J_x}{J_y}\right) + \frac{U_3}{J_y} \tag{3-59}$$

$$\dot{x}_5 = \dot{y} = x_6 \tag{3-60}$$

$$\dot{x}_6 = \ddot{y} = \frac{U_1}{m}(cos\phi sin\theta sin\psi - sin\phi cos\psi) \tag{3-61}$$

$$\dot{x}_7 = \dot{\phi} = x_8 \tag{3-62}$$

$$\dot{x}_8 = \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) + \frac{U_2}{J_x} \tag{3-63}$$

$$\dot{x}_9 = \dot{z} = x_{10} \tag{3-64}$$

$$\dot{x}_{10} = \ddot{z} = \frac{U_1}{m}cos\phi cos\theta - g \tag{3-65}$$

$$\dot{x}_{11} = \dot{\psi} = x_{12} \tag{3-66}$$

$$\dot{x}_{12} = \ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{J_x - J_y}{J_z}\right) + \frac{U_4}{J_z} \tag{3-67}$$

State Vector X

$$x^T = [x \quad \dot{x} \quad \theta \quad \dot{\theta} \quad y \quad \dot{y} \quad \phi \quad \dot{\phi} \quad z \quad \dot{z} \quad \psi \quad \dot{\psi}] \tag{3-68}$$

The Input matrix U is given

$$U^T = [U_1 \quad U_2 \quad U_3 \quad U_4] \tag{3-69}$$

The Output matrix Y is given

$$Y^T = [x \quad y \quad z \quad \phi \quad \theta \quad \psi] \tag{3-70}$$

The quadcopter dynamics can be written in compact form

$$\dot{x} = f(x) + g(x)u \tag{3-71}$$

Where,

$$f(x) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ p + qsin(\phi)\,tan(\theta) + rcos(\phi)\,tan(\theta) \\ qcos(\phi) - rsin(\phi) \\ qsin(\phi)\,sec(\theta) + rcos(\phi)\,sec(\theta) \\ 0 \\ 0 \\ -g \\ \frac{I_y - I_z}{I_x} qr \\ \frac{I_z - I_x}{I_y} pr \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix} \tag{3-72}$$

And

$$g(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{cos(\phi)\,sin(\theta)\,cos(\psi) + sin(\phi)\,sin(\psi)}{m} & 0 & 0 & 0 \\ \frac{cos(\phi)\,sin(\theta)\,sin(\psi) - sin(\phi)\,cos(\psi)}{m} & 0 & 0 & 0 \\ \frac{cos(\theta)\,cos(\psi)}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{bmatrix} \tag{3-73}$$

3.3. Linear Model

With the nonlinear model detailed, the next step is to linearize the plant in order to utilize the linear control approaches described in this research.

The hover condition $P = [x \quad y \quad z]^T, \eta = [0 \quad 0 \quad 0]^T$, are the point equilibrium from the linearization derived and the yaw angle is also regarded as zero. It should be linearized could be performed for the other situations as well-chosen due to its simplicity. The following approximations are obtained by reducing the Taylor series to the first order term:

$$\begin{aligned} &cos\phi \approx cos\theta \approx cos\psi \approx 1 \\ &sin\phi \approx tan\phi \approx \phi \\ &sin\theta \approx tan\theta \approx \theta \\ &sin\psi \approx tan\psi \approx 0 \end{aligned} \tag{3-74}$$

3.3.1. Position X And Pitch Subsystem State Space Representation

$$\ddot{x} = \frac{U_1}{m}(cos(\phi)\, sin(\theta)\, cos(\psi) + sin(\phi)\, sin(\psi)) \tag{3-75}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{Jz-Jx}{Jy}\right) + \frac{U_3}{Jy} \tag{3-76}$$

Using Taylor Series Approximation

$$cos\phi \approx cos\theta \approx cos\psi \approx 1 \tag{3-77}$$

$$sin\phi \approx tan\phi \approx \phi \tag{3-78}$$

$$sin\theta \approx tan\theta \approx \theta \tag{3-79}$$

$$sin\psi \approx tan\psi \approx 0 \tag{3-80}$$

$$U_1 \approx mg \tag{3-81}$$

$$U_3 \approx \tau_\theta \tag{3-82}$$

$$\ddot{x} \approx \frac{mg}{m}(\theta + 0) = g\theta \tag{3-83}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{Jz-Jx}{Jy}\right) + \frac{U_3}{Jy} \approx \frac{U_3}{Jy} \approx \frac{\tau_\theta}{Jy} \tag{3-84}$$

The state space representation

$$x_x = [x \quad \dot{x} \quad \theta \quad \dot{\theta}]^T \tag{3-85}$$

$$U_3 = \tau_\theta \tag{3-86}$$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & g & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ \frac{1}{J_y}\end{bmatrix} \qquad 3\text{-}87$$

3.3.2. Position Y And Roll Subsystem State System Representation

Using Taylor series approximation

$$cos\phi \approx cos\theta \approx cos\psi \ \approx 1$$
$$sin\phi \approx tan\phi \approx \phi \qquad 3\text{-}88$$
$$sin\theta \approx tan\theta \approx \theta$$
$$sin\psi \approx tan\psi \approx 0$$
$$U_1 \approx mg \qquad 3\text{-}89$$
$$\ddot{y} = \frac{U_1}{m}(cos\phi sin\theta sin\psi - sin\phi cos\psi) \approx \ -g\phi \qquad 3\text{-}90$$
$$\ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) + \frac{U_2}{J_x} \approx \frac{\tau_\phi}{J_x} \qquad 3\text{-}91$$

State space representation for

$$x_y = [y \quad \dot{y} \quad \phi \quad \dot{\phi}]^T \qquad 3\text{-}92$$
$$U_2 = \tau_\phi \qquad 3\text{-}93$$
$$\begin{bmatrix}\dot{x}_5\\ \dot{x}_6\\ \dot{x}_7\\ \dot{x}_8\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & -g & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_5\\ x_6\\ x_7\\ x_8\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ \frac{1}{J_x}\end{bmatrix}U_2 \qquad 3\text{-}94$$

3.3.3. Yaw And Height Subsystem State Space Representation

using Taylor series approximation

$$\ddot{z} = cos\phi cos\theta \frac{T}{m} - \ g \ \approx \ \frac{1}{m}(T - mg) \qquad 3\text{-}95$$
$$\ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{J_x - J_y}{J_z}\right) + \frac{U_4}{J_z} \approx \frac{U_4}{J_z} \qquad 3\text{-}96$$

State space representation

$$x_2 = [z \quad \dot{z} \quad \psi \quad \dot{\psi}]^T \qquad 3\text{-}97$$
$$U_1 = T - mg \qquad 3\text{-}98$$
$$U_2 = \tau_\psi \qquad 3\text{-}99$$

$$\begin{bmatrix} \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad \textit{3-100}$$

The state space representation for dynamic quadcopter system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{J_z} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad \textit{3-101}$$

$$Y = CX_{(t)} + DU_{(t)} \quad \textit{3-102}$$

$$\begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad \textit{3-103}$$

CHAPTER FOUR

4. PID CONTROLLER DESIGN

The most common type of control system is the Proportional-Integral-Derivative (PID) control system. The PID controller is known as a closed loop feedback system. The controller calculates the difference between the actual and desired state and produces an error value. The measurements from the quadcopter's sensors are fed back to compute this error signal, closing the loop from output to input. This controller is popular because it is somewhat intuitive to tune, easy to implement in code with little processing power, and utilizes measurements that are available on-board most systems. The output of any PID control system is a control value u that will drive the system closer to the desired state. The composition of the PID controller is shown below in both the time domain and frequency domain

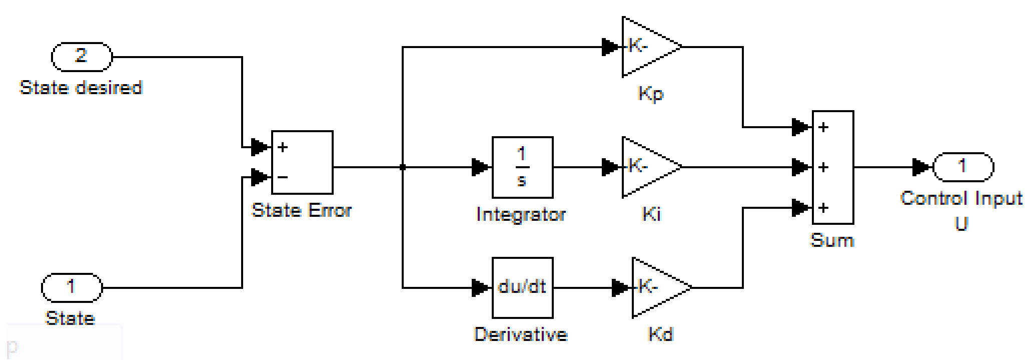


Figure 4.1 PID Control Stricture

Where,

U = control input

P = Proportional control term

I = Integral control term

D = Derivative control term

K_p = Proportional control gain

K_i = Integral control gain

K_D = Derivative control gain

t = instantaneous time

$x_d(t)$ = desired value of state x at time t

$x(t)$ = measured value of state x at time t

$e(t)$ = error value at time t

$\dot{e}(t)$ = derivative of error at time t

$$u = P + D + I \tag{4-1}$$

$$e(t) = x_d(t) - x(t) \tag{4-2}$$

$$P = k_p e(t) \tag{4-3}$$

$$I = k_I \int_0^t e(t-1)dt \tag{4-4}$$

$$D = k_D \dot{e}(t) \tag{4-5}$$

$$u(t) = k_p e(t) + k_I \int_0^t e(t-1)dt + k_D \dot{e}(t) \tag{4-6}$$

$$U(s) = k_p + \frac{k_I}{s} + k_D s \tag{4-7}$$

4.1. PID Controller Design for Altitude, Heading and Attitude Control of Quadcopter

A closed-loop control uses a feedback controller to compensate for changes in the input. Figure 4.2 shows an entire control system for quadcopter trajectory tracking using a PID feedback control system. A proportional, integral, and derivative (PID) control action is used to control the quadcopter's roll, pitch, and yaw angles, as well as altitude (force along the z-axis) and translation along the *X* and *Y* axes. A proportional-differential-integrator (PID) controller is used to regulate the quadcopter system's roll, pitch, and yaw angles, as well as its altitude and location along the X and Y axes.

Nested Control Loops

When nested control loops are used, it means that there are multiple levels of control loops operating simultaneously. The outer loop controls a higher-level process or system, while the inner loop controls a lower-level aspect of that process or system.

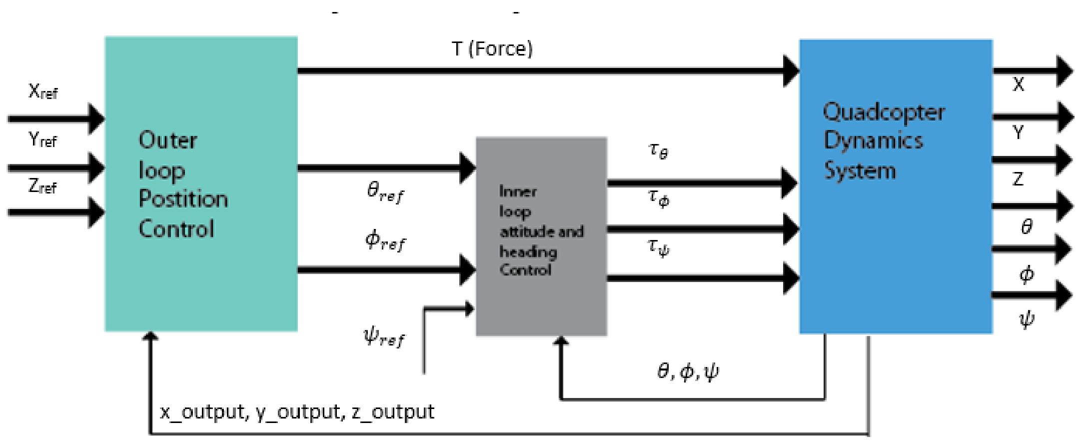


Figure 4.2 Cascade Control Stricture of Quadcopter

4.2. PID Equations

In this section, an overview of the PID control algorithm will be presented. The main objective is to formalize a design for the PID controller to stabilize the flight of the UAV. The control input u used to maintain stability of the UAV is designed as

$$u(t) = k_p e(t) + k_I \int_0^t e(t-1)dt + k_D \dot{e}(t) \qquad \textit{4-8}$$

The gain K values are associated to their respective control mechanism:

Proportional gain K_p, integral gain K_I, and derivative gain K_D. the error e(t) is formulated by

$$e(t) = s_P - P_V(t) \qquad \textit{4-9}$$

The desired position s, is subtracted from the measured process variable P_V to provide a correction value to apply to the PID. It is required that the PID controller maintain stability in yaw, pitch and roll while simultaneously controlling the error.

Efficient tracking of the UAV allows for a smooth and stable flight path. In figure 4.3 below, a cubed space in the inertial frame that tracks the instantaneous position of the UAV through every adjustment is presented.

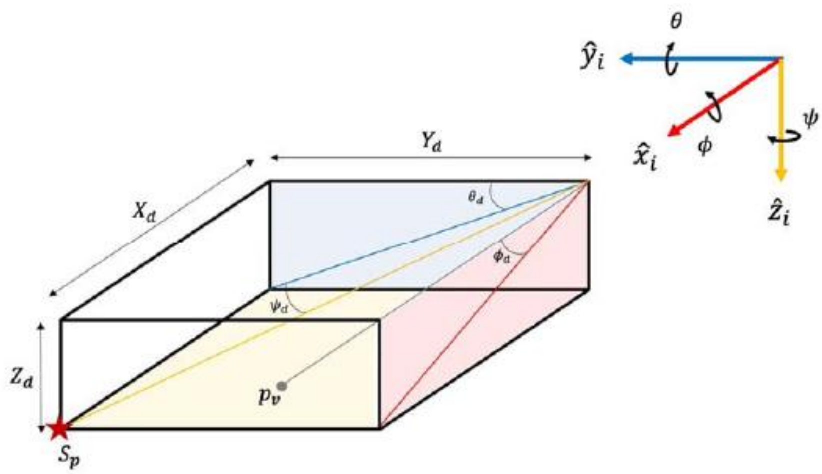


Figure 4.3 The Drone Rotational Angles Description

Each adjustment brings the UAV closer to the desired position. Achieving the desired position is accomplished by using the desired coordinate positions, X_d, Y_d and Z_d, and the orientation angles, ψ_d, θ_d *and* ϕ_d. To formulate the desired angles, the use of geometry depicted in Figure 4.3 is used with

$$\phi_d = tan^{-1}\left({}^{Z_d}/_{Y_d}\right), \quad \text{4-10}$$

$$\theta_d = sin^{-1}\left(\frac{X_d}{\sqrt{X_d^2+Z_d^2}}\right) \quad \text{4-11}$$

$$\psi_d = cos^{-1}\left({}^{Y_d}\Big/_{\sqrt{X_d^2+Y_d^2+Z_d^2}}\right) \quad \text{4-12}$$

There are six control inputs that also must be formulated. They are described in [4-8] by

$$u_x = K_p(X_d - X) + K_i \int_0^t (X_d - X)\, dt + K_d \frac{d(X_d - X)}{dt} \quad \text{4-13}$$

$$u_y = K_p(Y_d - Y) + K_i \int_0^t (Y_d - Y)\, dt + K_d \frac{d(Y_d - Y)}{dt} \quad \text{4-14}$$

$$u_z = K_p(Z_d - Z) + K_i \int_0^t (Z_d - Z)\, dt + K_d \frac{d(Z_d - Z)}{dt} \quad \text{4-15}$$

$$u_\phi = K_{p\phi}(\phi_d - \phi) + K_{i\phi} \int_0^t (\phi_d - \phi)\, dt + K_{d\phi} \frac{d(\phi_d - \phi)}{dt} \quad \text{4-16}$$

$$u_\theta = K_{p\theta}(\theta_d - \theta) + K_{i\theta} \int_0^t (\theta_d - \theta)\, dt + K_{d\theta} \frac{d(\theta_d - \theta)}{dt} \quad \text{4-17}$$

And

$$u_\psi = K_{p\psi}(\psi_d - \psi) + K_{i\psi} \int_0^t (\psi_d - \psi)\, dt + K_{d\psi} \frac{d(\psi_d - \psi)}{dt} \quad \text{4-18}$$

The gains associated with each of the angle control inputs differ based on the stability criteria for each motion. The position is commanded to the UAV with individual gain values for each specific orientation angle. After several iterations of these specific control inputs, the UAV arrives to the desired position.

4.2.1. PID for Roll Angle Correction

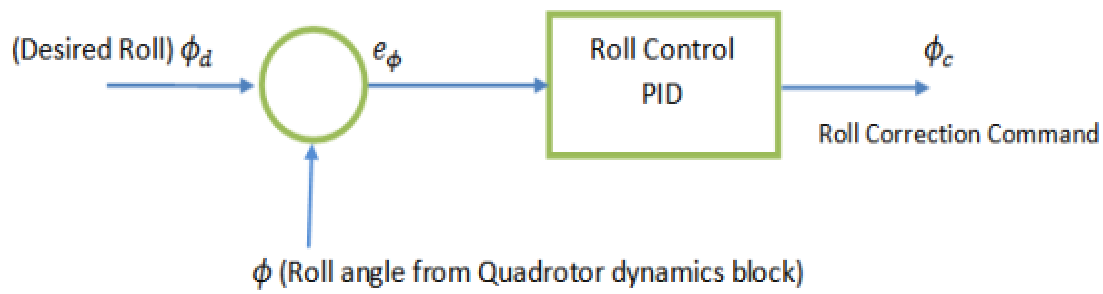


Figure 4.4: PID for Roll Angle Correction

$$e_\phi = \phi_d - \phi \quad \text{4-19}$$

$$u_\phi = K_{p\phi}(\phi_d - \phi) + K_{i\phi}\int_0^t(\phi_d - \phi)\,dt + K_{d\phi}\frac{d(\phi_d - \phi)}{dt} \quad \text{4-20}$$

Error in roll angle is calculated by taking difference between desired roll angle and current roll angle. Then PID is applied to minimize error as shown in Figure 4.4. is set to decrease the rise time but it causes overshoot in the system. So, is set to reduce overshoot. If there exist some steady state error then will be used for reducing the steady state error.

4.2.2. PID Controller for Pitch Angle

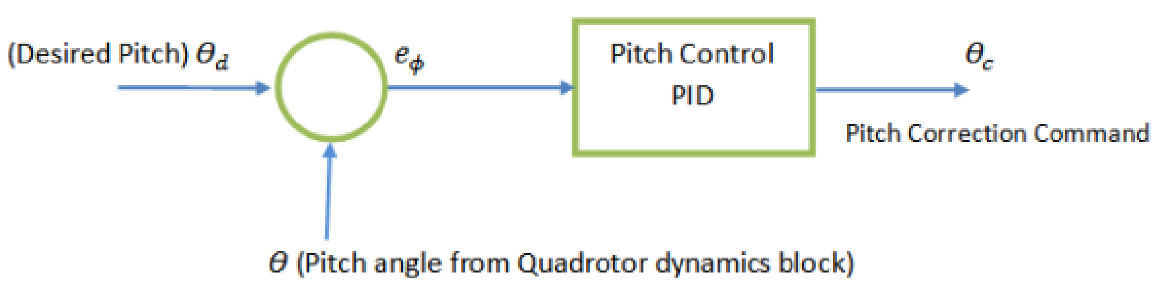


Figure 4.5: PID for Pitch Angle Correction

$$e_\theta = \theta_d - \theta \quad \text{4-21}$$

$$u_\theta = K_{p\theta}(\theta_d - \theta) + K_{i\theta}\int_0^t(\theta_d - \theta)\,dt + K_{d\theta}\frac{d(\theta_d - \theta)}{dt} \quad \text{4-22}$$

Error in pitch angle is calculated by taking difference between desired pitch angle and current roll pitch angle. Then PID is applied as shown in Figure 4.5. Instead of taking derivative of errors in roll angle, error in pitch angle and error in yaw angle, I directly take the roll angle rate, pitch angle rate and yaw angle rate from the quadcopter dynamic block

4.2.3. PID Controller for Yaw Angle

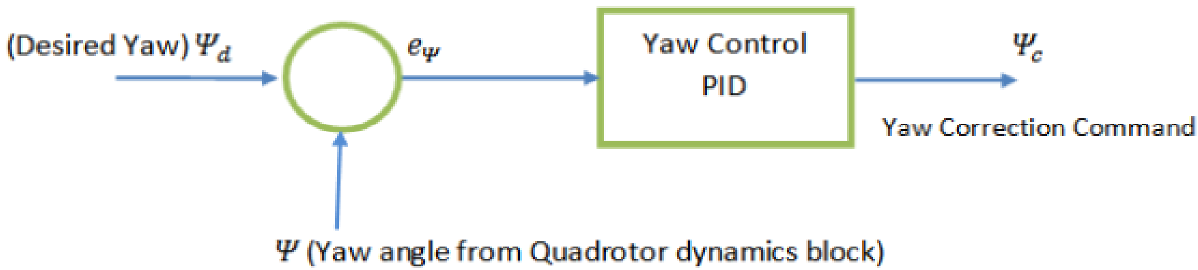


Figure 4.6: PID for Yaw Angle Correction

$$e_\psi = \psi_d - \psi \qquad \text{4-23}$$

$$u_\psi = K_{p\psi}(\psi_d - \psi) + K_{i\psi}\int_0^t(\psi_d - \psi)\,dt + K_{d\psi}\frac{d(\psi_d - \psi)}{dt} \qquad \text{4-24}$$

Error in yaw angle is calculated by taking difference between desired yaw angle and current yaw angle. Then PID is applied as shown in Figure 4.6. Integrator limit is applied to prevent the output from saturation because it can damage the actuator of the system and also have a bad impact on system energy as it causes instability in the system.

4.3. Designing The Altitude Controller for The Simplified Model

$$m\ddot{h}(t) = 4kT\,u(t) - mg \qquad \text{4-25}$$

Where m is the quadcopter mass, h is the quadcopter altitude, k_T is the thrust coefficient, u is the control output (a single motor command), g is the gravitational constant, and t is time. Figure 4.7 below is the block diagram for this simplified model.

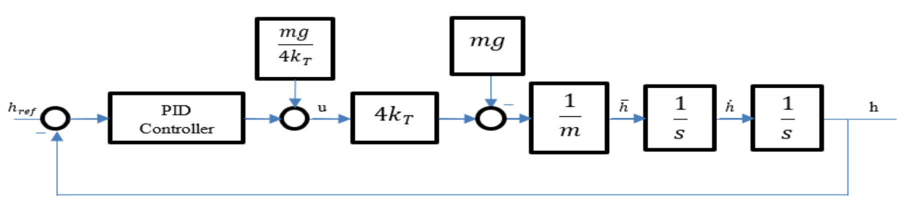


Figure 4.7: Simplified Model of Quadcopter

$$u(t) = K_i \int_0^t [h_{ref}(\tau) - h(\tau)]d\tau + K_p[h_{ref}(t) - h(t)] - K_d\dot{h}(t) + \frac{mg}{4k_T} \qquad 4\text{-}26$$

where h_{ref} is the reference altitude and K_P, K_I, and K_D are the proportional, integral, and derivative gains to be determined.

To determine the stability of the system, the relation between $u_{throttle}$ and the height h has been determined. The following calculation are done assuming that ϕ and θ are small, and that the angle feed-forward will handle angular deviations. Assuming that the thrust from the motors is a linear function of the control signal, the thrust force is $F_{throttle}=K_t\,u_{throttle}$, for some parameter Kt. When hovering, $u_{throttle} = u_0$ and $F_{throttle}=mg$. Hence,

$$K_t u_0 = mg ==> K_t = \frac{mg}{u_0} \qquad 4\text{-}27$$

Also, using F=ma

$$a = \frac{F}{m} = \frac{K_t u_{throttle} - mg}{m} = \frac{K_t u_0 + K_t u_{PID} - mg}{m} = \frac{K_t u_{PID}}{m} = \frac{u_{PID}\, g}{u_0} \qquad 4\text{-}28$$

Using u_{PID}, a differential equation can be written as

$$a = \ddot{h} = \frac{g}{u_0}\left(K_i \int_0^t [h_{ref} - h]d\tau + K_p[h_{ref} - h] - K_d\dot{h}(t)\right) \qquad 4\text{-}29$$

$$\frac{u_0}{g}\ddot{h} + K_p h + K_I \int_0^t h dt + K_D\dot{h} = K_P h_{ref} + K_I \int_0^t h_{ref} dt \qquad 4\text{-}30$$

This can be rewritten into a transfer function according to

$$\left(\frac{u_0}{g}s^3 + K_D s^2 + K_P s + K_I\right)h = (K_P s + K_I)h_{ref} \qquad 4\text{-}31$$

Into

$$\frac{h}{h_{ref}} = \frac{K_P s + K_I}{\frac{u_0}{g}s^3 + K_D s^2 + K_P s + K_I} \qquad 4\text{-}32$$

CHAPTER FIVE

5. SIMULATION RESULT AND ANALYSIS

The PID controller approach for the Quadrotor is tested using the MATLAB simulation software. The simulation is done using MATLAB/Simulink software. The proposed controller strategy for the quadrotor is implemented in this section in order to verify the accuracy and stability system using simulation. The program chosen to model proposed control and explain the quadrotor behavior system. In this section, we evaluate the proposed controller's capability to stabilize the quadrotor system by taking into consideration different time parameterized reference signals for quadrotor control system through computer simulation in MATLAB/Simulink. The entire Simulink diagram for quadrotor control system shown in Figure 5.1 for PID control. Figure 5.2 for the quadrotor system as shown below.

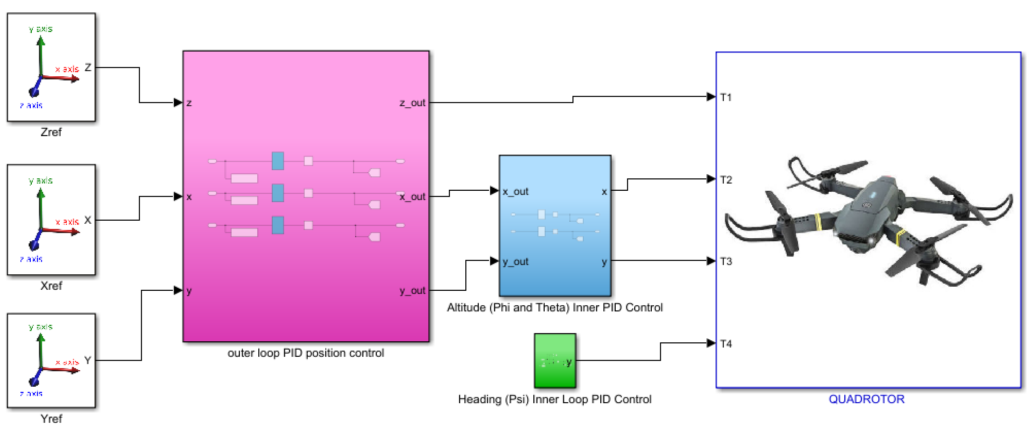


Figure 5.1 Simulink Block Diagram for Entire Inner-Outer Loop PID Control Of Quadrotor

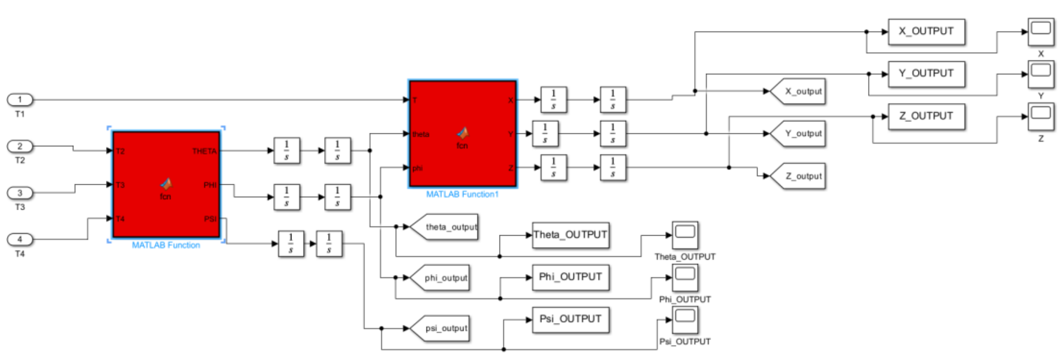


Figure 5.2 Simulink Block Diagram for Quadrotor System

The value of numerical parameters used to test proposed controller was show in the table 5-1 below

Table 5-1: Physical Parameters of Quadcopter

Parameter	*Value and unit*
Arm Length [L]	0.127 [meter]
Total mass [m]	0.46 [kg]
I_x	$2.2*10^3$[kg m^2]
I_Y	$2.9*10^3$[kg m^2]
I_Z	$5.3*10^3$[kg m^2]
Gravitational acceleration [g]	9.81 [m/s^2]

In this section, we have seen in detail that the proposed control system-tracking performance giving desired trajectories to states. The following figures show the performance of the PID performance for each of all states independently at the same time. In this paper, we are used reference trajectories by selecting reasonably for simulation purpose.

$$z_d = \begin{cases} 0m, & t \leq 0 \; and \; t > 5 \\ 1m, & otherwise \end{cases} \tag{5-1}$$

$$x_d = \begin{cases} 0m, & t \leq 0 \; and \; t > 5 \\ 1m, & otherwise \end{cases} \tag{5-2}$$

$$y_d = \begin{cases} 0m, & t \leq 0 \; and \; t > 5 \\ 1m, & otherwise \end{cases} \tag{5-3}$$

$$\psi_d = \begin{cases} 0rad, & t \leq 0 \; and \; t > 5 \\ 0.2\,rad, & otherwise \end{cases} \tag{5-4}$$

Initially the quadrotor is at rest starting from its flight (0, 0, 0, 0, 0, 0) state and follow the above desired trajectory with time, the time of simulation is taken 30 second. The controller can have managed quadrotor at any time as we want.

5.1. Position Controller Tracking Performance (Z, X And Y)

Table 5-2 shows the controller for PID Position control of the quadrotor system. Here the PID parameter is chosen so that the amount energy requires will be equal in controller in order to achieve the tracking problem.

Table 5-2: Controller Parameters

Controller	K_p	K_I	K_D
PID [Z]	0.085	0.004	0.448
PID [X]	0.09	0.031	0.22
PID [Y]	4.8	1.134	3.2

Table 5-3: Altitude Controller Performance Comparison

Controller	Time domain specification				
	t_r	t_s	t_p	overshoot	undershoot
PID [Z]	0.8911	16.9197	1.1072	10.7184	0
PID [X]	2.0504	22.2588	1.2495	24.9467	$1.0948e^{-44}$
PID [Y]	1.2044	17.1432	1.1194	11.9445	$1.2042e^{-44}$

Here, in this paper the performance of controllers evaluated domain specification namely, overshoot, rise time, settling time and under shoot. As shown in Table 5-3 PID respect the parameters addressed above under time domain specification with value.

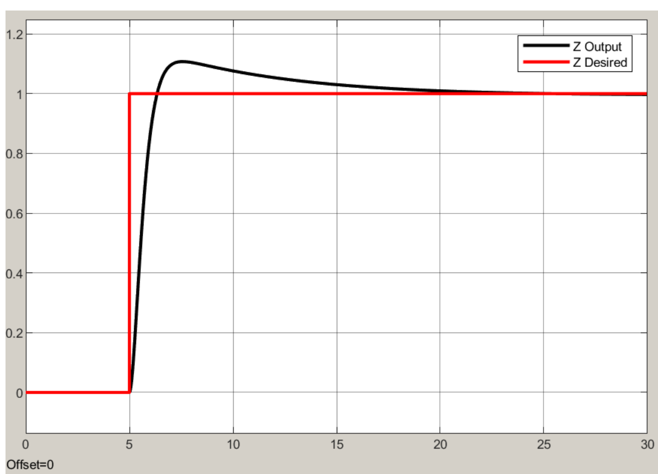


Figure 5.3 Quadcopter Altitude Control System

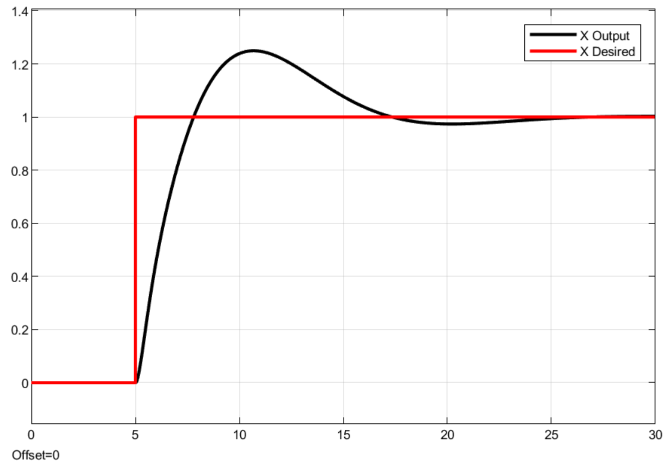


Figure 5.4 Quadcopter X Axis Position Control System

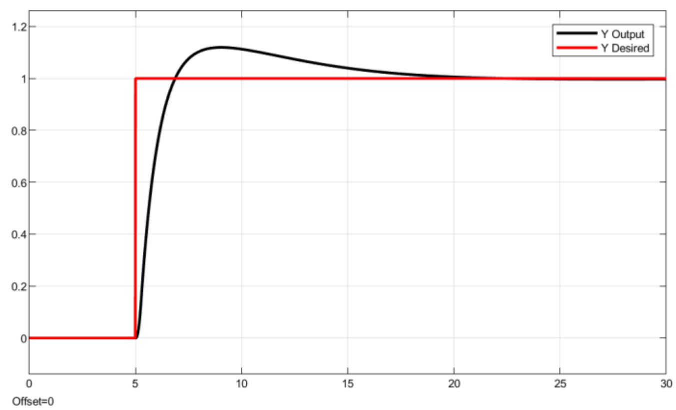


Figure 5.5 Quadcopter Y Axis Position Control System

5.2. Heading (Yaw) Controller Tracking Performance ψ

Table 5-4 shows the inner loop controller parameters for PID the heading control of the quadrotor system. Here the PID parameter is chosen so that the amount energy requires will be equal in both to achieve the tracking problem.

Table 5-4: Controller Parameters

Controller	K_I	K_P	K_d
PID	$4.36e^{-05}$	0.0009	0.005

In this paper the performance of controller evaluated using time domain specification namely, overshoot, rise time, settling time and under shoot.

Table 5-5: Yaw Controller Performance

Controller	T_r	T_s	T_p	Over shoot	Under shoot
PID	0.8911	16.9197	0.2214	10.7184	0

From Figure 5.6, the proposed controller tracking performance seen compared to the PID controller for the heading (ψ) control of the quadrotor system.

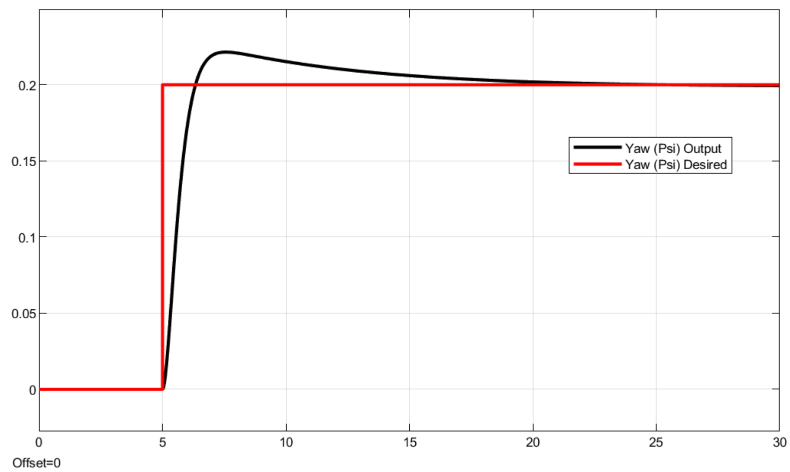


Figure 5.6: Quadrotor Yaw (Heading) Control System

5.3. Pitch Controller Tracking Performance (θ)

Table 5-6 shows the controller parameters for inner loop PID for the pitch control of the quadrotor system. Here the PID parameter is chosen so that the amount energy requires will be equal in controller in order to achieve the tracking problem.

Table 5-6: Controller Parameters

Controller	K_I	K_P	K_d
PID	0.01	0.009	0.02

From Figure 5.7, the proposed controller tracking performance seen compared to the PID controller for the pitch (θ) control of the quadrotor system. As shown from the results proposed system have good behaviors with good performance with less overshoot and oscillation with respect to the PID one.

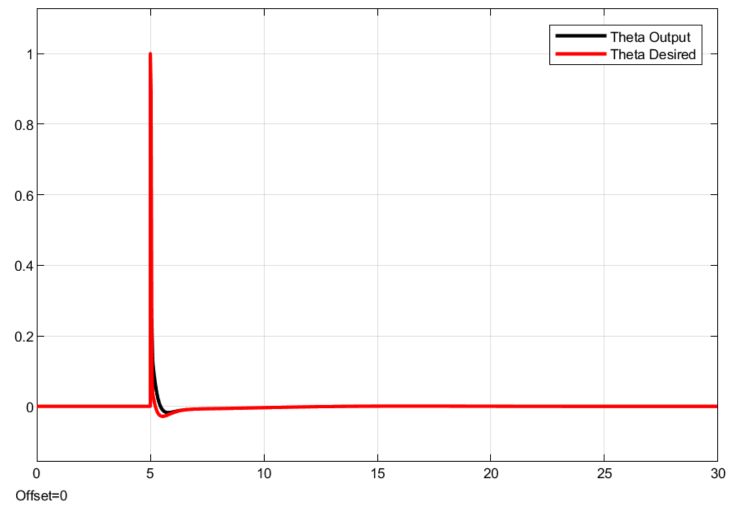


Figure 5.7: Quadrotor pitch PID control system

5.4. Control Of Roll in the Tracking Performance

Table 5-7 shows the controller parameters for inner loop PID and for the roll control of the quadrotor system. Here the PID parameter is chosen so that the amount energy requires will be equal in both controllers in order to achieve the tracking problem.

Table 5-7: Controller Parameters

Controller	K_I	K_P	K_d
PID	5.963	1.9	0.15

From Figure 5.8, the proposed controller tracking performance seen compared to the PID controller for the roll (θ) control of the quadrotor system.

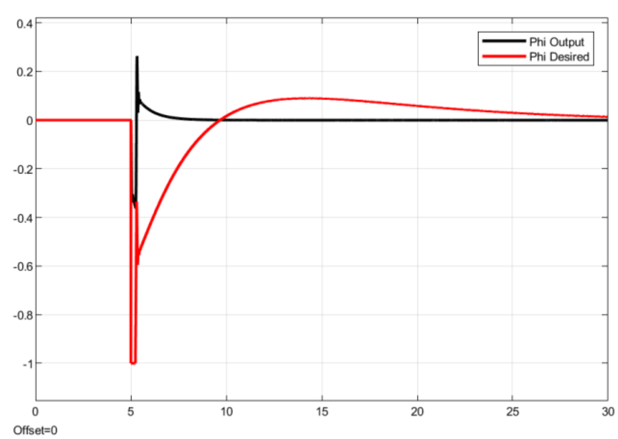


Figure 5.8 Quadrotor PID Roll control system.

In optimal control system the energy expenditure one most important control objectives such that this quantity one parameters which going to be minimized despite good time and frequency domain specifications. Therefore, the controller parameters for PID chosen so that the torque required achieving the tracking as shown from Figure 5.9 below which varies from nearly from 7Nm up to 4.5Nm for the altitude trust control of quadrotor and from -1Nm up to 1Nm for remaining torques.

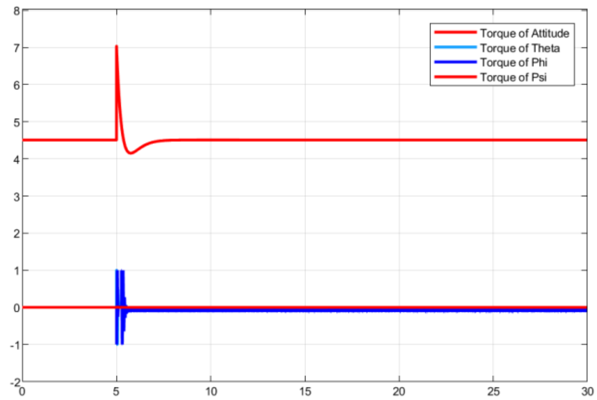


Figure 5.9 PID Torque Control Signals

Evaluate the performance the proposed control architecture trajectory was created and given as a reference for subsystem as shown in Figure 5.10, 5.11 and 5.12 for Z, X and Z respectively.

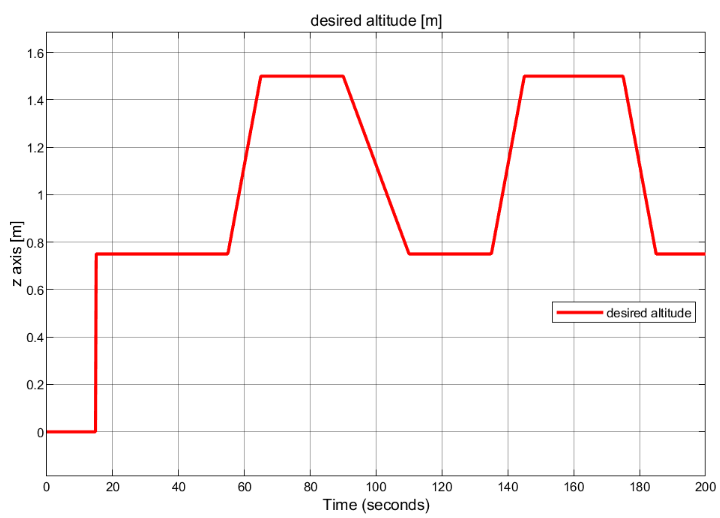


Figure 5.10 Reference of Altitude

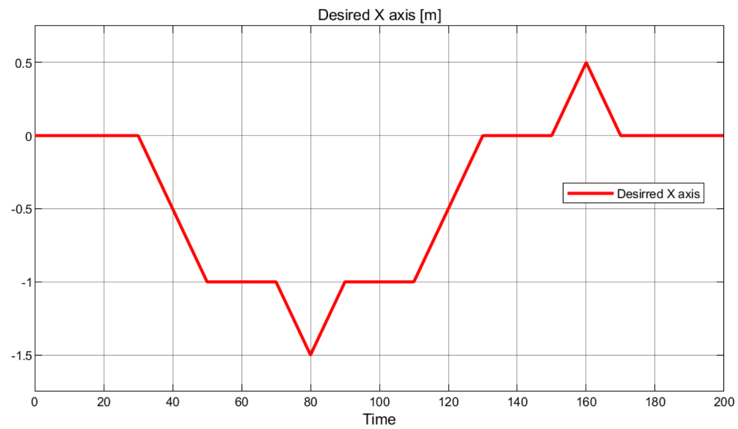


Figure 5.11 Reference of Position X Desired

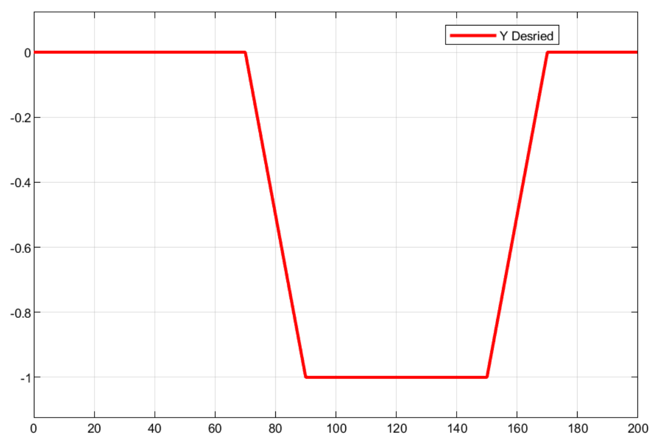


Figure 5.12 Reference of Position Y Desired

The performance of controller for given reference will be seen in the following Figure 5.13, 5.14 and 5.15 which shows good performance for the specified trajectory tracking.

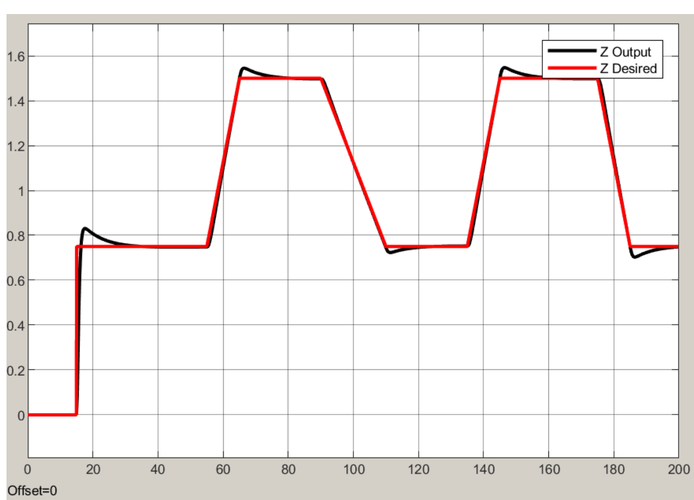


Figure 5.13 Altitude Tracking.

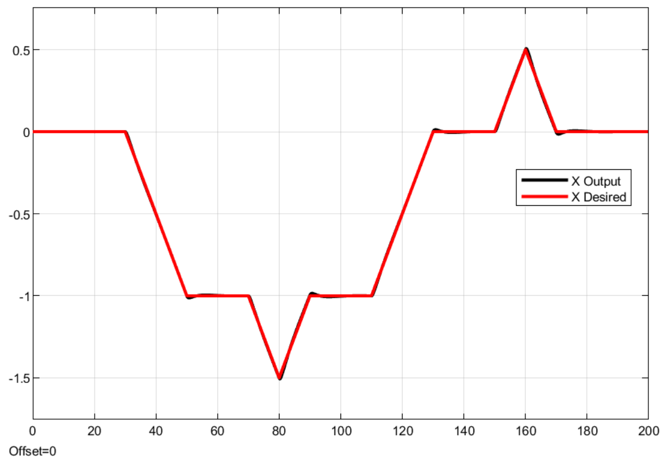


Figure 5.14 Positions Along X Axis Tracking

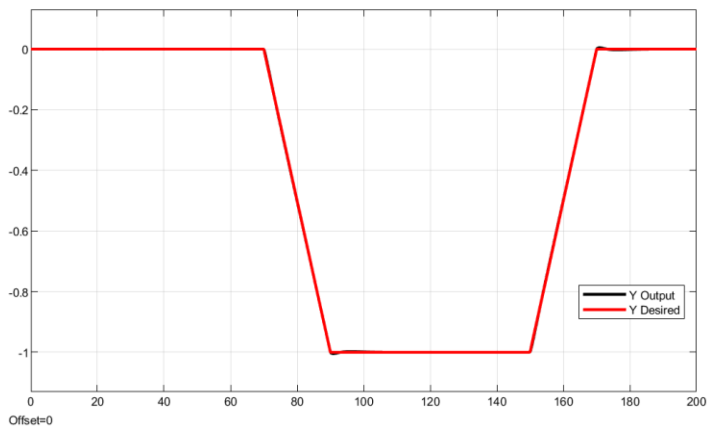


Figure 5.15 Position Along Y Axis Tracking

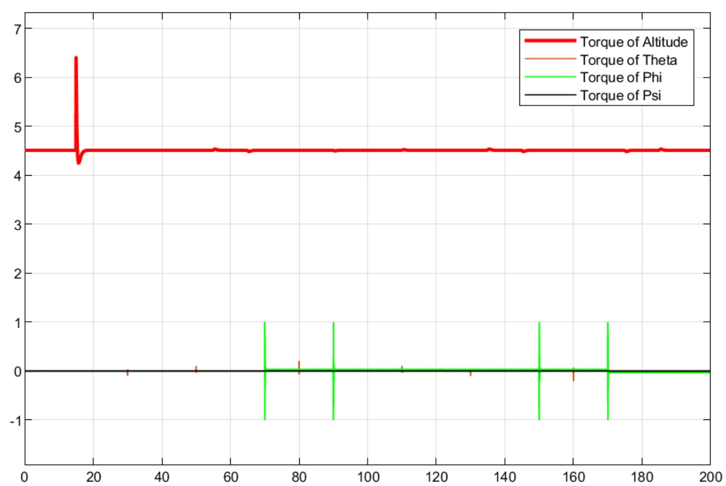


Figure 5.16 PID Control Signals

From Figure 5.17 the three-dimensional movement of quadrotor in the space will be indicated and shows with respect to PID controller

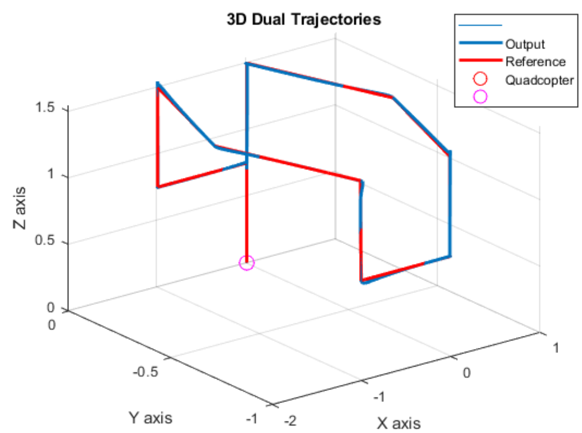


Figure 5.17 3D Tracking of Quadrotor System

The following reference trajectory will be used to test controller for more complex reference system for helix trajectory. In this case, the helix tracking performance of the designed using three controlled variables of X, Y and Z of quadrotor system. The controller shown in Figure 5.18 below and shows that especially helix reference trajectory.

$$Halix\ trajectory = \begin{cases} X_{ref} = 4 * sin\ (u) \\ Y_{ref} = 5 * cos\ (u) \\ Z_{ref} = 2 + u \end{cases} \tag{5-5}$$

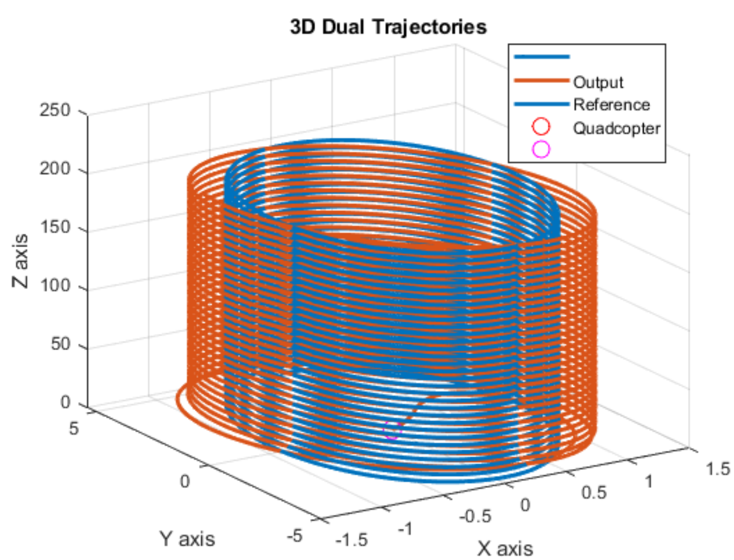


Figure 5.18 3D Helix Tracking of Quadrotor System

5.5. Position Control Tracking Performance (Z, X, Y) Under Matching Model Uncertainty.

For robustness designed controller uncertainty due to the matching model uncertainty of 50% deviation from the nominal value used here. As shown in Figure 5.19, 5.20 and 5.21, the proposed controller is totally insensitive to model uncertainty.

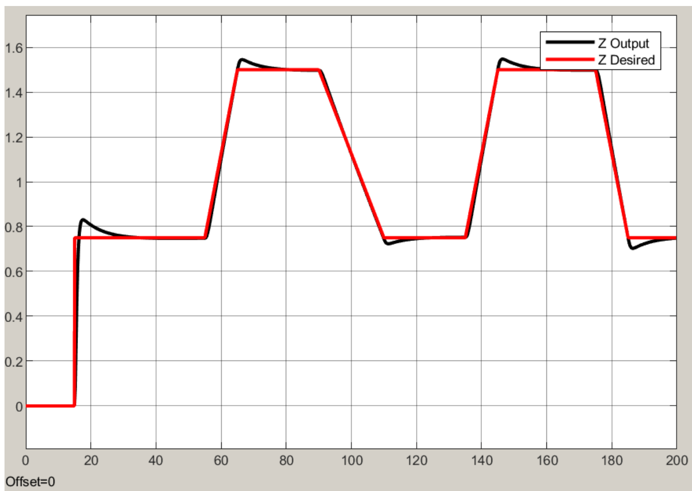


Figure 5.19 Altitude Control System Under Model Uncertainty.

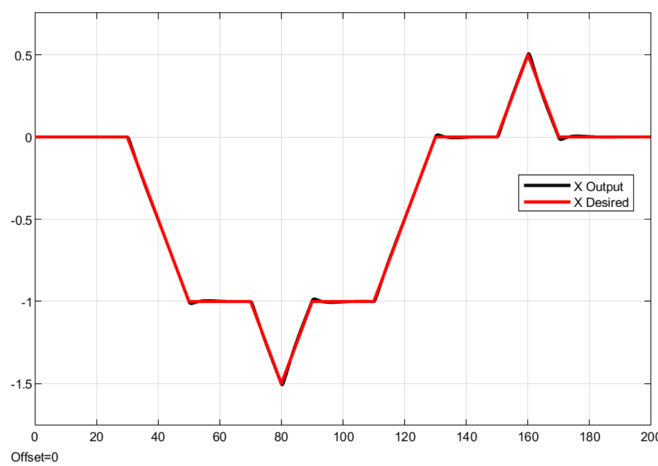


Figure 5.20 Position Along X Axis Control System Under Model Uncertainty

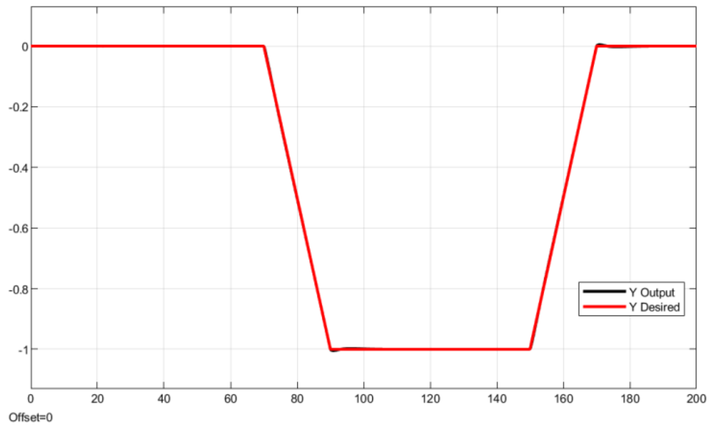


Figure 5.21 Controls of Heading In The Trajectory Tracking Under Model Uncertainty

5.6. Heading (Yaw) Controller Tracking Performance Under Matching Uncertainty.

Test robustness of the designed controller uncertainty due to the matching model uncertainty of 50% deviation from the nominal value used here. As shown in Figure. 5.22, the proposed controller is totally insensitive to model uncertainty

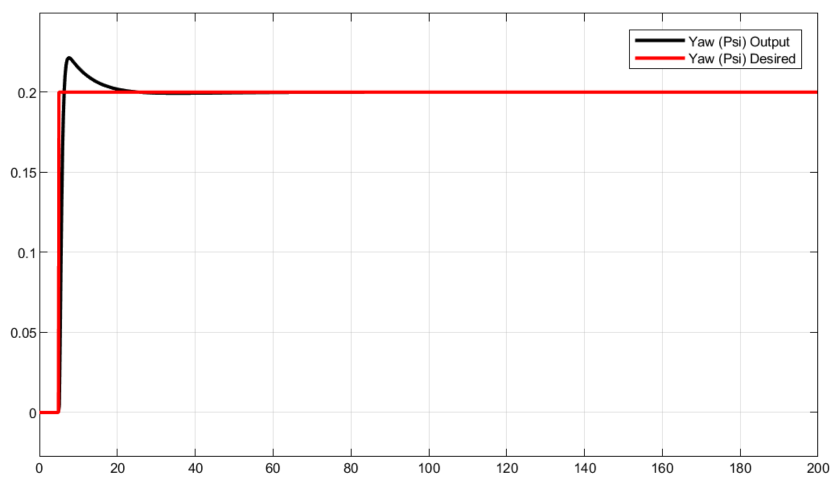


Figure 5.22 Yaw Control System Under Model Uncertainty

CHAPTER SIX

6. CONCLUSIONS AND WORK IN THE FUTURE

6.1. Conclusion

In this document dynamic model of quadrotor derived using Newton-Euler formulation and Linearized model obtained. Cascaded inner-outer loop trajectory tracking designed for quadrotor trajectory tracking. Outer loop position controller i.e., altitude, position along X and Y, heading and attitude motion control of quadrotor using PID controllers tracking designed. The performance of proposed controller compared to PID controller with different time domain metrics like rise time, settling time and overshoot. Moreover, error between all control variable indicated using absolute error histogram is to compare the results. The performance PID shows good work in both time domain specification and absolute error histogram in terms of rise, settling time and overshoot.

6.1.1. Future Works

The following ideas are suggested for future projects.

- Quadcopter controlling using LQR controller is more efficient and more stable compared to PID.
- Changing the controller to an Artificial Neural Network may also improve the performance of tracking a variable trajectory.
- Using the developed quadrotor controllers and adding capabilities such as prediction of future air conditions and disturbances
- The neural network is utilized to forecast weather conditions, and then the controller takes steps to reject air disturbances before the quadrotor takes flight.
- It can also be used in practice with the help of advanced controllers.
- By adding additional devices to the quadrotor, such as sensors, cameras, and actuators, the quadrotor can be used for of purposes, and the controller can take accurate and precise action. Furthermore, putting the designed controller into practice would be a very interesting extension of this work.

REFERENCES

[1] H. Kaushal and G. Kaddoum, “Applications of Lasers for Tactical Military Operations,” IEEE Access, vol. 5, pp. 20736–20753, 2017, doi: 10.1109/ACCESS.2017.2755678.

[2] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, “Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments,” pp. 21–45, 2016, doi: 10.1007/978-94-6265-132-6_2.

[3] K. Hassani and W. S. Lee, “Optimal tuning of linear quadratic regulators using quantum particle swarm optimization,” Int. Conf. Control. Dyn. Syst. Robot., no. 59, pp. 1–8, 2014.

[4] A. Sigalos, M. Papoutsidakis, A. Chatzopoulos, and D. Piromalis, “Design of a Flight Controller and Peripherals for a Quadcopter,” Int. J. Eng. Appl. Sci. Technol., vol. 04, no. 05, pp. 463–470, 2019, doi: 10.33564/ijeast. 2019.v04i05.067.

[5] G. Chowdhary and R. Jategaonkar, “Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter,” Aerosp. Sci. Technol., vol. 14, no. 2, pp. 106–117, 2010, doi: 10.1016/j.ast.2009.10.003.

[6] V. Raja, S. K. Solaiappan, P. Rajendran, S. K. Madasamy, and S. Jung, “Conceptual design and multi-disciplinary computational investigations of multirotor unmanned aerial vehicle for environmental applications,” Appl. Sci., vol. 11, no. 18, 2021, doi: 10.3390/app11188364.

[7] B. M. Simmons, “System identification for propellers at high incidence angles,” J. Aircr., vol. 58, no. 6, pp. 1336–1350, 2021, doi: 10.2514/1.C036329.

[8] Y. Gao, S. Zhen, L. Zhang, and Z. Liu, “A Novel Nonlinear Quadrotor Attitude Regulation with Disturbances and Preassigned Convergences,” Int. J. Aerosp. Eng., vol. 2022, 2022, doi: 10.1155/2022/3576996 (enezin le literature lay siletetekemin ketita yigebalu kidem teketelu endaykeyer)

[9] E. Kuantama and R. C. Tarca, “Quadcopter Attitude and Thrust Simulation Based on Simulink Platform QUADCOPTER ATTITUDE AND THRUST SIMULATION BASED ON SIMULINK PLATFORM,” no. April, 2016.

[10] S. Bharat, A. Ganguly, R. Chatterjee, B. Basak, D. Sheet, and A. Ganguly, “A Review on Tuning Methods for PID Controller,” Asian J. Converg. Technol., vol. V Issue I, no. I, pp. 1–4, 2019

[11] J. T. Drotar, “Transformation of covariance matrices between inertial and earth-fixed coordinates,” J. Guid. Control. Dyn., vol. 37, no. 1, pp. 317–321, 2014, doi: 10.2514/1.61319.

[12] K. Mazaheri and A. Ebrahimi, “Experimental investigation on aerodynamic performance of a flapping wing vehicle in forward flight,” J. Fluids Struct., vol. 27, no. 4, pp. 586– 595, 2011, doi: 10.1016/j.jfluidstructs.2011.04.001.

[13] “Grey-Box System Identification of a Quadrotor Unmanned Aerial Ve- hicle Qianying Li”

[14] K. S. Chia and Z. Nichols, “Ziegler-Nichols Based Proportional-Integral-Derivative Controller for a Line Tracking Robot,” vol. 9, no. 1, pp. 221–226, 2018, doi: 10.11591/ijeecs.v9.i1.pp221-226.

Appendix A – MATLAB Code for Plotting 3D Graphs

```
% Initial plot

figure;

plot3(X_OUTPUT, Y_OUTPUT, Z_OUTPUT, 'b-'); % Path

hold on;

xlabel('X axis');

ylabel('Y axis');

zlabel('Z axis');

title('3D Figure-Eight Trajectory');

grid on;

% Load and create the drone model

global Quad;

load Quad_plotting_model;  % Load model data

plot_quad_model();  % Initialize the drone body

% Check if Quad is properly initialized

if ~isfield(Quad, 'X_arm') || ~isvalid(Quad.X_arm)

    error('Quad model is not properly initialized.');

end

% Loop through each point

for k = 1:length(X_OUTPUT)

    % Check if the figure still exists

    if ~ishandle(Quad.X_arm)

        warning('Figure has been closed. Stopping animation.');

        break;

    end

    updateDrone(X_OUTPUT(k), Y_OUTPUT(k), Z_OUTPUT(k));
```

```
    drawnow;

    pause(0.01);

end

hold off;

% Function to initialize the drone model

function plot_quad_model()

    global Quad;

    % Assuming Quad structure is loaded and contains necessary fields

    Quad.X_arm = patch('xdata', Quad.X_armX, 'ydata', Quad.X_armY, 'zdata', Quad.X_armZ, 'facealpha', .9,
'facecolor', 'b');

    Quad.Y_arm =
patch('xdata',Quad.Y_armX,'ydata',Quad.Y_armY,'zdata',Quad.Y_armZ,'facealpha',.9,'facecolor','b');

    Quad.Motor1 =
patch('xdata',Quad.Motor1X,'ydata',Quad.Motor1Y,'zdata',Quad.Motor1Z,'facealpha',.3,'facecolor','g');

    Quad.Motor2 =
patch('xdata',Quad.Motor2X,'ydata',Quad.Motor2Y,'zdata',Quad.Motor2Z,'facealpha',.3,'facecolor','k');

    Quad.Motor3 =
patch('xdata',Quad.Motor3X,'ydata',Quad.Motor3Y,'zdata',Quad.Motor3Z,'facealpha',.3,'facecolor','k');

    Quad.Motor4 =
patch('xdata',Quad.Motor4X,'ydata',Quad.Motor4Y,'zdata',Quad.Motor4Z,'facealpha',.3,'facecolor','k');

end

function updateDrone(x, y, z)

    global Quad;

    % Update the position of each component of the drone

    % X Arm

    set(Quad.X_arm, 'xdata', x + Quad.X_armX, 'ydata', y + Quad.X_armY, 'zdata', z + Quad.X_armZ);

    set(Quad.Y_arm, 'xdata', x + Quad.Y_armX, 'ydata', y + Quad.Y_armY, 'zdata', z + Quad.Y_armZ);

    set(Quad.Motor1, 'xdata', x + Quad.Motor1X, 'ydata', y + Quad.Motor1Y, 'zdata', z + Quad.Motor1Z);
```

```
    set(Quad.Motor2, 'xdata', x + Quad.Motor2X, 'ydata', y + Quad.Motor2Y, 'zdata', z + Quad.Motor2Z);

    set(Quad.Motor3, 'xdata', x + Quad.Motor3X, 'ydata', y + Quad.Motor3Y, 'zdata', z + Quad.Motor3Z);

    set(Quad.Motor4, 'xdata', x + Quad.Motor4X, 'ydata', y + Quad.Motor4Y, 'zdata', z + Quad.Motor4Z);

end
```

Appendix B – MATLAB Code for Plotting 3D Trajectories

```
% Assuming x_total, y_total, z_total are 1x1 double timeseries objects

% Extract data from timeseries objects

X_OUTPUT2 = x_total.Data;

Y_OUTPUT2 = y_total.Data;

Z_OUTPUT2 = z_total.Data;

% Assuming X_OUTPUT, Y_OUTPUT, Z_OUTPUT are already defined and are numeric arrays

% Create the initial plot

figure;

plot3(X_OUTPUT, Y_OUTPUT, Z_OUTPUT,"LineWidth",2); % The first path in blue

hold on; % Keep the plot

plot3(X_OUTPUT2, Y_OUTPUT2, Z_OUTPUT2, "LineWidth",2); % The second path in green

dronePlot1 = plot3(X_OUTPUT(1), Y_OUTPUT(1), Z_OUTPUT(1), 'ro', 'MarkerSize', 10); % Initial position of first drone

dronePlot2 = plot3(X_OUTPUT2(1), Y_OUTPUT2(1), Z_OUTPUT2(1), 'mo', 'MarkerSize', 10); % Initial position of second drone

xlabel('X axis');

ylabel('Y axis');

zlabel('Z axis');

title('3D Dual Trajectories');

grid on;

hold off;
```