

# **Modelling and Control of Quadrotor System**

A REPORT

on

**Project & Thesis Part-II (EE861)**

By

**Somik Dhar: 510617005**

UNDER THE GUIDANCE OF

**Dr. Ashoke Sutradhar**



DEPARTMENT OF ELECTRICAL ENGINEERING  
INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY,  
SHIBPUR

**May 2021**

## DECLARATION BY THE CANDIDATE

I/We certify that to the best of my knowledge

- i) The work contained in the thesis has been done by myself/ourselves under the general supervision of my/our supervisor/supervisors.
- ii) I/We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- iii) Whenever I/we have used materials (data, theoretical analysis and text) from other sources, I/we have given due credit to them by citing them in the text of the report and giving their details in the references.
- iv) Whenever I/we have quoted written materials from other sources, I/we have put them under quotation marks and given due credit to the sources by citing them and giving details in the references.

Sl. No.	Name	Examination Roll No.	Signature
1	Abanti Howlader	510617001	Abanti Howlader
2	Somik Dhar	510617005	S Dhar
3	Shivam Singh	510617056	Shivam
4	Yashu Ranjan	510616029	yashu Ranjan

Date: 15.05.2021

Place: IEST, Shibpur

## CERTIFICATE

I, do hereby, forward the report on Project & Thesis Part-II entitled **Modelling and Control of Quadrotor System** prepared by **Abanti Howlader, Somik Dhar, Shivam Singh** and **Yashu Ranjan** under my guidance, as partial fulfillment of requirements for the completion of their Degree of B. Tech. (Electrical) at IEST, Shibpur.



17/05/21

---

Dr. Ashoke Sutradhar  
Professor  
Department of Electrical Engineering  
IEST, Shibpur



---

Prasad Syam  
Professor and Head  
Department of Electrical Engineering  
IEST, Shibpur

## **Acknowledgement**

We would like to express our deepest gratitude to our guide, Dr. Ashoke Sutradhar, for his valuable guidance, consistent encouragement, personal caring, timely help and providing us with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule and with limited resources in our hands and in this time of pandemic, he has extended cheerful and cordial support to work and provide insight on this topic in spite of his busy schedule.

Abanti Howlader  
Somik Dhar  
Shivam Singh  
Yashu Ranjan

## Abstract

This is the final project report about modelling and control of quadrotor. It introduces a mathematical model for simulation and control of such systems. Based on the mathematical model, Proportional, Integral, Derivative (PID) control is used to design and simulate controllers along this work. The dynamic model is evolved from a simple set of equations, valid only for hovering, to a more complex mathematical model with more realistic aerodynamic coefficients and sensor and actuator models.

**Keyword:** Quadrotor, dynamic modelling, PID control

# Table of Contents

<b>DECLARATION BY THE CANDIDATE</b> .....	1
<b>CERTIFICATE</b> .....	2
<b>Acknowledgement</b> .....	3
<b>Abstract</b> .....	4
<b>List of Figures</b> .....	7
<b>List of Symbols</b> .....	8
<b>1. Introduction</b> .....	9
1.1 Motivations and Objectives.....	9
<b>2. Literature Survey</b> .....	10
2.1 System Modelling.....	10
2.2 System Control.....	10
2.3 Hardware Design.....	10
<b>3. Overview of the System</b> .....	11
3.1 General Description.....	11
3.2 Various Components.....	11
3.2.1 Mechanical Parts.....	11
3.2.2 Quadrotor Electronic Components.....	12
<b>4. System Modelling</b> .....	14
4.1 Some Assumptions for Modelling a Quadrotor.....	14
4.2 Coordinate System.....	15
4.2.1 Rotation Matrix.....	15
4.3 Modelling with Newton-Euler Formulation.....	17
4.3.1 Rotational Dynamics.....	17
4.3.2 Translational Dynamics.....	19
<b>5. Simulink Model of Quadrotor System</b> .....	21
5.1 The Plant Model.....	21
5.1.1 Orientation and Motors.....	22
5.1.2 Rotational Dynamics.....	23
5.2 Linear Dynamics.....	25
5.3 Practical System Limits.....	28

<b>6. Controller Design for Quadrotor System</b> .....	30
6.1 General Control System Strategy.....	30
6.1.1 PID Control.....	30
6.2 Controller Tuning.....	33
6.2.1 PID Tuning for Altitude Control.....	33
6.2.2 PID Tuning for Roll Angle ( $\phi$ ) and Pitch Angle ( $\theta$ ) Control.....	34
6.2.3 PID Tuning for Yaw ( $\psi$ ) Control.....	35
6.3 Trajectory Control.....	37
<b>7. Conclusion and Future Scope</b> .....	40
7.1 Concluding Remarks.....	40
7.2 Future Scope.....	41
<b>8. References</b> .....	42

# List of Figures

Fig. 1: Types of Frame .....	11
Fig. 2: Propellers .....	12
Fig. 3: Brushless DC Motor .....	12
Fig. 4: Electronic Speed Controller.....	13
Fig. 5 : Quadcopter structure and coordinate systems .....	15
Fig. 6: Plant Model Subsystem .....	21
Fig. 7: Plant Model Subsystem opened .....	21
Fig. 8 : Orientation and Motors Subsystem .....	22
Fig. 9: Orientation and Motors Subsystem Inside.....	22
Fig. 10: Angular Acceleration subsystem .....	23
Fig. 11 : Angular Acceleration subsystem Inside .....	23
Fig. 12: Roll Angle subsystem.....	24
Fig. 13: Pitch Angle subsystem .....	24
Fig. 14: Yaw Angle subsystem.....	25
Fig. 15: Linear acceleration subsystem .....	25
Fig. 16: Linear Acceleration Subsystem Inside View .....	26
Fig. 17: $x$ - acceleration subsystem .....	26
Fig. 18: $y$ -acceleration subsystem .....	27
Fig. 19: $z$ -acceleration subsystem .....	27
Fig. 20: Applying internal limits .....	28
Fig. 21: Applying angular velocity limits .....	29
Fig. 22: Limiting Euler angle positions .....	29
Fig. 23: PID Controller .....	31
Fig. 24: Input Commands and controller subsystem.....	32
Fig. 25: Controller Subsystem.....	32
Fig. 26: Tuned Altitude PID Controller Response .....	33
Fig. 27: Tuned Altitude Real World Response.....	34
Fig. 28: Roll and Pitch PID Controllers Tuned Response .....	35
Fig. 29: Tuned Yaw Controller Response .....	36
Fig. 30: Signal Builder Blocks.....	37
Fig. 31: Signal Commands and Signal Responses .....	38
Fig. 32: Quadcopter 3D Trajectory .....	39

## List of Symbols

<b>Symbol</b>	<b>Description</b>	<b>Unit</b>
$\theta$	Pitch Euler Angle	<i>rad</i>
$\dot{\theta}$	Pitch Euler Angular Velocity	<i>rad/s</i>
$\ddot{\theta}$	Pitch Euler Angular Accel.	<i>rad /s<sup>2</sup></i>
$\phi$	Roll Euler Angle	<i>rad</i>
$\dot{\phi}$	Roll Euler Angular Velocity	<i>rad/s</i>
$\ddot{\phi}$	Roll Euler Angular Accel.	<i>rad /s<sup>2</sup></i>
$\psi$	Yaw Euler Angle	<i>rad</i>
$\dot{\psi}$	Yaw Euler Angular Velocity	<i>rad/s</i>
$\ddot{\psi}$	Yaw Euler Angular Accel.	<i>rad /s<sup>2</sup></i>
$X$	Position in X	<i>m</i>
$\dot{X}$	Velocity in X	<i>m/s</i>
$\ddot{X}$	Accel. in X	<i>m /s<sup>2</sup></i>
$Y$	Position in Y	<i>m</i>
$\dot{Y}$	Velocity in Y	<i>m/s</i>
$\ddot{Y}$	Accel. in Y	<i>m /s<sup>2</sup></i>
$Z$	Position in Z	<i>m</i>
$\dot{Z}$	Velocity in Z	<i>m/s</i>
$\ddot{Z}$	Accel. in Z	<i>m /s<sup>2</sup></i>
$\Omega_1$	Motor 1 Angular Velocity	<i>rad/s</i>
$\Omega_2$	Motor 2 Angular Velocity	<i>rad/s</i>
$\Omega_3$	Motor 3 Angular Velocity	<i>rad/s</i>
$\Omega_4$	Motor 4 Angular Velocity	<i>rad/s</i>
$I_{xx}, I_{yy}, I_{zz}$	Inertia moments	<i>Kg m<sup>2</sup></i>
$J_r$	Rotor inertia	<i>Kg m<sup>2</sup></i>
$l$	Rotor axis to quadcopter centre distance	<i>m</i>
$b$	Thrust coefficient	<i>N/s<sup>2</sup></i>
$d$	Drag coefficient	<i>Nm/s<sup>2</sup></i>
$m$	Mass	<i>kg</i>
$R_b^G$	Body to global coordinate frame	

# 1. Introduction

## 1.1 Motivations and Objectives

Flying objects have always exerted a great fascination on man encouraging all kinds of research and development. Today the robotics community across the globe have been showing a great progress in MAV (Micro Aerial Vehicle) and UAV (Unmanned Aerial Vehicle) development. A quadcopter has four rotors in square formation at the equal distance from the centre of mass of the vehicle. Speed of the rotors is manipulated to perform different manoeuvres, hovering, take-off and landing.

Due to growing interest in this field in the early 2000s, it is now a well-researched topic. The models that we shall be developing or likely to design has already been implemented. So, we intend this project for only for educational purposes and for our own understanding of quadcopters. We believe that we will gain a deep insight about aero-dynamic modelling and control systems and what are the practical limitations in modelling such systems.

However, to do this, a plant model of the quadrotor system, which is a system of equations that represent the dynamics of the quadcopter, is needed to simulate flight to prove the control system works prior to installing on the quadcopter. Creating a plant model and simulating it provides a prediction of what the real-world quadcopter will do. Quadcopter plant models can increase in accuracy depending on how involved the modeling process is.

## 2. Literature Survey

The quadrotor project required extensive research into similar systems. By reviewing others work, we used this insight to model and try to develop our system. To this end, research papers from various quadrotor groups were used as guides in the early development of the dynamics and control theory.

### 2.1 System Modelling

The modelling that we have studied here is on the basis of dynamic equations given in [Bouabdallah \[1\]](#), [Alia Zakriti \[2\]](#), [Ismail \*et. al.\* \[3\]](#) and [Usman \[4\]](#). A complete model was developed with a complete set of equations, describing the vehicle dynamics not only in hover but also in motion. This was achieved through an introduction of several effects like friction force and propeller rolling moment. The implementation of a simulator which includes aerodynamic coefficients. The model was implemented in MATLAB<sup>®</sup>/Simulink [\[5\]](#) and used to optimize the design and to tune the control parameters as discussed by [Ferry \[6\]](#).

### 2.2 System Control

Various PID control designs for quadrotor systems had been suggested by [Usman \[4\]](#), [Ferry \[6\]](#) and [Selby \[7\]](#). The control action on actual nonlinear model and also in linearized model was discussed.

### 2.3 Hardware Design

Basic design requirements for the quadrotors have been discussed by [Javir \*et. al.\* \[8\]](#), where all the basic components required to build a quadrotor system had been discussed in detail. Many different types of hardware implementations have been found in the literatures. Out of which the hardware models as discussed by [Zimmerman \[9\]](#) and a low-cost implementation given by [Adnan \*et. al.\* \[10\]](#) are worth mentioning.

## 3. Overview of the System

### 3.1 General Description

A quadrotor, also called a quadcopter, is basically a helicopter with four independently controlled rotors mounted on a rigid frame. Quadcopters have gained tremendous popularity in the field of Unmanned Autonomous Vehicle (UAV) research. Quadcopters depend on electronic sensors and control systems to stabilize their flight. The main advantage that a quadcopter has over a helicopter is that since it requires four small rotors, each rotor possesses less kinetic energy and hence will cause less damage if it hits something. Each motor must lift only a quarter of the weight so we can use cheaper motors. Also, quadcopters are easier to build and control and due to their small size, can be maneuvered easily in closed spaces too. Mechanical simplicity of quadrotors compared with other flying vehicles makes it quite attractive.

### 3.2 Various Components

#### 3.2.1 Mechanical Parts

1. **Frame:** A frame is a structure that holds all the components together. It should be rigid, and be able to minimize the vibrations induced by the motors.



Fig. 1: Types of Frame

2. **Propellers:** Propeller is a type of fan that converts rotational motion into thrust. In order to counter motor torque, Quadcopter require two clockwise and two anticlockwise rotating propellers. All the propellers used in quadcopter should have same diameter and pitch. Many motors come with propeller specifications so as to have optimum power consumption. If propeller specifications are not mentioned on motor then we have to use trial and error method.



Fig. 2: Propellers

### 3.2.2 Quadrotor Electronic Components

As has been discussed by (Prof.A.V.Javir, 2015), the quadrotor components are as follows:

- 1) **Motors:** Brushless DC motors are most commonly used in Quadrotors due to their several advantages over their brushed counterpart in terms of having more torque per unit weight, reduced noise, increased reliability, longer life time and increased efficiency.

- Motor Calculations:

The motor should be selected such that it satisfies the thrust/weight relationship

$$\text{Ratio} = \text{thrust/weight} = a / g$$

The vertical take-off and landing are only possible if the ratio is greater than 1.

For simplicity purpose as the quadrotor is symmetrical,

$$\text{Thrust produced by the four motors} = 2 * (\text{weight of the quadrotor})$$

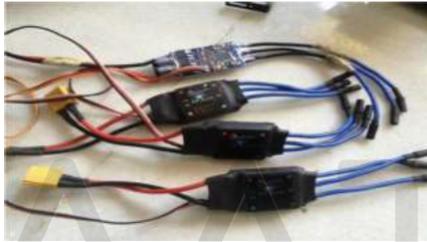
$$\text{Thrust produced by each motor (assuming each motor to be identical)} = \frac{\text{Total Thrust}}{4}$$



Fig. 3: Brushless DC Motor

- 2) **Electronic Speed Controllers:** Low voltage and current is provided by the microcontroller which is insufficient to drive motors. To drive the motors at specific speed, we need the motor driver to supply the right amount of voltage and current to the motors, which is done by electronic speed controllers.

- **Usually**  $ESC \text{ rating} = [1.2 \text{ to } 1.5] * (\text{max. ampere rating of motor})$



**Fig. 4: Electronic Speed Controller**

- 3) **Battery:** Usually Lithium Polymer (Li-Po) batteries are used in quadcopter, as these batteries are rechargeable and also have low weight and high voltage capacity as compared to other type of batteries. Battery chargers are required for charging the battery.
- 4) **Flight Controller:** To maintain balance, the quadcopter should continuously take measurements from the sensors and make adjustment accordingly to the speed of the rotors to keep the body level.
- 5) **Transmitter and receiver:** The Transmitter (Tx) and Receiver (Rx) system allows the Quadcopter to be remotely controlled through a wireless signal. The aircraft controls would typically include throttle, pitch, roll, yaw, and mode settings. 2.4GHz TX and RX system is used for its better performance, because it will not experience signal conflicts from other radio frequency (RF) controllers. Receiver used is having 6C 2.4Ghz system which perfectly bonded with the 2.4Ghz transmitter.
- 6) **IMU (Inertial Measurement Unit):** The IMU is an electronic sensor device which measures the velocity, orientation and acceleration along different directions. This sensor allows the control system to navigate the bot in the environment. The readings of the IMU are fed to the main controller which are then compared with the set points and then appropriate action is taken by the motor controller system. The IMU is a combination of a 3-axis gyroscope and a 3-axis accelerometer, which together makes it a 6 degree of freedom sensor. Sometimes a 3-axis magnetometer is also included to get an absolute yaw control relative to the Earth's magnetic field. This makes the IMU a 9 degree of freedom sensor.

## 4. System Modelling

The concept followed in this report for dynamic modelling was to write physical equations, assuming some predefined parameters and identify the dynamics of the quadrotor. This approach makes it easy to build dynamic models of unstable systems since we don't have to perform closed loop identification in flight. Newton-Euler formalism, model identification and momentum theories were used for modelling the physical system.

### 4.1 Some Assumptions for Modelling a Quadrotor

- The quadrotor structure is supposedly rigid.
- The centre of gravity and the body fixed frame origin are assumed to coincide.
- The structure is symmetrical about the centre of gravity.
- The propellers of the quadrotor systems are rigid.
- Thrust and drag force exerted on the quadrotor system are approximately proportional to the square of the propeller's speed.

Since quadrotor is assumed to be a rigid body,  $x$ - $y$ - $z$  Euler angle notation is used to construct the rotation matrix. The model of the quadrotor should contain all the important effects including the gyroscopic ones.

A short list of the main effects acting on the helicopter are described briefly in Table 4.1.

**Table 4.1**

Effect	Source	Formulation
Aerodynamic Effects	Propeller rotation Blade Flapping	$C_a \omega^2$
Inertial Counter Torques	Change in propeller rotation speeds.	$J \dot{\omega}$
Gravity Effect	Center of mass position	$Mg$
Gyroscopic Effect	Change in orientation of the rigid body	$I \theta \psi$
	Change in orientation of the propeller plane	$Jr \omega_r, \theta, \phi$
Friction	All helicopter motion	$A \dot{\phi}, \dot{\theta}, \dot{\psi}$

Once all the components of the equations are developed, it is possible to simplify the equations of motion after making several assumptions about the method which the quadrotor will be operated. The goal is to make a model that is simple while still being as realistic as possible.

## 4.2 Coordinate System

The basic quadrotor structure used for the model development is shown in the figure 5 below depicting the Euler angles of roll, pitch, and yaw, a body coordinate frame  $b$ , and the global coordinate frame  $G$ .

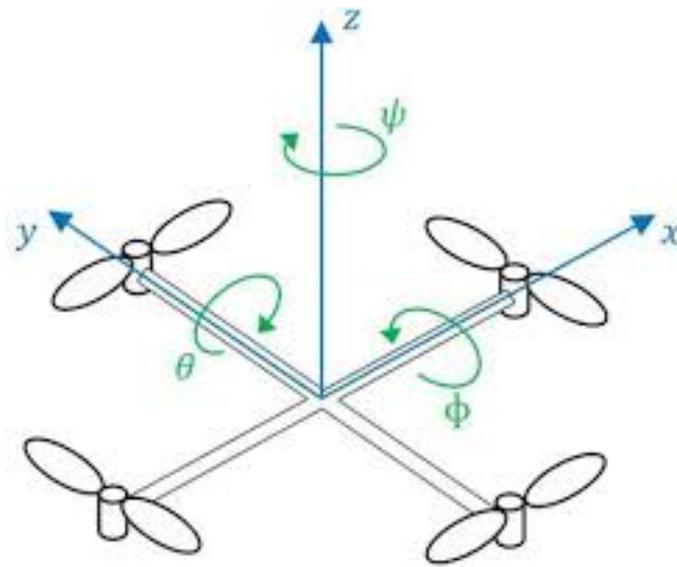


Fig. 5 : Quadcopter structure and coordinate systems

### 4.2.1 Rotation Matrix

In aerospace engineering, the axes are directed as for a craft moving in the positive  $x$ -axis with the right side corresponding to the positive  $y$ - direction and the vertical underside corresponding to the positive  $z$ -direction. These three angles are individually called roll, pitch and yaw.

Considering a right-hand oriented coordinate system, the three single rotations are described separately by:

- $R(x, \Phi)$ , rotation around  $x$ -axis
- $R(y, \theta)$ , rotation around  $y$ -axis
- $R(z, \psi)$ , rotation around  $z$ -axis

They are represented by as follows:

$$\begin{aligned}
 R(x, \Phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & \sin \Phi \\ 0 & -\sin \Phi & \cos \Phi \end{bmatrix} \\
 R(y, \theta) &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \\
 R(z, \psi) &= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

The complete rotation matrix is the product of the previous three successive rotations:

$$R(\Phi, \theta, \psi) = R(x, \Phi).R(y, \theta).R(z, \psi) \quad \text{Eq. 4-1}$$

Which results in:

$$R_G^b = \begin{bmatrix} \cos \Psi \cos \theta & \sin \Psi \cos \theta & -\sin \theta \\ \cos \Psi \sin \theta \sin \Phi - \sin \Psi \cos \Phi & \sin \Psi \sin \theta \sin \Phi + \cos \Psi \cos \Phi & \cos \theta \sin \Phi \\ \cos \Psi \sin \theta \cos \Phi + \sin \Psi \sin \Phi & \sin \Psi \sin \theta \cos \Phi - \sin \Phi \cos \Psi & \cos \theta \cos \Phi \end{bmatrix} \quad \text{Eq. 4-2}$$

Also,

$$\begin{aligned}
 \begin{bmatrix} X^G \\ Y^G \\ Z^G \end{bmatrix} &= R_b^G \begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} = (R_G^b)^T \begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} \\
 R_b^G &= \begin{bmatrix} \cos \Psi \cos \theta & \cos \Psi \sin \theta \sin \Phi - \sin \Psi \cos \Phi & \cos \Psi \sin \theta \cos \Phi + \sin \Psi \sin \Phi \\ \sin \Psi \cos \theta & \sin \Psi \sin \theta \sin \Phi + \cos \Psi \cos \Phi & \sin \Psi \sin \theta \cos \Phi - \sin \Phi \cos \Psi \\ -\sin \theta & \cos \theta \sin \Phi & \cos \theta \cos \Phi \end{bmatrix} \quad \text{Eq. 4-3}
 \end{aligned}$$

As the angular rates are defined in the body frame and the Euler angles are defined in intermediate coordinate frames, we can use the rotation matrix derived above to determine the relationship between the angular rates and the time derivatives of the Euler angles as shown below. The angular velocities are vectors pointing along each axis of rotation and are not equal to the time derivative of the Euler angles. The derivation below assumes that the time derivative of each Euler rate is small ([Bouabdallah \[1\]](#)).

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = R(\Phi)R(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R(\Phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} \quad \text{Eq. 4-4}$$

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

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \Phi \tan \theta & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \frac{\sin \Phi}{\cos \theta} & \frac{\cos \Phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \text{Eq. 4-5}$$

### 4.3 Modelling with Newton-Euler Formulation

The dynamics of a rigid body under external forces applied to the center of mass and expressed in the body fixed frame are in Newton Euler formulation ([Usman \[4\]](#)).

$$\begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \quad \text{Eq. 4-6}$$

#### 4.3.1 Rotational Dynamics

The rotational equations of motion are defined in the body frame so that the rotation can be computed about the quadrotor's center and not the center of the global coordinate frame. Since we assumed the quadrotor to be symmetrical, the quadrotor moment of inertia matrix is symmetrical.

- **Thrust equations**

**Table 4.2**

Input	Thrust Equation	Description	
$U_1$	$b (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$	General Thrust	Eq. 4-7
$U_2$	$b (\omega_4^2 - \omega_2^2)$	Roll Thrust	Eq. 4-8
$U_3$	$b (\omega_1^2 - \omega_3^2)$	Pitch Thrust	Eq. 4-9
$U_4$	$d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$	Yaw Thrust	Eq. 4-10

There are 4 main thrust equations, vertical thrust, roll thrust, pitch thrust, and yaw thrust.  $U_1$  applies a general thrust to each of the motors to increase the altitude.  $U_2$  changes thrust between motor 2 and motor 4 to roll the quadcopter.  $U_3$  changes thrust between motor 3 and motor 1 to pitch the quadcopter.  $U_4$  changes the thrust to motors 2 and 4 and equally changes the thrust to motors 1 and 3 to yaw the quadcopter.

The quadcopter inputs can also be defined in the inertial orientation as given in Table 4.3.

**Table 4.3**

<b>Input</b>	<b>Thrust Equation</b>	<b>Description</b>	
$U_1$	$b (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$	General Thrust	Eq. 4-11
$U_2$	$b \sin(\frac{\pi}{4})(\omega_4^2 - \omega_2^2)$	Roll Thrust	Eq. 4-12
$U_3$	$b \sin(\frac{\pi}{4})(\omega_1^2 - \omega_3^2)$	Pitch Thrust	Eq. 4-13
$U_4$	$d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$	Yaw Thrust	Eq. 4-14

The overall angular velocity of the all the propellers can also be defined. This will be used in angular acceleration equations. In most cases the opposite side motors are spinning in the same direction (motor 1 and 3 in the same direction, and 2 and 4 in the same direction).

$$\omega_r = \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \quad \text{Eq. 4-15}$$

Starting with the moments of a quadcopter, which include, rolling moments, pitching moments, and yawing moments, each of these moments consist of body gyro effects, propeller gyro effects, and actuator action.

- Body gyroscopic effects - changes in the quadcopter's orientation
- Propeller gyroscopic effects - propeller rotation coupled with quadcopter frame orientation
- Actuation action - forces produced by the rotors
- **Gyroscopic effects**

$$\tau_{gp} = J_r \omega_r \begin{bmatrix} \dot{\phi} \\ -\dot{\theta} \\ \mathbf{0} \end{bmatrix}$$

Where,  $\omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$

Let us consider, the quadcopter is at a perfectly still hover. This means that the pitch and yaw angular velocities are zero, so the body gyro effects and the propeller gyro effects are zero. This leaves the roll actuator action ( $U_2$ ) input, which would be zero also if the quadcopter is still, thus the roll angular acceleration is zero. Now if the user applies an

input signal to  $U_2$ . Since this input only applies thrust in the rolling axis the pitch and yaw angular velocities will be zero, thus the body gyro and propeller gyro effects again are zero. The actuator action is not zero due to the user input, thus the quadcopter will have a rolling angular acceleration equal to  $l(U_2)/I_{xx}$  and the quadcopter will begin to roll. It is worthy to note within the equations of motion, it doesn't matter if the system is using a body or ground/inertial frame of orientation, thus the subscript will not be used.

The equation of motion that governs the rotational motion of the quadrotor is given by:

$$\begin{bmatrix} \ddot{\Phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 1/I_{zz} [(I_{yy} - I_{zz})\dot{\theta}\dot{\psi} - J_r\theta\omega_r + l(U_2)] \\ 1/I_{yy} [(I_{zz} - I_{xx})\dot{\phi}\dot{\psi} - J_r\phi\omega_r + l(U_3)] \\ 1/I_{xx} [(I_{xx} - I_{yy})\dot{\phi}\dot{\theta} + (U_4)] \end{bmatrix} \quad \text{Eq. 4-16}$$

### 4.3.2 Translational Dynamics

The linear equation of motion is defined in the global reference frame

- **Newton's Second Law of Motion**

$$m\ddot{X}^G = -F_g + U_1 - F_d \quad \text{Eq. 4-17}$$

Where,  $\ddot{X}^G = \begin{bmatrix} \ddot{X}^G \\ \ddot{Y}^G \\ \ddot{Z}^G \end{bmatrix}$

- **Gravity in global coordinate frame**

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}$$

- **Drag Force and the Friction**

$$F_d = \begin{bmatrix} -A_x\dot{x} \\ -A_y\dot{y} \\ -A_z\dot{z} \end{bmatrix}$$

❖ The equation of motions that governs the translational motion of the quadrotor is thus given by:

$$\begin{bmatrix} \ddot{X}^G \\ \ddot{Y}^G \\ \ddot{Z}^G \end{bmatrix} = \begin{bmatrix} 1/m ([-\cos \Phi \cos \Psi \sin \theta + \sin \Phi \sin \Psi]U_1 - A_x \dot{X}^G) \\ 1/m ([-\cos \Phi \sin \Psi \sin \theta + \sin \Phi \cos \Psi]U_1 - A_y \dot{Y}^G) \\ 1/m ([\cos \Phi \cos \Psi]U_1 - A_z \dot{Z}^G) - g \end{bmatrix} \quad \text{Eq. 4-18}$$

# 5. Simulink Model of Quadrotor System

## 5.1 The Plant Model

Now that the basic equations of motion have been generated, the Simulink plant model can now be formed. A subsystem for the plant modelling is created with  $x$ ,  $y$ ,  $h$  or  $z$ , and  $\Psi$  as outputs and  $M1$ ,  $M2$ ,  $M3$  and  $M4$  as inputs (Fig. 6). Plant model subsystem will have several subsystems to represent different parameters depending on their equations. The purpose is to create a model consisting of all the dynamic equations of the vehicle and then build a control system for the model.

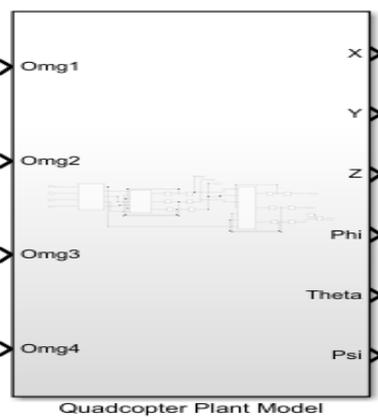


Fig. 6: Plant Model Subsystem

On opening the plant model subsystem, it looks as follows (Fig. 7)

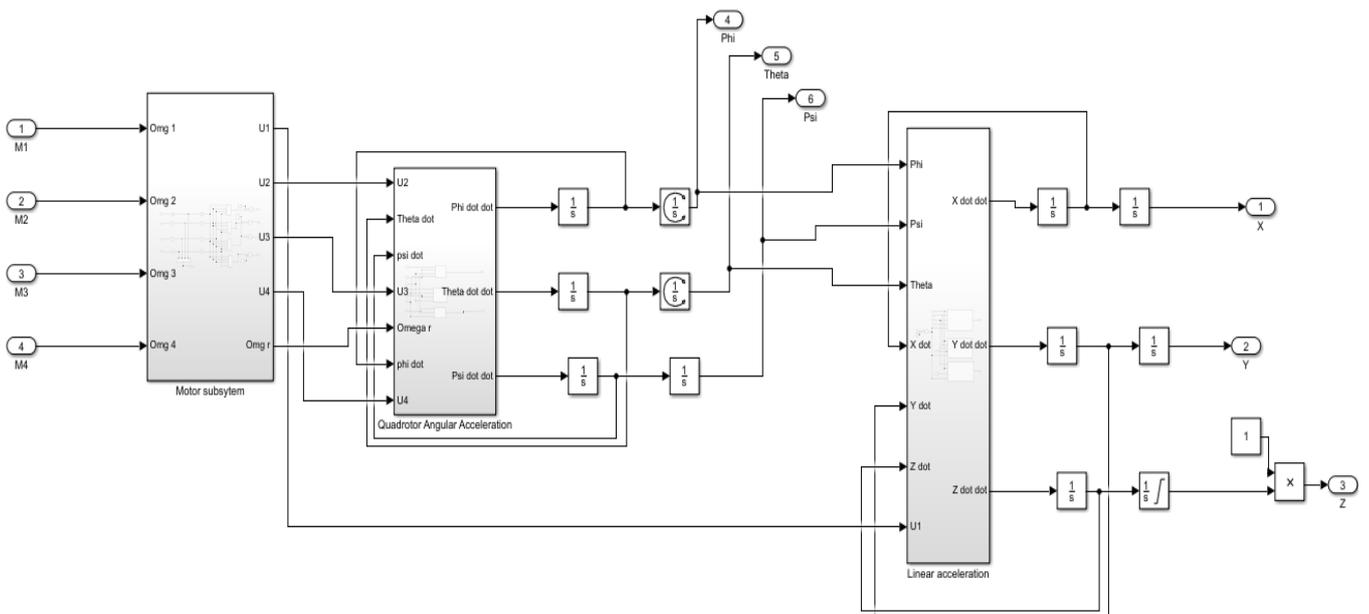
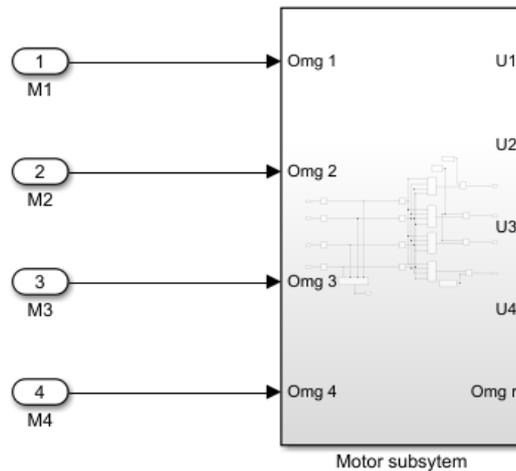


Fig. 7: Plant Model Subsystem opened

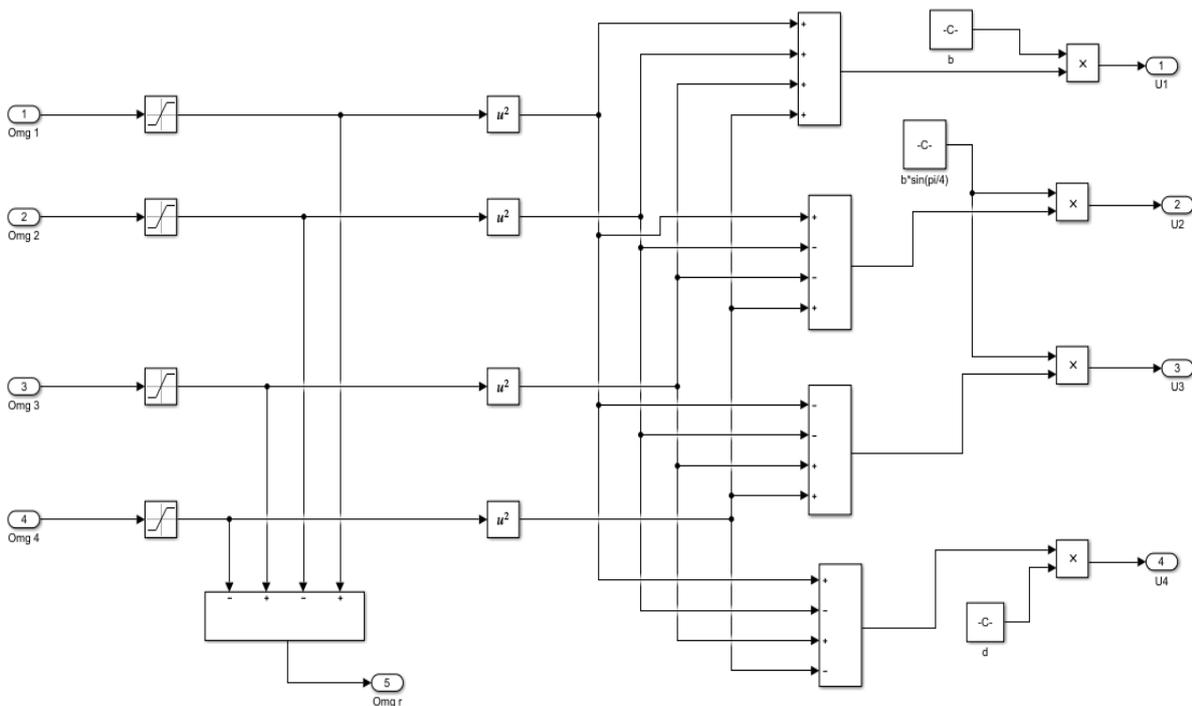
### 5.1.1 Orientation and Motors

Orientation and Motors is a subsystem (Figure 8) under Plant Model subsystem. It depends on the moment equations for the specific orientation of the vehicle, the quadcopter has two orientations body and inertial. We have derived equations for inertial orientation of our vehicle.



**Fig. 8 : Orientation and Motors Subsystem**

The Orientation and Motors subsystem looks as follows on opening (Fig. 9). This subsystem takes the motor commands and converts them to moments to control the vehicles orientation and altitude.  $Omg\_r$  is the total angular velocity of the quadcopter; the propellers rotate in the opposite direction to balance the total angular velocity.



**Fig. 9: Orientation and Motors Subsystem Inside**

### 5.1.2 Rotational Dynamics

This subsystem (Figure 10) is based on equation 4-16 in which motors thrust is converted into angular rates. It takes  $U_2, U_3, U_4$  and  $\omega$  from motors subsystem and outputs  $p, q$  and  $r$ . While  $U_1$  is fed straight to the next subsystem, as  $U_1$  is the total thrust of all motors in the  $z$  direction.

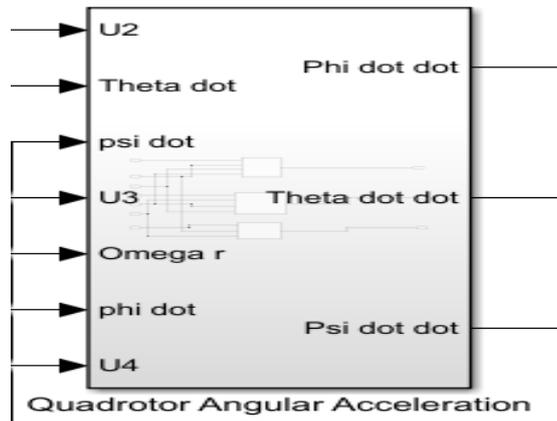


Fig. 10: Angular Acceleration subsystem

Angular acceleration subsystem has three more subsystems to control roll, pitch and yaw acceleration separately, containing the equations. Subsystems can be opened to see the equations and their arrangement.

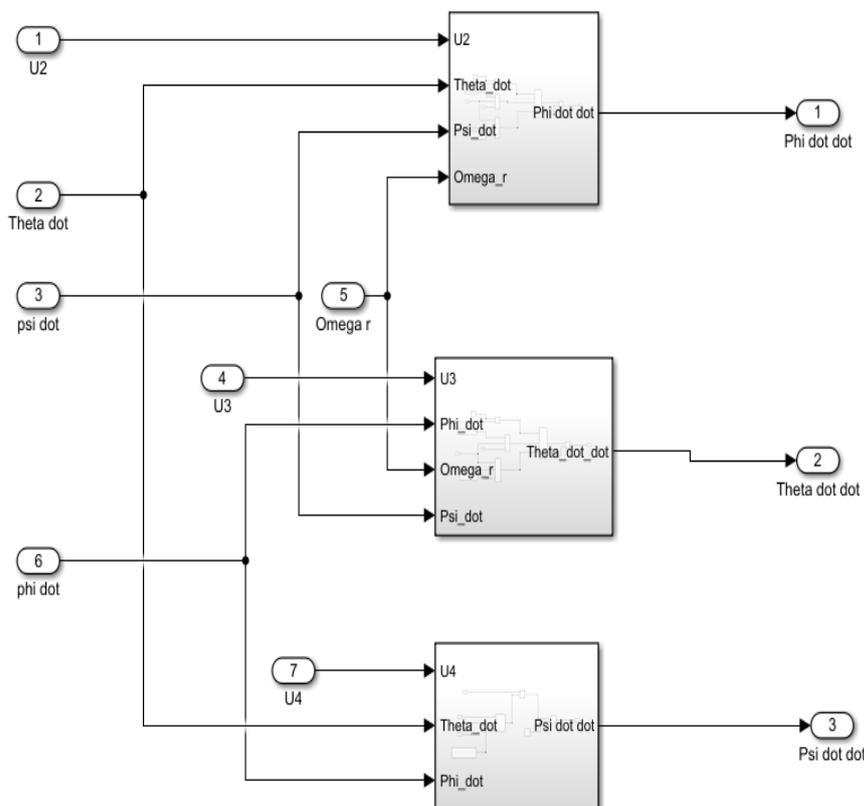


Fig. 11 : Angular Acceleration subsystem Inside

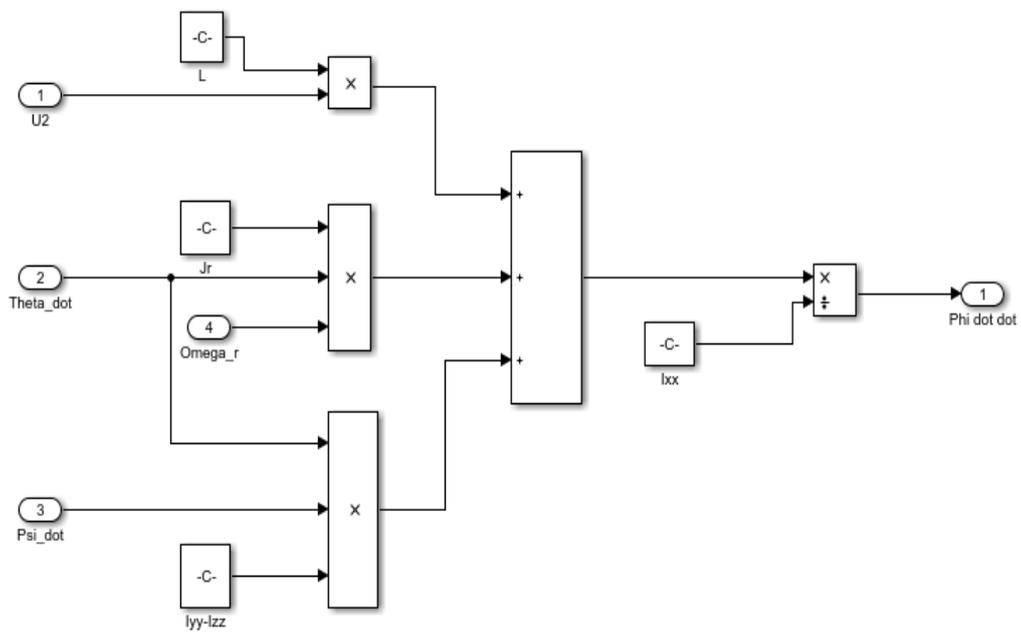


Fig. 12: Roll Angle subsystem

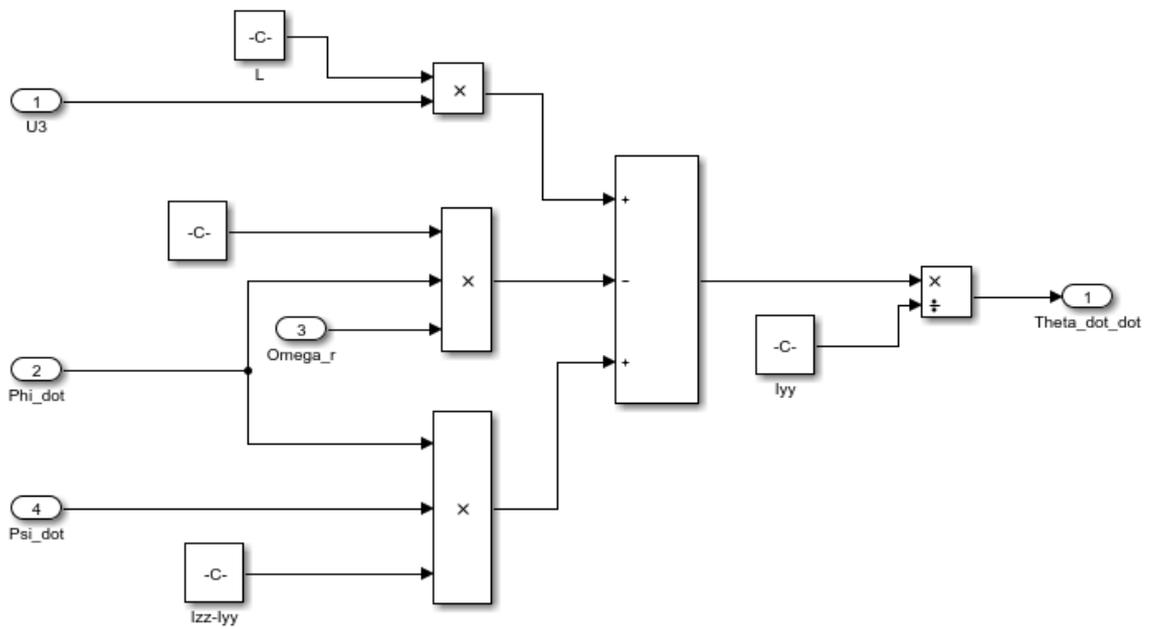


Fig. 13: Pitch Angle subsystem

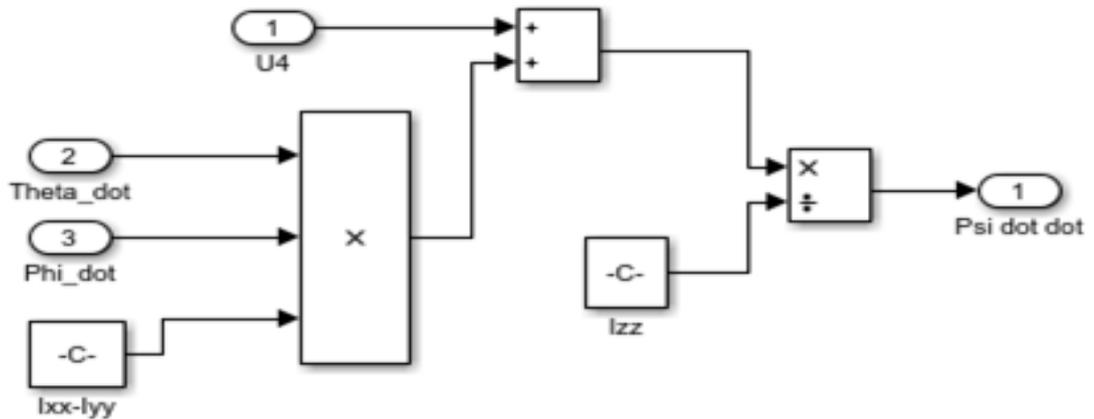


Fig. 14: Yaw Angle subsystem

## 5.2 Linear Dynamics

The angular rates from the previous subsystem are converted into angles using integrators. Now the linear acceleration subsystem (Figure 15) converts angles into the linear acceleration in the inertial frame depending on navigation coordinate equations 4-18. To navigate the vehicle in the inertial coordinate frame.

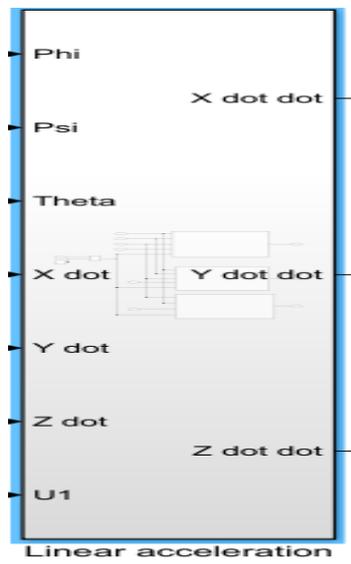


Fig. 15: Linear acceleration subsystem

Linear acceleration subsystem also has three subsystems for x, y and z acceleration separately, containing equations for each (Fig. 16, 17,18 and 19).

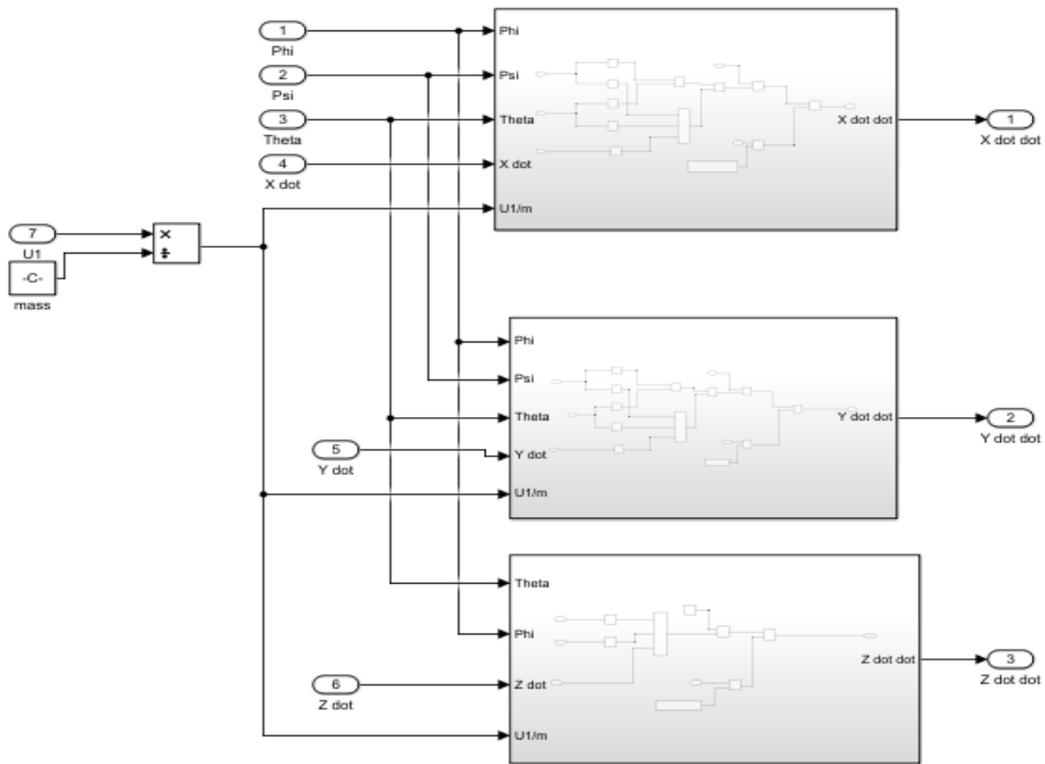


Fig. 16: Linear Acceleration Subsystem Inside View

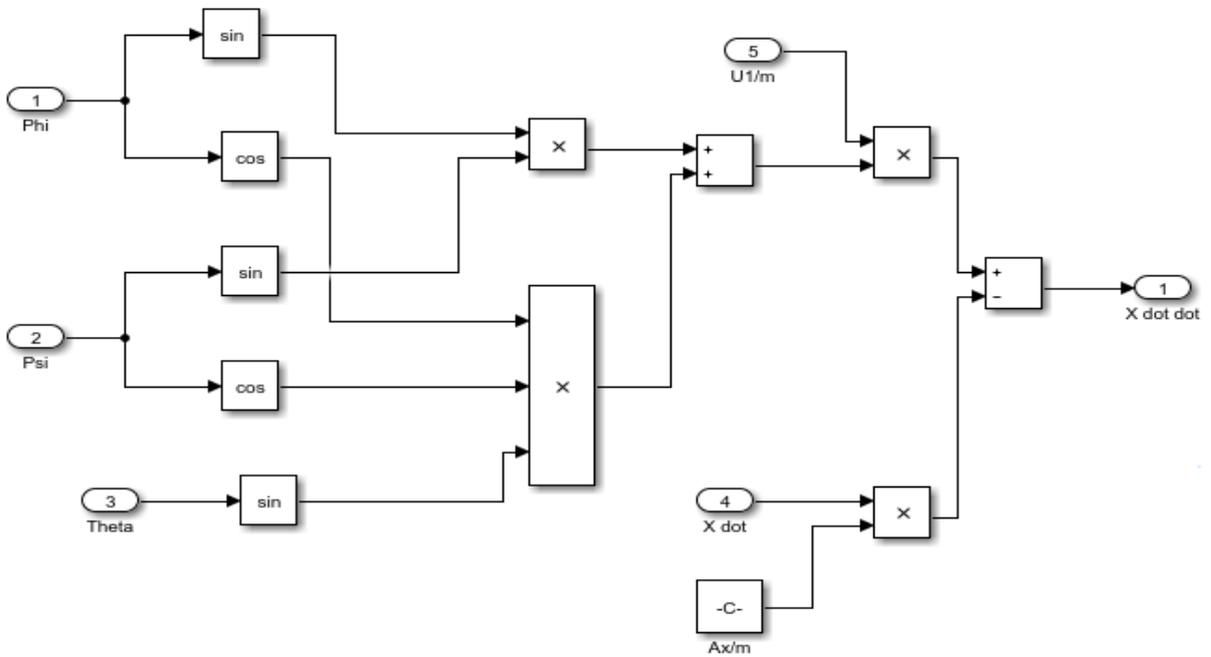


Fig. 17: x- acceleration subsystem

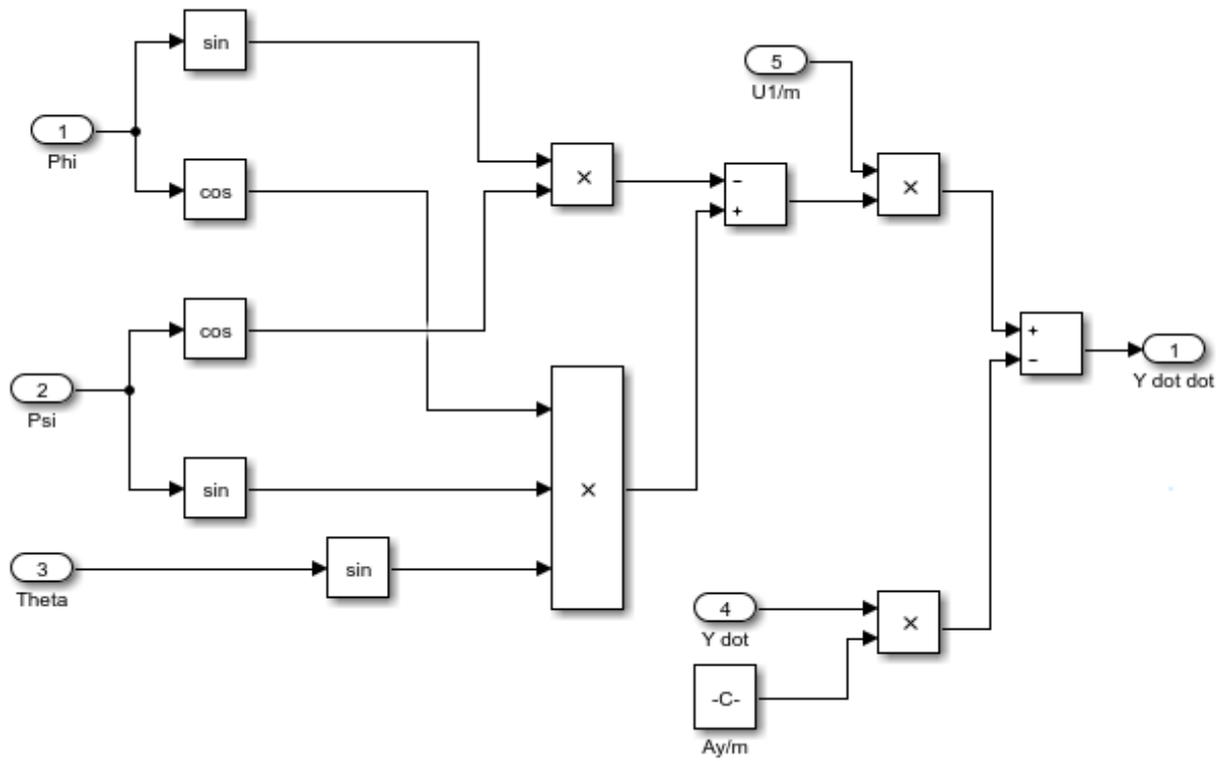


Fig. 18: y-acceleration subsystem

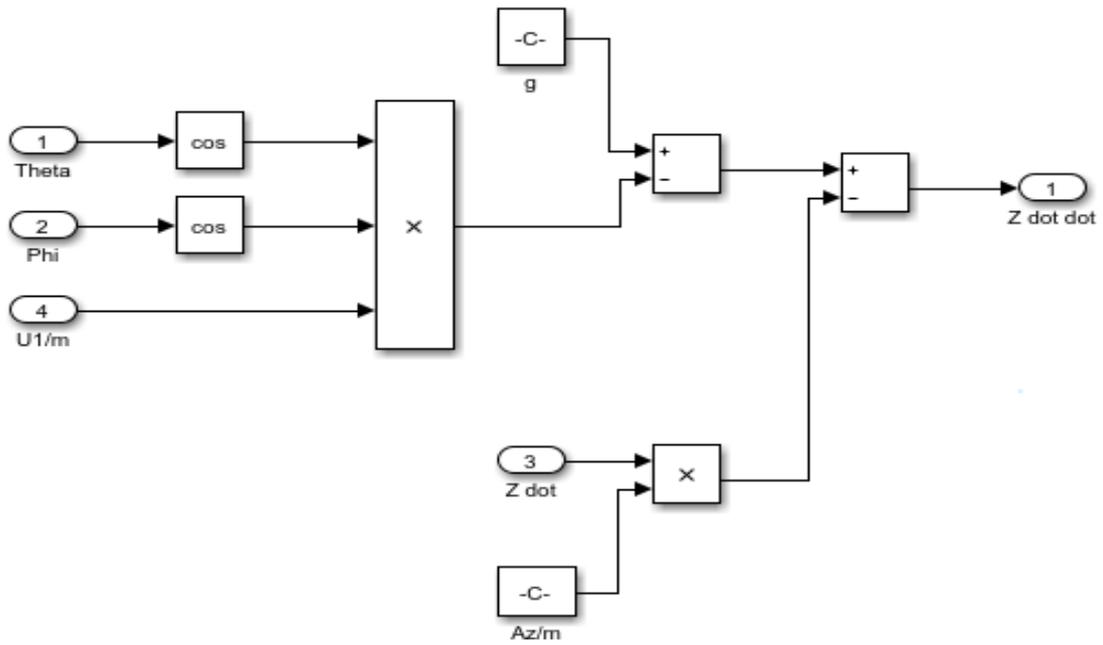


Fig. 19: z-acceleration subsystem

### 5.3 Practical System Limits

At this point a general plant model has been created by simply using the equations of motion, along with input thrust equations. An important aspect to realize about the model is that the equations of motion do not perfectly represent a quadcopter. For instance, in many quadcopters the motors only spin in one direction, or that the quadcopter cannot physically be in the positive Z coordinate (depending on frame of reference of course). Because of this, limits need to be applied to the model. First, the frame of reference needs to be considered. For this model the X and Y directions will be free, and the Z direction will be limited from zero to infinity. This is because the Z direction is up. There are multiple methods that can be used to implement these limits to the model. The saturation block can be used, or integral limits can be applied.

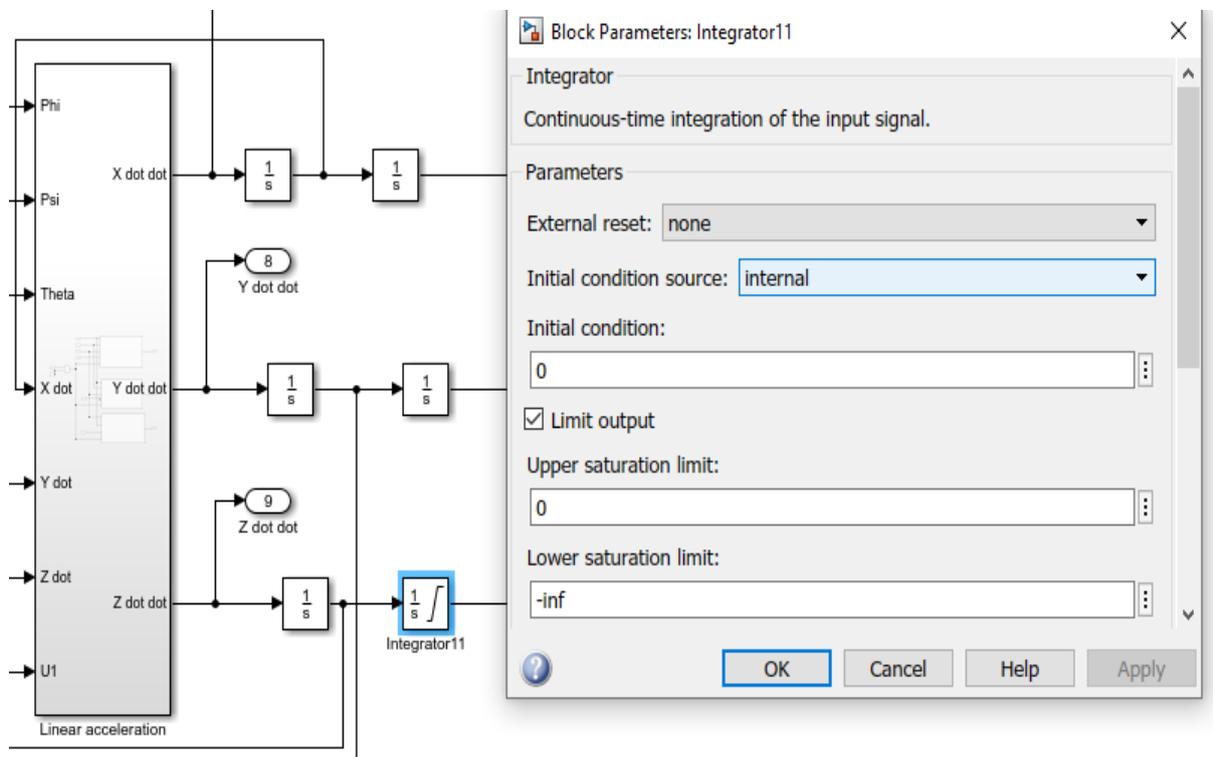


Fig. 20: Applying internal limits

Next, limits should be applied to the angular velocity of the rotors. Some quadcopters possess the ability to reverse the velocity of their rotors midflight, however, the model being created this shall only possess the ability to rotate in one direction. These limits should be applied in the subsystem block where the angular velocity was found. Here, saturation blocks will be used, limited from zero to infinity. In later sections the angular velocity will no longer have an upper limit of infinity due to motor dynamic limitations.

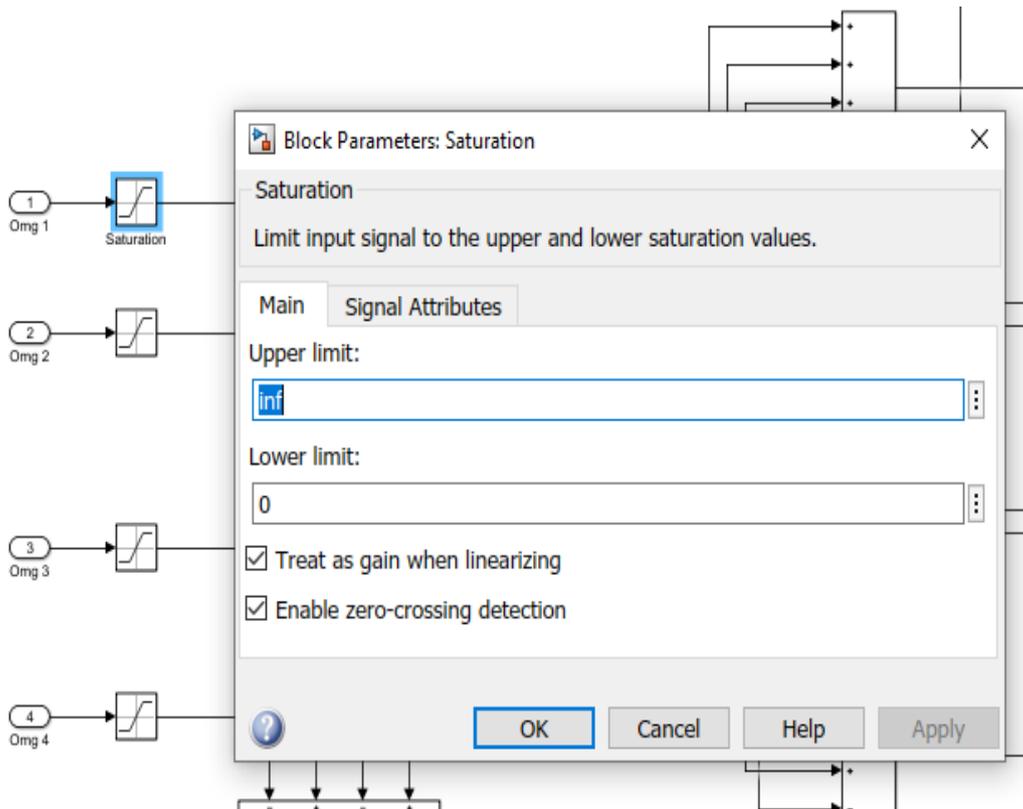


Fig. 21: Applying angular velocity limits

Lastly, the Euler angles can be wrapped. This limits the angles to be between  $-\pi$  and  $\pi$ . This limit can be placed in the model in the integrals from angular velocity to angular position.

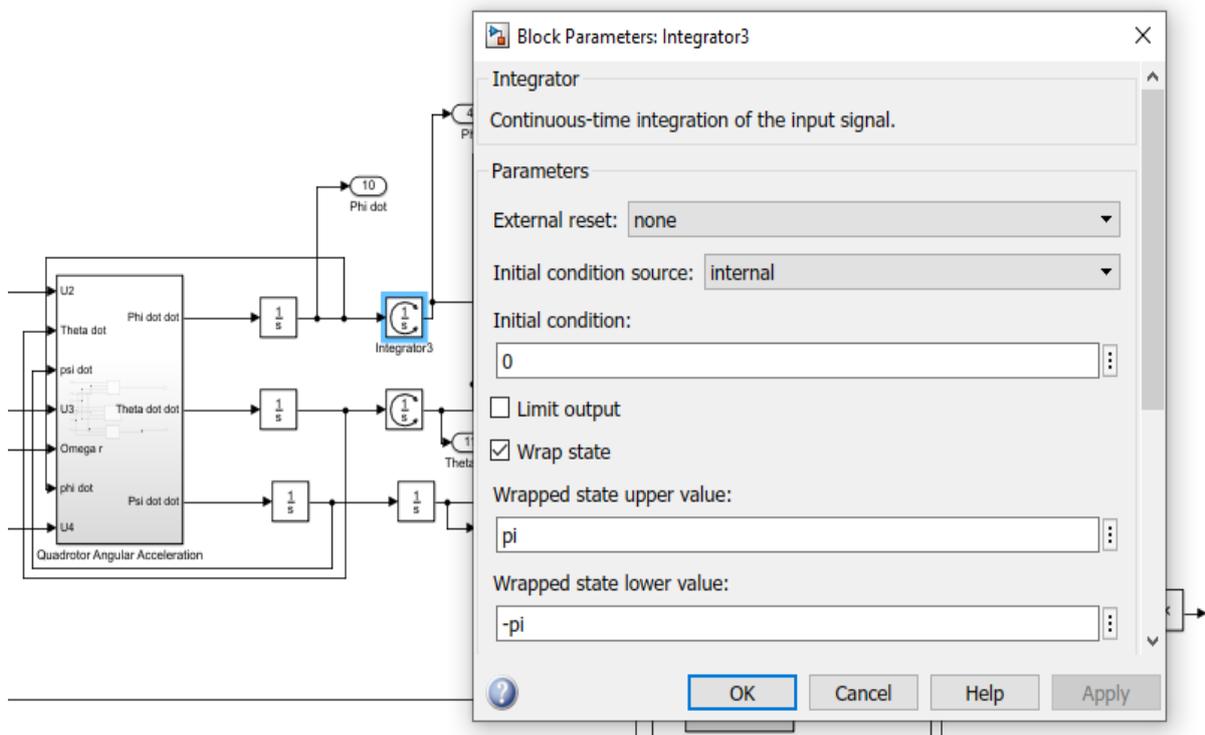


Fig. 22: Limiting Euler angle positions

## 6. Controller Design for Quadrotor System

The control system developed here are based on the quadrotor model derived as above, and implemented in a simulated environment.

### 6.1 General Control System Strategy

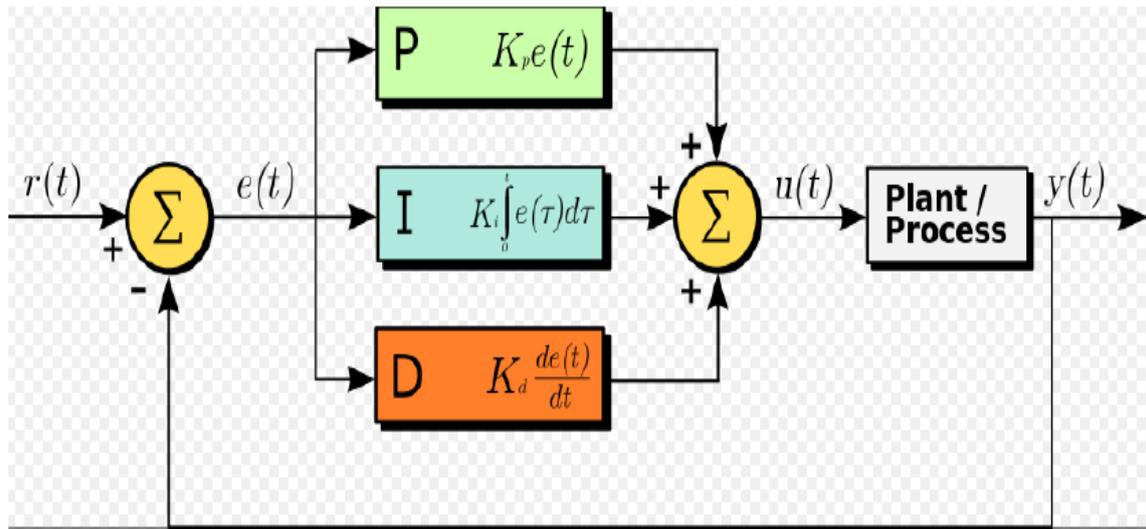
The quadrotor is a classic underactuated system. The quadrotor is able to move in 6 degrees of freedom (3 translational and 3 rotational) but only has 4 control inputs (the speeds of each motor). While the quadrotor can move directly on the vertical  $z$  axis without changing any other state, it must change its attitude to move on the horizontal  $x$  and  $y$  axes. Since it is not possible to control 6 degrees of freedom with only 4 control inputs, we have instead design controllers to stabilize around desired  $x$ ,  $y$ ,  $z$  positions and a desired heading. The quadrotor should be able to safely move to this desired position while maintaining stable roll and pitch angles.

Controller is very important for a vehicle to work properly. Without a controller the quadcopter will be unstable. There are many control strategies and algorithms but for this project, a feedback control system comprising of PID controllers is utilized.

#### 6.1.1 PID Control

A PID (Proportional Integral and Derivative) controller has three control terms (Figure 23). The input to the PID controller is an error signal (difference between actual and desired value), which is multiplied by PID gains to acquire the control of the plant or process. Proportional term compares desired and actual value and simply multiply the error value to a constant. Integral term removes steady state error by integrating the error value over time until it reaches zero. Derivative term estimates about the future error and removes it. It estimates the error by the rate of change of error with time and multiply it with derivative constant.

A good representation of the PID control scheme can be seen in Figure 23 below.



**Fig. 23: PID Controller**

$$u = P + I + D$$

$$e(t) = r(t) - y(t)$$

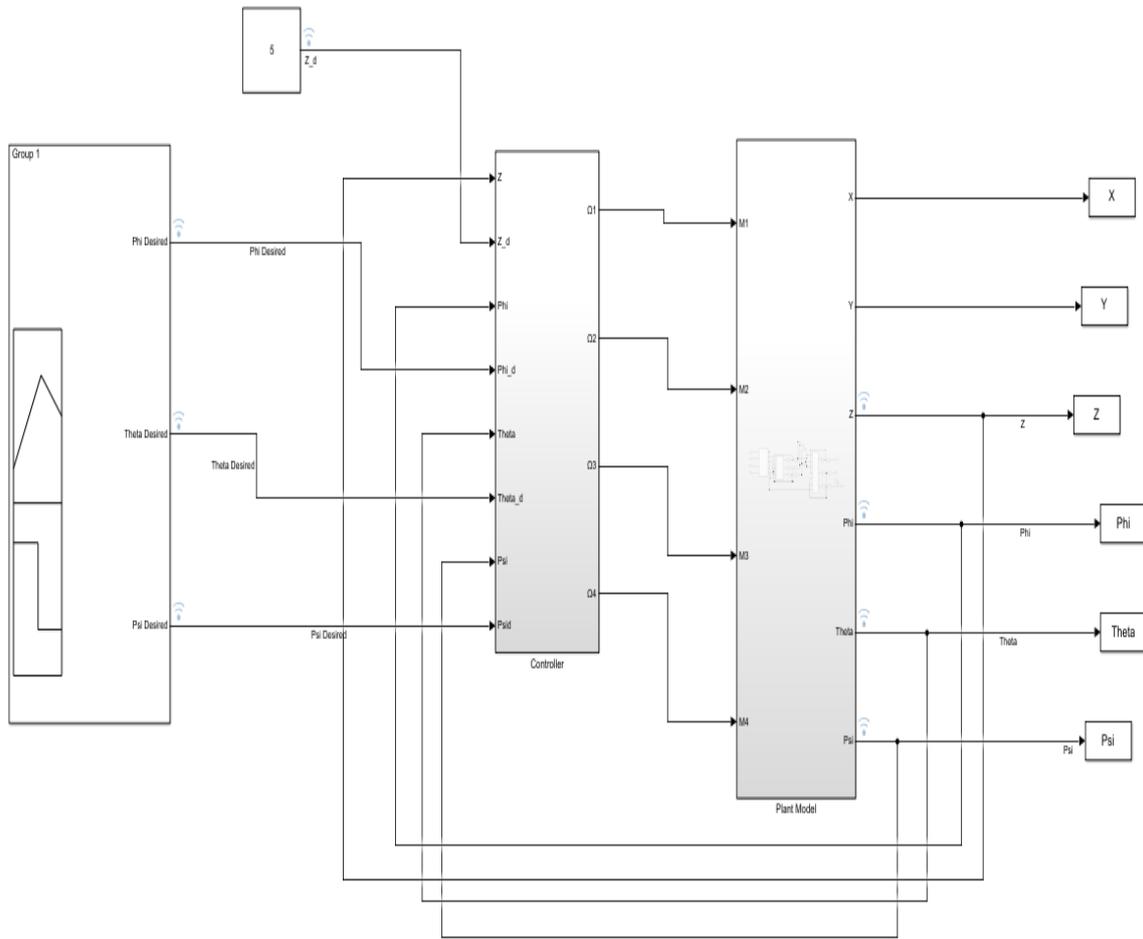
$$P = K_p e(t)$$

$$I = K_I \int_0^t e(t-1) dt$$

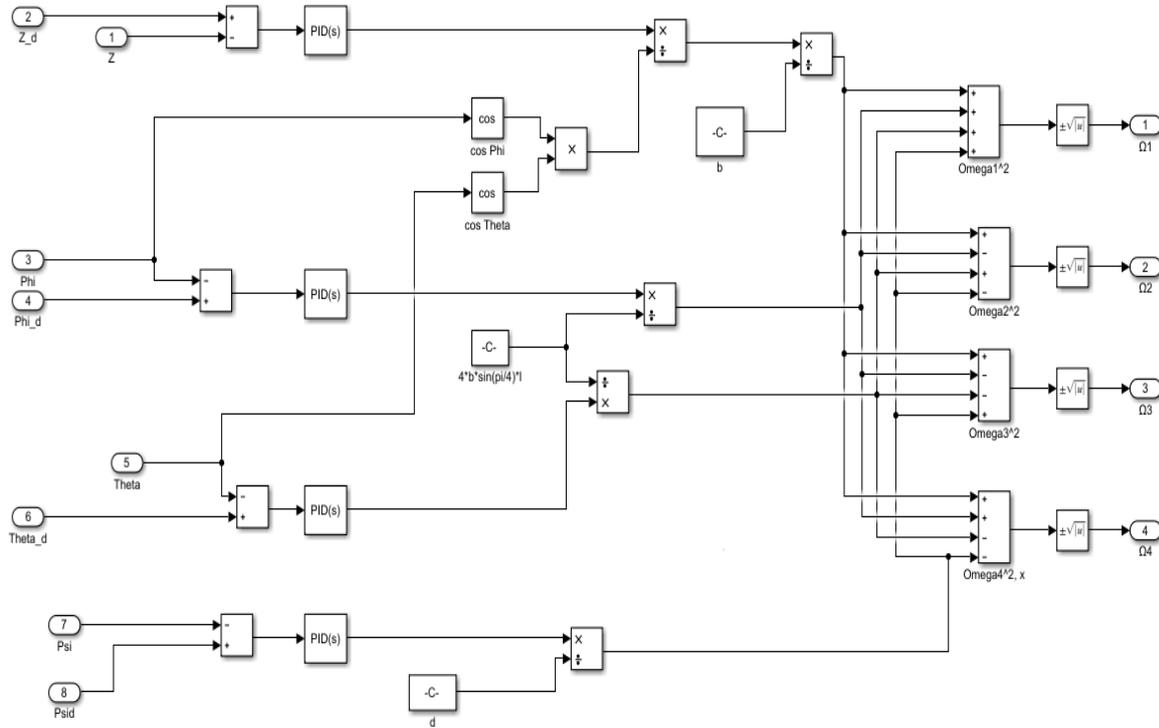
$$D = K_D \dot{e}(t)$$

$$u(t) = K_p e(t) + K_I \int_0^t e(t-1) dt + K_D \dot{e}(t) \quad \text{Eq. 6-1}$$

Now, we are familiar with the basic working principle of the PID controller and have verified the plant model obeying the real-world constraints. The purpose of the project is to control the attitude of the copter, but a controller for the altitude is also used because it is necessary to control the altitude to tune the attitude (roll, pitch and yaw) controllers. A PID controller can be built by using simple gains in Simulink and there is also a built-in PID block in Simulink. Built-in PID blocks are utilized in the controller building of the vehicle. A PID based controller is applied to control the actual values obtained from the plant model according to the desired input values. In total, we have four PIDs for altitude, roll, pitch and yaw. Then, the output of the PID controllers' splits into motor commands for x orientation of the quadcopter. The plant model with controller looks as follows (Figure 24 and 25).



**Fig. 24: Input Commands and controller subsystem**



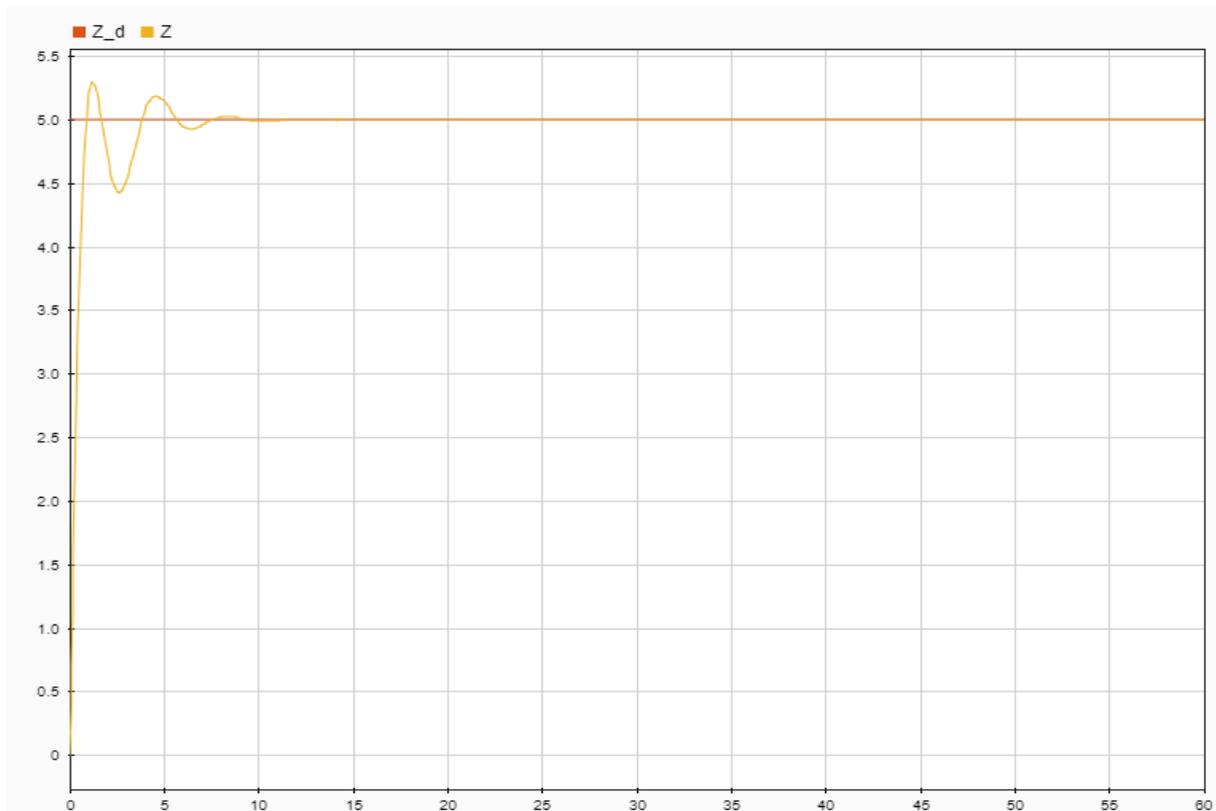
**Fig. 25: Controller Subsystem**

## 6.2 Controller Tuning

A PID controller can be tuned manually or using auto-tuning function in Simulink. Each of the PID is tuned individually for each of the desired input control.

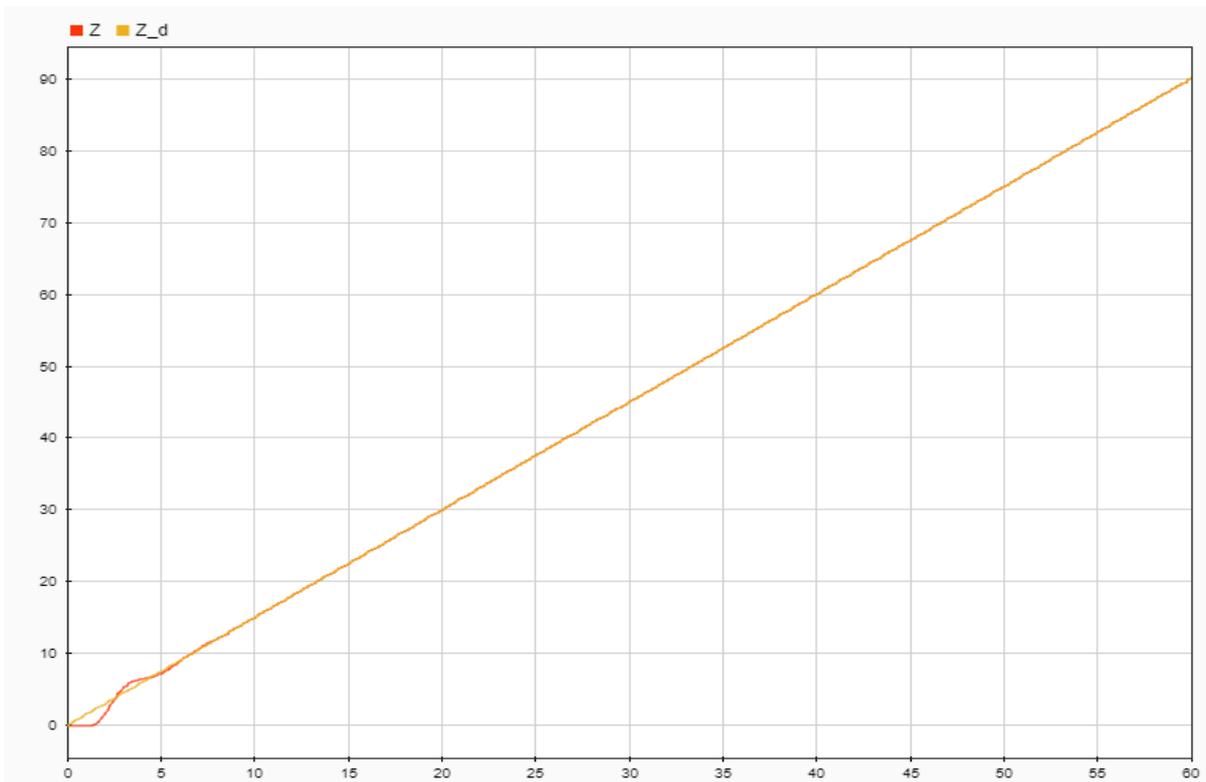
### 6.2.1 PID Tuning for Altitude Control

After doing some adjustments and fixings, the fully tuned PID gave the following results (Figure 43). The results are quite satisfying, the quadcopter is stabilizing at an altitude of 5 m in less than 10 seconds. Same goals can be achieved in even less time, but the behaviour of the vehicle will be very aggressive.



**Fig. 26: Tuned Altitude PID Controller Response**

Figure 27 gives a more realistic representation of the vehicle behavior. Previously (Figure 26), 5 m altitude command was given to the vehicle from the beginning of the simulation which is unlikely to happen, contrary to this in the real-world a pilot will take off rather slowly, to mimic this in simulation a ramp block (Figure 27) is used as an altitude command, signal starts from zero and keeps increasing with a slope of 1.5.



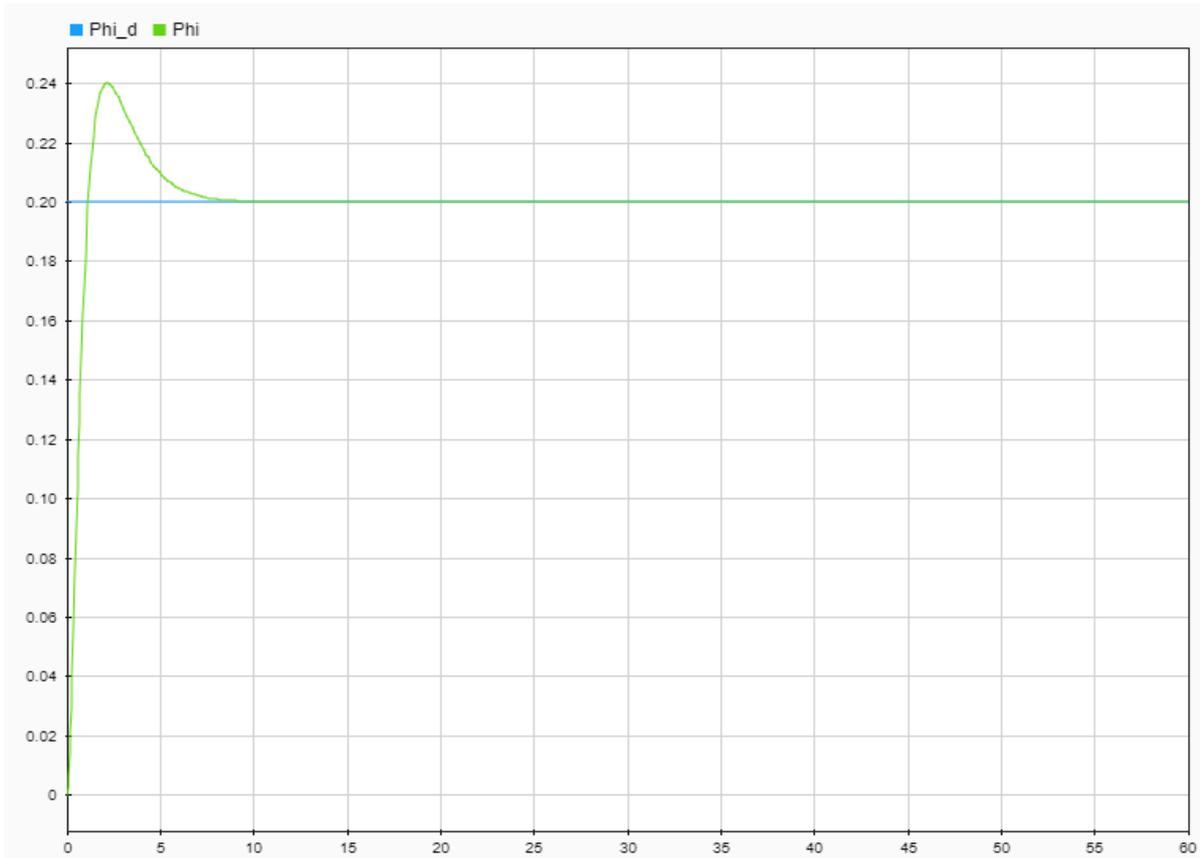
**Fig. 27: Tuned Altitude Real World Response**

**Table 6.1: Altitude control PID gains**

P	4.3
I	4.5
D	2

### 6.2.2 PID Tuning for Roll Angle ( $\phi$ ) and Pitch Angle ( $\theta$ ) Control

Roll and pitch PIDs have same gain values and the same responses, as they work on the same principle. Fig. 28 shows the roll or pitch response of the vehicle, when the desired angle is 12 degrees and the vehicle is trying to attain the altitude of 1m at the same time. Table 6.2 shows the PID gains of tuned roll and pitch PID controller.



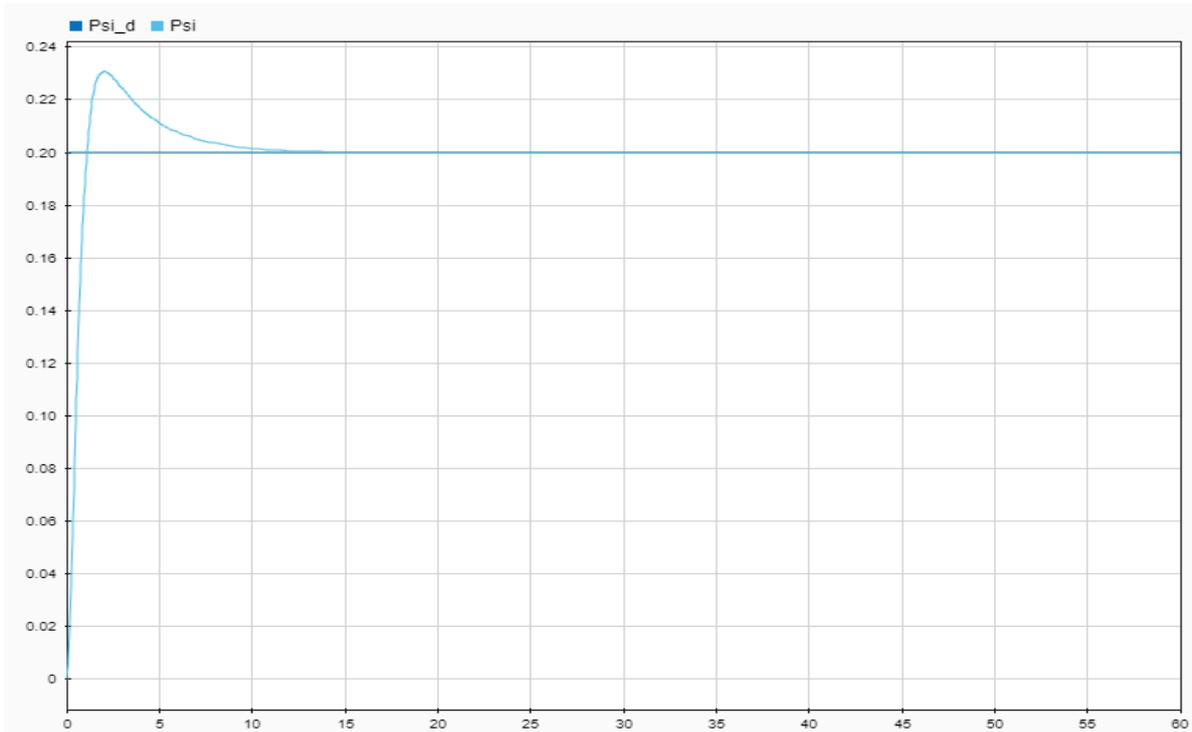
**Fig. 28: Roll and Pitch PID Controllers Tuned Response**

**Table 6.2: Roll and Pitch control PID Gains**

P	0.006
I	0.000009
D	0.013

### 6.2.3 PID Tuning for Yaw ( $\psi$ ) Control

Yaw PID values differ from roll and pitch, needed to be turned separately. It is difficult to tune for the yaw PID, as it depends on torque produced by all four rotors. Figure 29 shows the tuned response of the yaw controller, when the desired yaw rate is 12 degrees and the vehicle altitude command is 1 m at the same time. Table 6.3 gives the gains values of a tuned yaw PID controller.



**Fig. 29: Tuned Yaw Controller Response**

**Table 6.3: Yaw control PID Gains**

P	0.01
I	0.0002
D	0.03

So far, all the PIDs of the controller are tuned, which works fine in the simulation, but implementation on real hardware may need to change these values slightly, because with these gain values the quadcopters response to each command will be a bit aggressive.

### 6.3 Trajectory Control

A trajectory is planned using a signal-builder block (Figure 30) in Simulink, to visualize the performance of the vehicle, when over one signal is applied at the same time. The inertial trajectory is random, because roll, pitch and yaw are controlled in body coordinate system, instead of X, Y and Z in inertial coordinate system.

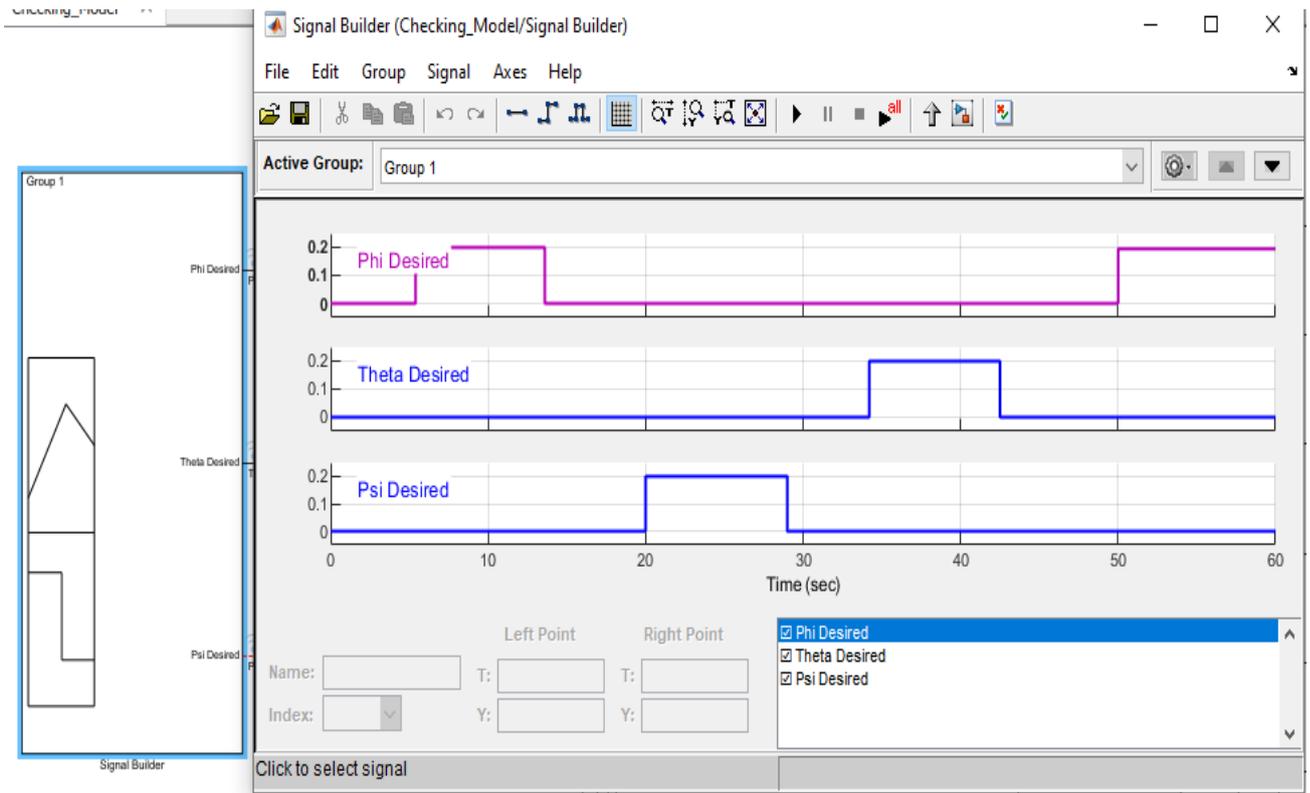
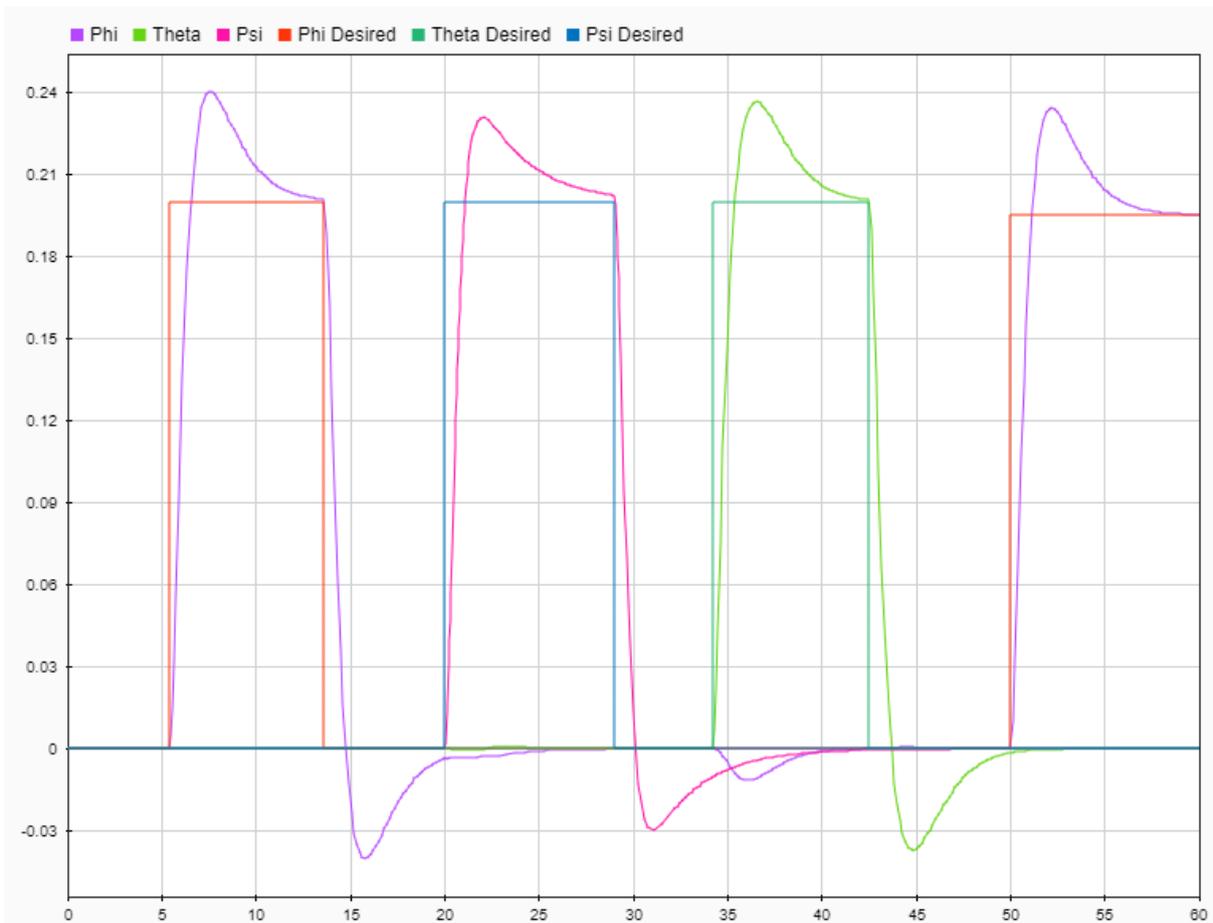


Fig. 30: Signal Builder Blocks



**Fig. 31: Signal Commands and Signal Responses**

It can be seen that the vehicle is responding to the commands very well (Fig. 31), verifying that the PID's are tuned properly. Response graphs (Fig. 31 and Figure 27), make sure that the controller is working and vehicle should fly obeying the commands, but we do not know, how the quadcopter will behave in the real world, to see this vehicle's trajectory is plotted in 3D (Figure 32).

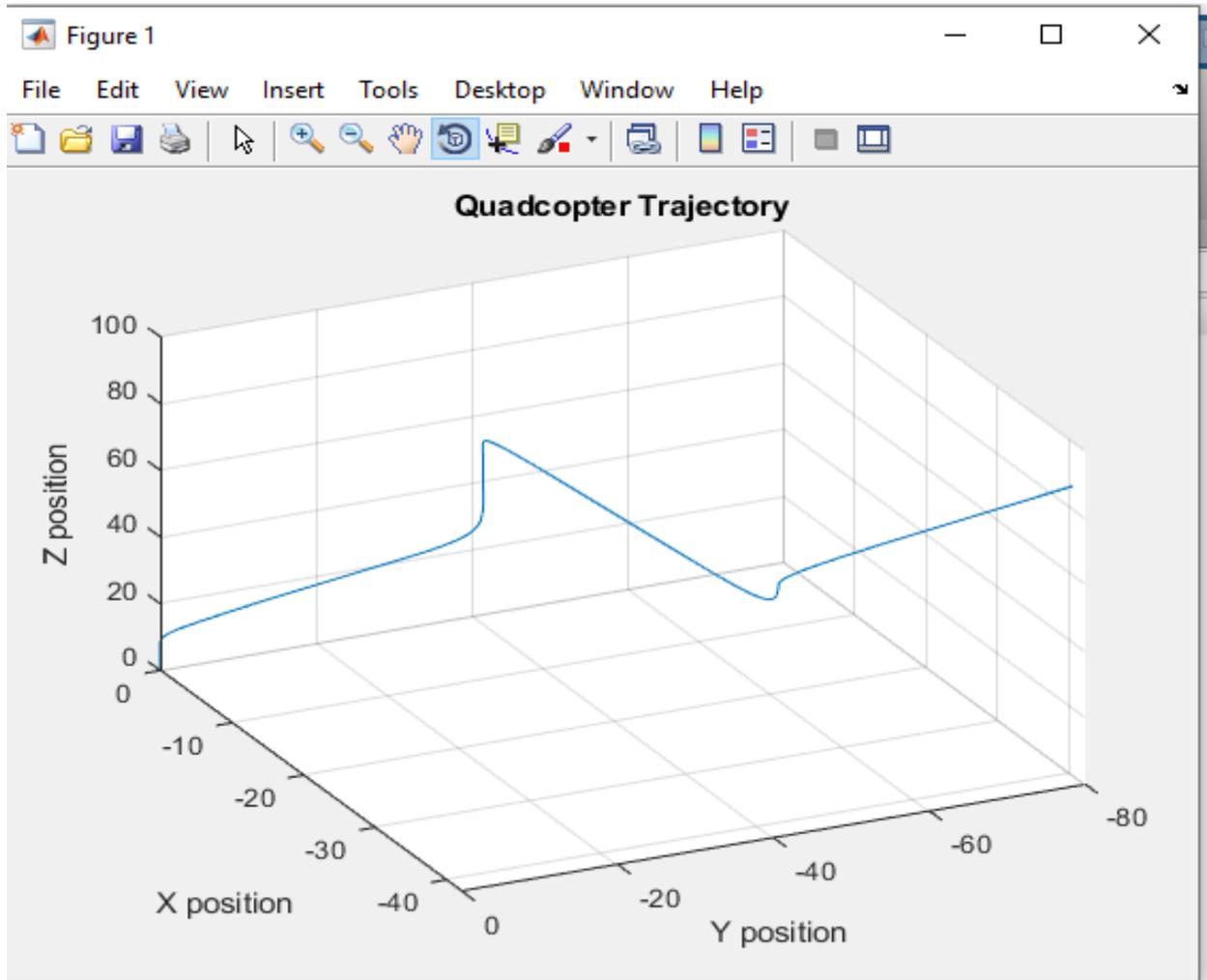


Fig. 32: Quadcopter 3D Trajectory

## 7. Conclusion and Future Scope

### 7.1 Concluding Remarks

In the present work the nonlinear model of a quadrotor system and its controller have been developed. The basic plant model contained simple thrust and torque coefficients along with the equations of dynamic motion. The basic plant model also had the choice of being in the inertial or body coordinate system which was added with the thrust equations within the model. Here we have considered the nonlinear dynamic equations of the system in closed loop. But the motor dynamics, sensor dynamics and battery dynamics were not considered.

The controller for a quadrotor system may be of many types, PID, LQR, Sliding Mode Control, etc. In this project, PID controller has been used to regulate four variables; the altitude ( $z$ ), roll angle ( $\phi$ ), pitch angle ( $\theta$ ) and yaw ( $\psi$ ) in a stable region. The tuning method for PID controllers is also relatively simple. The method presented here is manual control by controlling the above mentioned four variables. Here we have used auto-tune command of Simulink and sometimes fine-tuned some parameters manually to get better performance. It yielded a stable, controllable system. After the PID controllers were tuned individually for the control loops of all these four variables, a better ramp response was observed compared to the step responses.

The nonlinear plant model along with the controller was implemented in MATLAB<sup>®</sup>/Simulink platform. Each of the nonlinear equations were developed in Simulink in a modular way. The system responses with various desired reference inputs of the four variables were found satisfactory. MATLAB/Simulink is also useful due to the ease of converting a system from a block model to C or C++ code which can be explored later.

In summary, the present work deals with the development of the nonlinear quadrotor plant model, design and tuning of the PID controllers, implementation in MATLAB<sup>®</sup>/Simulink and testing of the same. It was observed that all the modules operated in closed loop when tested for individual variables and produced satisfactory results. Finally, a trajectory control of the quadrotor was also tested and found satisfactory 3-dimensional motion.

## 7.2 Future Scope

The modeling in this project considered a bit simplified one as per the system motion dynamic equations. The other characteristics that could be added to the model, such as hub forces, and ground effects. The addition of these in a model will increase the accuracy of the model to a practical quadrotor. Along with increased accuracy in the model, a range of control techniques may also improve the control of the quadcopter. Here, we have used PID control, which may not always produce robust performance as desired by the user. In such cases advance control techniques such as LQR, LQG, model predictive, sliding mode control etc. can be explored.

The implementation of the entire project can be done on an embedded system mounted on the quadrotor system itself, which could not be done due to lack of time and inaccessibility of the hardware lab in the department in this pandemic situation. However, we have thought of to develop the entire control and instrumentation system around an arduino board with gyro and 3D accelerometer sensors. This would require the additional modelling of various motors, and sensors like IMU etc. to have a definitive mathematical model and also need to establish communication protocol for the remote operation of the quadrotor. This entire hardware implementation part may be considered as one of the future work.

An aspect about the quadcopter modeling process that was not considered here was the use of complimentary filters to achieve the true Euler angles. In reality, this complimentary filter is used such that the accelerometer in conjunction with the gyroscopes to provide more accurate angle readings. This would be beneficial to add to a plant model to achieve a more practical model, which may be taken up as a future work. Wind resistance can also be added as a noise to achieve more robust performance of the quadrotor system. This would be effective when the quadcopter is flying against the wind gust.

Along with the additions in complexity to the model and using advanced control methods, it would be beneficial to stabilize and control the quadcopter in a better form when the data from the quadcopter could be continuously accessed and the model parameters could be updated. In many cases parameters used in the plant model are not perfect; estimation of the parameters from the input-output data from the quadcopter can make the model more acceptable one. This is another avenue of future work.

## 8. References

- [1] Bouabdallah, S. (2007); Design and Control of Quadrotors with application to autonomous flying. *Zurich: Ecole Polytechnique Federale De Lausanne.*
- [2] Alia Zakriti, S. A. (2019); Modeling of a Quadcopter Trajectory Tracking System Using PID Controller. *The 12th International Conference Interdisciplinarity in Engineering*, 564–571.
- [3] Ismail M. Khairuddin, A. P. (2014); Modelling and PID Control of a Quadrotor Aerial Robot. *Advanced Materials Research*, 327-331. doi:10.4028/www.scientific.net /AMR.903.327.
- [4] Usman, M. (2020); Quadrotor Modelling and Control with MATLAB/SIMULINK Implementation. *Bachelor Thesis, LAB University of Applied Science Technology Lappeenranta, Mechanical Engineering and Production Technology.*
- [5] MATLAB®/Simulink Toolbox, Series R2018a
- [6] Ferry, N. (2017, December ); Quadcopter Plant Model and Control System Development With MATLAB/Simulink Implementation. *Rochester Institute of Technology, Department of Electrical and Microelectronic Engineering. Rochester, New York: Kate Gleason College of Engineering.*
- [7] Selby, W. (n.d.); Arducopter. Retrieved from <https://www.wilselby.com/research/arducopter/>
- [8] Prof. A. V. Javir, K. P. (2015); Design, Analysis and Fabrication of Quadcopter. *Journal of Advance Research in Mechanical and Civil Engineering*, 2(3), 16-27.
- [9] Zimmerman, N. (2016); Flight Control and Hardware Design of MultiRotor Systems. *Master Thesis, Marquette University, Milwaukee, Wisconsin.*
- [10] Adnan Rafi Al Tahtawi, M. Y. (2019); Low-cost Quadrotor Hardware Design with PID Control System as Flight Controller. *TELKOMNIKA*, 1923-1930.