A REPORT

ON

NON-LINEAR MODELLING AND SIMULATION OF A QUADROTOR TYPE UAV

BY

SAPTADEEP DEBNATH 2014AATS0061U ECE

AT

BITS, Pilani – Dubai Campus
Dubai International Academic City (DIAC)
Dubai, U.A.E

First Semester, 2017-2018

A REPORT

ON

NON-LINEAR MODELLING AND SIMULATION OF A QUADROTOR TYPE UAV

BY

SAPTADEEP DEBNATH 2014AATS0061U ECE

Prepared in Fulfillment of the
Project Course: ECE F377

AT

BITS, Pilani – Dubai Campus
Dubai International Academic City (DIAC)
Dubai, UAE

First Semester, 2017-18

BITS, Pilani – Dubai Campus
Dubai International Academic City (DIAC)
Dubai, UAE

Course Name: Design Project

Course No: ECE F377

Duration: 4 Months

Date of Start:

Date of Submission:

Title of Report: Non-Linear Modelling and Simulation of a Quadrotor Type UAV

Name: Saptadeep Debnath

ID Number: 2014AATS0061U

Discipline: Electronics and Communication Engineering

Name of Project Supervisor: Dr. R. Mary Lourde

Keywords: Quadrotor, UAV, Non-linear control.

Project Area: Non-linear Control and Aerial Robotics

Abstract: This project presents non-linear modelling and simulation of a quadrotor type Unmanned Aerial Vehicle. The quadrotor will be modelled taking in the non-linear dynamics of the system. Different mechanics models are being studied and implemented in the simulation platform. MATLAB-Simulink environment is used for developing the system and for testing.

Signature of the Student | **Signature of Faculty**
Date: | **Date:**

ACKNOWLEDGEMENTS

Firstly, I would like to express my heartfelt gratitude to the Director of BITS Pilani Dubai Campus, Dr. R.N. Saha, who has ushered a new light on our college.

Next, I would like to thank Dr. R. Mary Lourde, my Project Supervisor, for her faith in me even before I started working under her and for her motivation, support, guidance and encouragement throughout the course of this project. She has been a great influence, urging me to learn and to be innovative, pushing me to excel.

Lastly, I would like to thank Dr. Abdul Razzak Instructor-in-Charge of Design Projects, for taking his time to look into my work and for his support regarding my project.

Saptadeep Debnath
2014AATS0061U

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS USED

Symbol	Meaning
φ	Roll
θ	Pitch
ψ	Yaw
b	Thrust Factor
d	Drag Factor
l	Distance between the rotor and the center of mass of the quadcopter
I_M	Motor Inertia
m	Mass of the Quadcopter
g	Acceleration due to gravity

CHAPTER 1
INTRODUCTION

1.1 Introduction

The recent times have seen a huge rise in the research and development of unmanned aerial vehicles (UAV). To increase their productivity, and to make them more user friendly, much work is being done on making the vehicles autonomous. An autonomous vehicle is capable of making its own decisions on every aspect, which in turn is based on a set of protocols. For a system to be autonomous, it has to be robust in an unknown environment. There is extensive research being done on linear control of the UAVs in an unknown environment, using the different control techniques. This project focuses on developing a non-linear control system for a quadcopter type UAV.

This research is carried out in two folds. First, setting up the MATLAB-Simulink platform for carrying out the simulations, based on a defined mathematical model of the quadcopter. Second, enhancing and optimizing the performance of the proposed controller.

1.2 Objectives

This research is aimed at enhancing the performance of the existing controllers, by dwelling more on the basic mathematical models used to model the UAV. The existing mathematical models used for modelling the system, are studied and implemented on the research platform. A control block is included in the system, to enhance the performance and to make it robust against the disturbances.

CHAPTER 2
LITERATURE REVIEW

The recent times have seen a rapid development in the field of robotics, be it ground-based, aerial-based or humanoid. A generic quadcopter has an 'X' or a '+' shape. This type of shape helps it in maintaining symmetry in a plane. Four motors along with electronics speed controllers (ESCs), at the end of each of the arms, are powered by a portable battery, which provides the required thrust to make the quadcopter fly. As there are six ranges of motion (i.e. forward, backward, left, right, up and down), but is controlled by only four motors, this system is generally referred to as an under actuated system.

For the ease of computation, the system is assumed to be linear [1]. But by the use of modern non-linear control theory, the performance of the system can be enhanced. Basic model of the quadcopter can be defined in the 6 ranges of motion as discussed earlier, where (x, y, z) are calculated on the centre of mass of the vehicle. The Euler angles, (ψ, θ, φ) defines the orientation of the system. Using these, the system can be defined using the Euler-Lagrange equation. The mathematical model gives a backbone to the further research on this topic. Though widely assumed linear, the nonlinearities can surely be added to the developed mathematical model [2]. The stability of which can be verified by Lyapunov stability theorem. Essentially, three types of motion need to be controlled, namely attitude (roll, pitch and yaw), altitude (z) and position (x and y). The system can also be modelled using a different type of mechanic model, like Hamiltonian mechanics rather than the frequently used Lagrangian model or the Newtonian model [8].

Proportional-integral-derivative (PID) is the most commonly used controller, for this type of system. A benefit of using this type of controller is that, it does not rely on the accurate model of the quadrotor [3]. In a PID loop, the errors in the loop are compensated by a three-stage function, directly dealing with the error (proportional), dealing with the error accumulated over time (integral) and compensating for the future errors in the system (derivative). Another type of controller used is a sliding mode controller [4]. An advantage of using the sliding mode controller is insensitivity to the errors in the model and any other type of uncertainities. Some other controllers include backstepping method [5], nonlinear $H\infty$ controller [6], and model predictive controller [7].

CHAPTER 3
QUADCOPTER DYNAMICS

As shown in figure 1, there are three basic types of quadcopter configurations, quad +, quad X and quad H. “Quad +” is chosen as the research platform for this project.

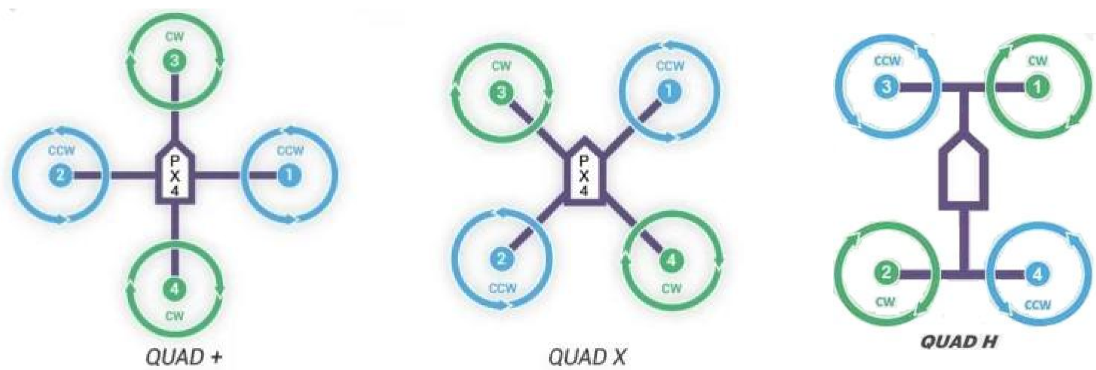

Figure 1 Different configurations for a quadcopter type UAV

3.1 Mathematical Approach

A quadcopter is an under actuated system, with four inputs (U1 ~ U4) and six output values (x, y, z, φ, θ, ψ). This system is modelled by using one of the four mathematical models, described below.

1. Newtonian Mechanics

Newtonian mechanics is the classical way of approaching a physical system in a dynamic environment. In this approach, one has to factor in the constraints and the geometrical nature of the system. Newton’s laws of motion govern the classical mechanics and its computation.

2. Lagrangian Mechanics

It is the reformulation of the classical mechanics approach, in which a function of the generalized coordinates called the Lagrangian (L) is used to formulate the given system. According to this mechanics model, given the total kinetic and potential energies of the system we can choose some generalized coordinates and blindly calculate the equation of motions.

$$\mathrm{L} = \mathrm{T} - \mathrm{V} \tag{1}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q} \tag{2}$$

3. Hamiltonian Mechanics

Regarded as the extension to the Lagrangian mechanics, Hamiltonian approach uses the velocity and coordinate relations that are given by second order differential equation. The Hamiltonian ($\mathcal{H}$) corresponds to the total energy of the system, which for a closed system is the sum of kinetic and potential energy. It can be obtained from the Legendre transformation of the Lagrangian (L).

$$\mathcal{H} = \mathrm{T} + \mathrm{V} \tag{3}$$

Position or the spatial coordinates (q) and momentum (p) are taken as the basis to formulate a given system.

$$T = \frac{p^2}{2m} \tag{4}$$
$$V = V(q) \tag{5}$$

4. **Routhian Mechanics**

Routhian mechanics is a coagulation of both Lagrange mechanics and Hamiltonian mechanics. It uses the independence of each particle, and thus solves a part of the system with Hamiltonian and some by Lagrangian approach. Routhian approach, as is expected, requires more of intelligence to reduce the work to a minimum. It is the smartest approach which is a little off the path for a machine.

3.2 Nonlinearities in the system

The dynamic model of the quadcopter factors in many nonlinearities, majority of which are mentioned below.

1. Drag Coefficient – It is a measure of resistance of an object in a fluid environment.
2. Inertia Matrix – Moment of inertia of the quadcopter system, in all the three axes (x, y, z) in a matrix format.
3. Vehicle Shape – The quadcopter platform is assumed symmetrical along the x and y-axes.
4. Relation between thrust and RPM – Thrust is assumed to be proportional to the square of the propellers' speed
5. Propeller Dynamics – Propellers are assumed rigid. The motors will provide unequal thrust if the propellers are not rigid.
6. Weight Distribution – The center of mass is assumed to be at the geometric center of the system.

3.3 Dynamic Model

The quadcopter design as represented in figure X, demonstrates the torques, angular velocities, forces and the motion in the different directions (x, y, z) & along the different angles (φ, θ, ψ). [9]

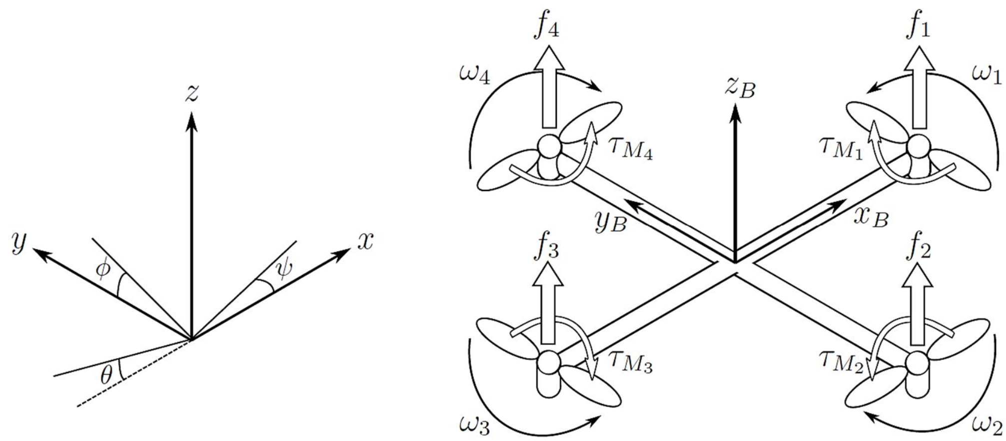

Figure 2 Inertial and Body frame

A rotation matrix (**R**) is used which relates the body frame to the inertial frame of the system. A vector $\vec{v}$ in the body frame is represented as $\mathbf{R}\vec{v}$ in the inertial frame. Rotation matrix is a powerful tool as it can negate the effect of roll, pitch and yaw in the body frame, and represent the orientation in the inertial frame. (C_x = Cos x, S_y = Sin y)

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{10}$$

As discussed previously, the quadcopter is assumed symmetrical in shape. Two of the arms align with the x-axis and the other two with the y-axis. Therefore, the moment of inertia across x-axis is same as of y-axis ($I_{xx} = I_{yy}$). The inertial matrix can be denoted as,

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{11}$$

The angular velocities of the individual motors produce a torque denoted as τ_{M_i}, where *'i'* specifies the motor numbers. The torques produced by the motors depend on two major non-linear components, the drag factor *'d'* and the inertia moment of the rotors I_M.

$$\tau_{M_i} = d\,\omega_i^2 + I_M\,\dot{\omega}_i \tag{12}$$

Apart from the torques produced by the individual motors, three additional torques are also produced with reference to the body frame. The torques corresponds to the body frame angles, namely roll, pitch and yaw.

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l\,b\,(-\omega_2^2 + \omega_4^2) \\ l\,b\,(-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix} \tag{13}$$

The four motors provide a thrust T, which allows the quadcopter to achieve a certain change in z-direction in the body frame.

$$T = b\ (\omega_1^2 + \ \omega_2^2 + \ \omega_3^2 + \ \omega_4^2) \tag{14}$$

The dynamics of the quadcopter discussed above are further used to derive the acceleration of the system along the x, y and z-axes using the Newton-Euler equations.

$$m\ddot{a} = G + RT_B \tag{15}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{T}{m} \begin{bmatrix} C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi S_\theta C_\phi - C_\psi S_\phi \\ C_\theta C_\phi \end{bmatrix} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{16}$$

In this representation, $\ddot{a}$ is a column matrix with the acceleration values across the x, y and z-axes. The gravitational constant G when added with the amount of thrust produced in the inertial frame, i.e. taking a product of the rotation matrix R with the thrust in the body frame T_B gives the total forces in the x, y and z-axes. Similar to the acceleration calculation in the x, y and z-axes, the acceleration for roll, pitch and yaw is denoted as follows.

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \left(I_{yy} - I_{zz}\right)\dot{\theta}\dot{\psi}/I_{xx} \\ (I_{zz} - I_{xx})\dot{\phi}\dot{\psi}/I_{yy} \\ \left(I_{xx} - I_{yy}\right)\dot{\theta}\dot{\phi}/I_{zz} \end{bmatrix} + I_M \begin{bmatrix} \dot{\theta}/I_{xx} \\ \dot{\phi}/I_{yy} \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix} \tag{17}$$

Where,

$$\omega_\Gamma = \ \omega_2 + \omega_4 - \omega_1 - \omega_3 \tag{18}$$

CHAPTER 4
CONTROLLER DESIGN

4.1 Attitude Control

A similar PD controller is used as in [9]. A proportional-derivative controller is preferred in this case as the system doesn't have many erratic disturbances.

$$\begin{aligned} \tau_{\phi} &= [K_{P,\phi}(\phi_d - \phi) + K_{D,\phi}(\dot{\phi}_d - \dot{\phi})]I_{xx}, \\ \tau_{\theta} &= [K_{P,\theta}(\theta_d - \theta) + K_{D,\theta}(\dot{\theta}_d - \dot{\theta})]I_{yy}, \\ \tau_{\psi} &= [K_{P,\psi}(\psi_d - \psi) + K_{D,\psi}(\dot{\psi}_d - \dot{\psi})]I_{zz}, \end{aligned} \tag{21}$$

The corrected angular velocities can now be calculated from the equations (13) and (14) with the result from (21).

$$\omega_i^2 = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \frac{T}{4b} + \frac{1}{2dl}\begin{bmatrix} -\tau_{\theta} \\ -\tau_{\phi} \\ \tau_{\theta} \\ \tau_{\phi} \end{bmatrix} + \frac{1}{4d}\begin{bmatrix} -\tau_{\psi} \\ \tau_{\psi} \\ -\tau_{\psi} \\ \tau_{\psi} \end{bmatrix} \tag{22}$$

4.2 Position Control

The purpose of a position control is to adequately have a change in the position of the system without having any deviation from the set route and set points. This is done by changing the four control inputs to the quadcopter, i.e. the angular velocities of the four motors. This control loop is generally referred to as the outer control loop, as this control loop is not typically integrated in a commercial flight controller, but is rather implemented on an onboard/offboard computer. A PD controller is implemented for the x, y and z coordinates as well.

$$\begin{aligned} X_d &= K_{P,x}(x_d - x) + K_{D,x}(\dot{x}_d - \dot{x}), \\ Y_d &= K_{P,y}(y_d - y) + K_{D,y}(\dot{y}_d - \dot{y}), \\ Z_d &= K_{P,z}(z_d - z) + K_{D,z}(\dot{z}_d - \dot{z}), \end{aligned} \tag{23}$$

The approach for trajectory control is to calculate the required angular velocities to move the quadcopter from the current position to a desired position. The desired angular velocities can be calculated by the equation (22), using the thrust and the torque values. The torque value are in turn calculated by the equation (21). Since it can be noticed in the equation (21), the desired values of the roll, pitch and yaw is required. We require a relation between the desired values of roll, pitch and yaw with the desired values of x, y and z, which is shown in [10].

$$\phi_d = \arcsin\left(\frac{X_d S_{\psi} - Y_d C_{\psi}}{X_d^2 + Y_d^2 + (Z_d + g)^2}\right),$$

$$\theta_d = \arcsin\left(\frac{X_d C_\psi + Y_d S_\psi}{Z_d + g}\right), \tag{24}$$

$$T = m\,[X_d(C_\psi S_\theta C_\phi + S_\psi S_\phi) + Y_d(S_\psi S_\theta C_\phi - C_\psi S_\phi) + (Z_d + g)(C_\theta C_\phi)],$$

The interactions between the different physical quantities defined in the equations (16), (17), (21), (22), (23) and (24), is represented in the Figure 13. First, the desired position coordinates are fed to a PD controller. The controlled input is then converted to the desired roll, pitch, yaw and the thrust value. The desired roll, pitch and yaw values then go through a PD controller, which is the inner control loop, which was described in the previous section. The corresponding torque values in addition with the thrust is used to calculate the angular velocities of the individual motors. The angular velocities with the help of the calculated roll, pitch, and yaw values are used to then find the current x, y and z coordinates.

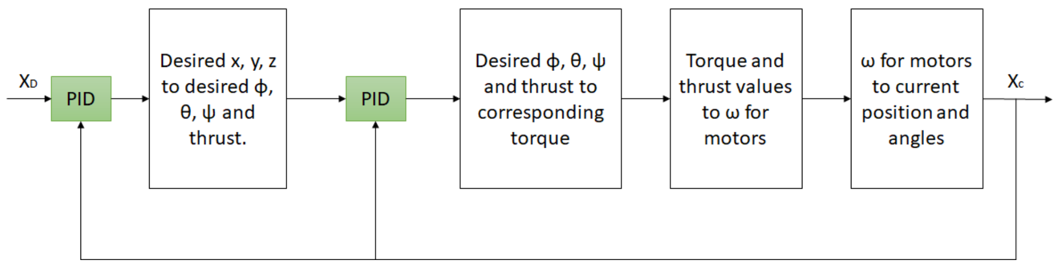

Figure 3 Interactions between different physical quantities

CHAPTER 5
SIMULINK ENVIRONMENT

As shown in figure 2, the complete Simulink block consists of two main subsystems, with one output terminal.

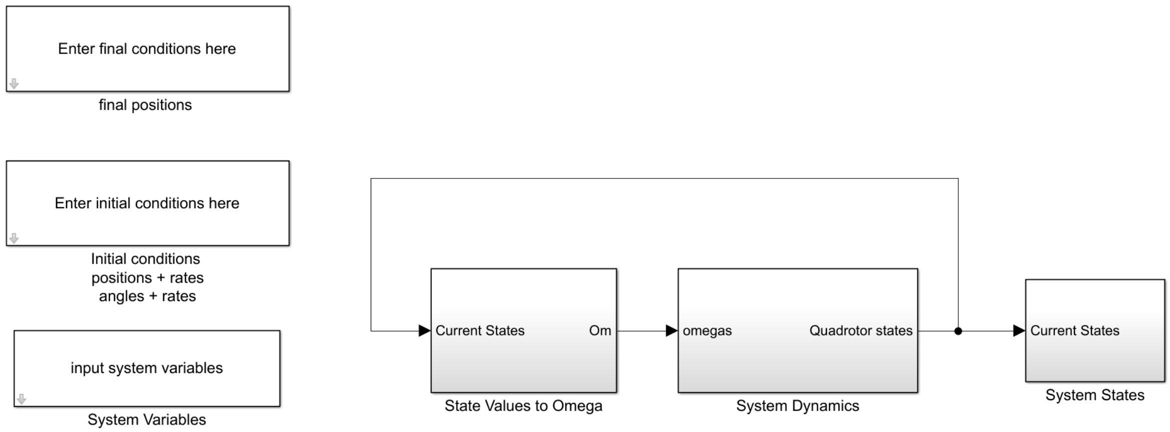

Figure 4 Complete Simulink Block Diagram

5.1 Input Blocks

There are three set of input variables; initial attitude variables, initial system variables and the final position values.

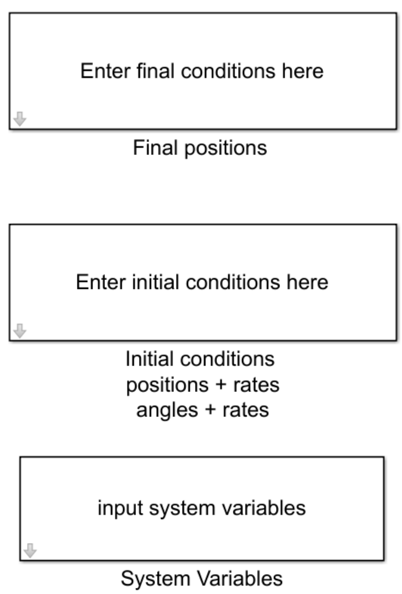

Figure 5 Input Blocks

1. **Initial Attitude Variables**

This block takes in the initial values and the rate of change of x, y, z, roll, pitch and yaw.

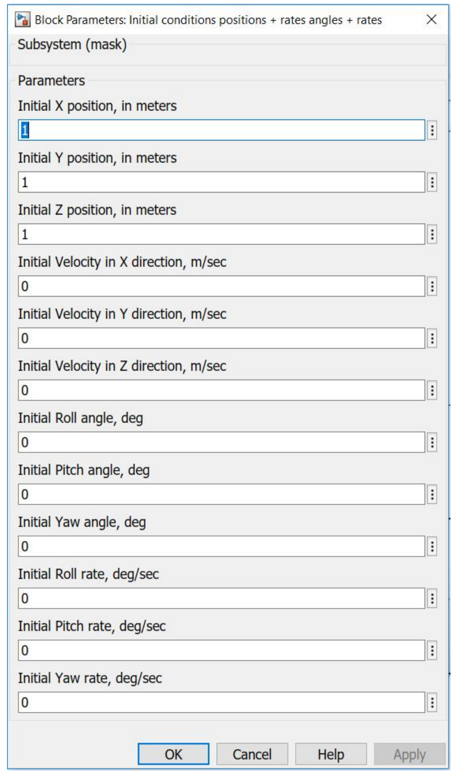

Figure 6 Initial Conditions Block

2. Initial System Variables

The system variables comprises of, quadcopter mass, thrust factor, drag factor, rotor inertia, length of each chord and the moment of inertia along each axes.

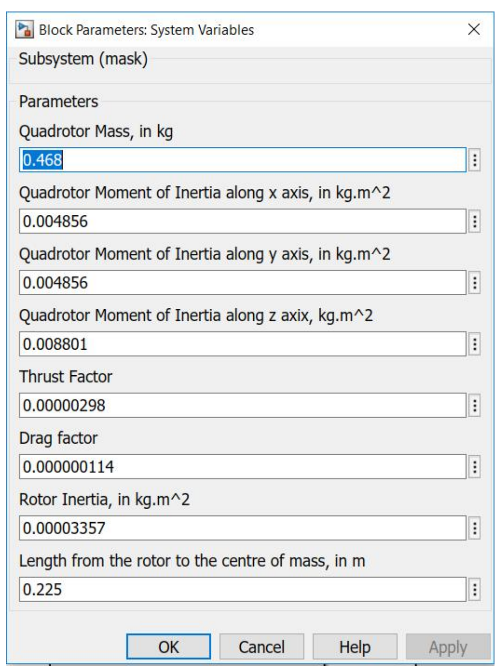

Figure 7 System Variables Block

3. Final Position Values

The desired coordinate points are entered in this block, which takes in the final position values of x, y and z.

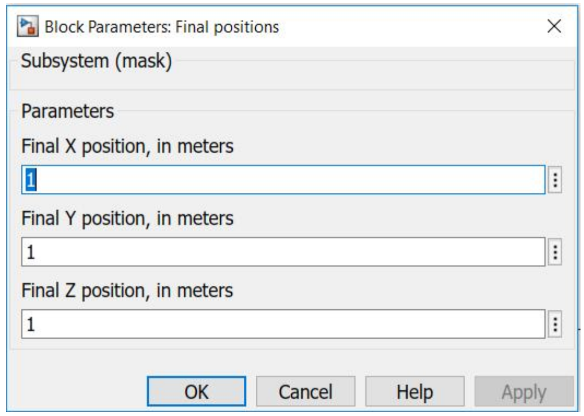

Figure 8 Final Position Block

5.2 Dynamics Blocks

There are two main blocks which define the whole dynamics of the system. First the ‘State Values to Omega’ block and the ‘System Dynamics’ block.

1. State Values to Omega

This subsystem is responsible for converting the desired positions, with the help of the current state values and the initial system variables, to the required omega (RPM) values for each of the four motors. As described in the Controller design, there are two levels of control mechanism in this system; the inner control loop and the outer control loop.

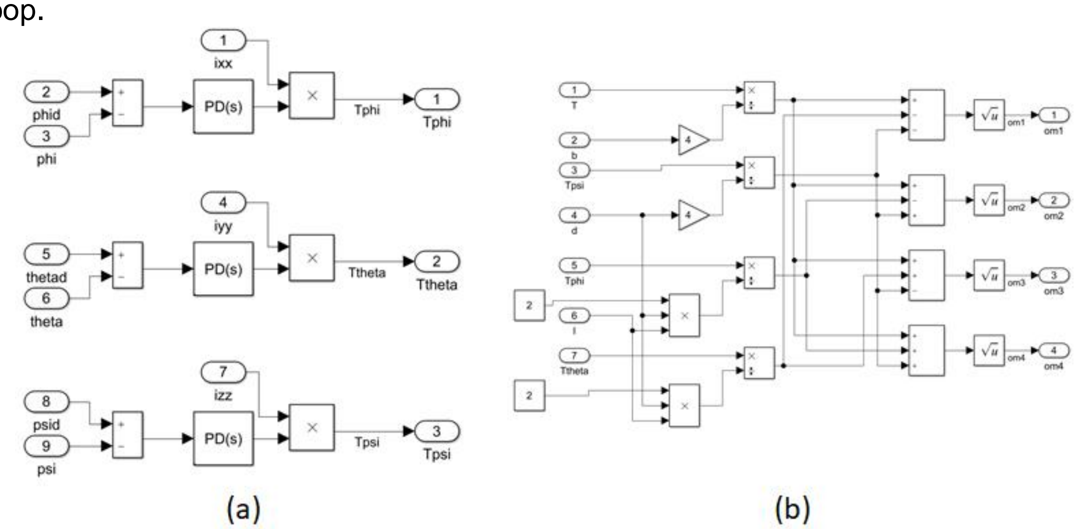

Figure 9, (a) Attitude control block, PD control for roll, pitch and yaw, (b) Conversion of thrust and torque values to corresponding omega (RPM) values

The attitude control block as shown in the figure x (a), is modelled based on the equation (21). This block converts the desired roll, pitch and yaw values to their corresponding torque values. The torque and the thrust obtained is then parsed into the next block as shown in figure x (b), which then calculates the equivalent omega values, according to the equation (22).

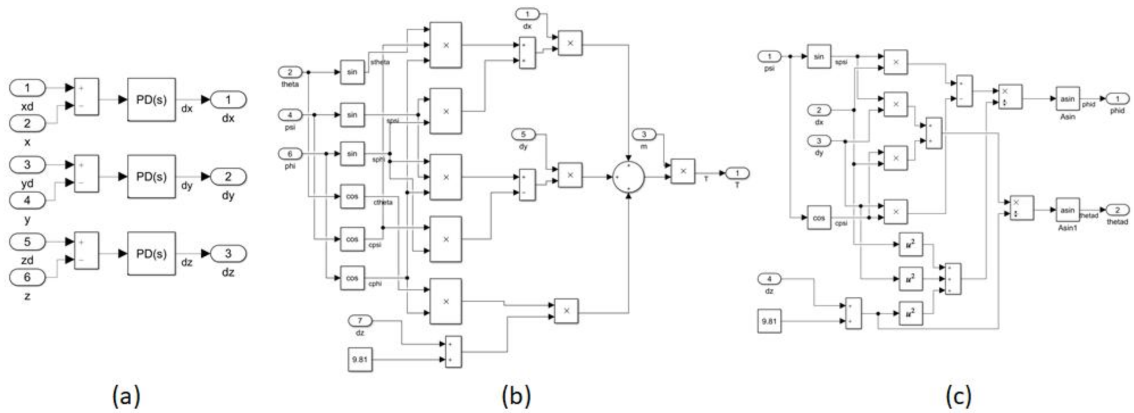

Figure 10, (a) Position control block, PD control for x, y, z (b) Thrust calculation (c) Desired Roll and Pitch calculation

The outer loop controls the position values (i.e. x, y and z). The position control block as shown in the figure x (a) is defined by the equation (23). Using the values obtained from this block, the required thrust, roll and pitch values are calculated as shown in figure x (b) and (c), which are modelled by the equation (24).

The overall model is shown below,

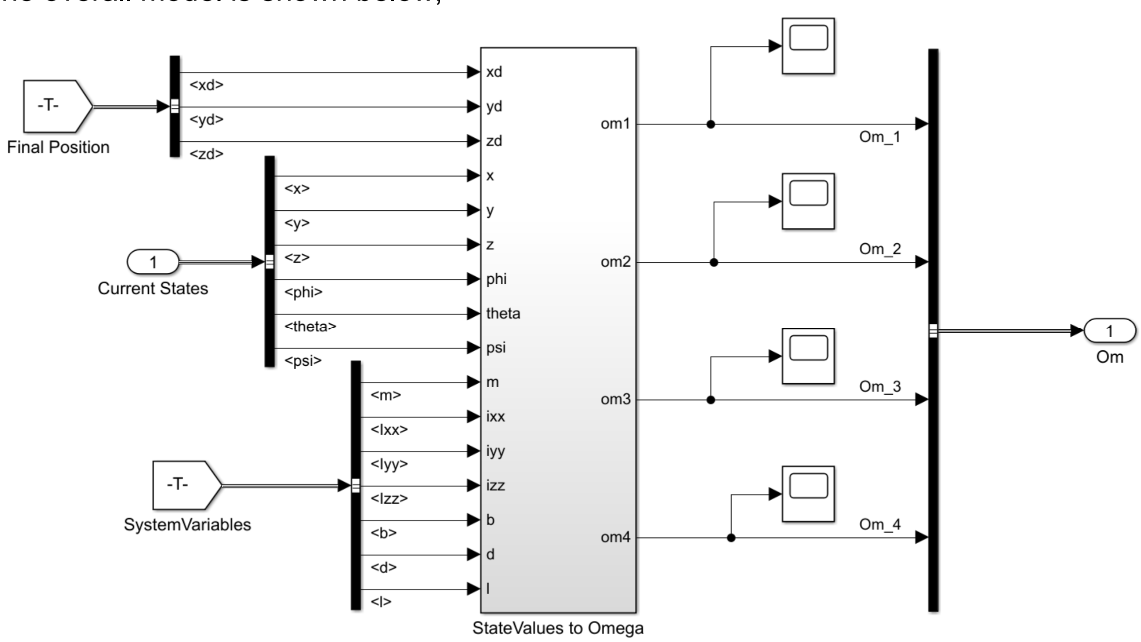

Figure 11 Overall model for state values to RPM values

2. **System Dynamics**

The individual omega values obtained from the previous block is further used to calculate the current states. Equations 16 and 17 are used to calculate the acceleration values across the six coordinates; i.e. x, y, z, roll, pitch and yaw. The double derivatives are then integrated to get the velocity values, and integrated further to obtain the attitude values.

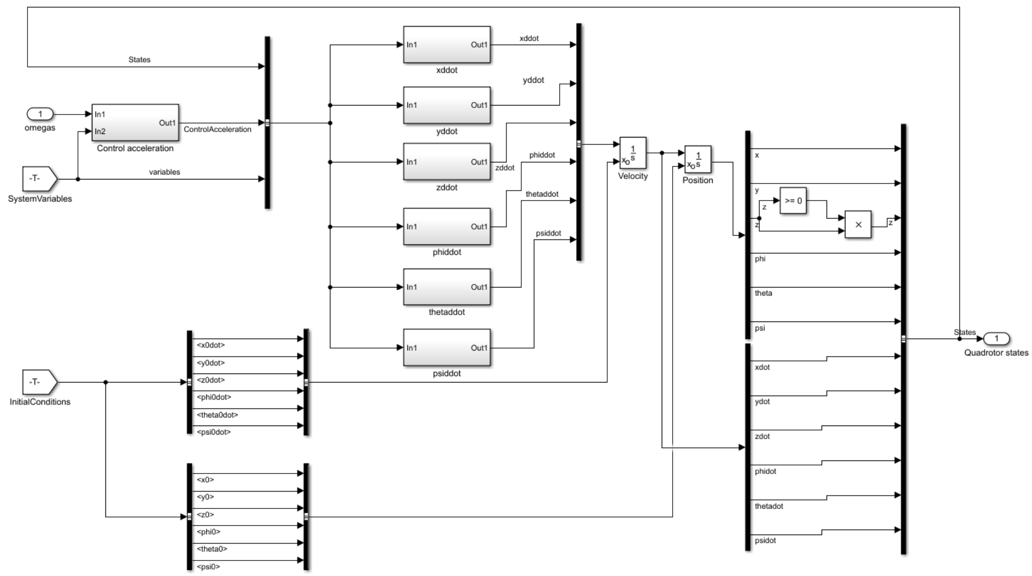

Figure 12 System Dynamics Model

As can be seen from the figure X, an additional condition has been included in the system, defining that the z position of the quadcopter cannot be less than zero. The ground level is taken as the zero position for the coordinate system.

5.3 Output Block

The current state values are parsed into this block. Display blocks and graphs are used to visualize the values in a systematic manner.

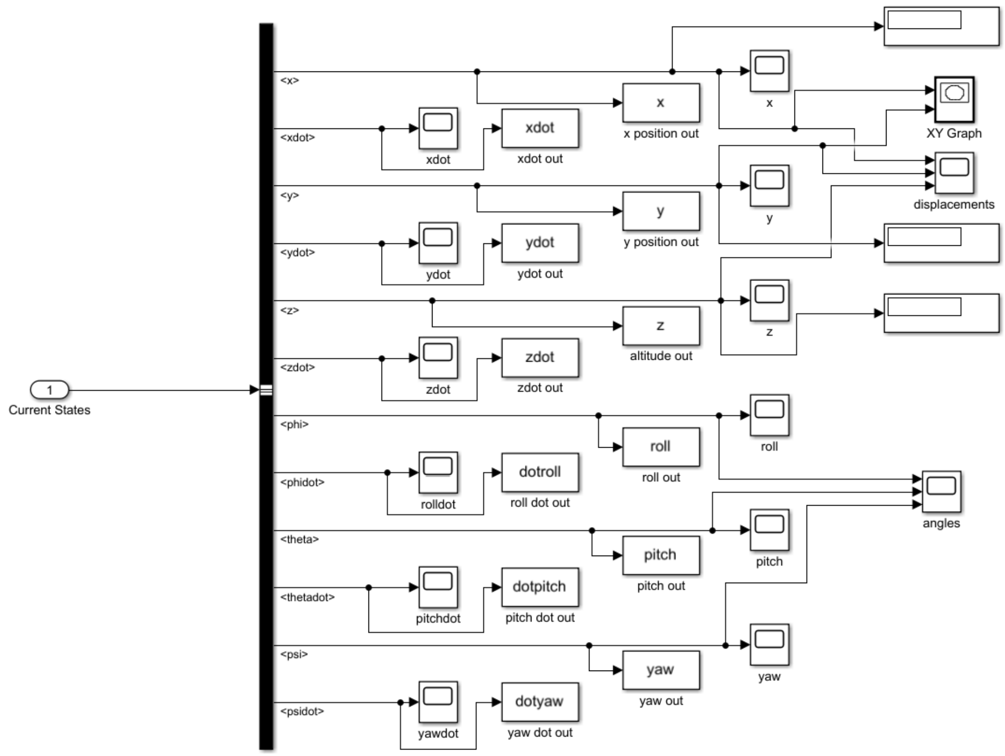

Figure 13 Output Block

CHAPTER 6
SIMULATION RESULTS

This section concentrates on tuning the control parameters as defined in the section 4 (Controller Design). There are six individual PD controllers present in the system, which are for x, y, z, roll, pitch and yaw. Tuning these parameters is done in a three step process. During the course of the simulations, the simulation time is taken as 40 seconds, and graphs are plotted for x, y, z, roll, pitch and yaw. Auto-tuning method in MATLAB is used for tuning the system.

First the z-control, or the altitude control is tuned. In this simulation the initial coordinates are set as [1, 1, 1] and the desired coordinates as [1, 1, 8].

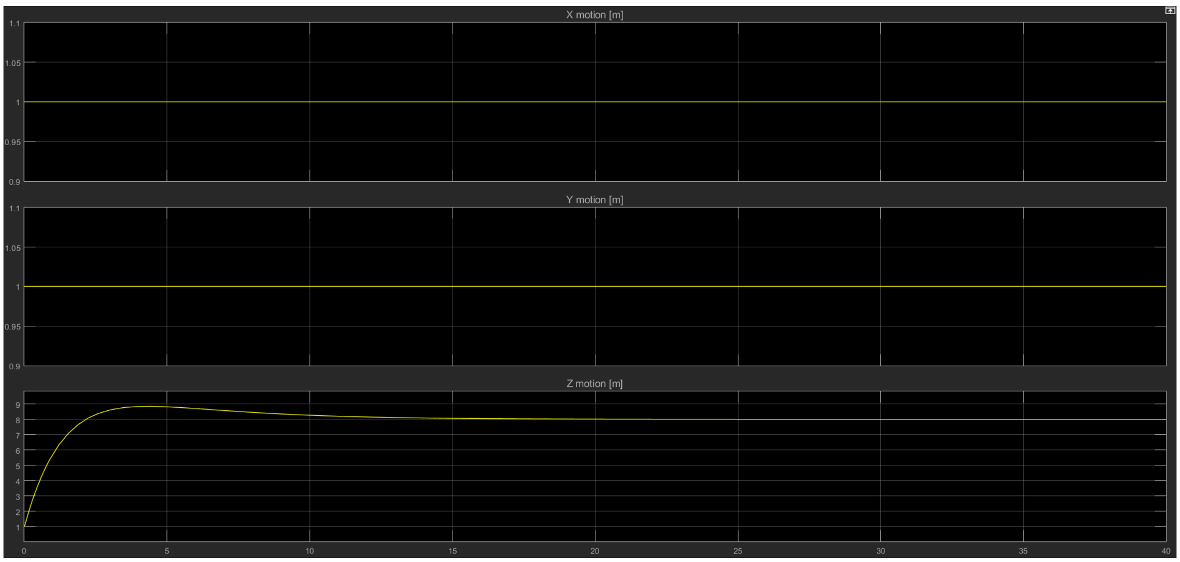

Figure 14 Position Graph for simulation 1

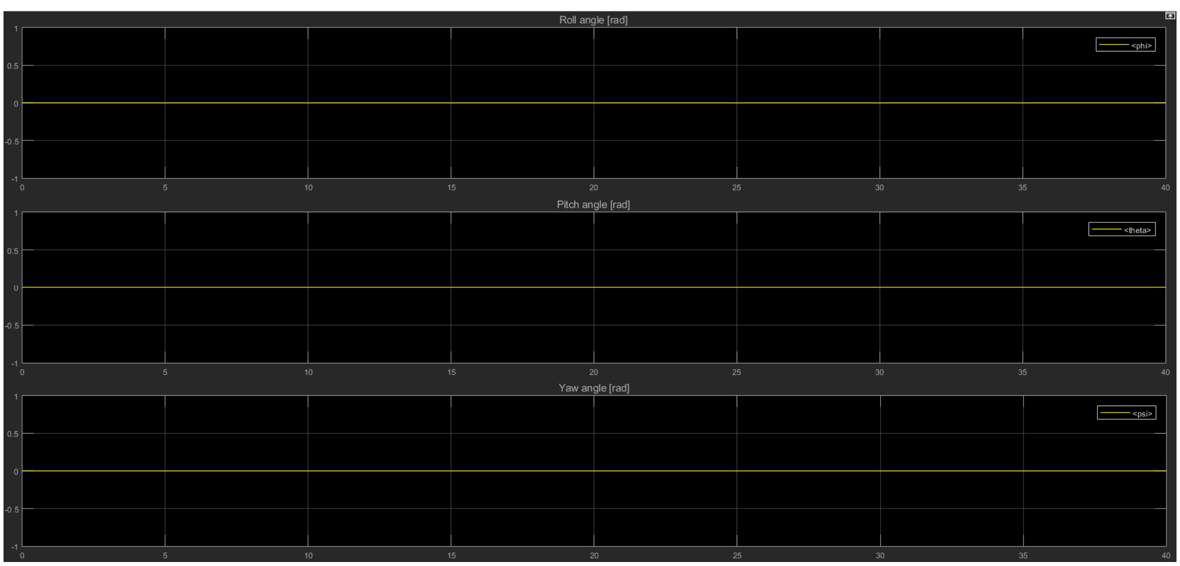

Figure 15 Attitude Graph for simulation 1

As seen from the figure x, there is no change observed in the x and y axis, as was desired. Consequently, there is no change in either roll, pitch or yaw. The following parameters are achieved after tuning.

Parameter	Value
$K_{P,z}$	0.21
$K_{D,z}$	0.976

Table 1 Control Parameters for z-axis

In the second simulation, the x axis is tuned. As the x-axis motion is solely depended on the pitch motion, so along with x, the pitch controller is tuned. The initial coordinates are set as [1, 1, 1] and the desired coordinates as [8, 1, 1]. This results in a forward motion of the quadcopter.

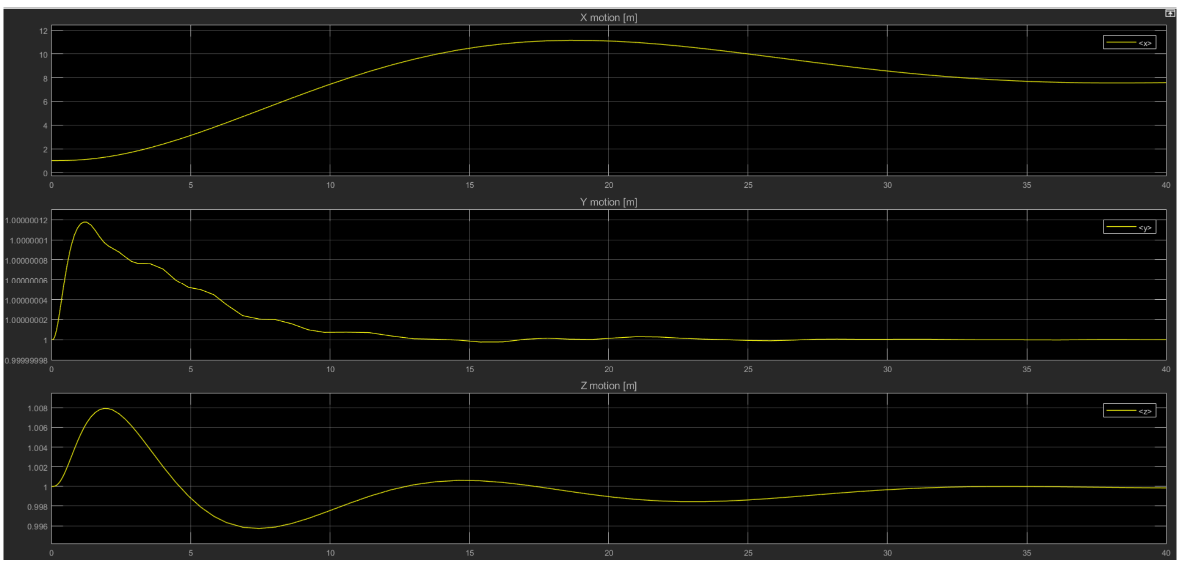

Figure 16 Position Graph for simulation 2

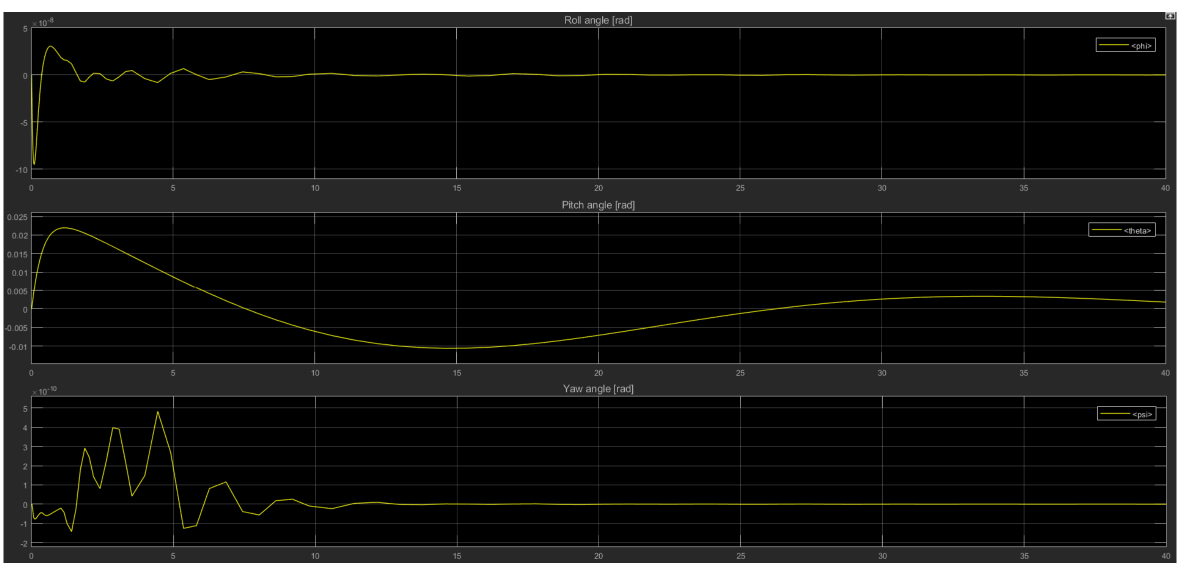

Figure 17 Attitude Graph for simulation 2

As expected there is a desirable change in the x-axis, as shown in the figure x. in addition to that there are minute disturbances which can be observed in the y and z-axis. The following control parameters are achieved after the tuning.

Parameter	Value
$K_{P,x}$	0.0058
$K_{D,x}$	0.1866
$K_{P,\theta}$	0.000256
$K_{D,\theta}$	0.0134

Table 2 Control Parameters for x-axis and pitch (theta)

The final simulation is carried out for tuning the y-axis motion. As y-axis motion is defined by the roll motion, the controller for roll angle is also controlled in the process. The initial coordinates for the simulation as taken as [1, 1, 1] and the desired coordinates as [8, 8, 8].

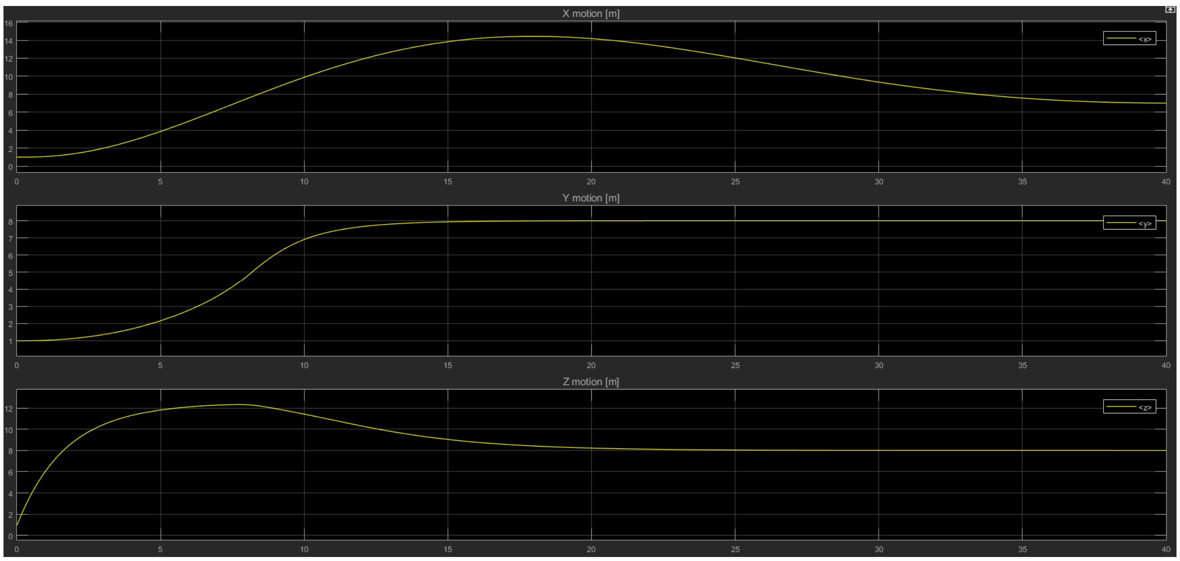

Figure 18 Position Graph for simulation 3

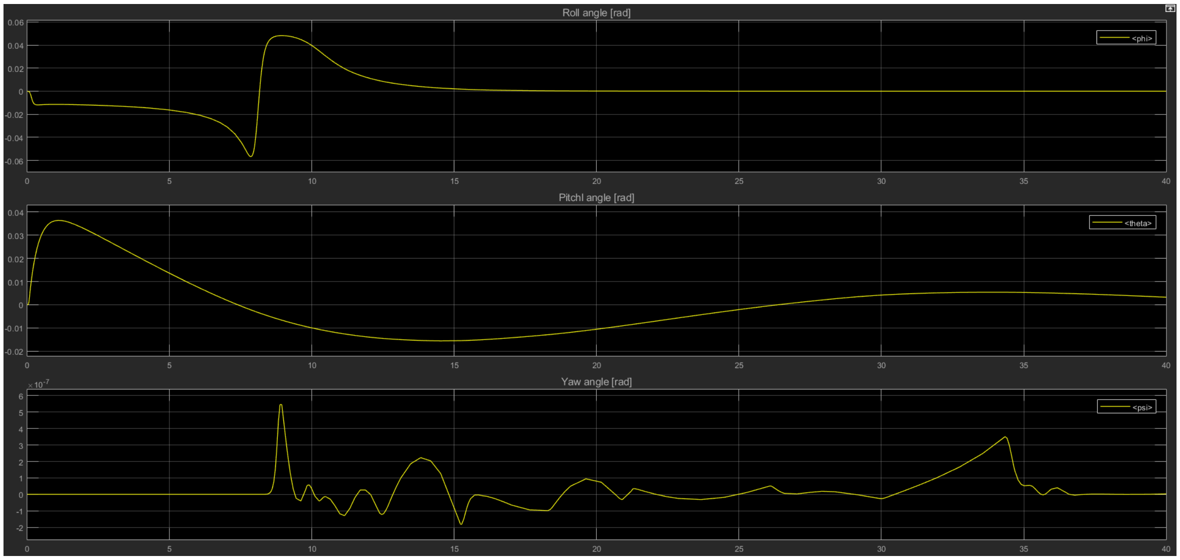

Figure 19 Attitude Graph for simulation 3

From the figure x, it can be observed that the desired coordinate values for x, y and z are achieved by the system. In addition to that the changes in the roll, pitch and yaw angles also negate down to zero when the simulation stops. During this simulation the yaw angle is also tuned, which gives the following tuning parameters.

Parameter	Value
$K_{P,y}$	12.828
$K_{D,y}$	27.984
$K_{P,\phi}$	2.477
$K_{D,\phi}$	0.94
$K_{P,\psi}$	234.031
$K_{D,\psi}$	68.748

Table 3 Control Parameters for y-axis, roll (theta) and yaw (psi)

All the control parameters are shown in the table below.

Parameter	Value	Parameter	Value
$K_{P,x}$	0.0058	$K_{P,\theta}$	0.000256
$K_{D,x}$	0.1866	$K_{D,\theta}$	0.0134
$K_{P,y}$	12.828	$K_{P,\phi}$	2.477
$K_{D,y}$	27.984	$K_{D,\phi}$	0.94
$K_{P,z}$	0.21	$K_{P,\psi}$	234.031
$K_{D,z}$	0.976	$K_{D,\psi}$	68.748

Table 4 Tuned Control Parameters

CHAPTER 7
SUMMARY AND CONCLUSION

The first part of this report deals with the basics of the quadcopter dynamics, the study of different types of mechanics model and the nonlinearities in a quadcopter system.

Further, the system dynamics and the control architecture is incorporated in the Simulink platform as separate subsystems. A PD controller is used to control the six coordinates as defined in a quadcopter system. This includes a two level control system, an inner control loop and an outer control loop. For the testing phase the system dynamics are chosen to be governed by the Euler-Lagrange model.

The latter half deals with the simulation results from the Simulink block, by feeding in the desired values and observing the anticipated motion. A three step simulation is carried out to tune the six control blocks. The three motions being the upward motion, forward motion and a diagonal motion (i.e. a change in all the three coordinates). The system designed responds properly for a localized coordinate system, as the parameters are tuned for a localized coordinate system. This design serves as an ideal non-linear simulation model for a plus '+' configured quadcopter.

REFERENCES

[1] Castillo, Pedro, Rogelio Lozano, and Alejandro Dzul. "Stabilization of a mini-rotorcraft having four rotors." Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 3. IEEE, 2004.

[2] Choi, Young-Cheol, and Hyo-Sung Ahn. "Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests." IEEE/ASME transactions on mechatronics 20.3 (2015): 1179-1192.

[3] Salih, Atheer L., et al. "Modelling and PID controller design for a quadrotor unmanned air vehicle." Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on. Vol. 1. IEEE, 2010.

[4] Xu, Rong, and Umit Ozguner. "Sliding mode control of a quadrotor helicopter." Decision and Control, 2006 45th IEEE Conference on. IEEE, 2006.

[5] Madani, Tarek, and Abdelaziz Benallegue. "Backstepping control for a quadrotor helicopter." Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006.

[6] G. V. Raffo, M. G. Ortega, and F. R. Rubio, “An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter,” Automatica, vol. 46, no. 1, pp. 29–39, 2010.

[7] M. Bangura and R. Mahony, “Real-time Model Predictive Control for Quadrotors,” IFAC Proceedings Volumes, vol. 47, no. 3, pp. 11 773 – 11 780, 2014, 19th IFAC World Congress.

[8] Hu, Kaijian, Yuhu Wu, and Xi-Ming Sun. "Attitude controller design for quadrotors based on the controlled Hamiltonian system." Control And Decision Conference (CCDC), 2017 29th Chinese. IEEE, 2017.

[9] Dikmen, I. Can, Aydemir Arisoy, and Hakan Temeltas. "Attitude control of a quadrotor." Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on. IEEE, 2009.

[10] Zuo, Zongyu. "Trajectory tracking control design with command-filtered compensation for a quadrotor." IET control theory & applications 4.11 (2010): 2343-2355.

[11] Luukkonen, Teppo. "Modelling and control of quadcopter." Independent research project in applied mathematics, Espoo (2011).

[12] Hehn, Markus, and Raffaello D'Andrea. "Quadrocopter trajectory generation and control." IFAC Proceedings Volumes44.1 (2011): 1485-1491.

[13] Gibiansky, Andrew. "Quadcopter dynamics, simulation, and control." Andrew Gibiansky:: Math→[Code] 21 (2012).

[14] Carrillo, Luis Rodolfo García, et al. "Modeling the quad-rotor mini-rotorcraft." Quad Rotorcraft Control. Springer London, 2013. 23-34.

[15] Mellinger, Daniel. Trajectory generation and control for quadrotors. University of Pennsylvania, 2012.

[16] Bhatkhande, Pranav, and Timothy C. Havens. "Real time fuzzy controller for quadrotor stability control." Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on. IEEE, 2014.