A MID SEM REPORT

ON

FLIGHT DATA ANALYSIS OF AN UNMANNED AERIAL VEHICLE

BY

SAPTADEEP DEBNATH 2014AAPS061U ECE

AT

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

Second Semester, 2015-2016

A MID SEM REPORT

ON

FLIGHT DATA ANALYSIS OF AN UNMANNED AERIAL VEHICLE

BY

SAPTADEEP DEBNATH 2014AAPS061U ECE

Prepared in Partial Fulfillment of the
Project Course: ECE F266

AT

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

Second Semester, 2015-16

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

Course Name: Study Project

Course No: ECE F266

Duration: 4 Months

Date of Start: 8th February 2016

Date of Submission: 24th May 2016

Title of Report: Flight Data Analysis of an Unmanned Aerial Vehicle

Name: Saptadeep Debnath

ID Number: 2014AAPS061U

Discipline: Electronics and Communication Engineering

Name of Project Supervisor: Dr. V. Kalaichelvi

Keywords: Pixhawk, Quadrotor, UAV, MATLAB, SimMechanics, Solidworks.

Project Area: Control Systems and Aerial Robotics

Abstract: This project presents the design, development and control parts of an Unmanned Aerial Vehicle of the type quadrotor. The proposed approach compares and validates the results achieved with the real time operation of a quadrotor to an analytical model of the quadrotor built in the Matlab/Simulink environment. The quadrotor used is equipped with an onboard autopilot system called Pixhawk. The real-time data is gathered from the Pixhawk's flight logs and analysed in Matlab. The dynamic model of the mechanical structure was imported from SolidWorks using SimMechanics in the Matlab/Simulink environment.

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, Prof. R.N. Saha, who has ushered a new light on our college.

Next, I would like to thank Dr. V. Kalaichelvi, 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. A. B. Chattopadhyay Instructor-in-Charge of Study Projects, for taking his time to look into my work and for his support regarding my project.

Saptadeep Debnath
2014AAPS061U

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Flying objects have always been a topic of fascination. Be it for amusements or for research and development. From the first flying machine to the invention of finger sized quadrotors, the field of aerospace is always growing.

In the recent past there is an increase in the market for the unmanned aerial vehicles, as they can replace the human interaction in extreme conditions. Such as a major natural calamity, wide spread health hazard, and other such major interactions where the human lives lies at stake. For such activities a Quadrotor is generally preferred, due to high stability (than the tricopter and hexcopter) and high maneuverability in comparison to the fixed wing planes. Once such quadrotor which is extensively used for research and development purposes is Parot AR.Drone. AR. Drone is used as the experimental model for this research work.

This research is carried out in two folds. First is the simulation of the model on the MATLAB environment. Second is the real time flight data analysis.

1.2 Objectives

This research is aimed at modelling a Controller for a stable flight for a Parot AR.Drone. It would be first simulated by modelling a 3D design of the AR.Drone in the Solidworks and importing the design to the MATLAB environment in a SimMechanics Model. A controller will then be added to the model and properly tuned so as to get the desired results. The second phase includes the implementation of the acquired controller values and comparing them with the real time flight values.

CHAPTER 2

LITERATURE REVIEW

There has been an exponential growth in the research and development of quadcopters and control aspect of the same. One such research paper [1] deals with the designing and simulation of an Unmanned Aerial Vehicle of the type quadrotor, and designing a PID control for the stability of the system. It explains the trends of the recent times and the advancements in the area of research in Unmanned Aerial Vehicles. In order to stabilize the quadrotor angles, separate PIDs were used for each angles, for which a closed loop control was designed. PID equations are then converted into linear system matrices. Depending on the angular values, the PID values were calculated and thus implemented on the real time model which was developed.

Research work carried out in Autonomous Systems Lab, ETH Zurich, Switzerland, also gives us a proof that apart from being manually controlled, a UAV can also be autonomously controlled with the use of onboard processors or controllers [2]. This paper is based on the results of modelling and control parts of the OS4 Project. It also describes about a Simulink model which takes into account, the variation in the aerodynamic coefficients during the flight. The final analysis part deals with the control aspect of the quadrotor. The approach used for controlling the quadrotor was Integral Backstepping. The results of the autonomous take-off, hover and landing were presented as conclusion.

Further on, research done in Helwan University, Cairo, Egypt [3], found out that PD-controller with low passes filter shows poor performance when it is controlling more than one angle at the same time. The research was also validated on real time experiments. Whereas some research groups tried to use vision based input [4] to control the quadrotor helicopter. A research done in the University of Derby, [5] was very extensive in their work, as they tried to compare the results from MATLAB simulations with the CFD (Computational Fluid Dynamics) analysis results on a 3-D Solidworks model of a quadrotor.

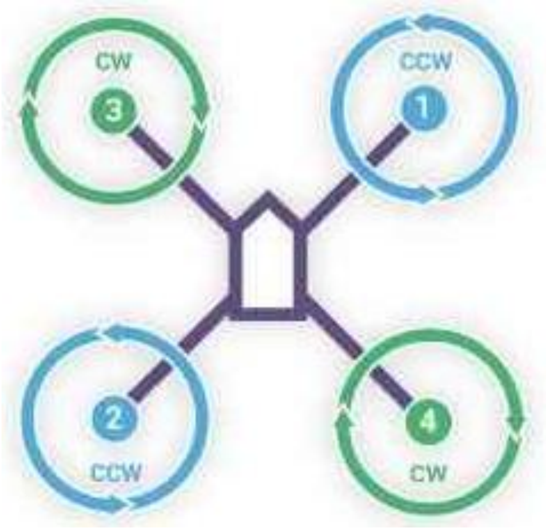
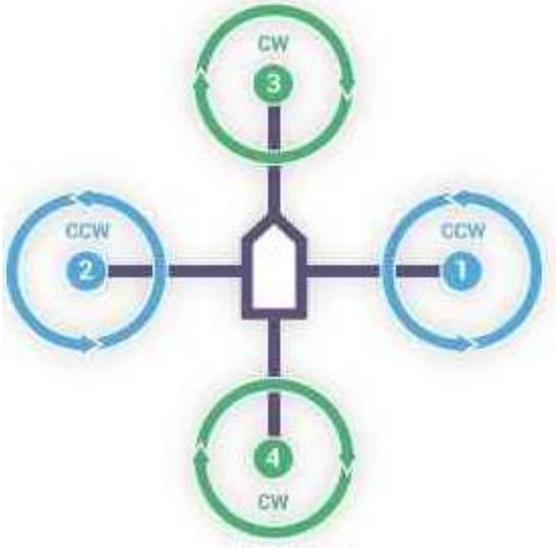
During the course of this project, topics outside of the generic quadrotor was also considered. This was done so as to get a better picture of the control aspect which is being implemented on the modern day robotic systems [8]; [9]. In these papers the importance and proper usage of MATLAB, Simulink and SimMechanics were presented in a very user friendly way.

CHAPTER 3

QUADROTOR

3.1 Quadrotor Configuration

A quadrotor can have different configuration according to its needs. The different configurations a quadcopter can have are shown in the Table 1.

 **Figure 1 Quad-X Configuration**	This configuration is considered to be the most stable of all the configurations available. It is also widely used for aerial photography.
 **Figure 2 Quad-Plus Configuration**	This type of model is generally used by the beginners because of the design being similar to a fixed wing plane. Analysis of the Quad-Plus configuration reveals that this type of model is very agile in itself.

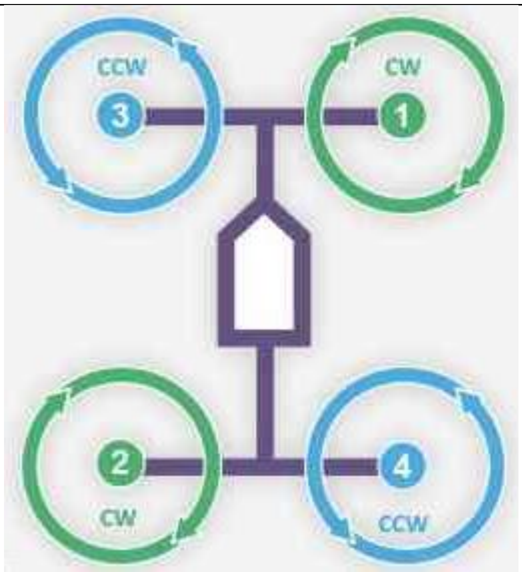
 **Figure 3 Quad-H Configuration**	Recently the H type configuration is gaining popularity in the aerial photography section, as the camera can easily be fitted in the quadcopter.

Table 1 Different configurations possible for a Quadcopter

3.2 Quadrotor Dynamic Modelling

A quadrotor has four rotors, which helps is to attain the VTOL (Vertical Take-Off and Lift) motion. These four rotors consists of four motors and four propellers. These propellers are actually two pairs, one in the clockwise direction and the other in the anti-clockwise direction. There are two main reason to have two sets of propellers having different directions:

- to counter the torque produced by the adjacent propellers
- to have an angular momentum of zero when applied equal thrust on all the motors.

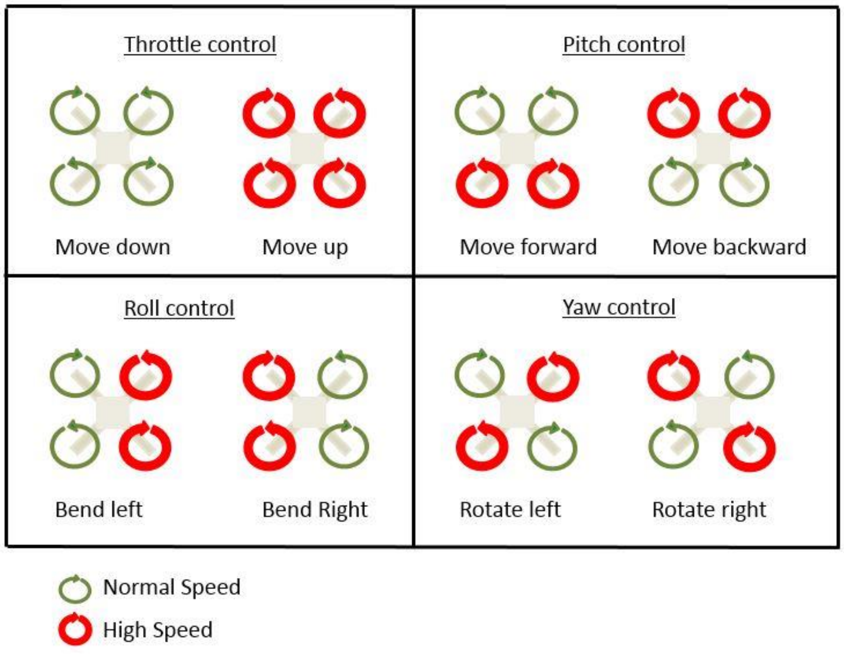


Figure 4 Illustrating the different motions achieved by a Quad-X configuration

As shown in the Figure 4, a quadrotor has four different types of motion, throttle, pitch, roll and yaw. Thus, it can be said that a quadrotor can have 3 angular motions (rotation about θ, ϕ and ψ) and 3 linear motions (along x, y and z axis), which in turn gives the quadcopter a 6-DOF (six-degree of freedom) motion.

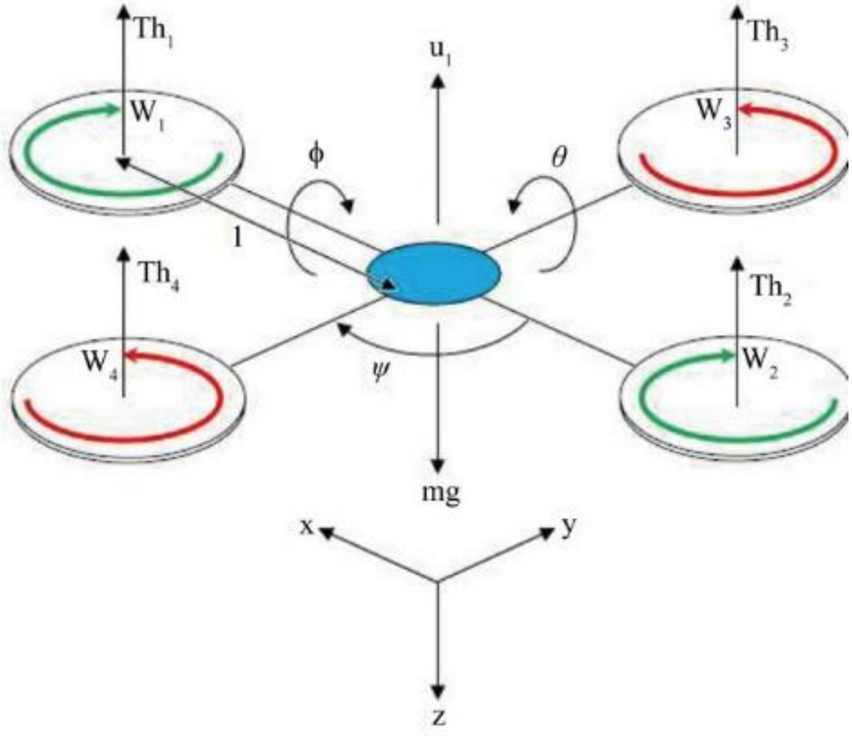


Figure 5 Dynamic model of a Quad-X configuration

3.3 Parot AR.Drone

In this research, the platform chosen is the Parot AR.Drone. The AR.Drone has 4 inrunner brushless dc motors, of rating 14.5W 28,500 RPM. The high speed rating of the AR.Drone makes it friendlier for research purposes, as the maneuverability and the response time of the drone increases with the increase in its speed rating.

The motherboard of the AR.Drone has 32-bit ARM Cortex A8 1GHz processor with 800MHz DSP TMS320DMC64x video, 1GB DDR2 RAM at 200MHz, 60 FPS QVGA vertical ground speed cameras for measurement, USB 2.0 high speed for extensions, Wi-Fi chips, HD Camera 720p 30 fps. The Inertial Measurement Unit (IMU) consists of three-axis gyroscopes 2000°/second precision, three axis accelerometers +/- 50 mg precision, three-axis magnetometers 6° precision, Pressure +/- 10 Pa precision sensors and ultrasound sensors for measurement of ground altitude. It uses Linux 2.6.32 version for all its internal computing.

It has an able Wi-Fi connection, with the help of which the user is able to get access of the drone’s internal computer with the computer or the mobile device (ground station).

Figure 6 Parot AR.Drone

CHAPTER 4
CONTROLLERS

A controller is a device monitors and physically alters the operating conditions of a given dynamical system. It all solely depends on the configuration of the system, which can be MIMO, which has multiple inputs and multiple outputs, or a SISO system, consisting of single input and single output. A MIMO requires multiple controllers to control the different processes in a system. The main purpose of including a controller in a system is to achieve the desired output signal. This is done by measuring the disparity between the desired output and the actual output value, which is the error value of the signal. This disparity or the error value is minimized by including a controller in the system.

In a quadrotor, the controlled outputs include course, roll, pitch and yaw. The controlling is done by actuating the motors at a desired speed to maintain a flight path on a safe and smooth trajectory.

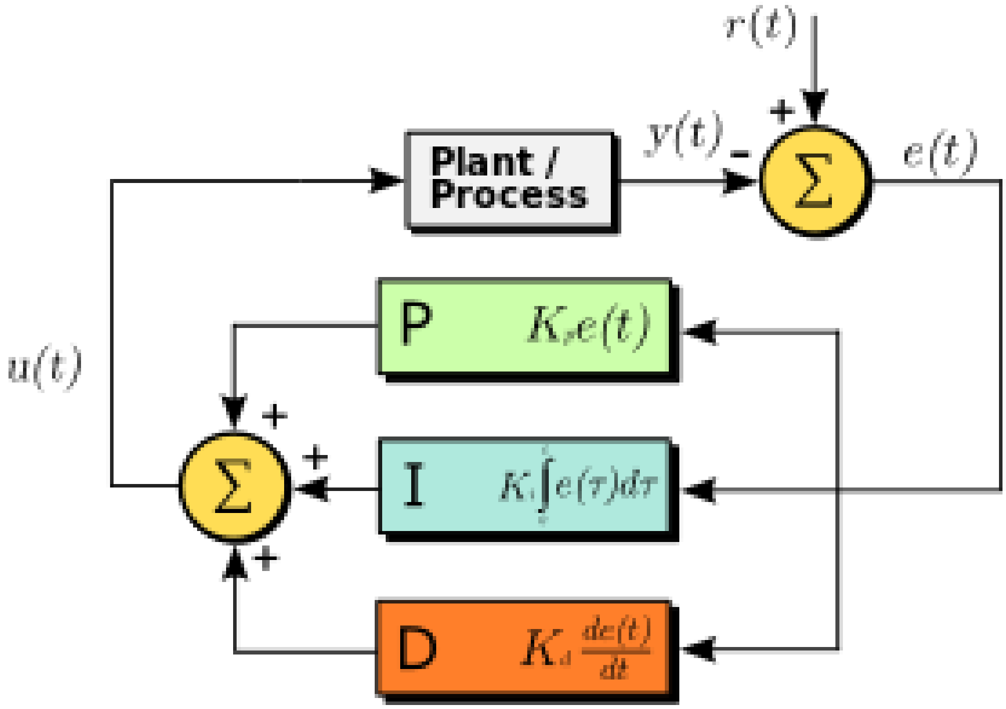


Figure 7 Closed loop for a PID controller

The conventional controllers which are generally used to control a system include the P, PI, PID controllers.

Each value of the PID controller has some major significance.

Proportional term

The proportional term produces an output value that is proportional to the current error value. The proportional term is given by:

$$Pout = Kp\, e(t) \tag{1}$$

It accounts for the present value of the error, and should contribute the bulk of the output change.

Integral term

The integral term contributes to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The integral term is given by:

$$Iout = Ki \int_0^t e(\tau) d\tau \tag{2}$$

It accounts for the past values of error and accelerates the movement of the process towards set point and eliminates the residual steady-state error.

Derivative term

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain. The derivative term is given by:

$$Dout = Kd\, \frac{de(t)}{dt} \tag{3}$$

It accounts for the possible future values of error. The derivative action predicts the system behavior and thus improves the settling time and thus the stability of the system.

Assigning u(t) as the controller output, we get the final equation as:

$$u(t) = MV(t) = Kp\, e(t) + Ki \int_0^t e(\tau)\, d\tau + \; Kd\, \frac{de(t)}{dt} \tag{4}$$

Where

K_p: Proportional gain, a tuning parameter

K_i: Integral gain, a tuning parameter

K_d: Derivative gain, a tuning parameter e: Error = SP - PV

SP: Set Point

PV: Process Variable

t: Time or instantaneous time (the present)

τ: Variable of integration; takes on values from time 0 to the present t.

CHAPTER 5
SIMULATION AND ANALYSIS

5.1 Solidworks Model

The AR.Drone model was modelled and assembled on Solidworks platform. The AR.Drone modelled consisted of 3 major parts:

- Crosslink,
- Arms, and
- Motor Assembly

The model of the AR.Drone is shown in Figure 8.

Figure 8 Solidworks model of AR.Drone

The motor assembly is the part which will be actuated to get the desired results. The motor assembly consists of the propellers, motor spool, gears and the motors.
Motor assembly is shown in the Figure 9.

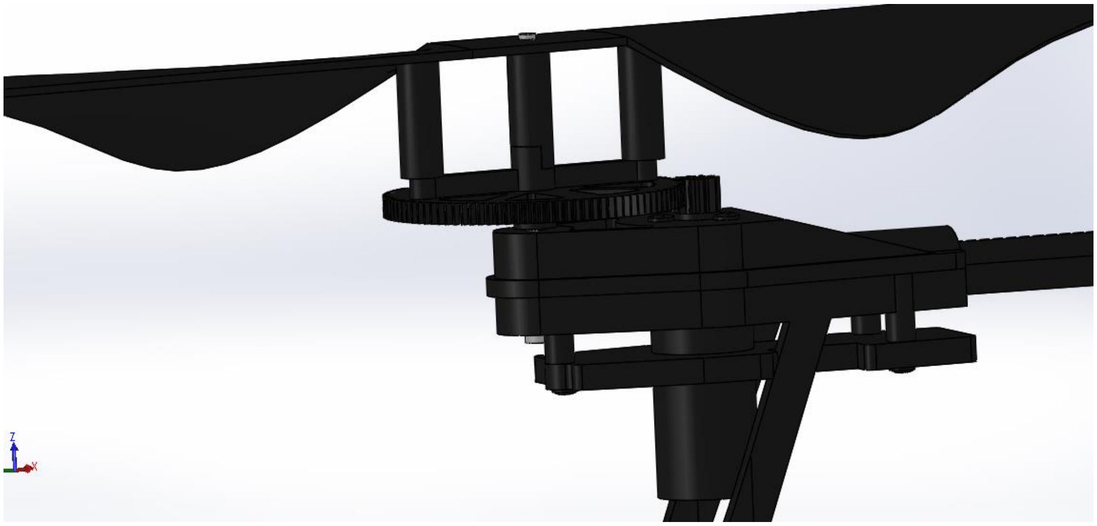


Figure 9 Motor Assembly model in Solidworks

5.2 SimMechanics Model

SimMechanics environment is a single arena for simulating 3D electro-mechanical systems with physical signals and control algorithms in Simulink. Designs can be imported from various CAD Softwares for simulation using MATLAB environment.

The Solidworks model is then imported to the MATLAB environment using the SimMechanics export link, which is available in Solidworks. An .xml file is generated, which can then be executed on MATLAB for further simulations.

SimMechanics link converts all constraints and mates into proper joints between the parts of the model. As the model is developed by the system, there can be few extra joints and a few extra links. After some inspection, the final Simulink model (Figure 10) is achieved on which further simulation will be carried on.

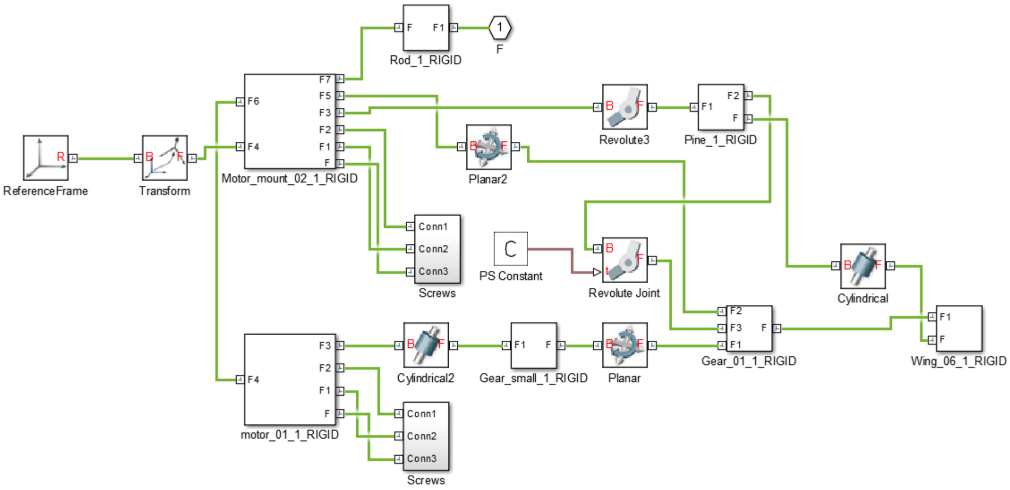


Figure 10 Simulink model of a motor assembly in MATLAB

The SimMechanics platform is capable of portraying the physical effects on a system which is depicted by the Simulink model. As can be seen in the Figure 11, the physical model is generated via the Simulink model.

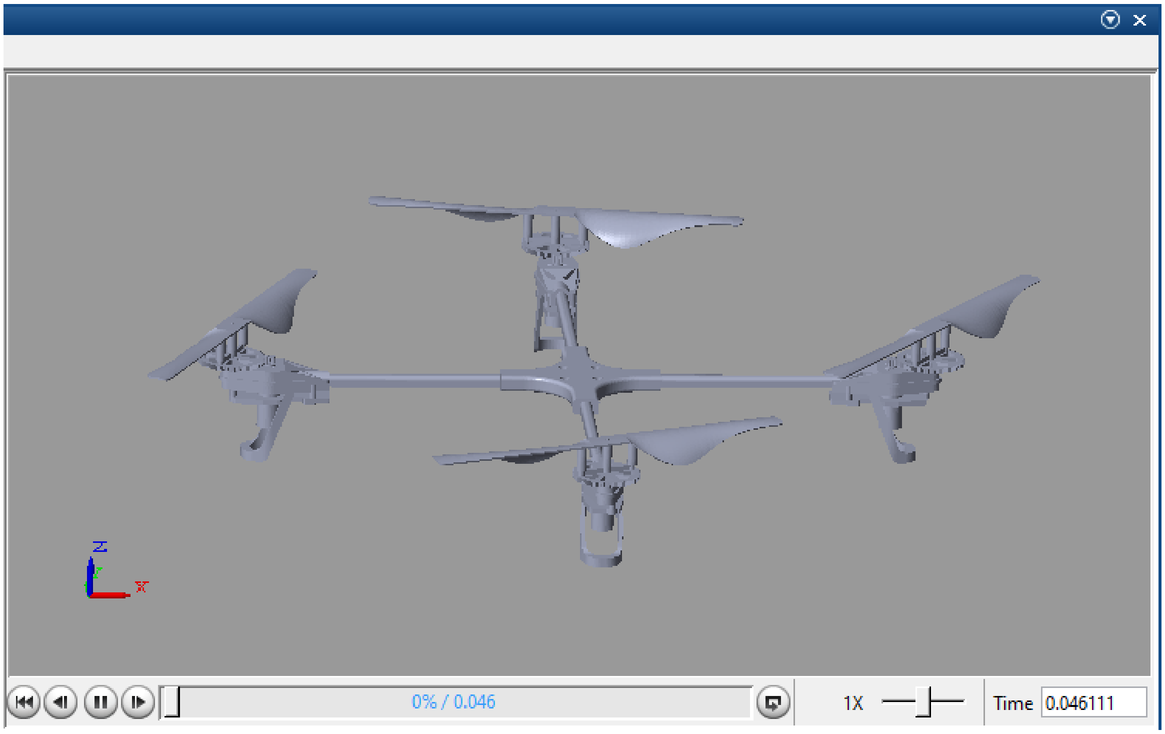


Figure 11 Physical Model Generated From the Simulink Model

After some simulation results it was found that the AR.Drone CAD model was taking excessive amount of time to complete its simulation. As a result of which the model for further analysis was changed, and the DJI F-450 was taken as the new platform. It was found that because of the simplicity in design of the model [Figure 12], the simulation time for this model was rather appropriate, than on the AR.Drone model.

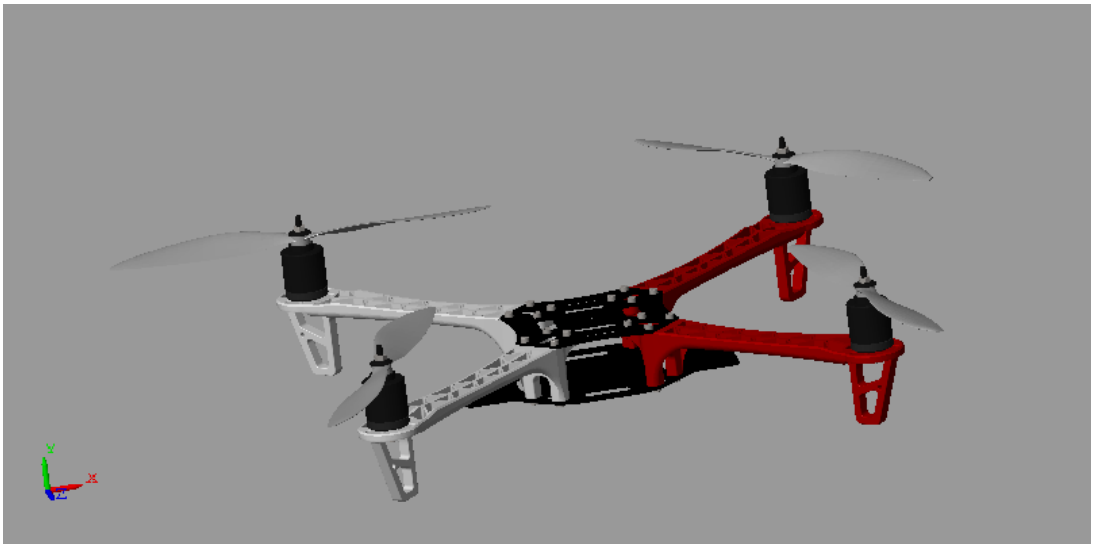


Figure 12 Physical Model Generated for DJI F-450 Model

5.3 Updated SimMechanics Block Diagram

As can be seen in the figure [Figure 13], the updated version of the SimMechanics block diagram mainly consists of 5 blocks, namely Inputs, GetErrors, Controller, Base and Signal, and Quad Model.

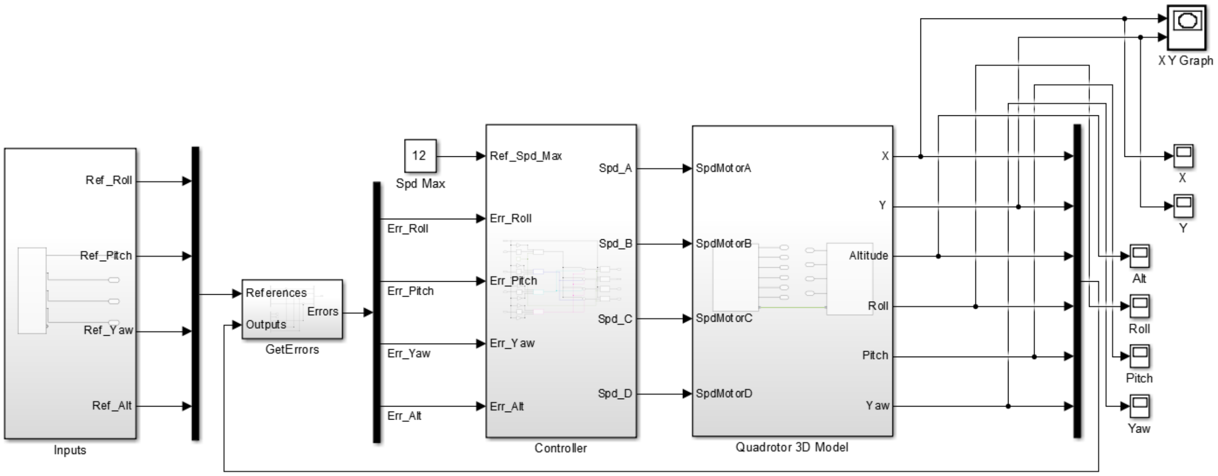


Figure 13 SimMechanics Simulink Model Generated for DJI F-450

1. <u>Inputs</u>

The Inputs block is essentially used to input signals to the system via a signal builder in Simulink. The signals are given in the following order to the quadrotor, Altitude, Yaw (rotation about the z-axis), Pitch and Roll. This block thus gives four different reference values as its output.

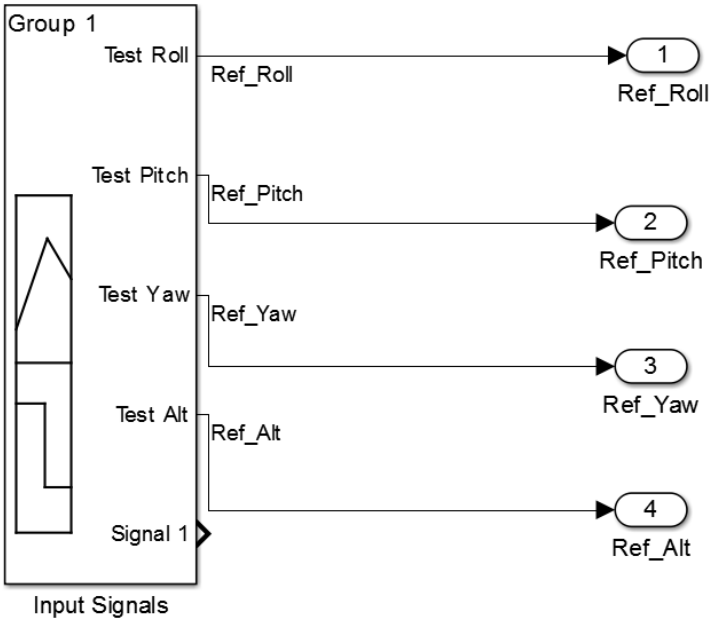


Figure 14 Inputs Block

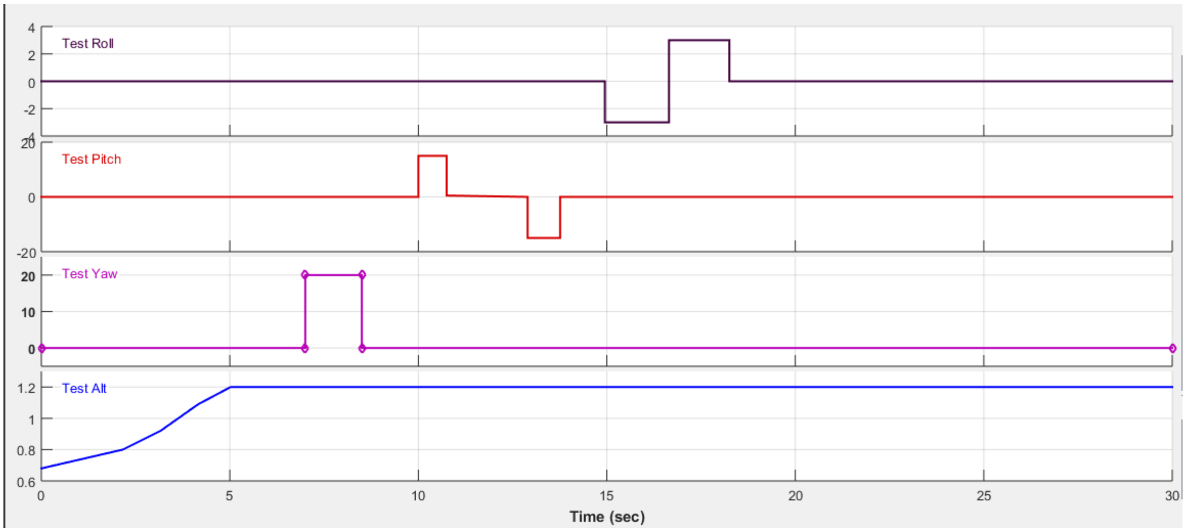


Figure 15 Input Signal Block

2. GetErrors

This block acts as the summing block for the system. It gives the error value from the reference signal, by subtracting the final output value obtained by the system. A total of two de-multiplexers and a single multiplexer is used in this block.

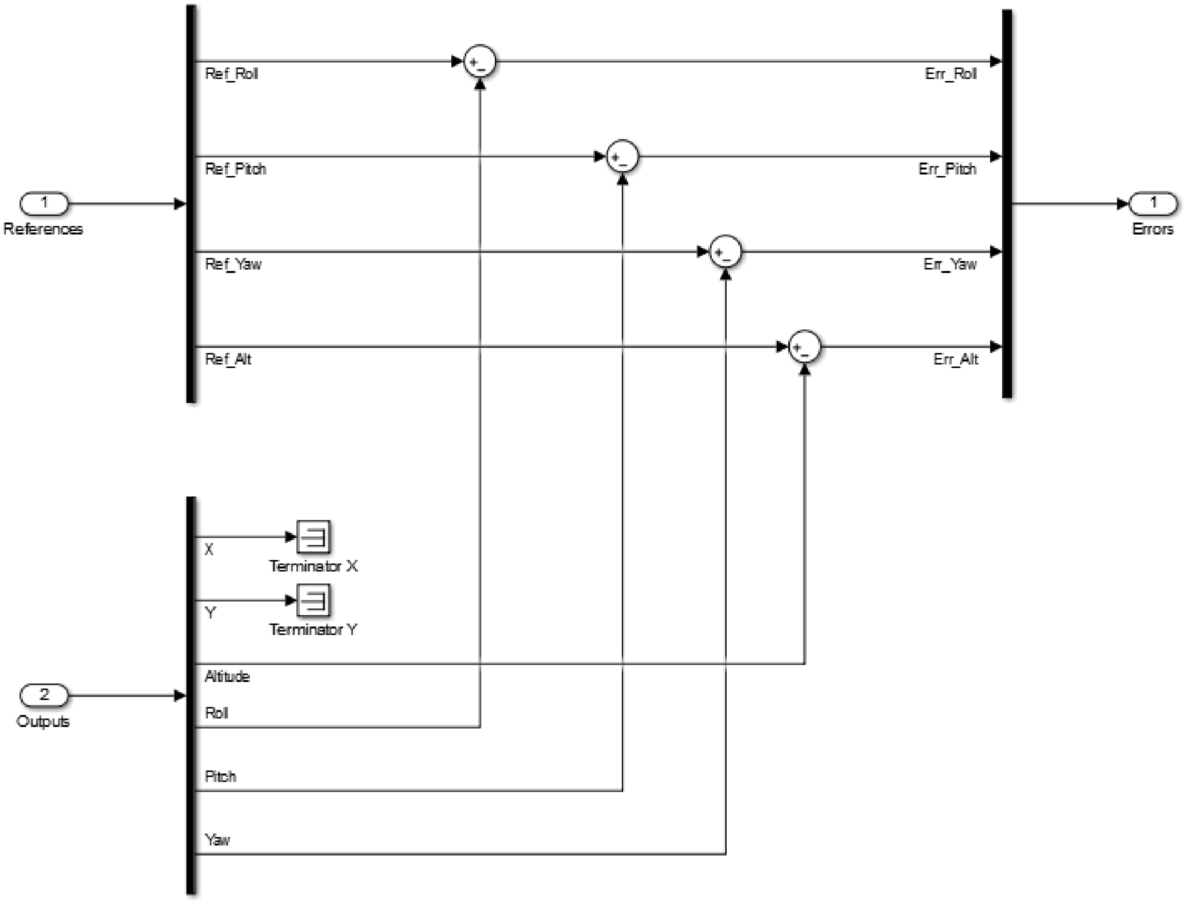


Figure 16 GetError Simulink Model

3. Controller

The error values along with a constant value for the maximum speed is provided to the Controller block. It uses two PID controllers for altitude control and yaw control. PD controller is used for the controlled motion of pitch and roll. A saturation dynamic block is used to limit the controlled value. The whole controller can be realized in the figure [Figure 17].

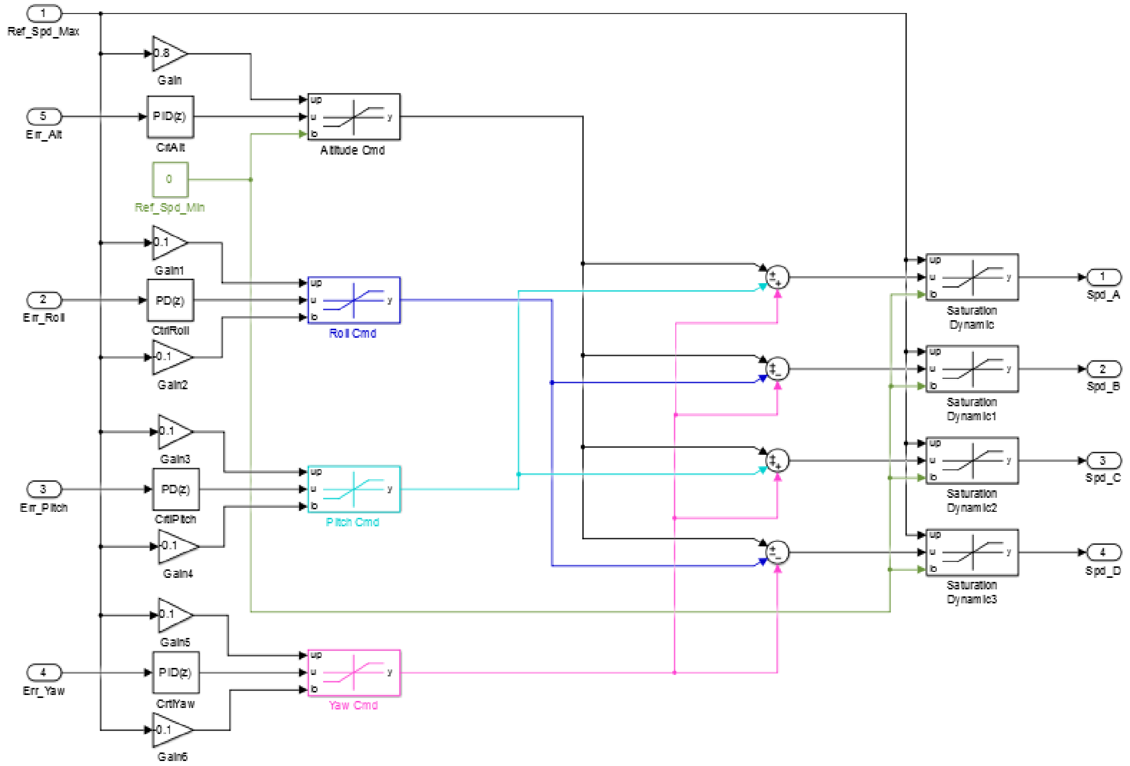


Figure 17 Complete Controller Model

4. Base and Sensors

The Base and Sensors block mainly consists of the two things which the name suggests, the base for the quadcopter and the sensors giving the angle and the position values.

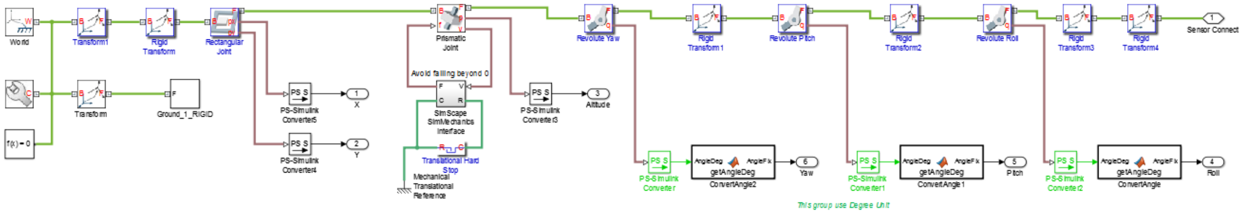


Figure 18 Base and Sensors

5. Quad Model

This block mainly consists of the SimMechanics model of the quadrotor. As can be seen in the Figure 19 the four motors are connected to a group, which consists of all non-rotating components of a quadcopter (such as screws and arms). The four motors are fed with the speed derived from the controller.

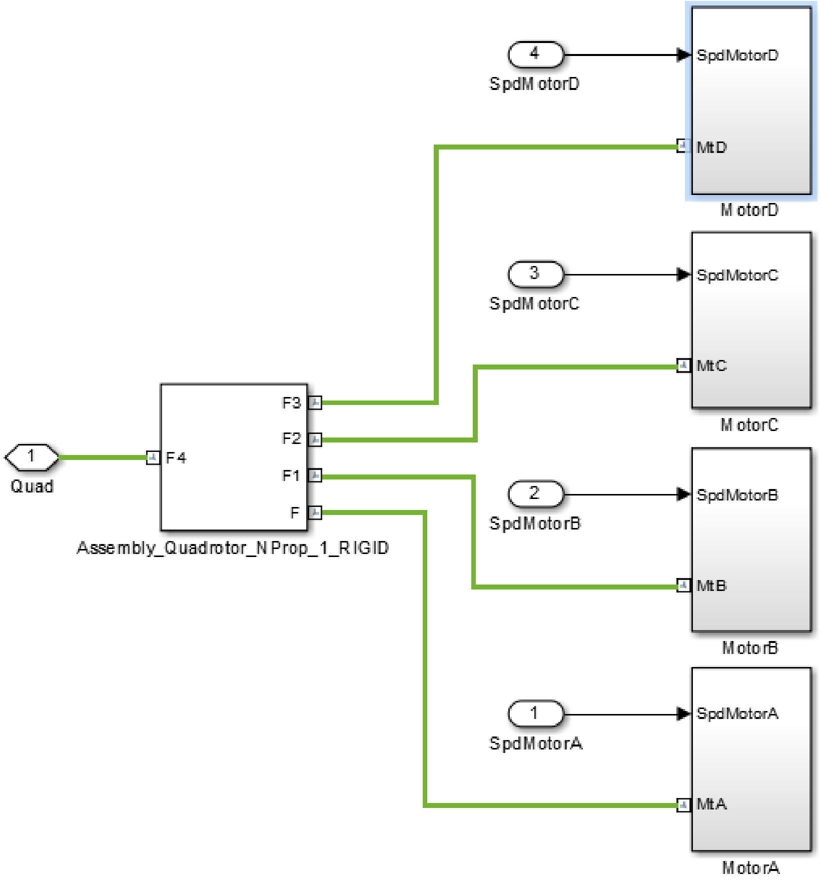


Figure 19 Quadcopter Simulink Model

The motors have a revolute joint interface with the non-rotating parts, which is driven externally by the speed derived from the controllers and multiplied by the gain value. The gain value is determined by the motor numbers. In this model, motors A and C have been given a gain value of “1” and motors B and D have been given a gain value of “-1”. This is equivalent to the physical model of a quadcopter, where the opposite motors spin in the same direction. Whereas the adjacent motors spin in the opposite direction. This is done to maintain a “zero angular momentum” at the centre of the system, and to cancel out the torque developed by the 4 motors.

As we have seen that the revolute joint is being actuated by the speed provided, it also has a sensing part, which helps in minimizing the error. The sensor part of the revolute joint consists of a Matlab function block. This function block has a small line of code which helps to maintain two things; minimizing the thrust of the motors as it starts to reach the desired value, and a check to see if the thrust is always positive. This sensor value is then fed to the External Force and Torque block via a Simulink to PS convertor and provided to the motor assembly which majorly consists of the propellers, motor top and motor base.

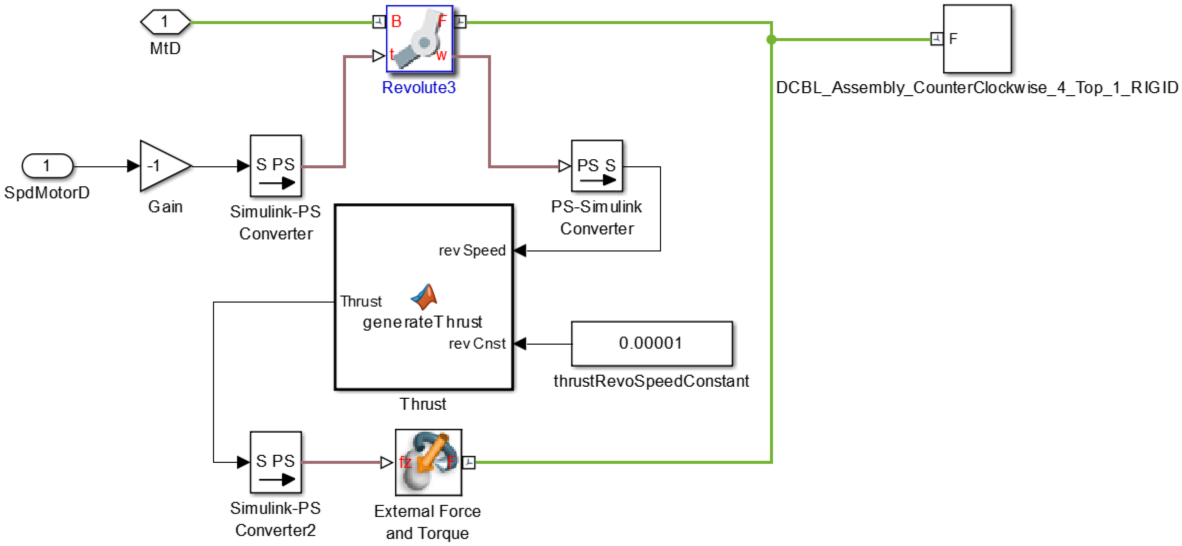


Figure 20 Motor Assembly in Simulink

CHAPTER 6

CONTROLLER DESIGN

A controller is essentially used in a system to achieve the desired output value with a reduced amount of error. From the different varieties of controllers available, in this simulation model two types of controllers are used:

- PID Controller
- PD Controller

In this system the PID controller is used for altitude control and yaw control. Whereas the PD controller is used to control the roll and the pitch motion of the quadcopter.

As can be seen in the Figure 21, there are 5 inputs to the Controller block (Ref_Spd_Max, Err_Alt, Err_Roll, Err_Pitch and Err_Yaw), and 4 outputs from this block (Spd_A, Spd_B, Spd_C and Spd_D). These speed values generated from this block is fed to the motors for the desired motion.

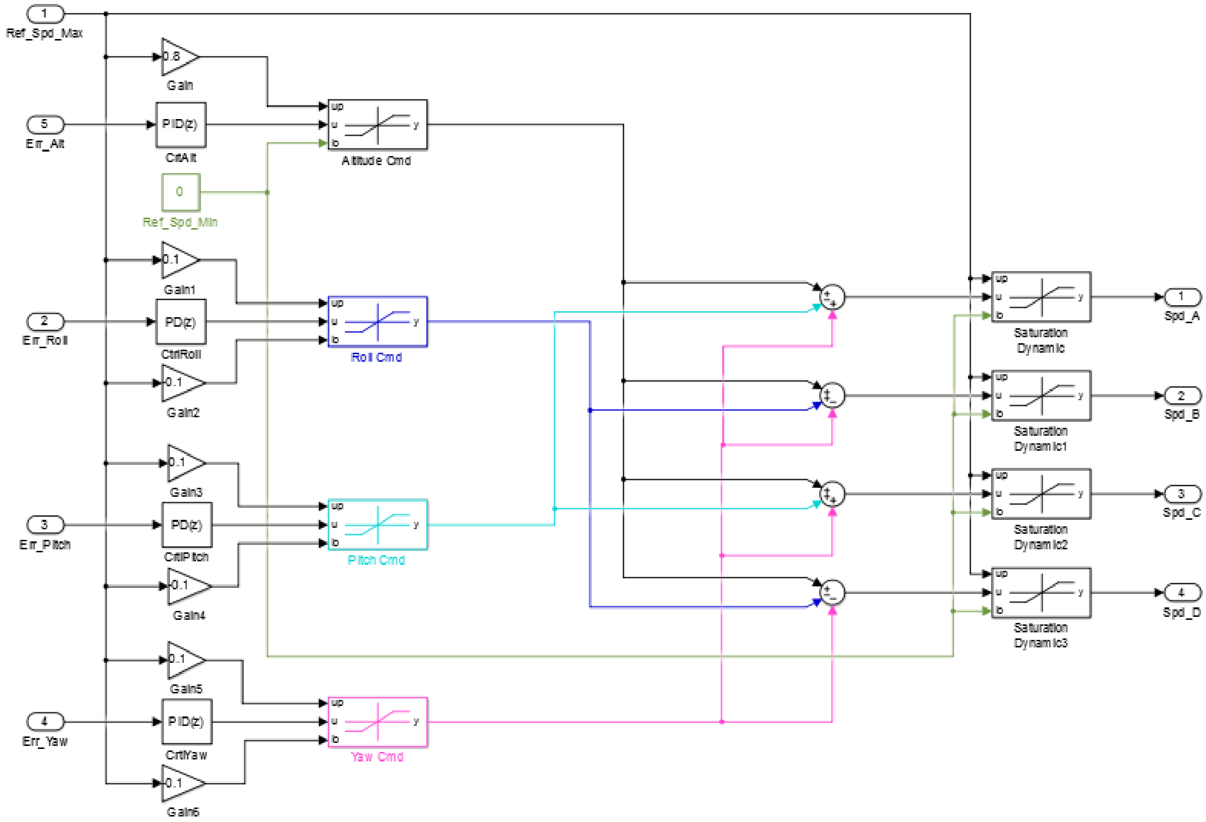


Figure 21 Complete Controller Model

The DJI-F450 model considered for this simulation is taken to be in plus (+) configuration, which means the Motor A is the front motor of this quadcopter and the Motor C acts at the rear part of the quadcopter. This is the reason why the pitch values have been fed only to motors A and C, whereas the roll values have been fed to motors B and D.

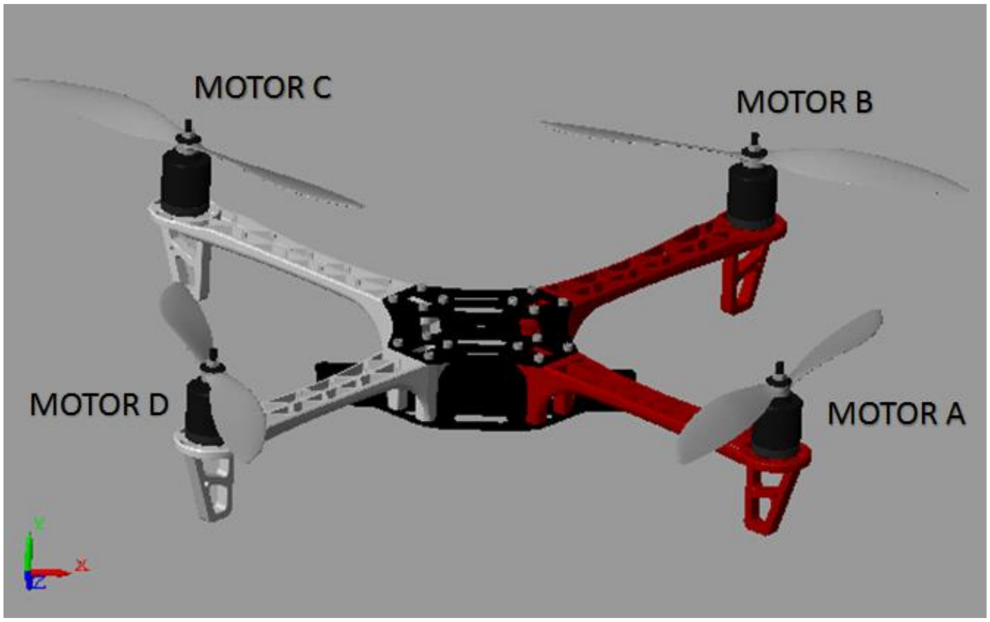


Figure 22 Motor Numbering in DJI-F450

CHAPTER 7

ANALYSIS AND SIMULATIONS

7.1 Tuning the Controllers

Incorporating a mere controller in the system is not enough to get the desired output values. The controller should also be tuned properly according to the requirement. The tuning can be done through many ways, manually, using verified mathematical techniques or using the automatic tuning in the MATLAB environment. Tuning for this system is done manually.

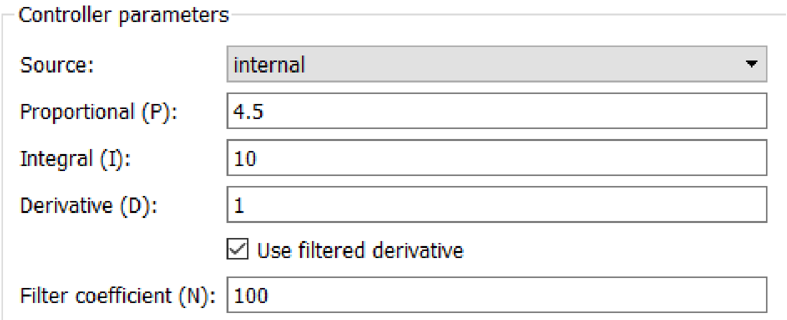
All the four controllers, function in the discrete time domain. The PID controllers have a Forward Euler integrator method, whereas all the controllers have the same Forward Euler filter method. Forward Euler method is used as it provides a better stability when the sampling times are small, as for this system the sampling time is taken as 0.001 sec.

The following values were obtained for the motion control.

Altitude Control

A PID controller is used to control the altitude of the quadcopter. The following values were used as the controller parameters:

Proportional (P) = 4.5
Integral (I) = 10
Derivative (D) = 1
Filter Coefficient (N) = 100



Controller parameters

Source: internal
Proportional (P): 4.5
Integral (I): 10
Derivative (D): 1
☑ Use filtered derivative
Filter coefficient (N): 100

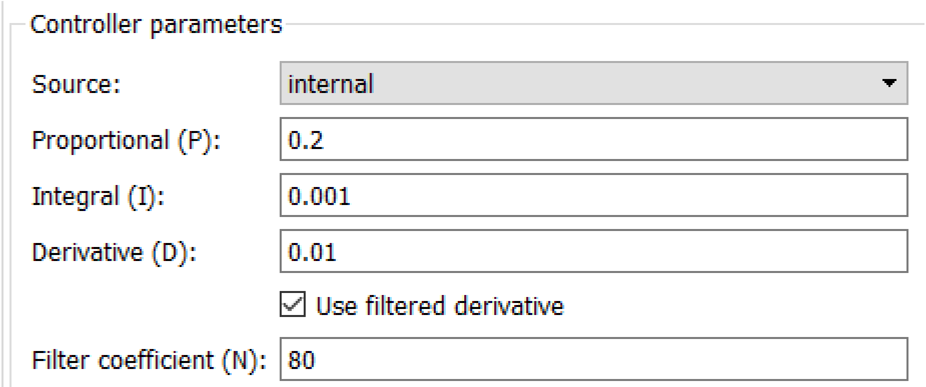
Figure 23 Controller Parameters for Altitude Control

Yaw Control

A PID controller is used to control the yaw of the quadcopter. The controlled value of speed is fed to all the four motors as all the four motors are responsible for the yaw motion of a quadcopter. The following values were used as the controller parameters:

Proportional (P) = 0.2

Integral (I) = 0.001
Derivative (D) = 0.01
Filter Coefficient (N) = 80



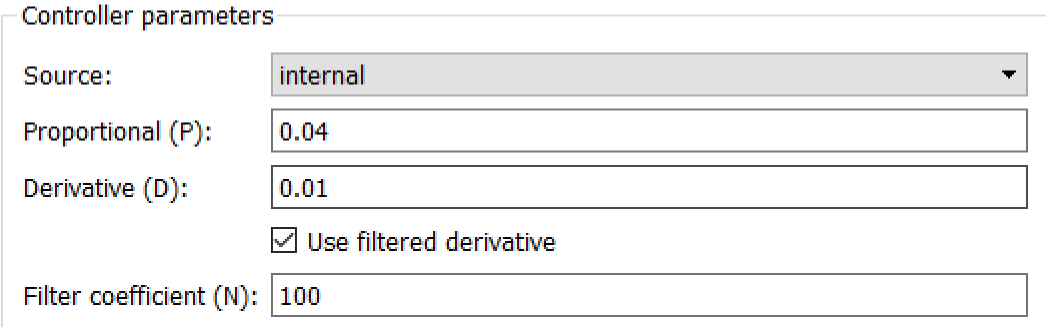
Controller parameters
Source: internal
Proportional (P): 0.2
Integral (I): 0.001
Derivative (D): 0.01
☑ Use filtered derivative
Filter coefficient (N): 80

Figure 24 Controller Parameters for Yaw Control

Roll Control

A PD controller is used to control the roll of the quadcopter. The following values were used as the controller parameters:

Proportional (P) = 0.04
Derivative (D) = 0.01
Filter Coefficient (N) = 100



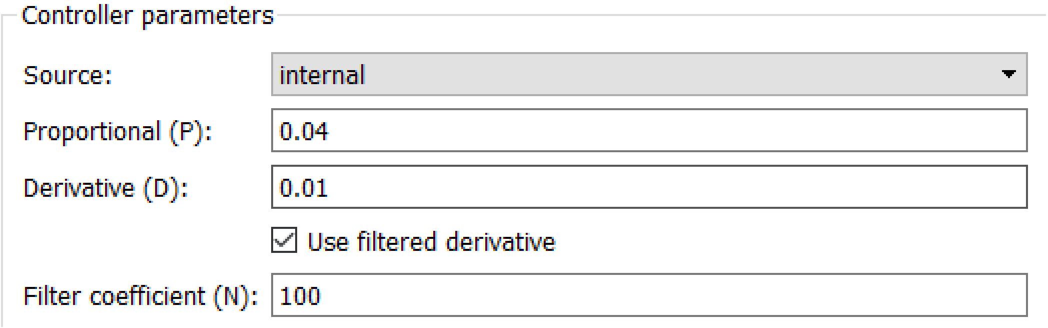
Controller parameters
Source: internal
Proportional (P): 0.04
Derivative (D): 0.01
☑ Use filtered derivative
Filter coefficient (N): 100

Figure 25 Controller Parameters for Roll Control

Pitch Control

A PD controller is used to control the pitch of the quadcopter. The following values were used as the controller parameters.

Proportional (P) = 0.04
Derivative (D) = 0.01
Filter Coefficient (N) = 100



Controller parameters

Source: internal

Proportional (P): 0.04

Derivative (D): 0.01

☑ Use filtered derivative

Filter coefficient (N): 100

Figure 26 Controller Parameters for Pitch Control

7.2 Simulation Results

Altitude Control

Following graph is obtained for the altitude control. As can be seen in the Figure 27, there is a little deviation from the desired value between 5 to 10 seconds. This happens because of the yaw motion, but it is further stabilized to the desired value.

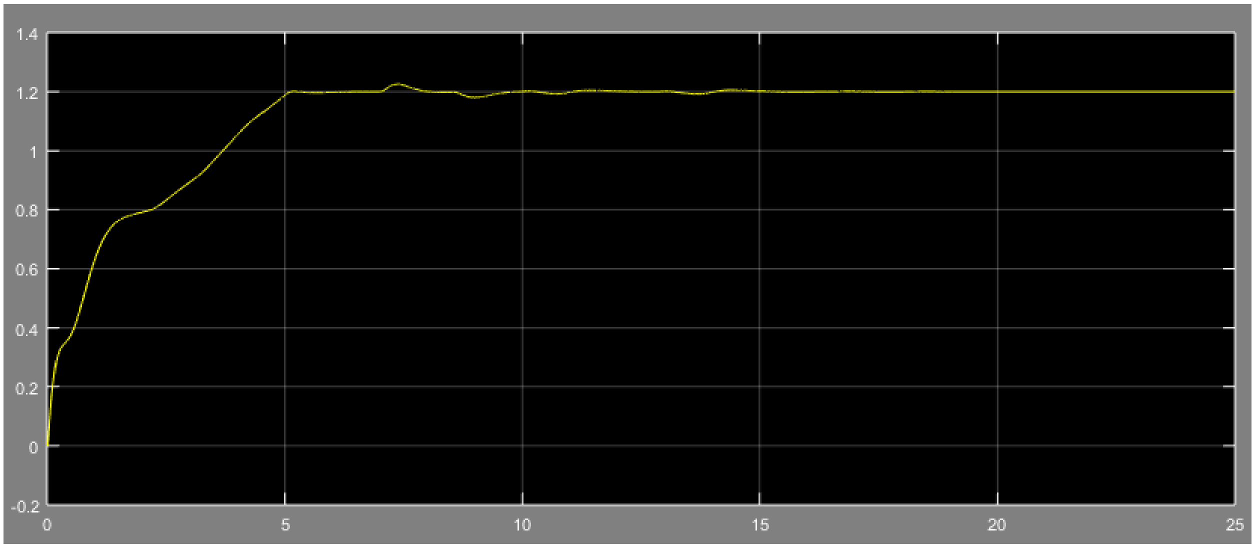


Figure 27 Altitude Graph

Yaw Control

The PID controller works exceptionally good for the yaw control as can be seen in the following figure 28. There is a gradual increase in the yaw direction which reaches a steady value and then the yaw direction is changed and the yaw angle becomes zero again.

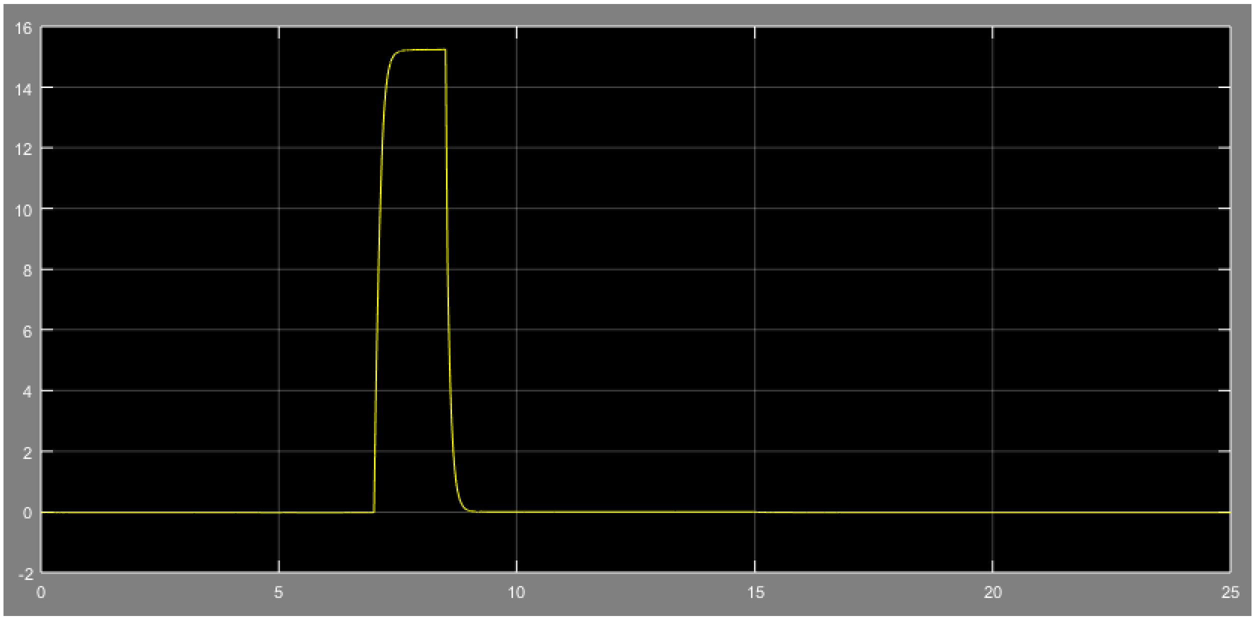


Figure 28 Yaw Angle graph

Roll Control

A PD controller is used to control the roll of the quadcopter. The following graph was obtained when the simulations were done.

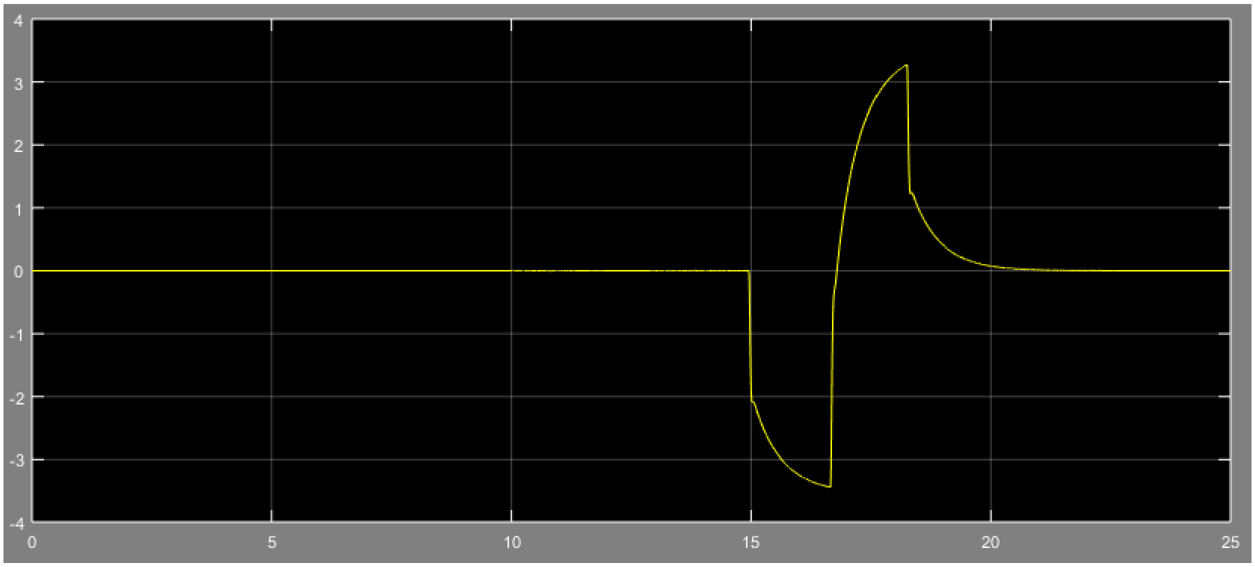


Figure 29 Roll Angle graph

Pitch Control

A PD controller is used to control the pitch of the quadcopter. The following graph was obtained when the simulations were done.

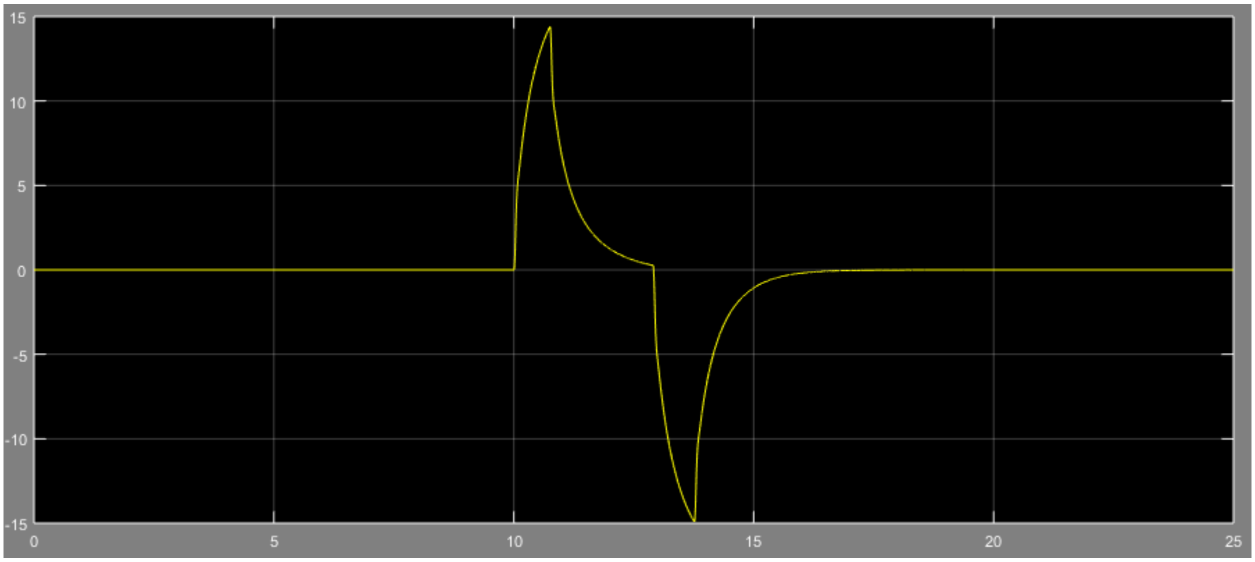


Figure 30 Pitch Angle graph

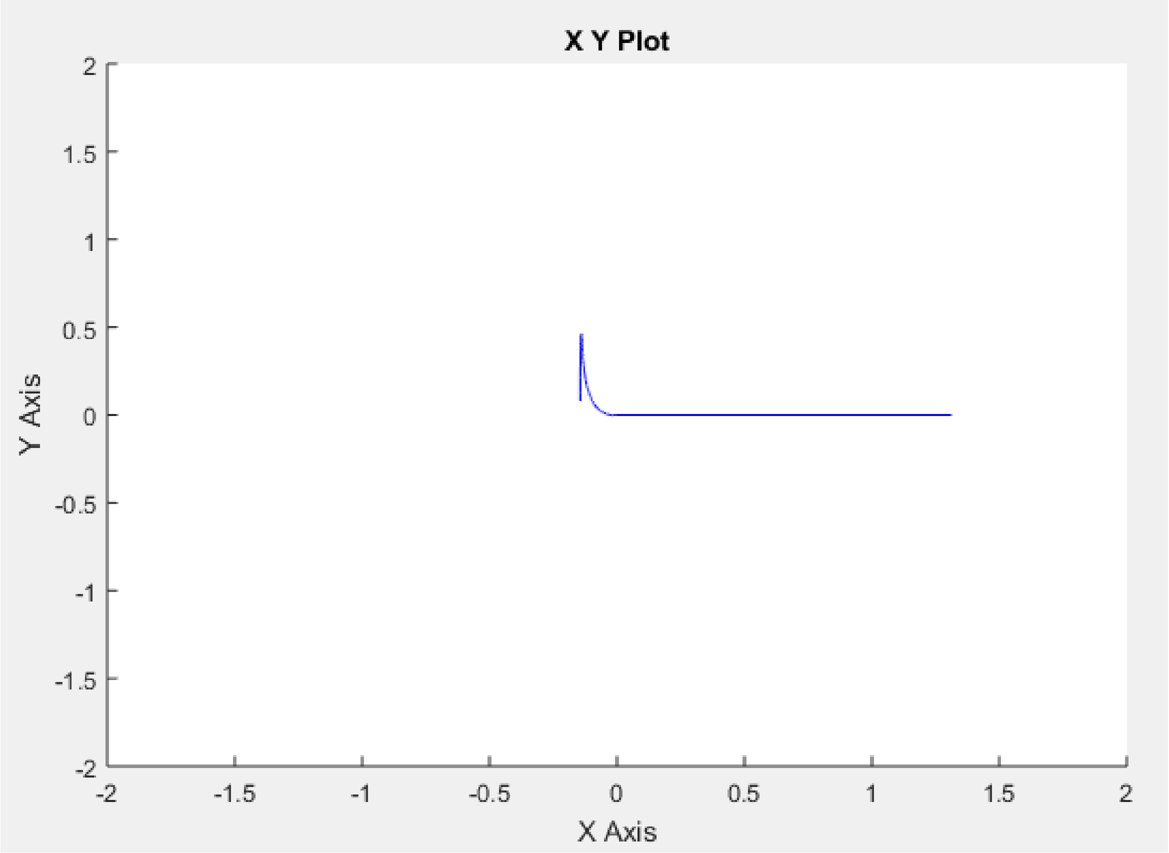


Figure 31 X-Y Plot the Quadcopter Motion

CHAPTER 8

SUMMARY AND CONCLUSION

The first part of this report deals with the basics of the quadrotor. Its modelling, flight dynamics and the various components of the Parot AR.Drone. Different types of configurations that are possible in a quadrotor is being explained. Further on we dwell on the problem statement, which deals with the designing of a controller for a stable flight. Before designing a controller, the different types of conventional controllers are studied so as to get a broader approach.

The second section includes the designing of a Simulink model of a quadrotor. Designing the CAD model of a quadrotor in the SolidWorks platform and then transforming the CAD model into a Simulink Model via a SimMechanics link. In beginning of the project Parot AR.Drone was considered as the test subject for the simulation, but due to various reasons it was dropped and DJI's quadcopter model, DJI F-450, was chosen as the test subject. SimMechanics model of the quadcopter is prepared and further simulations are run for the system.

A closed loop system is designed for the quadcopter model, which includes two PID controllers for Altitude and Yaw control, and two PD controllers for the Pitch and Roll control. The system is fed with appropriate input signals and simulated. The controller is manually tuned to get the optimum reading for Altitude, Yaw. Pitch and Roll Control. As the controller is properly tuned, as can be seen in the graphs, a stable flight is achieved by the quadcopter.

REFERENCES

[1] Cavalcante Sa, Rejane, et al. "Construction and PID control for stability of an unmanned aerial vehicle of the type quadrotor." *Robotics Symposium and Competition (LARS/LARC), 2013 Latin American*. IEEE, 2013.

[2] Bouabdallah, Samir, and Roland Siegwart. "Full control of a quadrotor." *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*. IEEE, 2007.

[3] Mahmoud, Osama E., Magdy R. Roman, and John F. Nasry. "Linear and Nonlinear stabilizing control of Quadrotor UAV." *Engineering and Technology (ICET), 2014 International Conference on*. IEEE, 2014.

[4] Erginer, Bora, and Erdinç Altuğ. "Modeling and PD control of a quadrotor VTOL vehicle." *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007.

[5] Poyi, Gwangtim Timothy, et al. "Validation of a quad-rotor helicopter Matlab/Simulink and solidworks models." *Control and Automation 2013: Uniting Problems and Solutions, IET Conference on*. IET, 2013.

[6] Gong, Xun, et al. "Trajectory tracking control of a quad-rotor UAV based on command filtered backstepping." *Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on*. IEEE, 2012.

[7] Jianping, Li, et al. "Control system design and simulation of an aerial three-axis isp system based on SimMechanics." *Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on*. Vol. 1. IEEE, 2011.

[8] Cretescu, Nadia, Mircea Neagoe, and Radu Saulescu. "Kinematic Modelling and VR Simulation of a 3DOF Medical Parallel Robot with One Decoupled Motion." *Advanced Materials Research*. Vol. 837. 2014.

[9] Stan, Sergiu-Dan, et al. "Kinematics analysis, design, and control of an Isoglide3 Parallel Robot (IG3PR)." *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*. IEEE, 2008.