UNIVERSITY OF SALFORD

School of Science, Environment and Engineering

Parrot Mambo Minidrone Flight Control and Simulation Using MATLAB and Simulink

Trimester Three Dissertation Project Report

DATE: 30th October 2020

NAME: PRAJYOT TADAS

STUDENT ID: @00570381

COURSE TITLE: M.Sc. ADVANCED CONTROL SYSTEMS

COURSE CODE: MST/CY/F

SUPERVISOR: DR. ANTHONY H. JONES

University of Salford
School of Science, Environment and Engineering

Name of Student: Prajyot Tadas

Course Code: MST/CY/F

Title of Project: Parrot Mambo Minidrone Flight Control and Simulation Using MATLAB and Simulink

I certify that this report is my own work. I have properly acknowledged all the materials that has been used from other sources, references, etc.

Signature of student: 

Date: 30th October 2020

Official Stamp:

Submission date (to be entered by relevant school office staff):

Abstract

The focus on the multi-copter is increasing every day. Minidrones are widely used as they are not only fast quadcopters with high manoeuvrability but also excellent research platforms for indoor usage. The advancement in its performance is important to keep up with modern civilisation. In this report a miniature unmanned aerial vehicle (UAV) is presented and the control aspect of the mini drone is studied, and the results are simulated using a software.

The miniature unmanned aerial vehicle (UAV) used is the Parrot Mambo mini drone quadcopter. The software used for studying and simulating the results is Mathworks MATLAB and Simulink. The main reason for using MATLAB is that it is easy to learn, and it has the Support Package for Parrot minidrones. The support package is very useful as it has everything required for the flight control part already built into it.

The report also details the development and history of Proportional, Integral and Derivative (PID) controllers and PID tuning methods. The main advantage of using the PID controllers is that it allows for autonomous flights of these quadcopters. A simulated model of the Parrot Mambo quadcopter is used to design PID control laws for roll, pitch, yaw angle and altitude. The primary purpose of this report is to test different PID gains within the flight control system (FCS) block and the estimator block for different sets of coordinates to obtain a stable and successful simulation so that it can be implemented in real life practical environments.

Acknowledgements

This project would not have been possible without the resources and aid from University of Salford. I would sincerely like to thank my supervisor, Dr Anthony H. Jones for giving me such guidance, time and attention. The project was possible because of the knowledge that Dr Jones have on MATLAB and Simulink along with his experience of working on Parrot Mambo minidrone. It was his support and teachings that helped me complete this project.

I would like to thank Mr. Daniel Lewis whose research was the only major reference available with me during the pandemic COVID-19. I would also like to thank my family for allowing me to study abroad and helping me to pursue my studies and ambitions. Finally, my gratitude goes to my friends and classmates for the extra support and motivation throughout my academic year.

Contents

List of Figures

Figure 42: Path Planning block showing set points X= 0; Y=1; and Z= -1.1

Figure 43: Path Planning block showing set points X= 1; Y=0; and Z= -1.1

List of Graphs

Graph 17: Inspecting the gravity feedforward/equilibrium thrust PD gains yaw output against time for P= 0.8 and D= 0.41

Graph 18: Inspecting the XY-to-reference-orientation PD gain Z axis output against time for P= [-0.29, 0.29] and D= [0.1, -0.1]

Graph 19: Inspecting the XY-to-reference-orientation PD gain X axis output against time for P= [-0.29, 0.29] and D= [0.1, -0.1]

Graph 20: Inspecting the XY-to-reference-orientation PD gain Y axis output against time for P= [-0.29, 0.29] and D= [0.1, -0.1]

Graph 21: Inspecting the XY-to-reference-orientation PD gain yaw output against time for P= [-0.29, 0.29] and D= [0.1, -0.1]

Graph 22: Inspecting the XY-to-reference-orientation PD gain Z axis output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]

Graph 23: Inspecting the XY-to-reference-orientation PD gain X axis output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]

Graph 24: Inspecting the XY-to-reference-orientation PD gain Y axis output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]

Graph 25: Inspecting the XY-to-reference-orientation PD gain yaw output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]

Graph 26: Inspecting the XY-to-reference-orientation PD gain Z axis output against time for P= [-0.15, 0.15] and D= [0.18, -0.18]

Graph 27: Inspecting the XY-to-reference-orientation PD gain X axis output against time for P= [-0.15, 0.15] and D= [0.18, -0.18]

Graph 28: Inspecting the XY-to-reference-orientation PD gain Y axis output against time for P= [-0.15, 0.15] and D= [0.18, -0.18]

Graph 29: Inspecting the XY-to-reference-orientation PD gain yaw output against time for P= [-0.15, 0.15] and D= [0.18, -0.18]

Graph 30: Output plots for default gains of set points X= 0.7; Y= 0; and Z= -0.7

Graph 31: Output plots for new gains of set points X= 0.7; Y= 0; and Z= -0.7

Graph 32: Output plots for default gains of set points X= 0.7; Y= 0.7; and Z= -0.7

Chapter 1: Introduction

An unmanned aerial vehicle (UAV) (also known as uncrewed aerial vehicle commonly known as Drone) is an aircraft without human pilot on board. UAVs are component of an unmanned aircraft system (UAS); which include a UAV, a ground-based controller, and a system of communications between the two. The flight of UAVs may operate with various degrees of autonomy; either under remote control by human operator, autonomously by onboard computers or piloted by an autonomous robot. UAVs were originally designed for missions considered to be too dull, dirty or dangerous for humans. While the UAVs were mostly used in military operations their use in commercial, scientific, recreational and agricultural applications is rapidly rising. The other applications for UAVs include surveillance, product deliveries, aerial photography, infrastructure inspections and drone racing [1].

MATLAB (an abbreviation of "matrix laboratory") is a proprietary multi-paradigm programming language and numerical computing environment developed by Mathworks. MATLAB allows matrix manipulations, plotting functions and data, implementation of algorithms, creation of user interfaces and interacting with programs written in other languages [2]. Simulink is a MATLAB based graphical programming environment for modelling, simulating and analysing multi domain dynamical systems. Its primary interface is a graphical block diagram tool and a customisable set of block libraries. It offers tight integration with the rest of MATLAB environment and can either drive MATLAB or be scripted from it [3]. For model-based design, automatic control and digital signal processing Simulink is widely used and hence the main reason of using it for this project. We use the Simulink support package for Parrot Mambo mini drone which is available and makes the control design of this project very easy.

A Proportional, Integral and Derivative (PID) controller is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control [4]. PID controllers are widely used because of their accuracy and cheap price. Once a PID controller is designed then its further tuning doesn't require skilled personnel. PID controllers can be used to play with mathematical models by tuning it with trial and error methods before real life implementations. In [5], the author obtains a state space model by using several phenomenon like gyroscopic effects for rigid bodies, propellers and rotors. The obtained model is then used to design PID control laws for roll, pitch, yaw angles and altitude.

As helicopters are controlled using swashplates which require complicated pitch variation rotors the multi rotor copters do not require such controls. Multi rotor copter are easy to design and create because they have fixed pitch rotors. The multiple rotor changes thrust to control the pitch, roll and yaw which is easier than the complications of pitch variation rotors. Depending upon the number of rotors the copters are named. The six-rotor copter is called hex copter, the eight-rotor copter is called octo-copter and the most famous and commonly used is the four-

rotor copter called the quadcopter. The advantages of adding more rotors is that it can handle the increasing payloads, and one or more engine failures. The advantages of more rotors mean inclusion of more power, more lift and greater safety [6].

1.1 Aim of Dissertation

The aim of this dissertation is to use MATLAB and Simulink to test and analyse different Proportional (P) and Derivative (D) gains on different sets of coordinates for stability of the flight control of the Parrot Mambo quadcopter.

1.2 Objectives

The main objectives of this dissertation are:

- To study the history and development of quadcopter over years.
- To study the MATLAB and Simulink software.
- To study the development of PID controllers and PID tuning methods.
- To carry out model-based tuning of PD control on hovering of simulated Parrot Mambo quadcopter over different sets of coordinates.

Chapter 2: Theory

2.1 Literature Review

One of the very basic UAVs (Unmanned Aerial Vehicles) is quadcopter. In [7] the author presented the design of the software in loop simulation framework for a quadcopter that was incorporated in an aircraft. The hybrid aircraft comprised of a quadcopter and a fixed wing with one forward thrust rotor. The author developed a split control system that utilised a typical quadcopter to control the four motors/ propellers and a supervisor controller to control the forward thrust rotor. The feedback signals from the quadcopter are taken by the supervisor controller and commands the fifth rotor for resolving problems like thrust saturation and stabilising the hybrid aircraft, the author concluded. For verifying overall control performances and safety of the hybrid aircraft before real life practical implementations on hardware, the software in loop simulation of a quadcopter was found to be a very effective method [7].

The authors in [8] proposed a model known as an X-4 flyer for the dynamics of a four-rotor vertical take-off and landing (VTOL). To obtain quasi-stationary flight conditions the model incorporated the airframe and motor dynamics along with aerodynamic and gyroscopic effects because of rotors. The approach taken by the author to obtain strong practical stability of the system involved separating the rigid body (airframe) dynamics from motor dynamics, developing separate control Lyapunov functions for the coupled systems and then bounding the perturbation error due to the interaction. The control strategy was novel in two ways by the authors. The first novel aspect, the system dynamics are controlled in two separate dynamic systems corresponding to rigid body dynamics and the motor dynamics. The separate system errors are combined into a single control Lyapunov function by a transient error bounding argument. Second novel aspect of the control is the use of quaternion representation of the rotation error in order to obtain a simple, smooth control design that contains only one singularity in error space corresponding to an error of 180 degrees in the rotation [8].

Drones are expected to be used extensively for delivery tasks in the future. Navigation from start point to the destination without any obstacles is very simple. In real life situations there will be obstacles in the path and the pilot must build a flight plan to avoid those obstacles. However, the task becomes more challenging when the obstacles in the path are unknown and are not at fixed positions. The authors in [9] suggest the use of artificial intelligence to overcome these challenges. The method suggested the author is called reinforcement learning. Reinforcement learning is the branch of artificial intelligence which is used for training machines. The application of reinforcement learning will allow drones with more intelligence eventually turning them into autonomous machines. In reinforcement learning, the agents are computerised systems that learn, and the trial and error experiences are obtained by interacting with the environment. Deep reinforcement learning proposes the use of neutral networks in the decision algorithm. Deep reinforcement learning is widely used in machine vision, parameter optimisation or path finding. The solution of deep reinforcement learning is fades on Double Deep Q Network (DDQN). The authors designed a neutral network that joins two state parts into a unique flow hence it was named Joint Neutral Network (JNN). The author concluded that

the results of Joint Neutral Network has got the better results by 50% and also reduced the variance of the results [9].

One of the tasks that drones are extensively used for is payload add and drop. The control parameters are for many drones are designed and tuned either for no load or constant load. The automatic control performance can deteriorate if the load changes are significant. The author in [10] proposed a robust adaptive control method to deal with this challenge. The aim was to achieve a consistent performance for the altitude control of a quadcopter and enhancing the robustness of the system subjected to signal noises. The proposed technique of adaptive control is suitable because of its capability of tracking a desired output signal even if parametric uncertainties are present. In adaptive control, if the plant parameters are unknown, the controller makes the plant output match that of the reference model. The model reference adaptive control method is to improve onboard Z-axis PI control such that consistent performance can be achieved before and after payload add or drop. The author concluded the superiority of the proposed controller over benchmark PID control by tracking performance comparisons [10].

Fully autonomous flights require high precision in aircraft positioning, especially in takeoff and landing phases. The problem becomes more complex if the landing needs to be accomplished on a moving target. In [11] the system allows a miniature UAV to takeoff and depart from a carrier vehicle, track it by holding a constant position, approach and landing on the vehicle. The author demonstrated how inexpensive infrared (IR) consumer electronic camera can be used as the main sensor for stable flight control. The camera is capable of detecting up to four infrared blobs and provides the pixel position of each blob. This is done by leaving the image processing out to the integrated circuits (ICs) on the camera. The control algorithms can run on an onboard microcontroller at a high frequency without external sensors or base stations. The key idea of this approach is to track a T shaped 3D pattern of infrared light attached to a moving target. The distinct pattern allows for the estimation of the current position relative to the target [11].

Quadcopters have been widely adopted as experimental platforms for research as they high non-linear and uncertain coupled dynamics and thus results in challenging control problems. In [12] the authors present design and deployment of PI-PD and fuzzy PI-PD (FPI-PD) structures to solve position control problem of the Parrot Mambo Minidrone. The non-linear mathematical model of the Parrot Mambo Minidrone is derived to obtain its control models. Then via the control models, firstly the PI-PD control systems are designed for altitude and position control systems, then to handle the coupled non-linearities FPI-PD controllers are designed and employed in the position control loop of the minidrone. The author concluded by results showing that performance of FPI-PD was better than the build in Parrot PD control system [12].

[13] deals with non-linear control design for Parrot Mambo Minidrone or Parrot Rolling Spider quadcopter using Simulink Support Package for Parrot Minidrones. For synthesis of the control the non-linear dynamics inversion and integrator back-stepping approaches are used. The Parrot Minidrones along with Simulink Support Package for Parrot Minidrones (SSPPM) contains all the default control system components such as quadcopter mathematical non-

linear model with the identified physical parameters, the state estimator subsystem that recovers state of the model from the measured data, a tuned PID controller to realise some basic angular and position reference motions. The researchers can test their own control systems and state estimations algorithms by replacing any of the systems components before testing them on real drone [13].

In [5], the author obtains a state space model by using several phenomenon like gyroscopic effects for rigid bodies, propellers and rotors. The obtained model is then used to design PID control laws for roll, pitch, yaw angles and altitude. Physical effects such as aerodynamic effects, gravity, gyroscopic effects, friction and inertial counter torques acting on the system makes helicopters and quadcopters most complex flying systems that exist. Helicopters and quadcopters are dynamically unstable and therefore to stabilise them suitable control methods are required. The author derived a MIMO model using Newton-Euler method. Then the model is simplified and re-written in 12 variable state space vector form. Then the model is decoupled and for each signal: roll, pitch and yaw angles and the altitude the PID controllers are designed. The obtained results for all controllers proved that the whole closed loop system correctly stabilises the motion of the Parrot [5].

2.2 What is a quadcopter?

A UAV is an unmanned aerial vehicle piloted by remote control. Unlike traditional aircraft, there do not carry any passengers or crew onboard. UAVs are also commonly referred to as drones, the two terms have become interchangeable in recent years [15].

The quadcopter is a new UAV in which thrust is generated by four quick turning rotors. Two of the rotors spin clockwise while the other two spin counter clockwise. Two sets of identical, fixed pitch propellers further support this process. Pilots control the quadcopter by remote control transmitters to change the speed of the rotor disks [14].

UAVs are generally divided into 6 categories: Civil and commercial, Research and development, Logistics, Combat, Reconnaissance, Target and decoy. The hobbyist and prosumer drones that we have grown to love are classified as civil and commercial UAVs. Up until around 2010, UAV development was reserved primarily for military use. Over the last decade, however, UAVs have become prevalent in enterprise data collection, aerial photography, and many more recently - with consumer drone hobbyists [15].

Quadcopters were among the first vertical take-off and landing vehicles (VTOL). Earlier helicopters used tail rotors to counterbalance the torque, or rotating force, generated by a single main rotor. But as this was insufficient, engineers developed quadcopters to solve the problems that helicopter pilots faced with making vertical flights [14].

2.3 History and timeline of drone and quadcopter

The fact origin of drones is hard to pinpoint. Some historians track drones back to mid 1800s when European armies used unmanned balloon aircrafts to deliver bombs. The invention of the fixed-wing aircraft in 1903 spurred a generation of drones. In 1917, during the First World War, the US developed drone weapons.

1849 - Austrian Balloons

Austrian balloons are the earliest recorded use of unmanned aerial vehicles. On August 22, 1849, Austrians used balloons loaded with explosives to attack the city of Venice, Italy. The UAV balloons were rather large, twenty-three feet in diameter, and exploded upon impact with the ground.

1907 - World's first quadcopter

The world's first quadcopter was created by Jacques and Louis Breguet and Professor Charles Richet. IT had its limitations that it required 4 people to control it make it steady. On its first flight it only lifted up to two feet.

1916 - World War 1 - Hewitt-Sperry, the first unmanned aircraft.

The first pilotless aircrafts were built during World War 1. In 196, the Hewitt-Sperry Automatic Airplane successfully flew demonstrating that unmanned aircraft flights were possible. The Hewitt-Sperry was controlled through gyroscopes. It was intended to act as flying bomb.

Also, during World War 1, the British military used aerial photography of German enemy trenches to aid their war strategy. The aircrafts were piloted but this was first use of aerial photography.

1917 - First pilotless winged aircraft with remote control (RC) technology

The Ruston Proctor aerial target was the first pilotless winged aircraft. It was radio controlled and based on remote control (RC) technology from inventor Nikola Tesla. Its goal was to act as a flying bomb which could be piloted into enemy territory, although it was never used in combat. However, it paved the way for today's military drones.

1918 - US Army Develops Kettering Bug

Following a demonstration of the Hewitt-Sperry the previous year, the US Army commissioned the development of Kettering Bug. The unmanned Kettering Bug was also intended to act as a flying bomb, but was much more successful, with ability to hit targets 40 miles away.

1930 - US Navy creates radio-controlled aerial aircraft

In the early 1930s, the US Navy began developing the first unmanned aircraft systems that were radio-controlled. They successfully created the Curtiss N2C-2 drone in 1937. The Curtiss N2C-2 drone was controlled from a nearby piloted aircraft.

1940 - Radioplane OQ2 - The first large scale UAV production

Reginald Denny and his Radioplane Company won US Army contract to mass produce their radio-controlled aircraft system. The Radioplane OQ2 UAV was manufactured for use in World War 2. Fifteen thousand drones were built for the army.

1943 - The Germans use FX-1400 in World War 2
The FX-1400 created for the German military in World War 2 was the first remote controlled weapon put into operational use with a 2300-pound bomb used to sink ships. It became the first military drone to be properly deployed and is considered the ancestor of modern anti-ship missiles and other precision guided weapons.

1956 - First propeller quadcopter.
George E Bothezat designed the Covertawings Model A quadcopter, which was the first to use propulsion or a propeller forward thrust to control an aircraft's roll, pitch and yaw.

1958 - Curtis Wright V27
The Curtis Wright V27 quadcopter was launched.

1960 - Popularity boom in RC planes
Breakthrough in transistor technology meant for the first-time miniaturised radio controlled (RC) components were available at low cost leading to popularity boom in RC planes.

1973 - Mastiff UAV - Unpiloted surveillance vehicle
In 1973, Israel developed a series of unmanned aircrafts intended specifically for surveillance and scouting. The surveillance drones were called the Mastiff and the IAA Scout.

1982 - Battlefield UAVs
In 1982, the world's perspective on the legitimacy of unmanned aircrafts changed when Israel's Air Force used them against Syria's Air Force. The implementation was extremely successful and because the aircrafts were unmanned, Israeli casualties were kept to minimum.

1985 - US- large scale UAV development
In 1985, the United States Military launches a large-scale UAV development program, intended to research and develop the technology further.

1986 - RQ2 Pioneer Reconnaissance drone
In 1986, United States and Israel formed a joint effort to produce RQ2 Pioneer Reconnaissance drone.

1993 - Environment and climate
Drones started to be used to monitor climate and environment.

1996 - Predator drone
The first predator drone was developed by the United States.

2001 - discussion for ethics and legalities around drone usage
The impact of 9/11 led to CIA flying armed drones over Afghanistan as part of the war against the Taliban. The first drone-based kill operation was in February 2002, when a drone was used to target a suspect, but mistakenly killed an innocent man. This was likely the first time that the ethics and legalities around drone usage started to come to be discussed.

2006 - UAVs permitted in US civilian airspace
A year after Hurricane Katrina, the US Federal aviation Administration authorises the use of UAV drones in civilian airspace for the first time, equipped with infrared camera which was able to detect heat signatures from a height as far as 10,000 feet.

2010 - Parrot AR drone is released
The Parrot AR Drone was the first smartphone-controlled quadcopter UAV available for consumers. It was first revealed at the International CES 2010 in Las Vegas. Also implemented for the first time, was an IOS application that acted as a control system. The AR Drone was 22 inches in diameter, a size that was manageable for consumer use.

2013 - DJI releases the Phantom 1 UAV
The release of DJI's Phantom 1 quadcopter drone, followed by the Phantom 2, saw camera equipped UAVs enter the market for the first time. The Phantom 1 was able to mount a Go Pro, opening a whole new world of aerial photography and cinematography possibilities to hobbyists. DJI's UAV products quickly gained popularity and made their way into the mainstream market.

2013 - Amazon drone delivery
Amazon's CEO, Jeff Bezos, states that he plans to implement delivery drones to send products to Amazon customers

2014 - Film and TV use
FAA permits Hollywood film and TV production companies to use drones on set.

2016 - Drone with machine learning technology
DJIs Phantom 4 drone introduced smart computer vision and machine leaning technology. It allowed the drone to avoid obstacles and intelligently track and photograph people, animals and objects rather than being limited to GPS signal. This was major milestone for a consumer and photographic drones. However, the same year saw a near-miss with commercial aircraft.

2017 - First passenger drone Ehang 184
Legalities and ethics started to become greater considerations as drones continue to increase in popularity. The first autonomous passenger drone, Chang 184 was unveiled. The Ehang 184's 8 propellers were powered by 8 electrical motors. This enabled it to carry one person and hand luggage.

[14] [15]

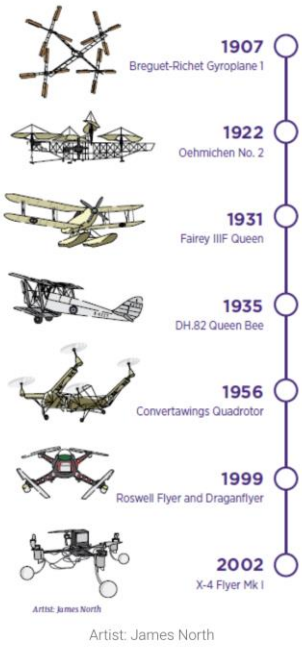


Figure 1: Evolving design of drones [16]

2. 4 Working of quadcopter

According to Newton's Third Law of Motion which says, "to every action there is an equal and opposite reaction (forces)." Forces always comes in pair known as "action-reaction force pair." As the name suggests quadcopter have four rotors. Out of the four, two rotates in clockwise direction while the other two rotates in counter clockwise direction. Because of these opposite rotations they cancel out all the forces and torques on the drone. The quadcopters centre of gravity (COG) is in the middle of its four rotors which helps to maintain its balance. However, if there are different payloads can impact on the centre of gravity which will cause stability problems and have effects on flight controls. To improve these different combinations of motor thrusts will be required.

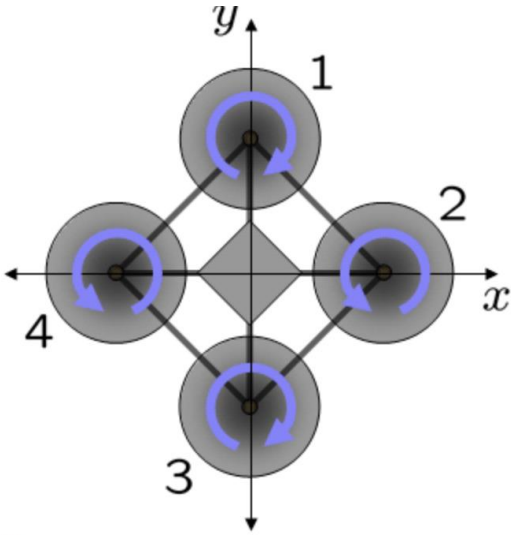


Figure 2: Working of quadcopter [2]

Most of the quadcopters are stabilised today because of three-axis gyroscope technology. As the number of users increases every day the stability problem of the drone becomes important. Most of the commercial drones are used for aerial photography and videography. In order to capture high quality and clear image rather than shaky and blurry, stability becomes important factor.

The propeller direction, together with the quadcopters motor rotation and speed, make flight and manoeuvrability possible. The quadcopters flight controller sends information to the motors van their electronic speed control circuits (ESC) information to thrust, RPM (revolutions per minute) and directions. The flight controller will also combine IMU, Gyro and GPS data before signalling to the quadcopter motors on thrust and rotor speeds [14] [17].

While the drone and quadcopter technology of today is modern, they still use the old principles of aircraft flight, gravity, action and reaction pairs. In the manufacture of quadcopters, propellers and motor design, the four forces which affect all flight that is weight, drag, lift and thrust are also important considerations.

2.5 Motors

The basic requirement for the quadcopter to fly are its motors. It is important that correct motors are selected. Most quadcopters use brushless DC motors. Brushless motors consist of a rotor with a permanent magnet and a number of electromagnets surrounding it which are called poles. Brushless motors usually have anywhere from 2 to 14 poles. The greater the number of poles, more precisely the motor can be controlled.

One of the key specifications of a quadcopter motor is its KV rating. Typical KV ratings of a quadcopter are between 500 and 2500. The maximum current rating of the motor is another important factor. The is important for choosing the electronic speed controller. Another important motor specification is the diameter of the shaft. This becomes important to determine what propellers to be used for the quadcopter [18].

Figure 3: Quadcopter motor [18]

2.6 Propellers

Diameter and pitch are two basic specifications of a propeller. Propellers come in variety of diameters and pitches which can be made of plastic, carbon fibre or wood. Smaller diameter propellers are used for racing and acrobatics with high KV motors. Larger diameter propellers are use with low KV motors in designs to lift heavy payloads such as video cameras.

High pitch propellers can generate slower rotations which can increase the vehicle speed, but this will consume more power. Lower pitch propellers can generate more torque, and this will make the motor to consume less current. The size of the propeller is referred by its diameter. On increasing the diameter, the propeller efficiency is increased but in return this consumes more current form the motor [18].

Figure 4: Quadcopter propeller [18]

2.7 Electronic Speed Controllers (ESC)

The device that controls the brushless DC motors is called Electronic Speed Controllers (ESC). Every motor has one electronic speed controller. ESC's are very small in size. They can vary a motor's speed, direction and braking actions. Quadcopters depend entirely on the variable speed of the motors driving the propellers. The quadcopter gets all necessary control to fly from variable motor speeds and RPM thrusts.

The ESCs have set of wires. These are three heavy-gauge wires that connect to the three wire on the brushless DC motors. The other wires are connected to power distribution board which supplies voltage to the ESC and the motors [14] [18].

Figure 5: Quadcopter Electronic Speed Controller (ESC) [18]

2.8 Flight Controllers

The brain of the quadcopter is the flight controller. This is the device that controls the speed of motor by sending signals to the electronic speed controller (ESC). Simple flight controllers contain only gyros which are sufficient for beginner level. Advanced flight controllers have more features and are coupled with sensors such accelerometer, sonar, GPS, magnetometers and gyros. Flight controllers are minicomputers which can be programmed and updated by connecting it with computer [18].

2.9 Radio Transmitters and Receivers

The quadcopter also has an on-board radio receiver which allows it to be controlled by a handheld radio transmitter. Both the transmitter and receiver are specified by number of supported channels and frequency upon which they operate.

The basic four channels are used as follows - one channel for throttle, one channel for turning left and right, one channel for pitching backwards and forwards and one for rolling right and left. In other words, throttle, yaw, pitch and roll. A channel can also be used to switch different flying modes [18].

2.10 Batteries

The battery selection is one of the most important factors to consider. Larger batteries will have larger capacities that will help the drone to stay in the air for longer periods of time. But this will add more weight which will lower the flight time as more current will be drawn from the motor to lift the drone. Heavier drones show less agility.

2.11 Frame

Frames can be made up of number of materials such as wood, plastic or carbon fibre. Carbon fibre can be very strong for its weight but can also be fragile. The selected frame should also have sufficient space and clearance to mount things like electronic speed controllers (ESC) and flight controllers along with additional items such as cameras [18].

2.12 Vertical Lift

For a quadcopter to get up in air, it requires a force which should be equal to or greater than the force of gravity. The quadcopter's flight involves upwards and downwards forces which is basically the Newton's Third Law of Motion. Newton's Third Law of Motion states that "to every action there is equal and opposite reaction", this means for the quadcopter to rise above the ground the vertical lift should be more than the force of gravity. The spinning of the propeller blades of the quadcopter pushes the air down which results in quadcopters flight.

The vertical lift is directly proportional to the rate of spinning of the rotors. This means higher vertical lift if the rotors spin faster and if the vertical lift is lower when the rotors spin slower. While in the vertical plane the quadcopter can do three things: hover still, climb ascend and vertical descend. The net thrust of the four rotors that push the quadcopter up must be equal to the gravitational force pulling it down for hovering still. By increasing the speed of the four rotors so the upward force is greater than the force of gravity the quadcopter will climb ascend. By decreasing the rotor speed which makes the force of gravity greater than rotor thrust the quadcopter descends vertically.

2.13 Roll, Pitch, Yaw

In a vehicle that travels fast on the ground like car or truck, or on the surface of the water like a boat, generally have travel in 2 dimensions - straight and level. On the other hand, like on aircraft, spacecraft or an underwater submarine, there is a third dimension of depth as well.

An axis can be thought of as a real or imaginary line about which an object such as an aircraft can and will rotate around. Pitch, Roll and Yaw are also known as the "Principal Axes" or "Axes of Rotation" covering Lateral Axis (Pitch), Longitudinal Axis (Roll) and Vertical Axis (Yaw). In case of aircrafts the rotations are achieved using flaps on the wings and tails, but in case of quadcopter, the rotations are achieved by changing the speed and power of the rotor.

PITCH - This moves the quadcopter on the side axis, so it would tilt up and down from front to back. By doing this the quadcopter move forwards or backwards depending upon which way it is tilted. ROLL - This moves the quadcopter on the long (longitudinal) axis, so it would tilt side to side. The causes the quadcopter move to one side or the other depending upon the tilt banking left or tight. YAW - This moves the quadcopter around in a clockwise or anticlockwise rotation as it stays level to the ground. This changes the direction of the quadcopter accordingly [19].

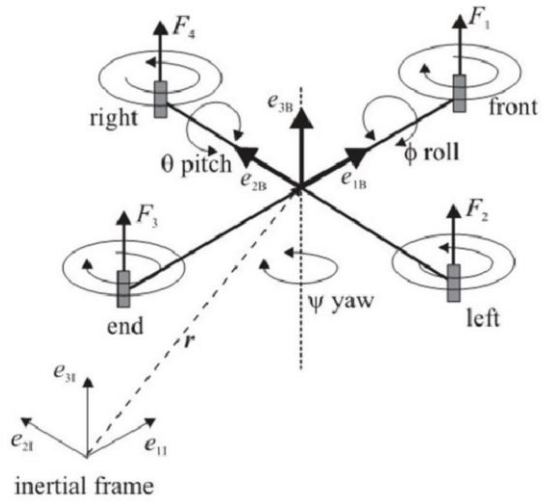


Figure 6: Pitch, Roll and Yaw of quadcopter

2.14 Quadcopter six degrees of freedom

The quadcopter has four motors and each motor has got its own propellers. Opposing set of propellers spin in the same direction. As the system is under-actuated the four set of propellers have to spin in particular ways to allow the quadcopter to be controlled in six degrees of freedom. The six degrees of freedom consists of three translational direction (up and down, right and left, forwards and backwards) and three rotational directions (Pitch, Roll and Yaw).

To express quadcopter's position and orientation in space, six variables are required. These variables are x, y, z, □, □ and □. The variables x, y and z represent quadcopters centre of mass along X, Y and Z axes respectively from a fixed reference frame. The other three variables are the three Euler angles which represents the quadcopter's orientation. □ is the angle about X axis and is called the Roll angle. □ is the angle about Y axis and is called Pitch angle. □ is the angle about Z axis and is called the Yaw angle [14] [20].

2.15 Reference Frame System

A reference point is either inside or outside the chosen coordinate system. A frame of reference or reference frame is reference point which serves as the origin for a coordinate system. A reference frame is like a fixed point. Properties of other objects such as position and velocity are measured using this point. This is because no point in the universe is stationary. The frames of references are classified into two types depending upon how they are moving. Those two types are called inertial and non-inertial frames of reference. An inertial frame of reference has no acceleration. The law of inertial holds in such a frame and no imaginary forces arise. A non-inertial frame of reference is an accelerating frame. The law of inertial does not hold in such a frame and imaginary forces may arise [21].

In case of a quadcopter there are two different reference frames, world reference frame (this is the inertial frame of reference) and body reference frame (this is the non-inertial frame of reference). The world reference frame or the inertial frame of reference is the earth fixed coordinate system with its origin at defined home location. The body reference frame or the non-inertial frame of reference has a coordinate system reference frame at the centre of mass of the quadcopter. The world reference frame remains in the same position even if the quadcopter turns and the body reference frame will remain in line with the quadcopter. This indicates that as the quadcopter rotates or take turn both the reference frame will no longer remain in line with each other [14] [20].

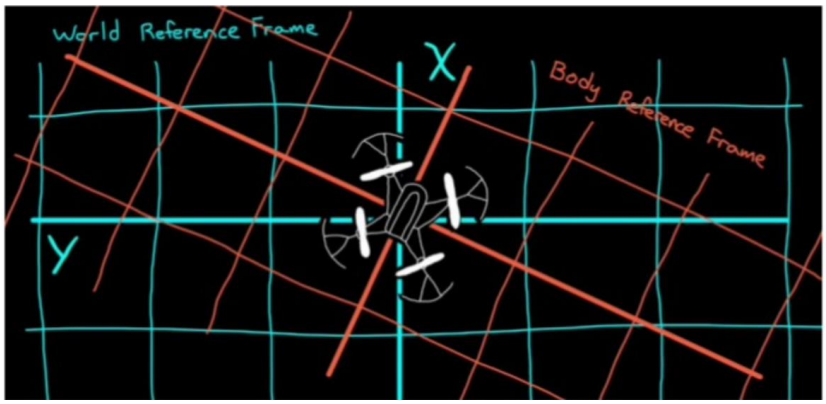


Figure 7: Quadcopter's reference frame [20]

2.16 PID feedback control system

The Proportional-Integral-Derivative (PID) controller is widely employed because it is very easy to understand, and it is quite effective. One attraction of the PID controller is that all engineers understand conceptually differentiation and integration, so they can implement the control system even without a deep understanding of control theory. Further, even though the compensator is simple, it is quite sophisticated in that it captures the history of system (through integration) and anticipates the future behaviour of the system (through differentiation).

A feedback control system is a system whose output is controlled using its measurements as a feedback signal. This feedback signal is compared with a reference signal to generate an error signal which is filtered by a controller to produce the system's control input. The main objectives of the feedback control is to ensure that variables of interest in a process or a system, thought off as the output signals, either 1) track reference trajectories (called tracing or servo), or 2) are maintained close to their set points (called regulation). The need of feedback control systems is basically for three reasons, 1) to counteract disturbance signals affecting the output, 2) to improve system performance in the presence of model uncertainty and 3) to stabilise an unstable plant [22].

A PID controller continuously calculates an error value e(t) as the difference between a desired set point (SP) and a measured process variable (PV) and applies a correction based on proportional, integral and derivative terms. In practical terms it automatically applies accurate and responsive correction to a control function. The block diagram shows the principles of how these terms are
generated and applied. The block diagram shows a PID controller, which continuously calculates an error value e(t) as the difference between a desired set point SP = r(t) and a measured process variable PV = y(t), and applies a correction based on proportional, integral and derivative terms. The controller attempts to minimise the error over time y adjustment of a control variable u(t), such as the opening of a control valve, to a new value determined by a weighted sum of the control terms [2].

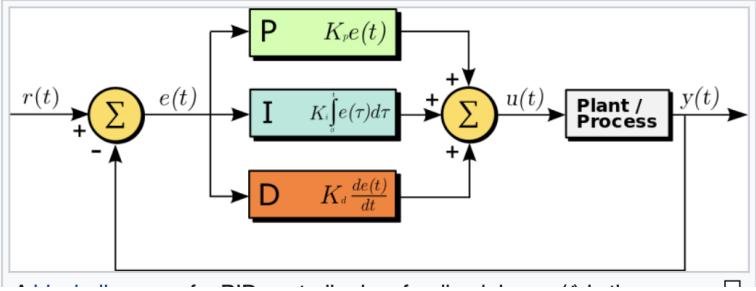


Figure 8: Block diagram of PID controller in feedback loop

The term P is proportional to the current value of the SP-PV error e(t). If the error is large and positive, the control output will be proportionally large and positive by gain factor 'K'. Using proportional control alone will result in an error between the set points and the actual process value because it requires an error to generate the proportional response. If there is no error,

there is no corrective response. The term I is integral and accounts for past values of SP-PV error and integrates them over time. If there is a residual SP-PV error after the application of proportional control, the integral terms seek to eliminate the residual error by adding a control effect due to the historic cumulative value of the error. When the error is eliminated, the integral term will cease to grow. This will result in the proportional error diminishing as the error decreases, but this is compensated by growing integral effect. The D term is a best estimate of the future trend of the SP-PV error, based on its current rate of change. It is sometimes called "anticipatory control", as it is effectively seeking to reduce the effect of the SP-PV error by exerting a control influence generated by the rate of error change. The more rapid the change, greater the controlling or dampening effect [2] [22]. The mathematical form of the overall control function is,

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(t')\,dt' + K_{\mathrm{d}} \frac{de(t)}{dt}$$

Figure 9: Equation for transfer function of PID controller

Where K_p, K_i and K_d, all non-negative, denote the coefficients for the proportional, integral and derivative terms respectively. In standard form of equation, K_i and K_d are respectively replaced by K_p/T_i and K_pT_d. The advantage of this being that T_i and T_d have some understandable physical meaning as they represent the integration time and derivative time respectively.

$$u(t) = K_{\mathrm{p}} \left(e(t) + \frac{1}{T_{\mathrm{i}}} \int_0^t e(t')\,dt' + T_{\mathrm{d}} \frac{de(t)}{dt} \right)$$

Figure 10: Equation for transfer function in time constant form

2.17 PID controller tuning methods

The balance of the proportional, integral and derivative effects is achieved by loop tuning to produce the optimal control function. The tuning constants are shown as "K" and must be derived for each control application, as they depend on the response characteristics of the complete loop external to the controller. Tuning a control loop requires the control parameters to be altered until the best values for the desired control response are achieved. Each of the three mathematical control functions in PID – Proportional, Integral and Derivative is governed by a user defined parameter. For optimisation of precision of control these parameters need to adjust. PID tuning is the process of determining the values of these parameters. The target of good tuning is to have the fastest response possible without causing instability.

If there are multiple challenging and complex objectives such as to achieve high stability and short transients, detaining and tuning of PID can be very difficult in practice. The initial designs are made and are adjusted by running computer simulations continuously until the desired closed loop performances are obtained. PID controllers often provide acceptable control using default tunings, but performance can generally be improved by careful tuning or performance may be unacceptable if the tuning is poor. There are several methods for tuning a PID loop. The most effective method generally involves development of some form of model, then choosing P, I and D based on the dynamic model parameters. Some of the commonly used methods of tuning rules are manual tuning on-site, Ziegler-Nichols reaction curve method, Ziegler-Nichols oscillation method and Cohen-Coon method [4] [14] [23].

Chapter 3: Parrot Mambo Fly Quadcopter

3.1 Hardware

Figure 11: Parrot Mambo Fly [24]

The Parrot Mambo quadcopter has optimised and ideal weight speed ratio which makes it the best machine for inspecting the effects of changing gains. This is due to the high efficiency of weight to power of the minidrone. The Parrot Mambo minidrone, when bumpers protecting blades attached, has a size of 18 cm by 18 cm. it weighs around 63 grams without any numbers or accessories. For better orientation and stable flight, the quadcopter is equipped with several stabilisation sensors. In total there are four sensors: ultrasound sensor, camera, pressure sensor and Inertial Measurement Unit (IMU) [14] [20] [24].

The Inertial Measurement Unit (IMU) is one of the main sensors of the Parrot Mambo minidrone. The Inertial Measurement Unit (IMU) is an electronic device which uses accelerometers and gyroscopes. It uses three axis accelerometer which is used to measure linear acceleration and three axis gyroscope that measures angular rate [20].

The Parrot Mambo quadcopter holds a pressure sensor which indirectly measures the altitude. It measures the pressure level of the atmosphere around itself and from this calculates the height of the quadcopter. The pressure sensor inside the min drone is a barometric pressure sensor and calculates the change of force on the quadcopter divided by the area of pressure sensor which is constant [14].

After flipping the drone, we can see two sensors. The first sensor is the one which is having a grid is the ultrasound sensor. The ultrasound sensor is used to measure vertical distance. The vertical distance is calculated by drone by sending high frequency sound pulses and calculating travel time from drone to ground and ground to drone pulse transmission [20].

The other sensor is the camera which takes images at 60 frames per second and uses image processor technique called optical flow to determine how object is moving from one frame to another frame. By using this the minidrone can estimate horizontal motion and can determine speed [20].

In simple words the ultrasound sensor is for distance above a surface, camera for horizontal motion and speed, pressure sensor for altitude and IMU for acceleration and angular rate. We can use ultrasound sensor and pressure sensor to determine altitude. We can use IMU and camera to determine rotational and translational motion [20].

We have four motors each with their own propellers. These four motors are laid out in X-configuration. The difference between X-configuration and +-configuration is that which motors to send commands to. The most ingenious part of the motor of the drone in motor configuration is its spin direction, opposing motors spin in the same direction as each other but opposite direction to the other pair. This is necessary to make sure that thrust, roll, pitch and yaw can be controlled individually. The motors produce thrust by speeding the propeller which pushes air down causing a reaction force in upward direction. If the motor is placed at centre of gravity (COG) then the object will move up with pure translation without rotation. If the force is equal to force from gravity, then the object will hover in place. A force away from the centre of gravity will produce both translational motion as well as torque rotating movement about the COG. If there are two forces at the end that are each half of COG, then again, the object will hover. The reason counter rotating motors are used is because how yaw or flat spinning motion interacts with roll and pitch [20].

The Parrot Mambo minidrone also has a 550 mAh LiPo battery that allows 9 minutes of autonomous flight, with no accessories or bumper attached and a 30-minute charge time with a 2.1 A charger. The Parrot Mambo minidrone also has accessories that can be added to it such as an FPV (First Person View) camera, a grabber and cannon [14] [24].

3.2 Software

The Parrot Mambo minidrone has stabilisation and flight control software that keeps the quadcopter stable and allows for autonomous flight. This also allows for the flight control system to be edited through MATLAB and Simulink and then uploaded to the quadcopter. The quadcopter can connect to its external accessories bumper, FPV camera, grabber or a cannon. The quadcopter can also connect to computers via Bluetooth. Using this Bluetooth, the code can be downloaded, and the flight control system can be edited [14] [24].

The Parrot mambo minidrone works with Windows operating system based computers and SDK (Software Development Kit) availability for Linux operating system-based computers. The quadcopter also has compatibility to work with iOS 7 or newer and Android 4.3 or newer versions with applications that allow for the quadcopter to be flown manually or to be made fly automatically using simple coding apps [14] [24].

3.3 MATLAB and Simulink

The Parrot Mambo quadcopter has support packages for both MATLAB and Simulink that allows for the quadcopter to be flown manually and autonomously from a computer that is connected wirelessly via "Bluetooth Low Energy' [14] [25] [26].

MATLAB has a support package for the Parrot Mambo which allows for the computer to send commands in order to control the speed, height and orientation of the quadcopter. MATLAB also allows for data such as speed, height and orientation, to be saved and then viewed in the form of matrices saved directly from the quadcopter's data logger or by graphing out the saved data. MATLAB also allows for images and videos to be "streamed" from the quadcopter's camera and also gives readings for the quadcopter's battery level [14] [25] [26].

Simulink also has a support package for the Parrot Mambo which allows for flight control algorithms to redesigned, built and then deployed wirelessly to the quadcopter via "Bluetooth Low Energy". The use of "Flight Control User Interface" allows for the flight control algorithm to be overwritten if required during flight as well as direct control of the quadcopter to be assumed by the "host computer". These algorithms can access the sensors on the Parrot Mambo, such as the Inertial Measurement Unit, the ultrasound sensor, the pressure sensor and the camera sensor, and control the quadcopter through the data collected from the sensors. Simulink also has the additional packages, such as the Aerospace Blockset, which allows for "6-DOF equations of motion" and the simulation of aircraft behaviour when it is put under numerous flight and environmental conditions. Another blockset, Simulink Coder, allows data from the quadcopter to be accessed, such as speed, height and orientation [14] [25] [26].

3.4 Model

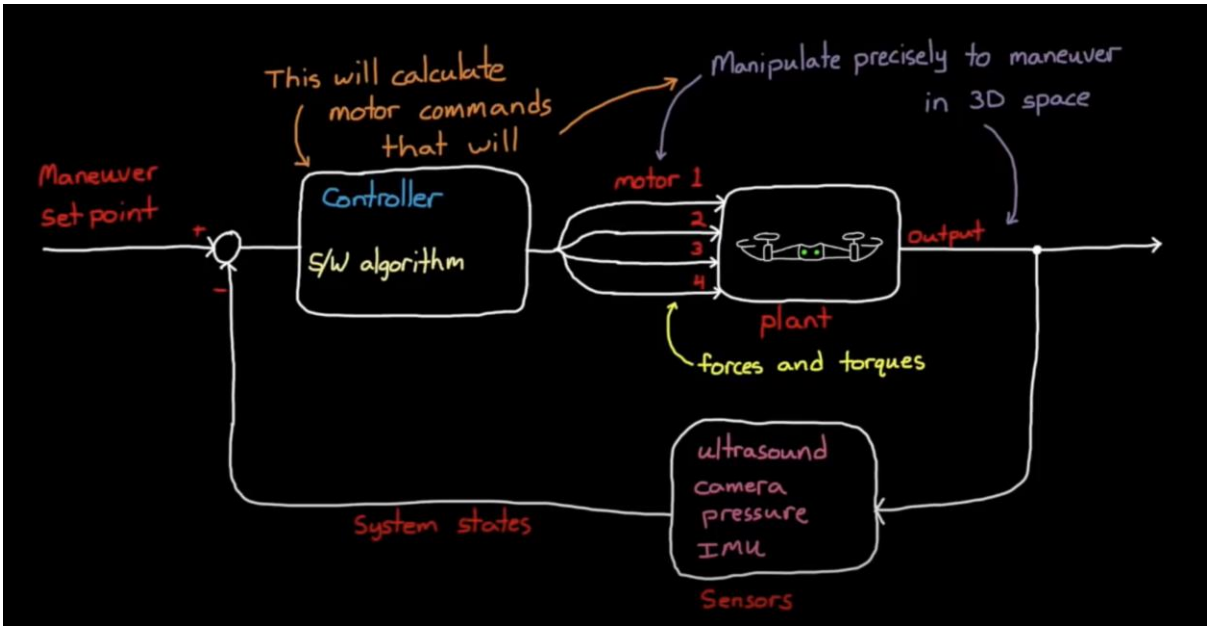


Figure 12: Overview of control problem [20]

Getting the control algorithm isn't straight forward because this is an under actuated system. We only have 4 actuators for six degrees of freedom. Three translational directions up/down, right/left, forward/backward and three rotational directions roll, pitch and yaw. Since we do not have actuators for each motion if means some directions are uncontrollable at any given time. For example, the minidrone cannot move left/right without rotating. This problem is dealt by designing a control system that couples rotation and thrust to accomplish the overall goal. The system states may include angular position/rates, altitude and horizontal velocity.

For building an autonomous feedback controller for the drone is started by focusing on thrust command. Thrust is always in the same direction relative to airframe. It is along the Z axis of the minidrone. Increasing thrust increases altitude rate and decreasing thrust decrease altitude rate. Here an assumption is made that roll and pitch angles are small so only thrust affects the altitude.

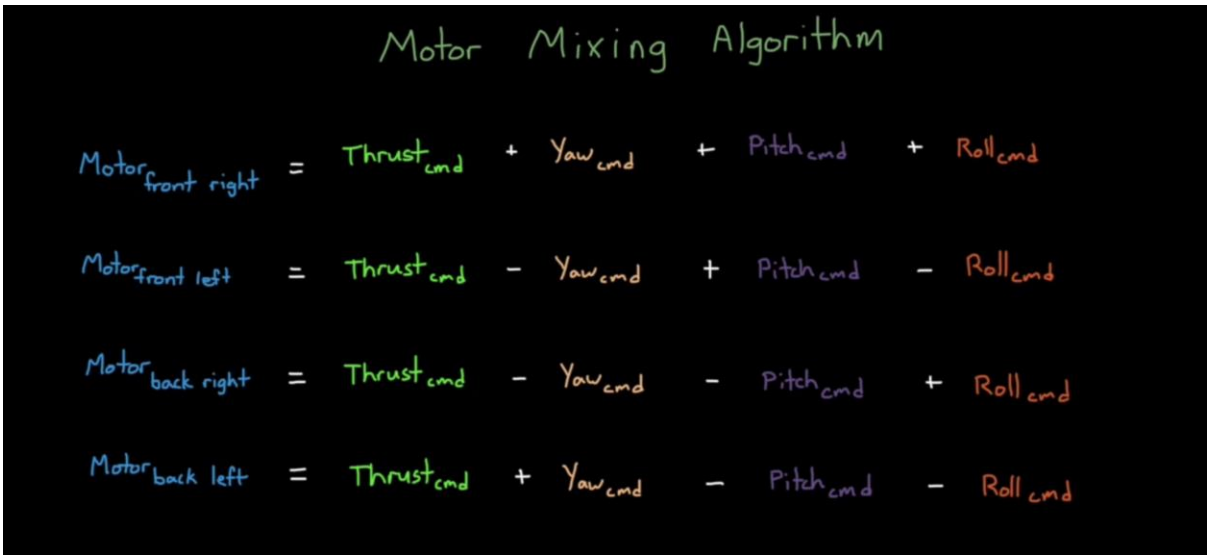


Figure 13: Motor Mixing Algorithm [20]

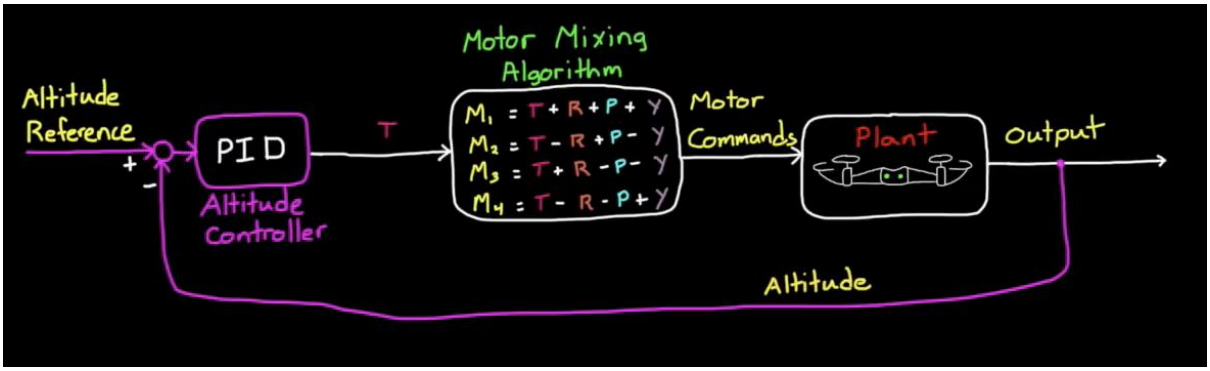


Figure 14: Controller 1 [20]

The image shows simple hovering control. If the altitude is more than the altitude reference, then all the motors will slow down and bring the minidrone down. If the altitude is less than altitude reference, then all the motors will speed up to bring the minidrone up in the air. But this is not possible because of external disturbances like wind that will induce little roll and pitch in the system. In this case the thrust will not only create altitude but horizontal motions as well. So, a controller is needed that will set roll and pitch angles to zero degrees [20].

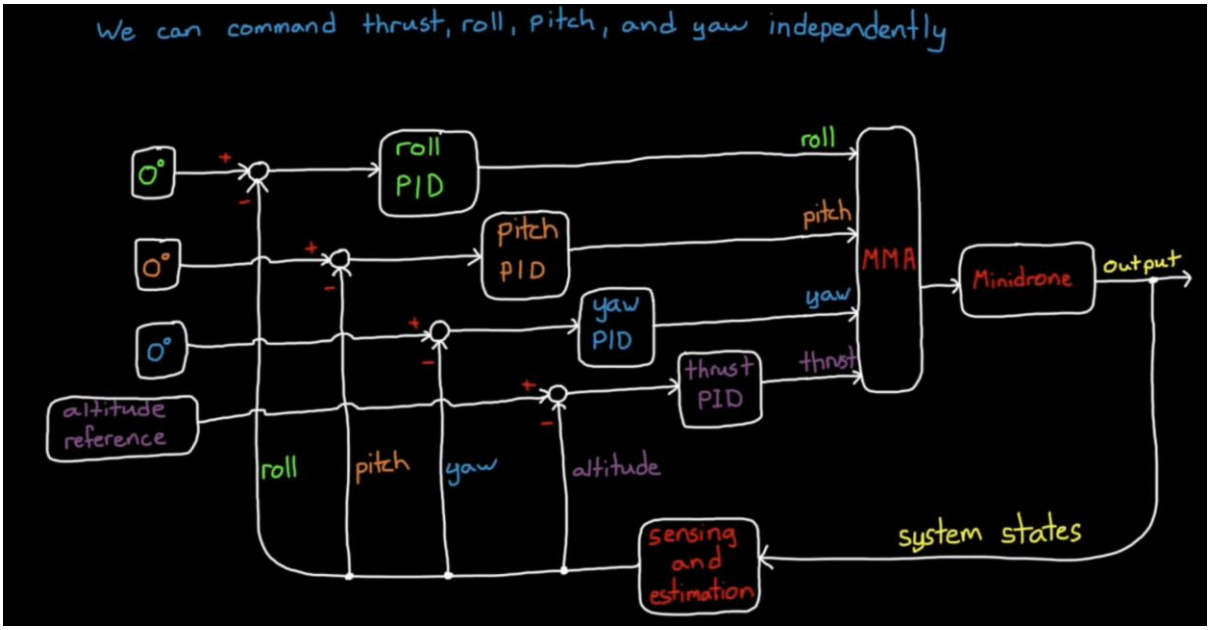


Figure 15: Controller 2 [20]

This is a better controller than the previous controller because it makes the roll, pitch and yaw angles to zero, but it will still not hover over a reference point. Let's assume a situation where there is wind on the system, which will create some roll, pitch and yaw. The system will make the roll, pitch and yaw angles to zero but by the time thrust would have acted and the minidrone would have moved away from the reference point [20].

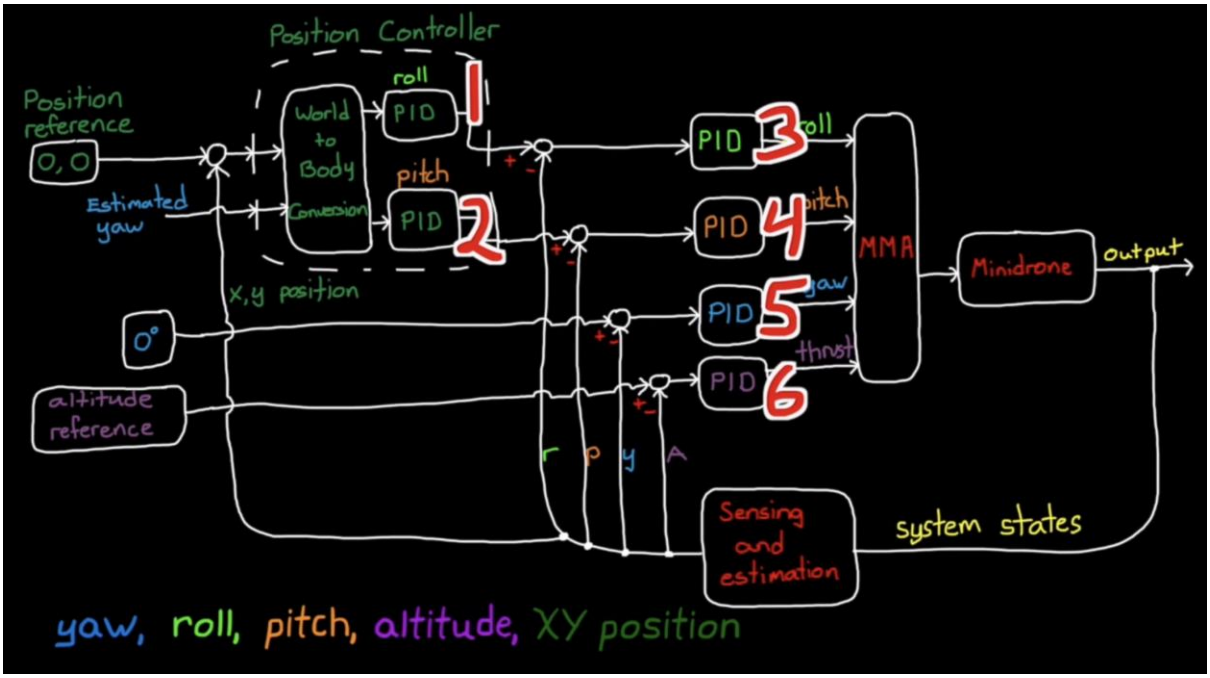


Figure 16: Controller 3 [20]

We can feedback the measured x, y position and compare it to get reference error. The position controller takes the position as input and outputs roll and pitch angles. These are reference angles that roll, and pitch are to follow. This means we are letting the position controller choose roll and pitch angles instead of us. The position controller is the outer loop and it is generating commands for the inner loop roll and pitch controllers [20].

3.5 Simulink model

The following images show block diagrams of different controllers inside the hover model of Simulink Support Package for Parrot Mambo minidrones.

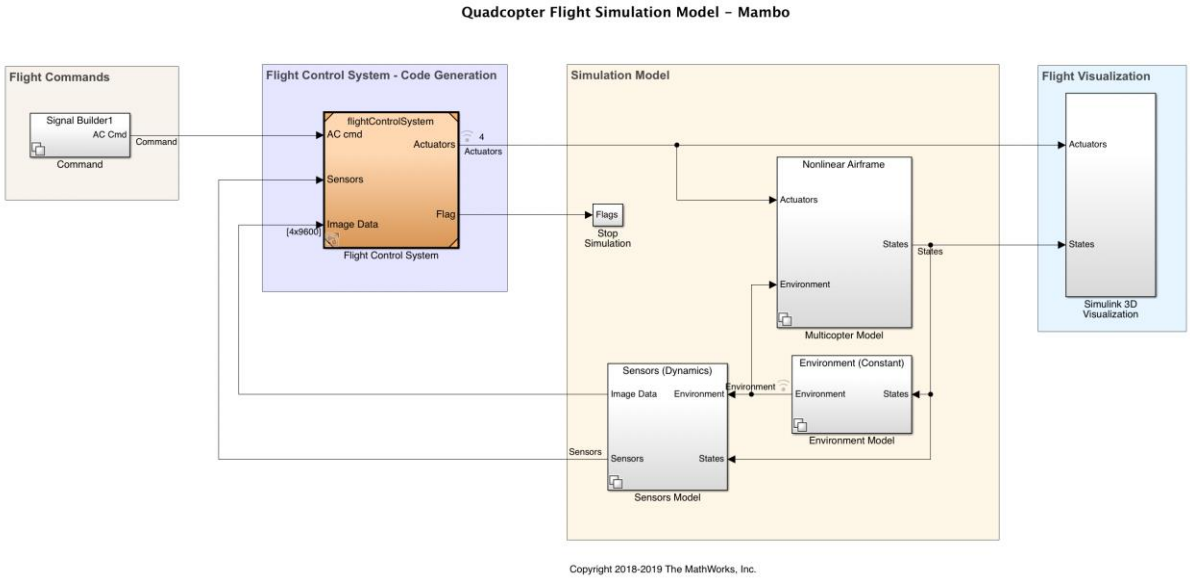


Figure 17: Quadcopter flight simulation model – Mambo

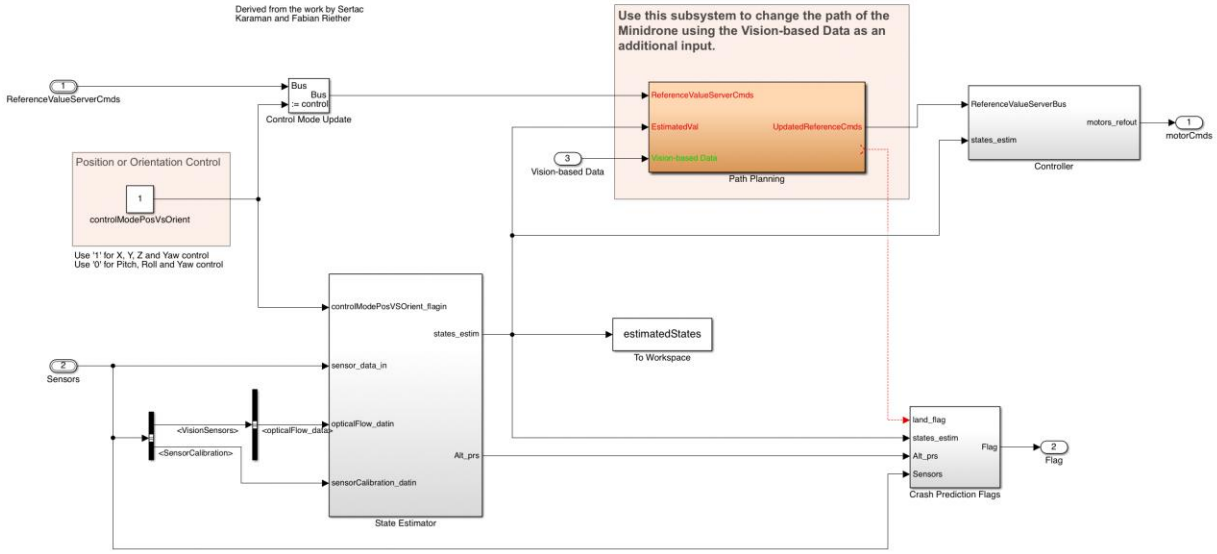


Figure 18: Flight Controller

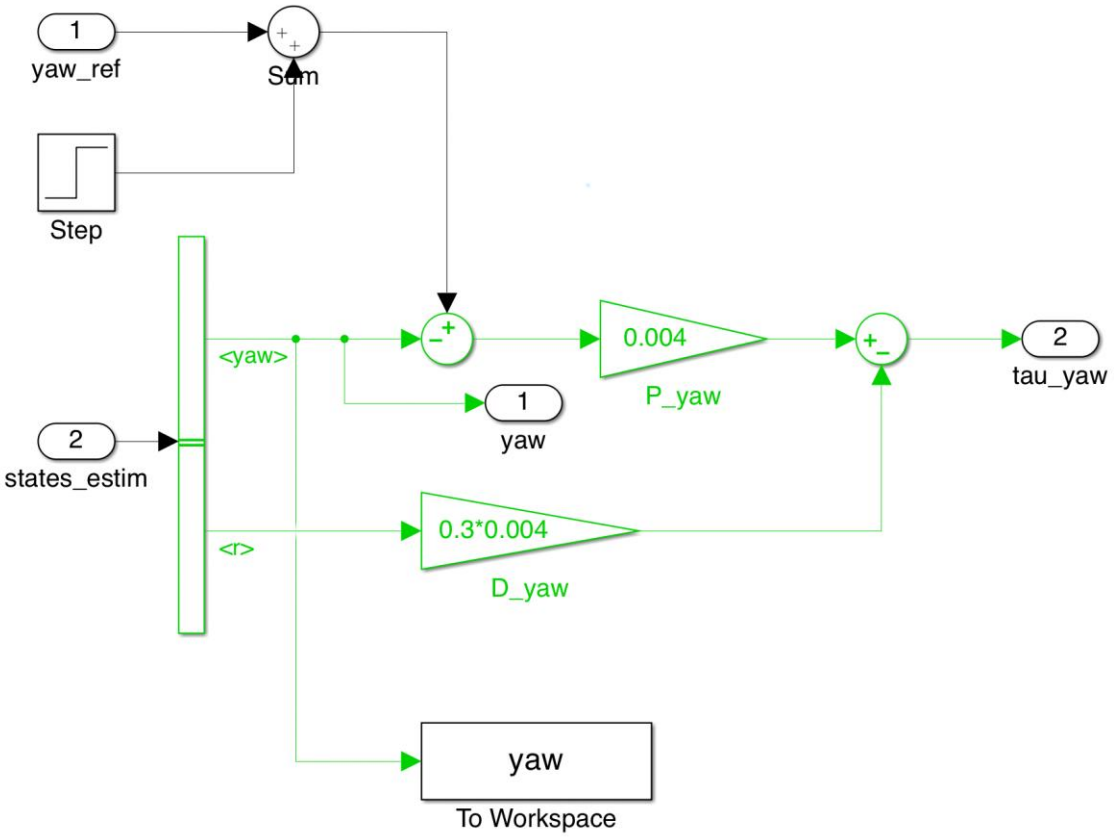


Figure 19: Yaw controller

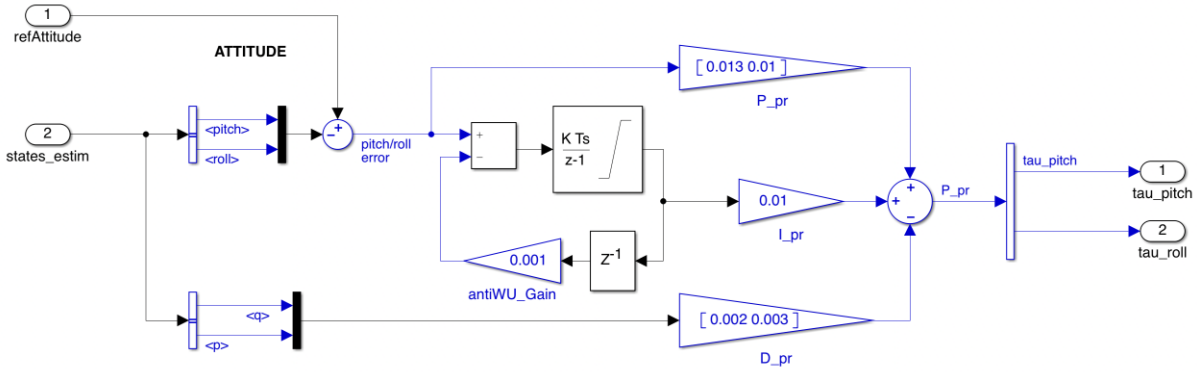


Figure 20: Altitude Controller

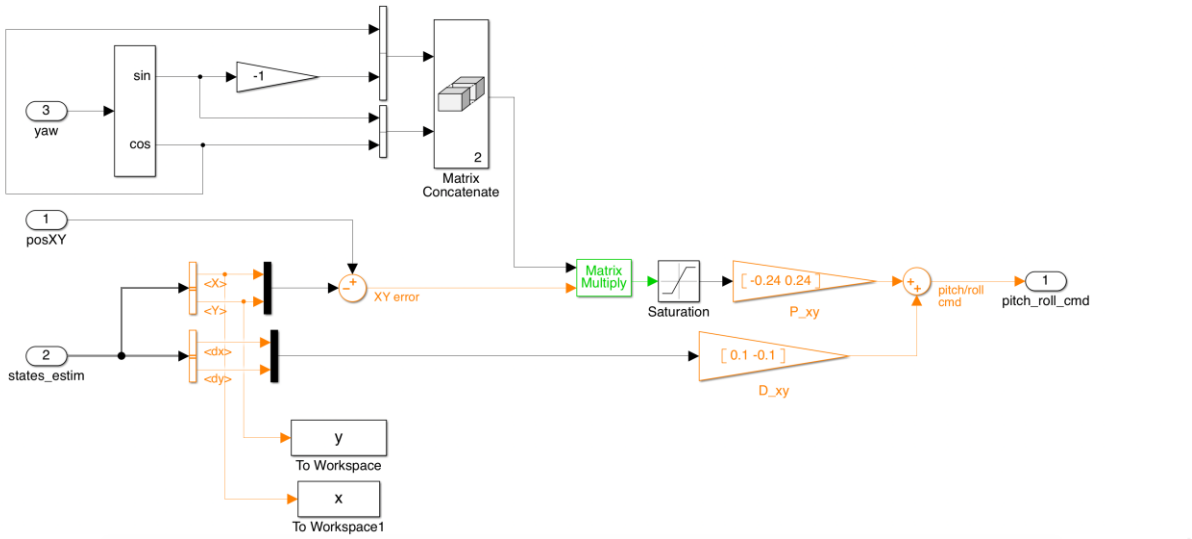


Figure 21: XY-to-reference-orientation block

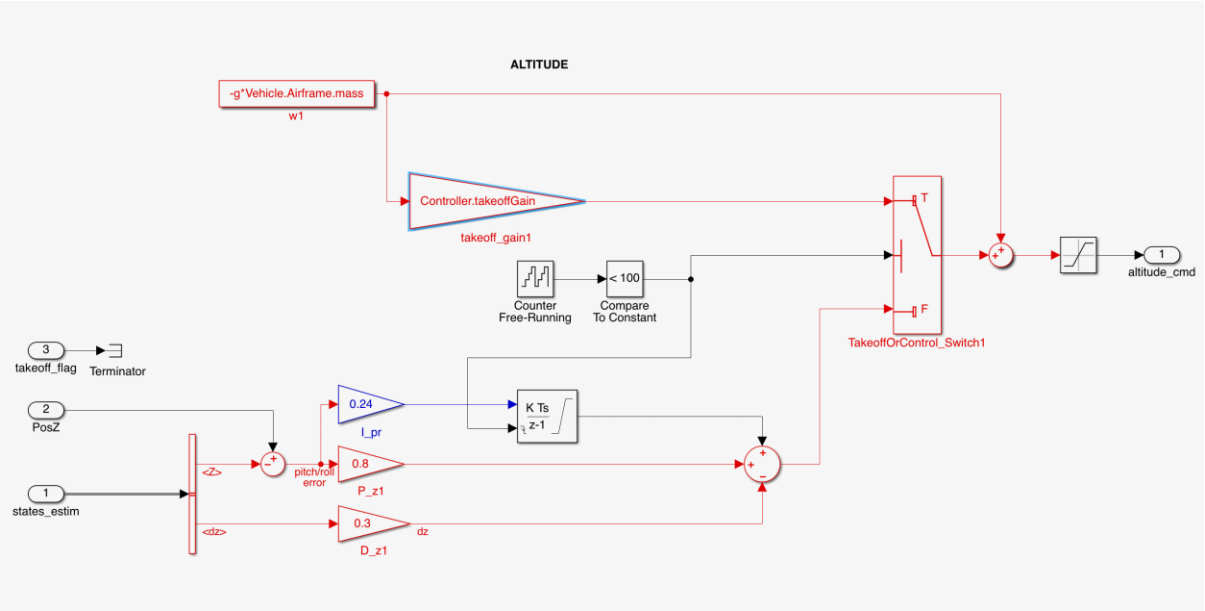


Figure 22: Gravity feedforward/equilibrium thrust block

Chapter 4: Results, Discussions and Inspections

To avoid the risks of damaging the quadcopter hardware simulations of the Parrot Mambo minidrone were performed on Simulink. This was done for selecting suitable values for the Proportional (P) and Derivative (D) gains by trial and error methods. The PD gains were changed in the Flight Control System (FCS) block. The Z axis stability of the quadcopter was tested for different PD gains during take-off and static hover. The initial set points were fixed at X=0; Y=0; and Z=-1.1. Z=-1.1 means height of 1.1 meter. After running the simulation for a set time the resulting graphs were studied and analysed.

4.1 Inspection of takeoff gain

The takeoff gain can be found inside the gravity feedforward/equilibrium thrust block. By changing the controller takeoff gain inside the gravity feedforward/equilibrium thrust block, controls the change in altitude. In XY-to-reference-orientation block the PD values used were the default values of P= [-0.24, 0.24] and D= [0.1 -0.1]

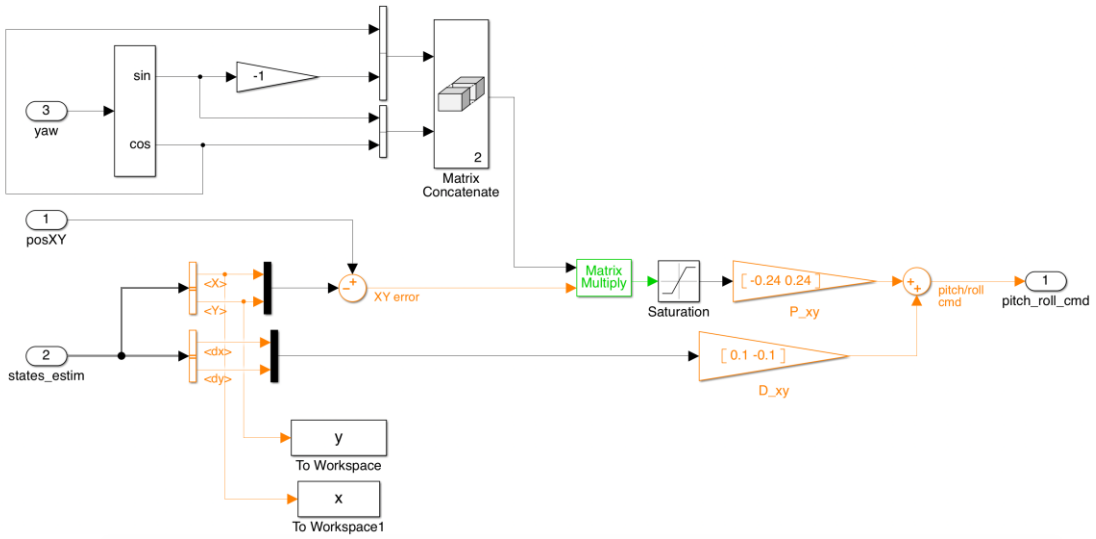


Figure 23: XY-to-reference=orientation block showing P= [-0.24, 0.24] and D= [0.1 -0.1]

In the gravity feedforward/equilibrium block the default PD gains of P=0.8 and D-0.3 were used.

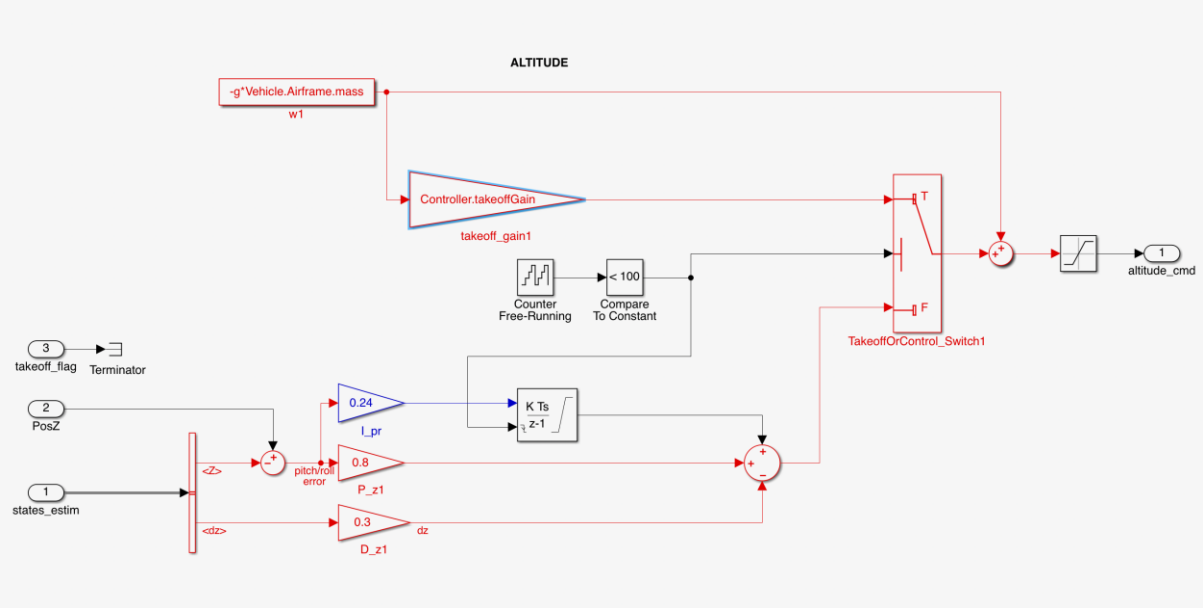
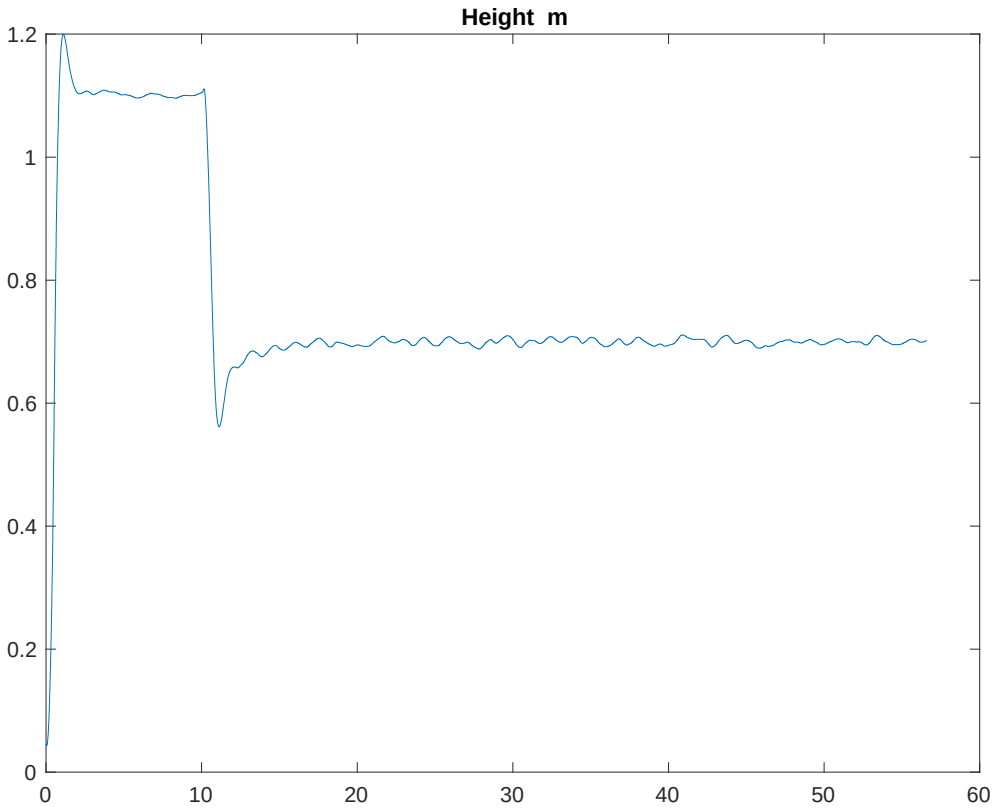
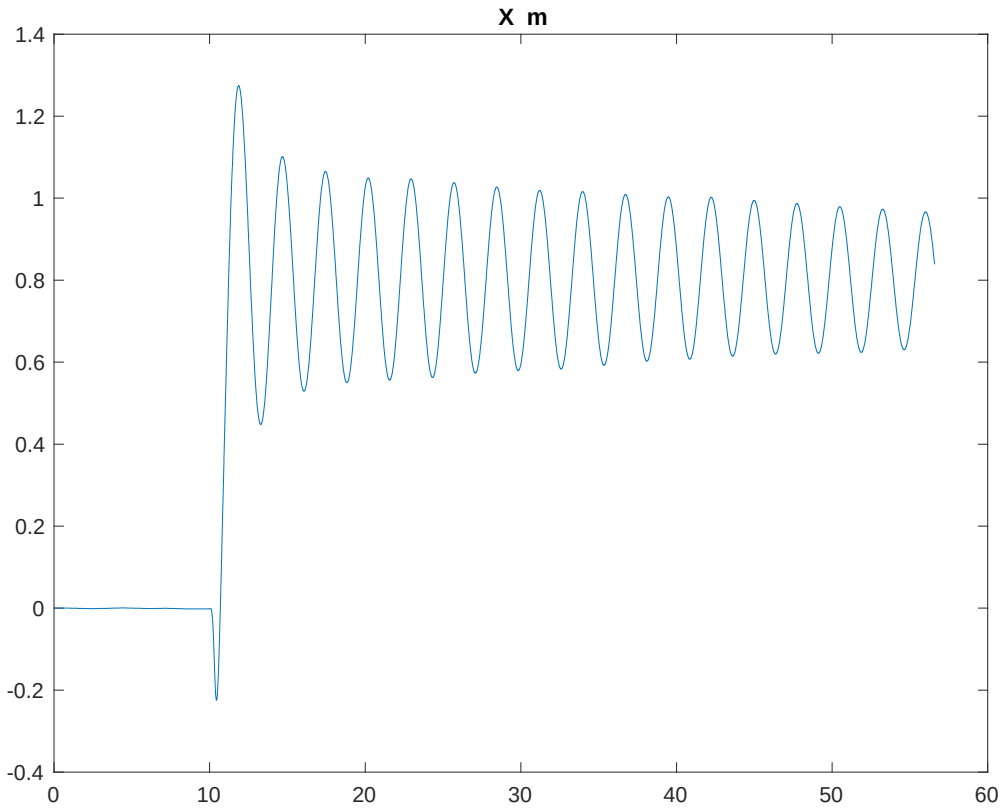


Figure 24: Gravity feedforward/equilibrium block showing P=0.8 and D=0.3. the takeoff gain used is default value

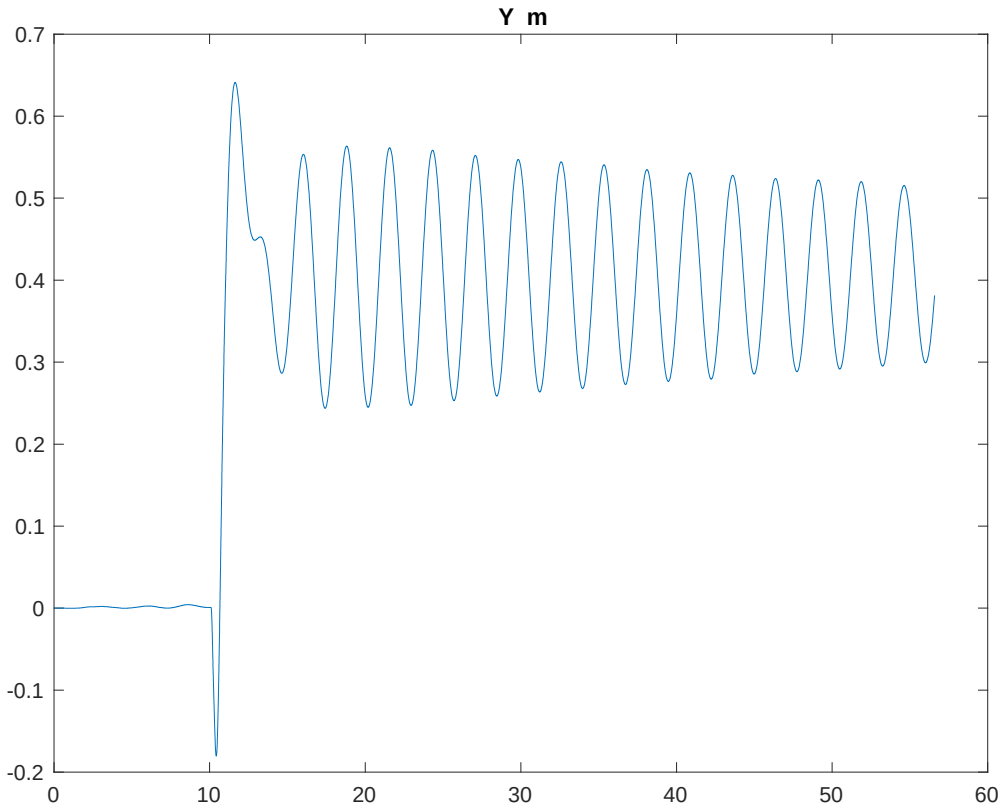
After running the simulation for T=50 the following graphs were obtained.



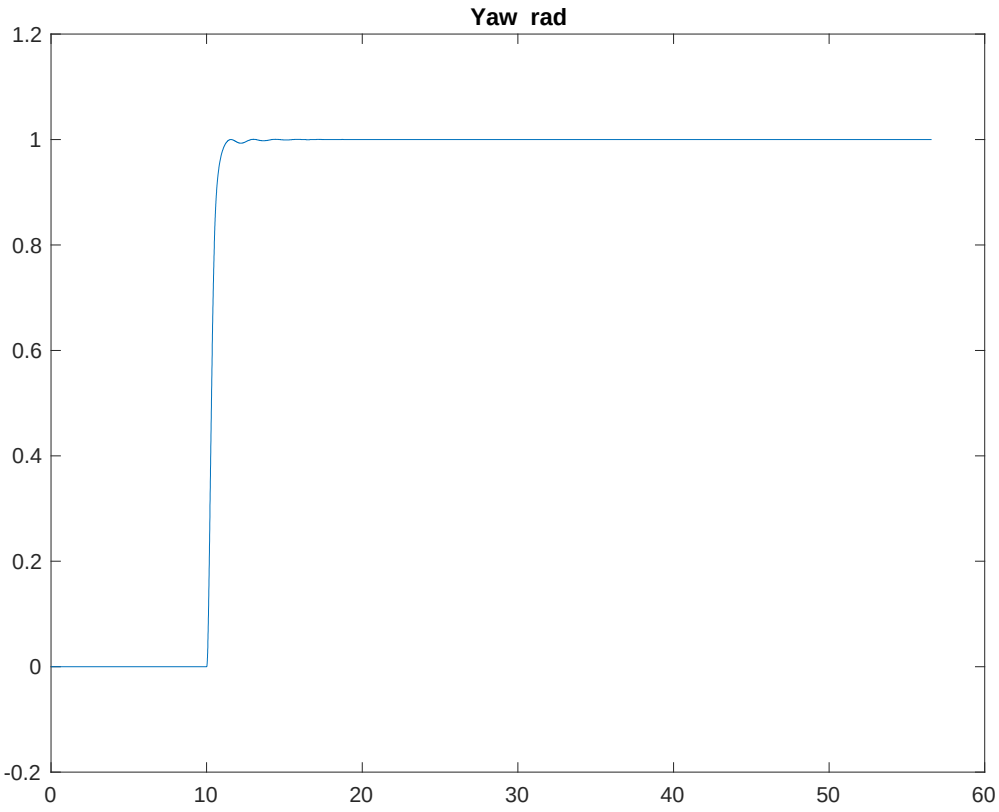
Graph 1: Inspecting the takeoff gain Z axis output against time for default takeoff gain.



Graph 2: Inspecting the takeoff gain X axis output against time for default takeoff gain.



Graph 3: Inspecting the takeoff gain Y axis output against time for default takeoff gain.



Graph 4: Inspecting the yaw output against time for default takeoff gain.

The Graph 1 shows that the overshoot is not much at around 1.2 meters which is a good sign, but the settling time is not good. Graph 2 shows the analysis of the takeoff gain X axis output against time. Graph 3 shows the analysis of the takeoff gain Y axis output against time. Graph 4 shows yaw output against time for the initial gains. A new gain is selected to test the performance of simulated quadcopter. The new takeoff gain selected is 0.8.

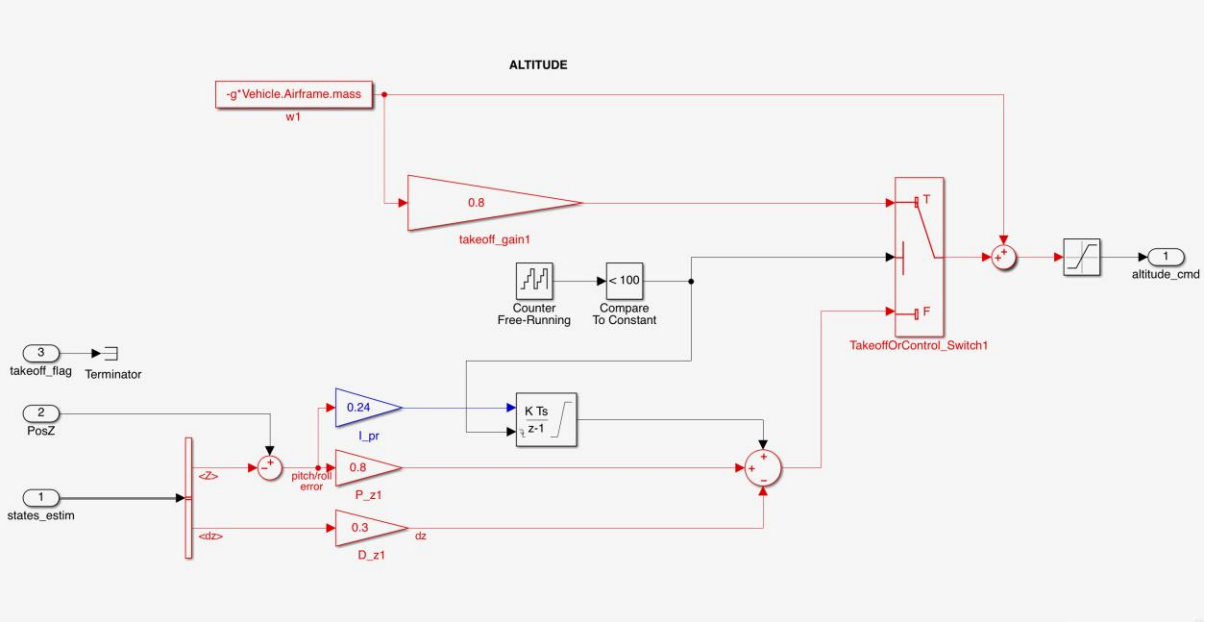
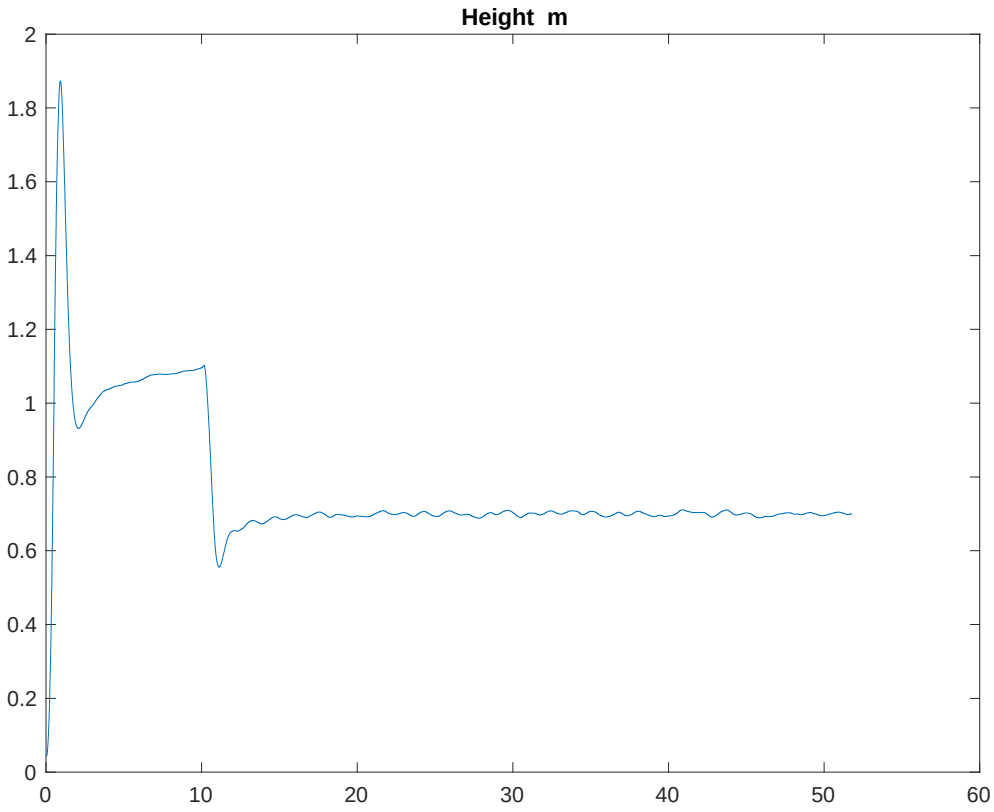


Figure 25: Gravity feedforward/equilibrium block showing P=0.8 and D=0.3. take off gain =0.8

After running the simulation for T=50 the following Z axis output against time is obtained.



Graph 5: Inspecting the takeoff gain Z axis output against time for takeoff gain = 0.8.

Graph 5 shows the plot for Z axis output against time for takeoff gain of 0.8. The plot shows overshoot of more than 1.8 meters for set height of 1.1 meters. This means a high value of takeoff gain won't do any good as the quadcopter might get out of control. A much lower value of takeoff gain is selected for the next test. The new takeoff gain value selected is 0.04.

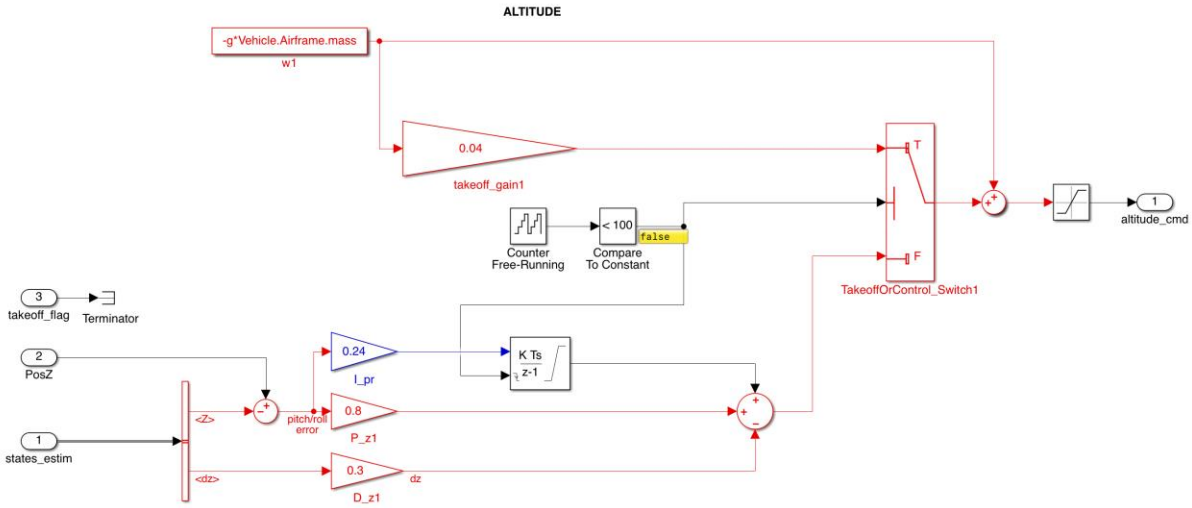
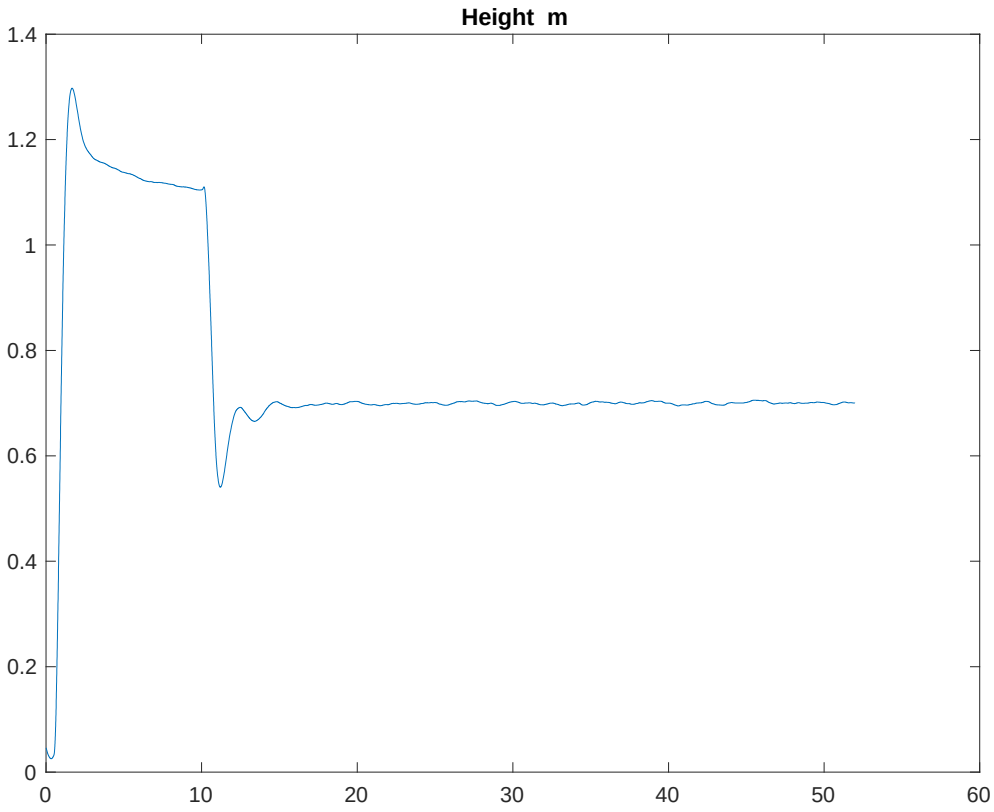
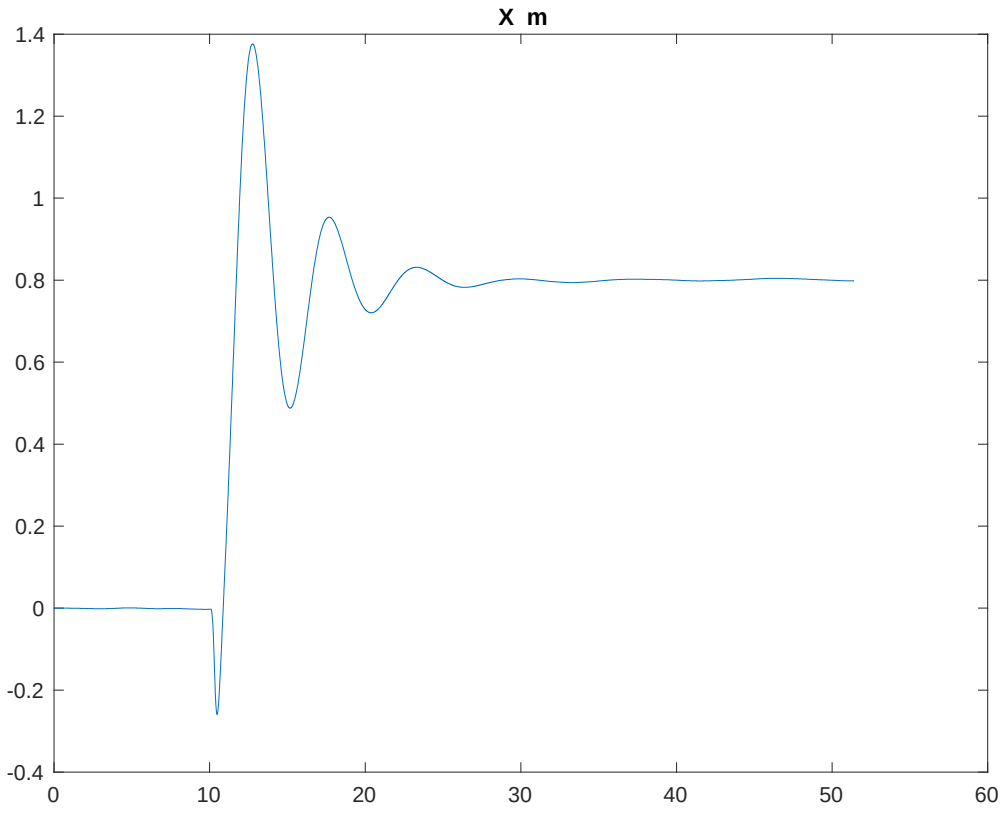


Figure 26: Gravity feedforward/equilibrium block showing P=0.8 and D=0.3. takeoff gain =0.04

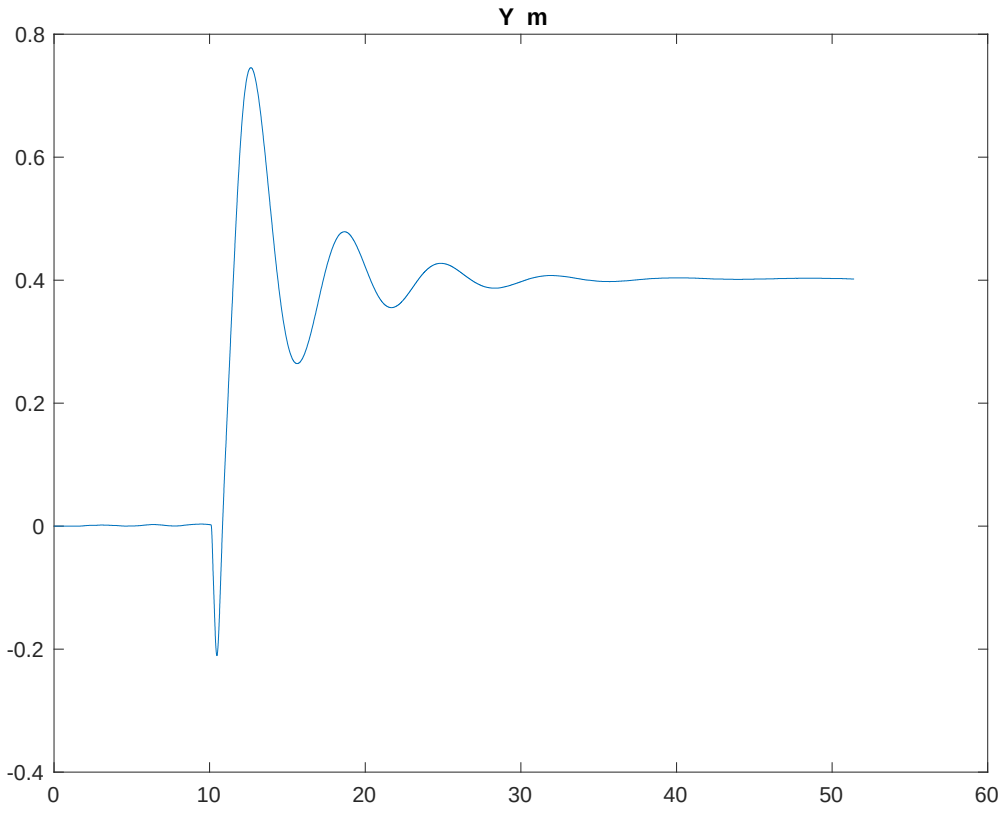
After running the simulation for T=50 the following graphs were obtained.



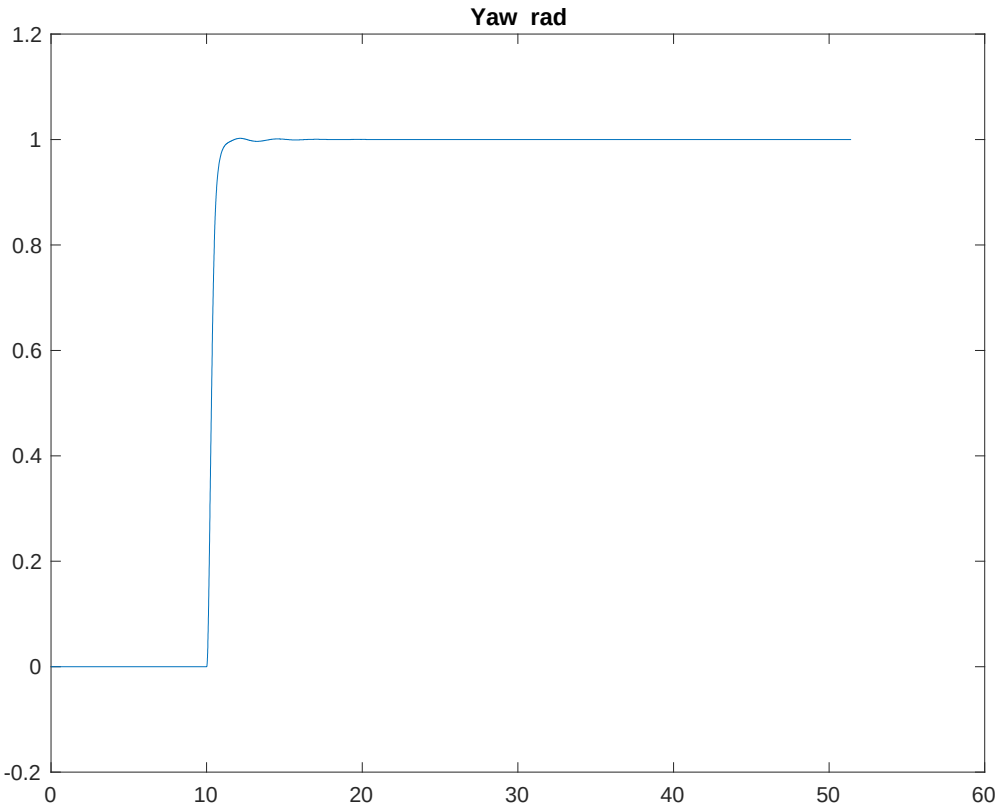
Graph 6: Inspecting the takeoff gain Z axis output against time for takeoff gain = 0.04.



Graph 7: Inspecting the takeoff gain X axis output against time for takeoff gain = 0.04.



Graph 8: Inspecting the takeoff gain Y axis output against time for takeoff gain = 0.04.



Graph 9: Inspecting the yaw output against time for takeoff gain = 0.04.

By inspecting Graph 6 we can conclude that the overshoot is increased quite a bit but there is significantly better settling time observed than those of the previous tests. Graph 7 shows analysis of takeoff gain X axis output against time. Graph 8 shows analysis of takeoff gain Y axis output against time. Graph 9 shows yaw output against time for takeoff gain of 0.04. This means a lower value of takeoff gain like 0.04 appears to be the best value for the simulated quadcopter to takeoff and settle at initial hovering height of 1.1 meter.

4.2 Inspection of gravity feedforward/equilibrium thrust PD gains

After changing the default takeoff gain the simulated quadcopter takes off and reach its initial height very quickly only with a smaller amount of overshoot. In this section, the inspection of the PD gains inside the gravity feedforward/equilibrium thrust block is done. Inspection of overshoot, settling time and stability is tested for Z axis, X axis, Y axis and Yaw against time for simulated Parrot Mambo quadcopter. The PD gains inside the XY-to-reference-orientation block are kept the same at P= [-0.24, 0.24] and D= [0.1, -0.1]. This is shown in the following image.

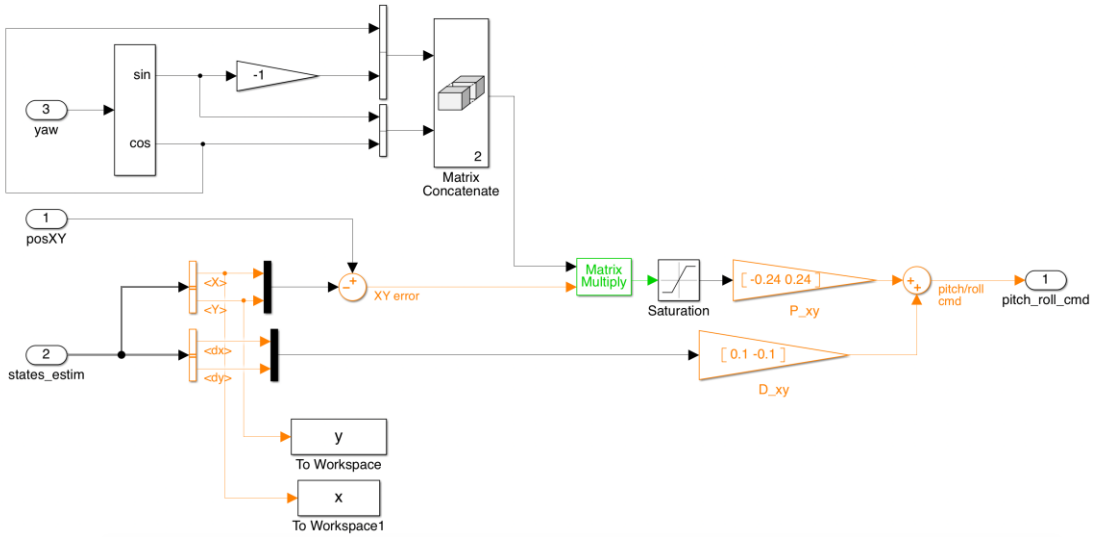


Figure 27: XY-to-reference=orientation block showing P= [-0.24, 0.24] and D= [0.1 -0.1]

The controller takeoff gain is set at 0.04 as obtained from previous test. The starting PD gains were set at P=0.8 and D=0.3. If we check the Z axis output against time, we will get Graph 6 as a result. The new PD gains selected for the test re P= 1.1 and D= 0.31. this is shown in the following image.

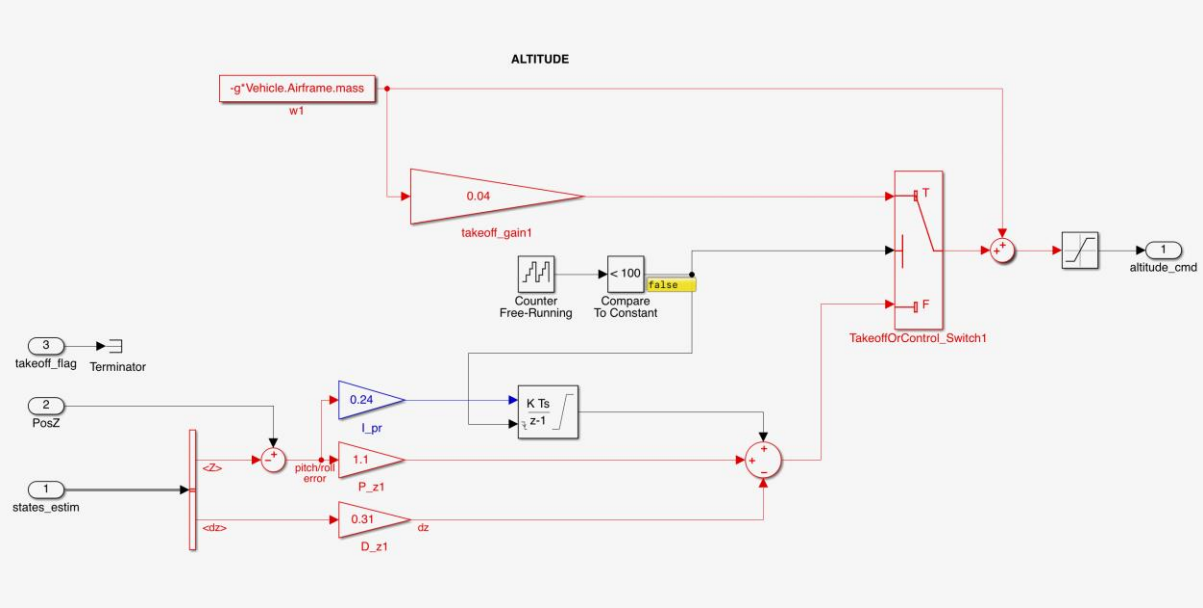
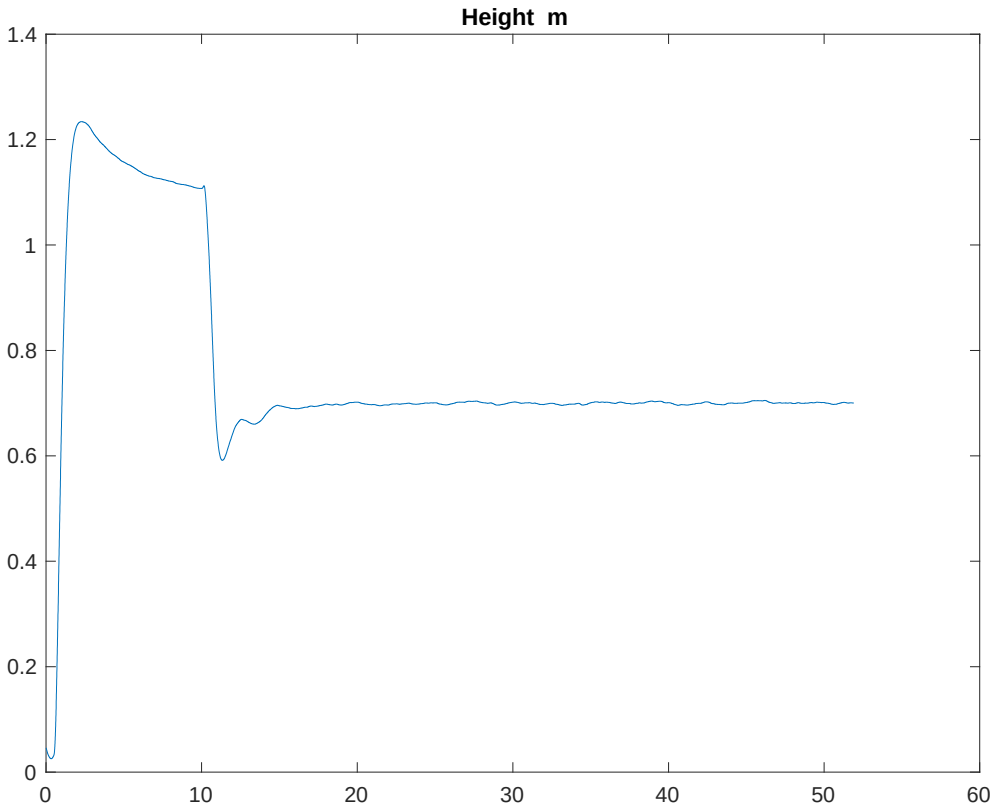


Figure 28: Gravity feedforward/equilibrium thrust block showing P= 1.1 and D= 0.31

After running the simulation for T= 50 the following Z axis output is obtained.



Graph 10: Inspecting the gravity feedforward/equilibrium thrust PD gains Z axis output against time for P= 1.1 and D= 0.31.

From Graph 10 we can conclude that these values for PD gains have resulted in a considerable amount of overshoot. The settling time and stability became better than the previous tests. For better stability experiments were done with different sets of PD gains as well. No changes were made in the PD gains inside of the XY-to-reference-orientation block. They remained same at P= P= [-0.24, 0.24] and D= [0.1, -0.1]. The PD gains inside the gravity feedforward/equilibrium block used were P= 0.21 and D= 0.31. this can be seen in the following image.

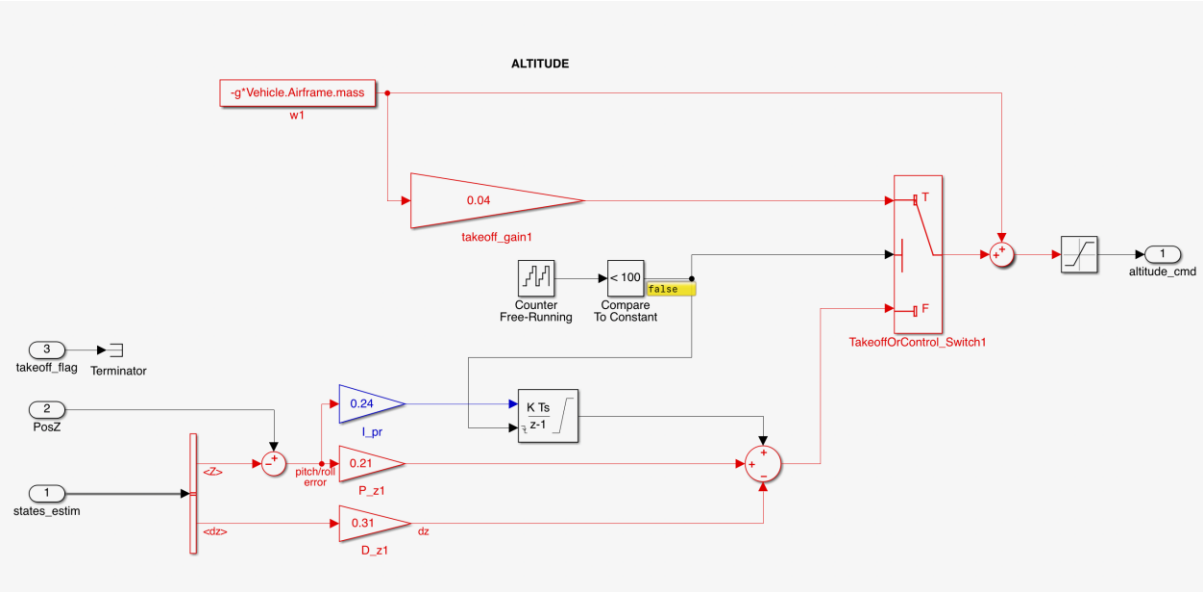
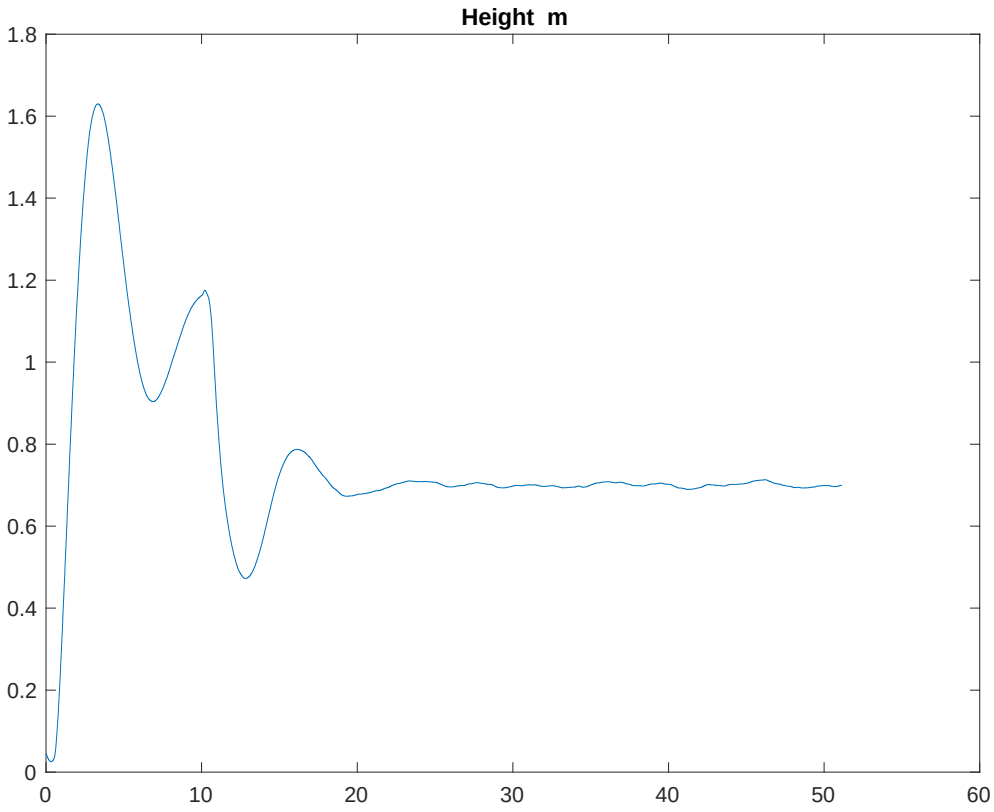


Figure 29: Gravity feedforward/equilibrium thrust block showing P= 0.21 and D= 0.31

After running the simulation for T= 50 the following Z axis output is obtained.



Graph 11: Inspecting the gravity feedforward/equilibrium thrust PD gains Z axis output against time for P= 0.21 and D= 0.31.

After analysing Graph 11 we can clearly conclude that the selected PD gains of P= 0.21 and D= 0.31 will not do anything good to the quadcopter. The Z axis output against time shows a very big overshoot even above 1.6 meters for the initial hover height of 1.1 meters. The stability and settling time also aren't that good. This means a low value of P gain inside the gravity feedforward/equilibrium thrust block is not desirable. The next test was performed by changing both the PD gains inside the gravity feedforward/equilibrium thrust block. The new PD gains selected were P= 0.41 and D= 0.21. this can be seen in the following image.

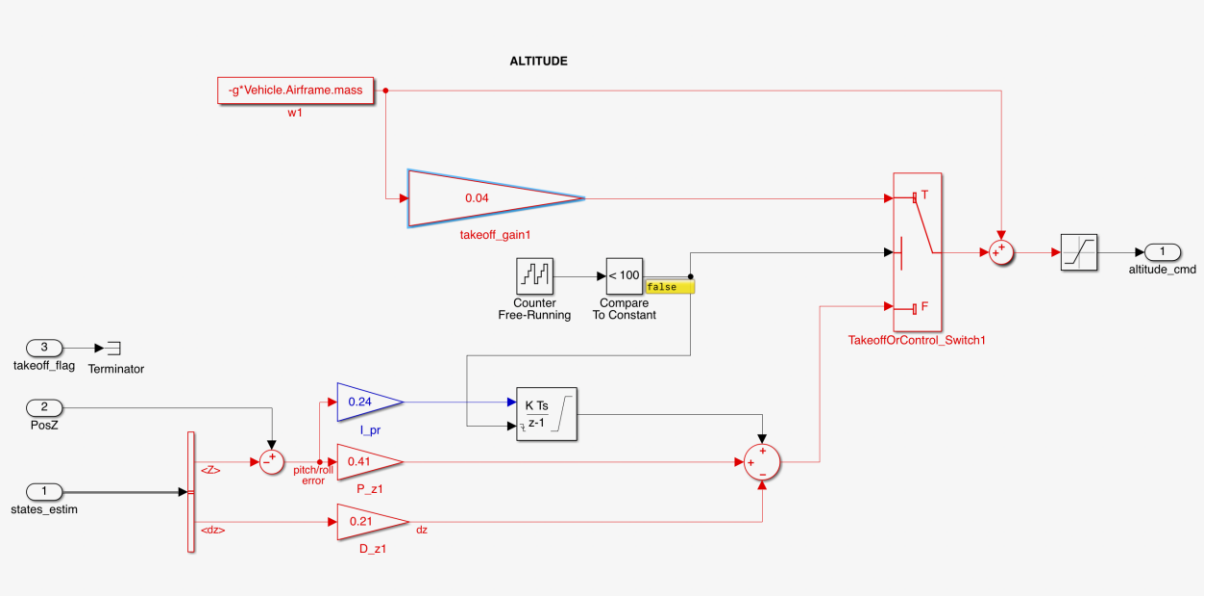
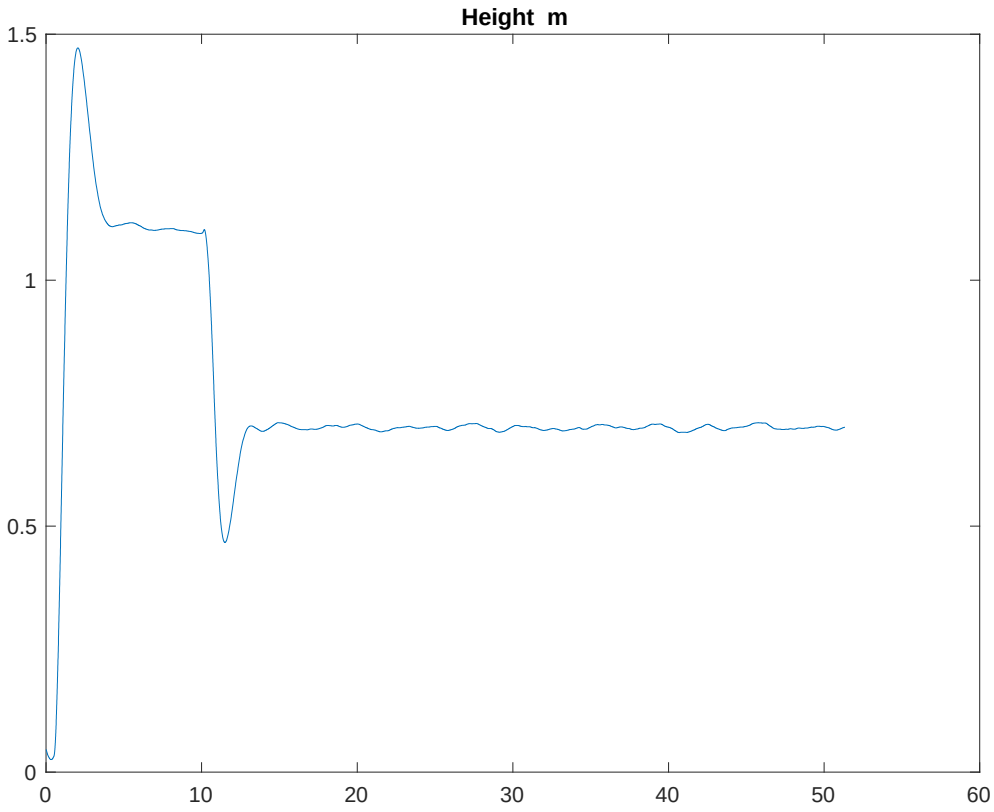


Figure 30: Gravity feedforward/equilibrium thrust block showing P= 0.41 and D= 0.21

After running the simulation for T= 50 the following Z axis output is obtained.



Graph 12: Inspecting the gravity feedforward/equilibrium thrust PD gains Z axis output against time for P= 0.41 and D= 0.21.

After analysing Graph 12 we can conclude that the new PD gains gave better results over the previous test but still have a bigger overshoot and poor settling time along with poor stability. For the next test the D gain was increased while keeping P gain the same and the results were analysed. The new D gain selected was increased from 0.21 to 0.31. this can be seen in the following image.

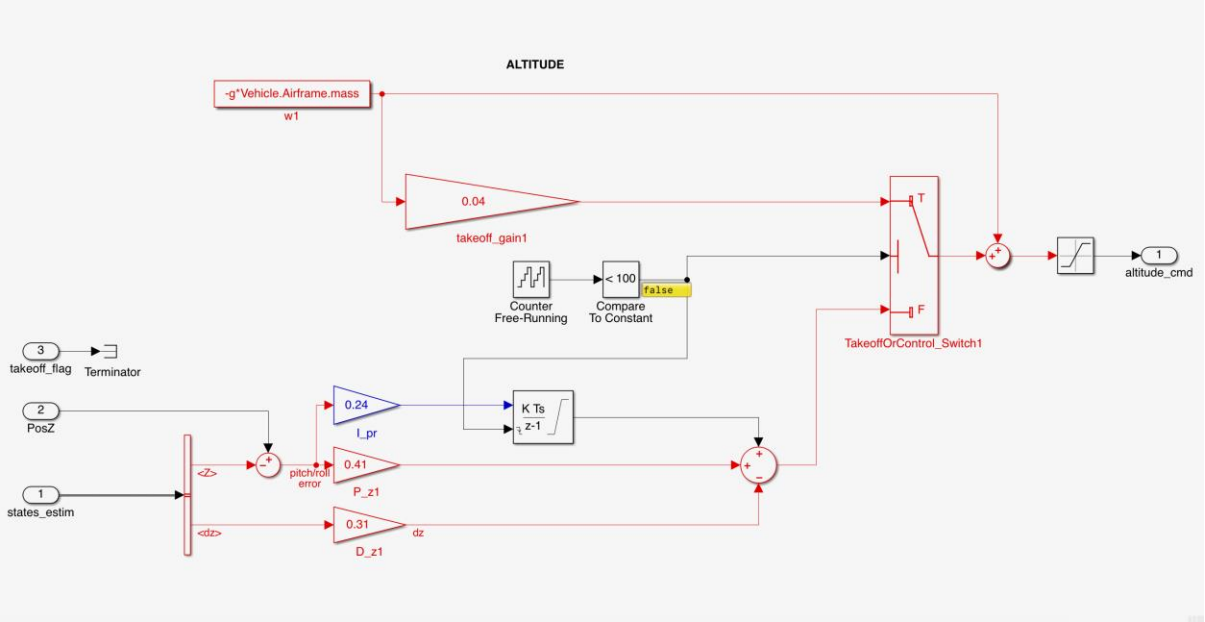
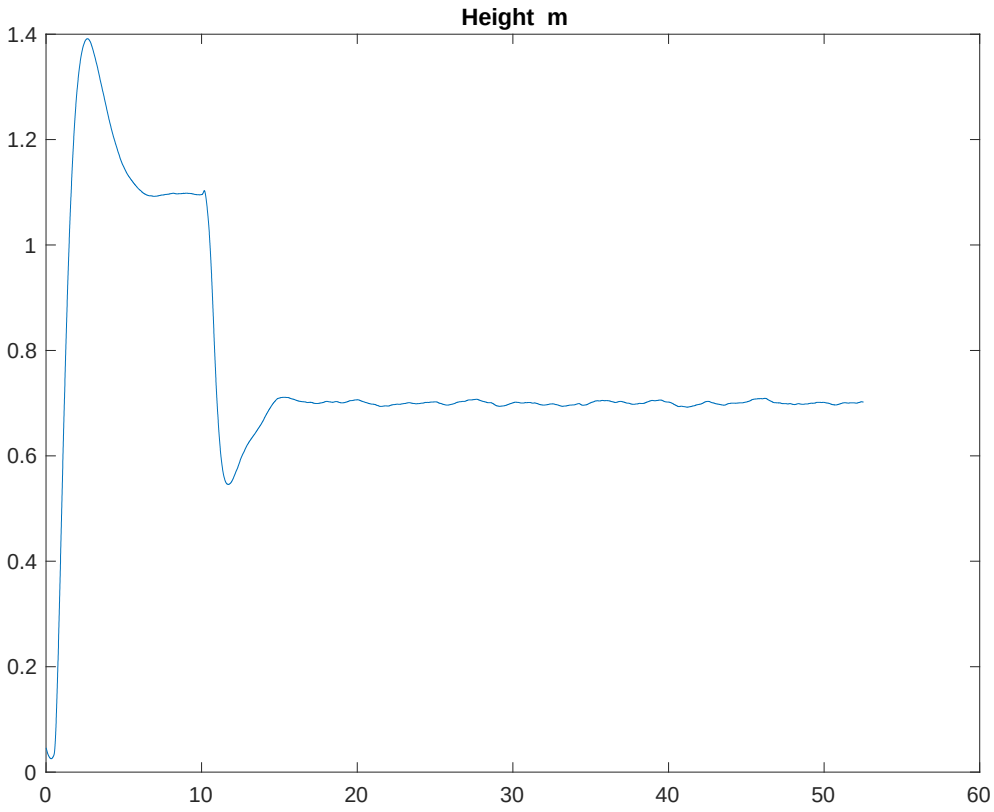


Figure 31: Gravity feedforward/equilibrium thrust block showing P= 0.41 and D= 0.31

After running the simulation for T= 50 the following Z axis output is obtained.



Graph 13: Inspecting the gravity feedforward/equilibrium thrust PD gains Z axis output against time for P= 0.41 and D= 0.31.

After analysing Graph 13 we can conclude that the overshoot has been reduced to a little less than 1.4 meters. The stability and settling time are better than the previous tests. After analysing this test, we can conclude that PD gains bigger than these values will give a better result in the performance of the Parrot Mambo minidrone. The new PD gains for gravity feedforward/equilibrium thrust block selected are P= 0.8 and D= 0.41. all the other gains in other blocks remain unchanged. The following image shows P= 0.8 and D= 0.41 inside gravity feedforward/equilibrium thrust block.

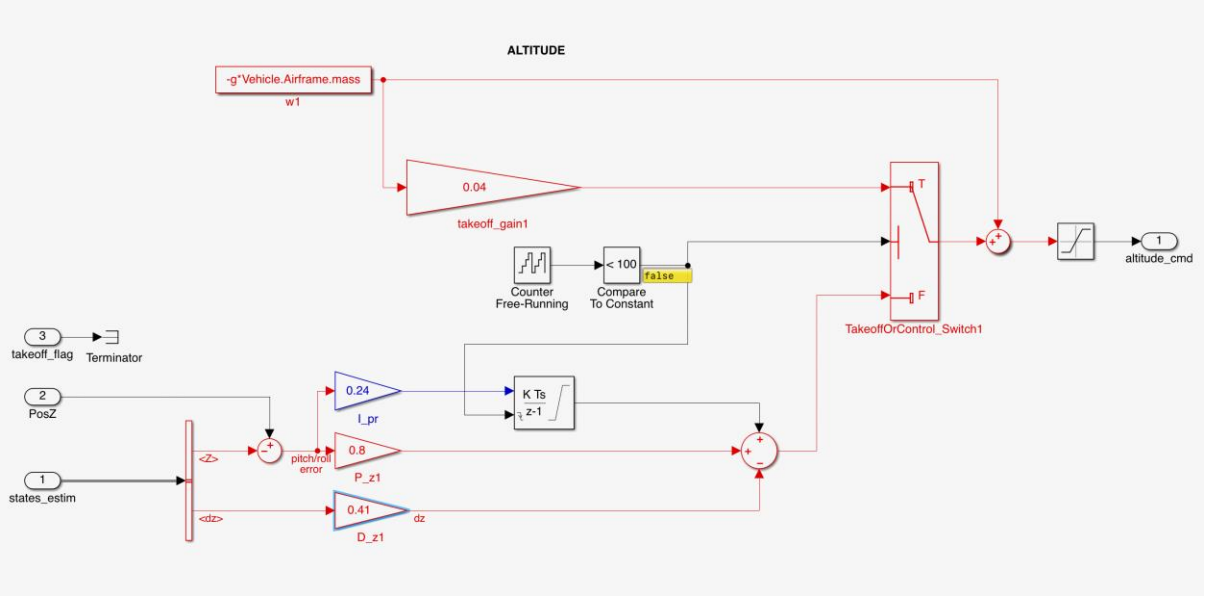
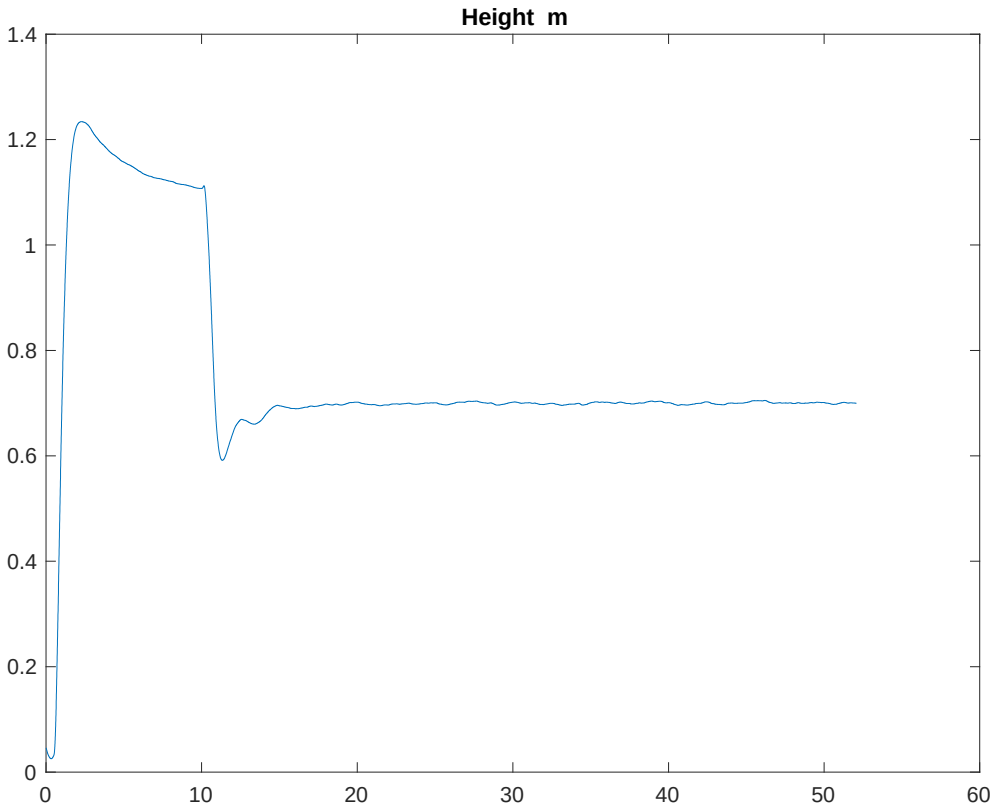
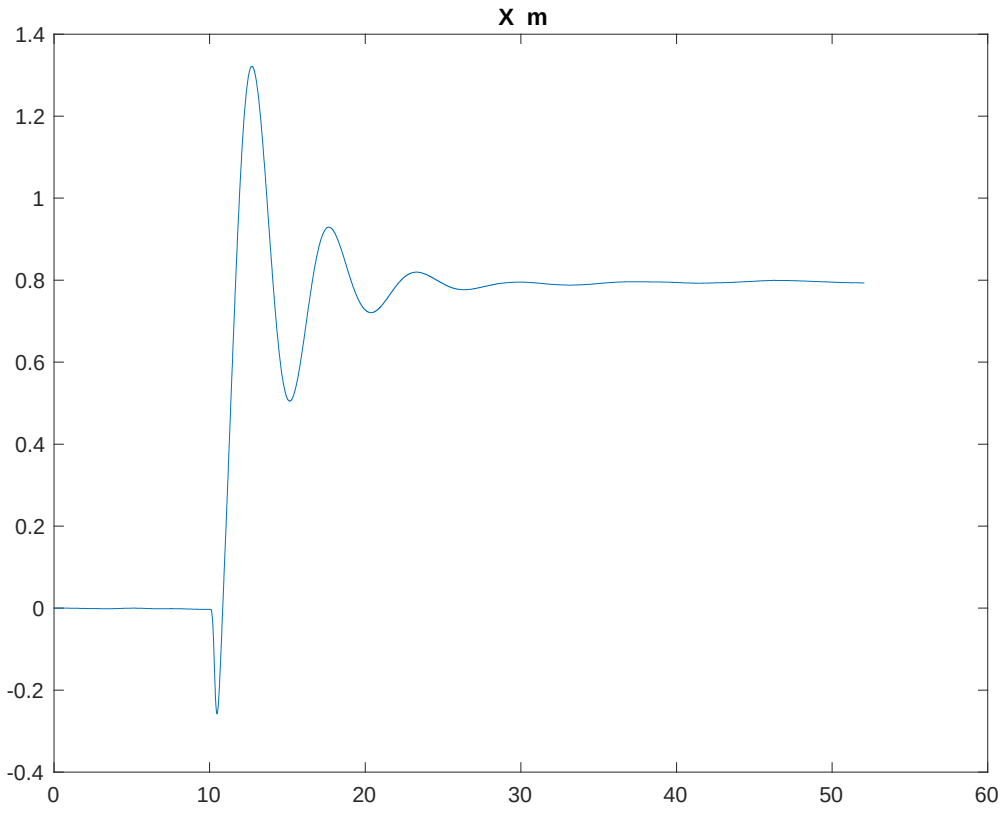


Figure 32: Gravity feedforward/equilibrium thrust block showing P= 0.8 and D= 0.41

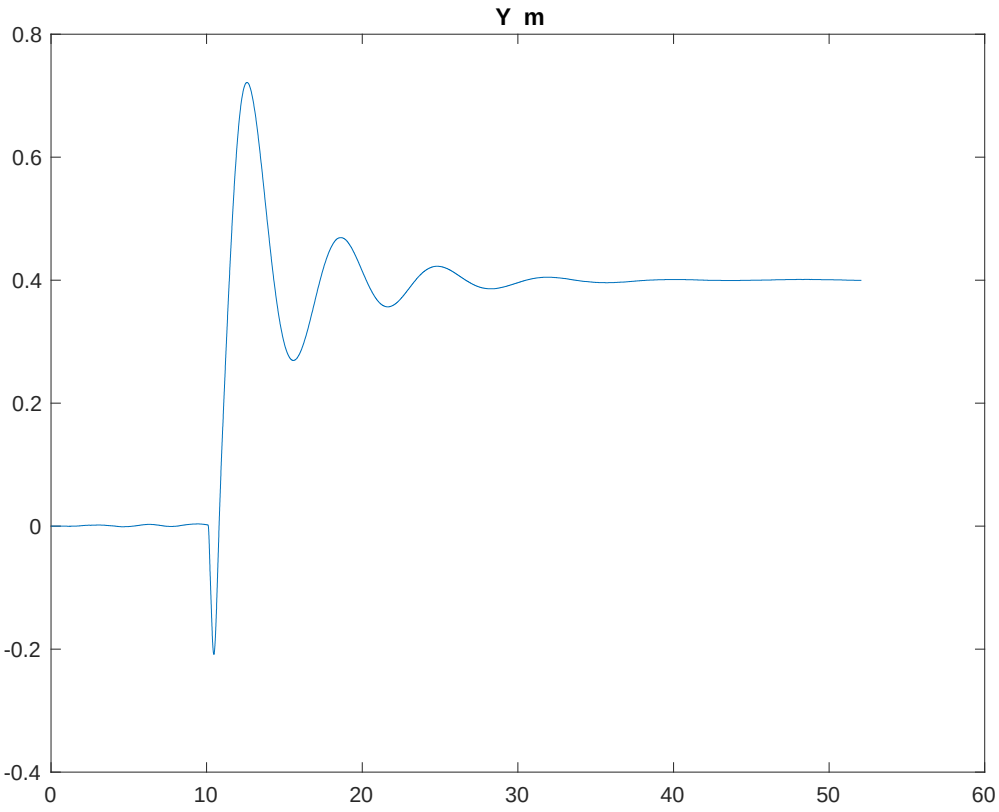
After running the simulation for T=50 the following graphs were obtained.



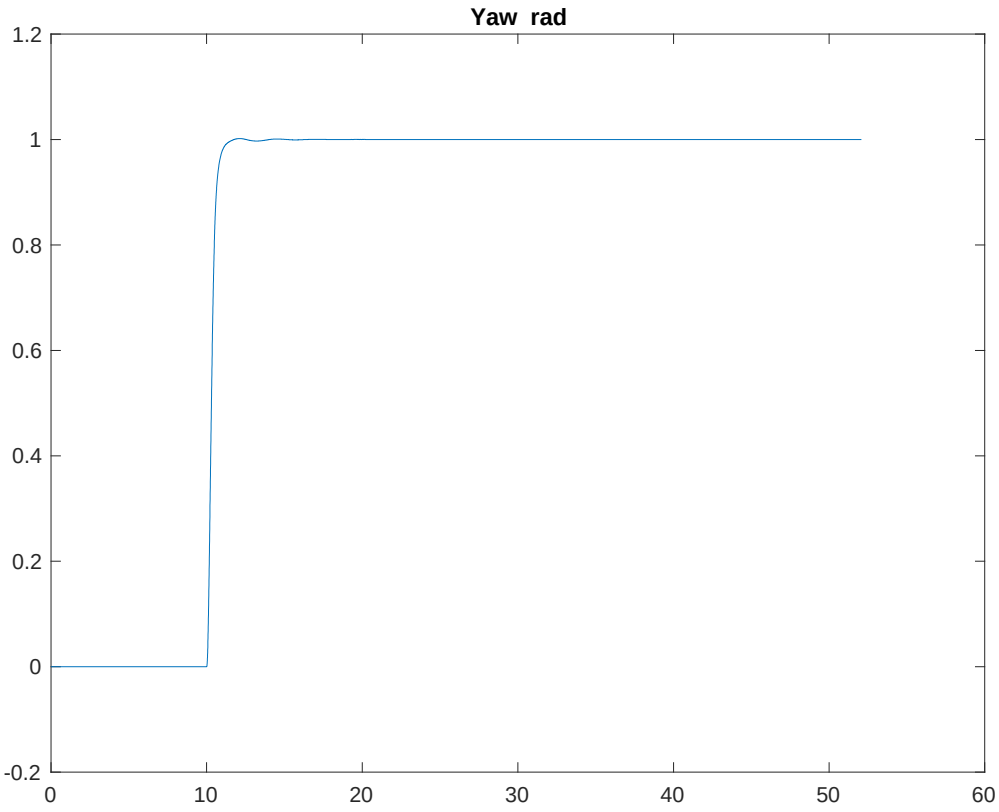
Graph 14: Inspecting the gravity feedforward/equilibrium thrust PD gains Z axis output against time for P= 0.8 and D= 0.41.



Graph 15: Inspecting the gravity feedforward/equilibrium thrust PD gains X axis output against time for P= 0.8 and D= 0.41.



Graph 16: Inspecting the gravity feedforward/equilibrium thrust PD gains Y axis output against time for P= 0.8 and D= 0.41.



Graph 17: Inspecting the gravity feedforward/equilibrium thrust PD gains yaw output against time for P= 0.8 and D= 0.41.

After inspecting Graph 14 we can clearly see that the overshoot has dropped significantly and is in the acceptable range. Graph 14 also shows better stability and settling time. With these values of PD gains inside the gravity feedforward/equilibrium thrust block the quadcopter performs and hovers quite well. Graph 15 shows the gravity feedforward/equilibrium thrust X axis output against time. Graph 16 shows the gravity feedforward/equilibrium thrust Y axis output against time. Graph 17 shows the gravity feedforward/equilibrium thrust yaw output against time for the selected gains.

We can clearly conclude that a lower value of PD gains for gravity feedforward/equilibrium thrust block is not desirable. This will cause instability and poor performance of the quadcopter which may result in crash. Hence the selected values of PD gains give better performance.

4.3 Inspection of XY-to-reference-orientation PD gains

After changing the default controller takeoff gains and selecting proper values for PD gains inside gravity feedforward/equilibrium thrust block the simulated quadcopter takes off and reach its initial height very quickly only with a smaller amount of overshoot. In this section, PD gains inside the XY-reference-orientation block is analysed and tested on simulated quadcopter model. Inspection of overshoot, settling time and stability is tested for Z axis, X axis, Y axis and Yaw against time for simulated Parrot Mambo quadcopter. Till now the tests were done with the gains inside the XY-reference-orientation block set at P= [-0.24,0.24] and D= [0.1, -0.1] as shown in the following image.

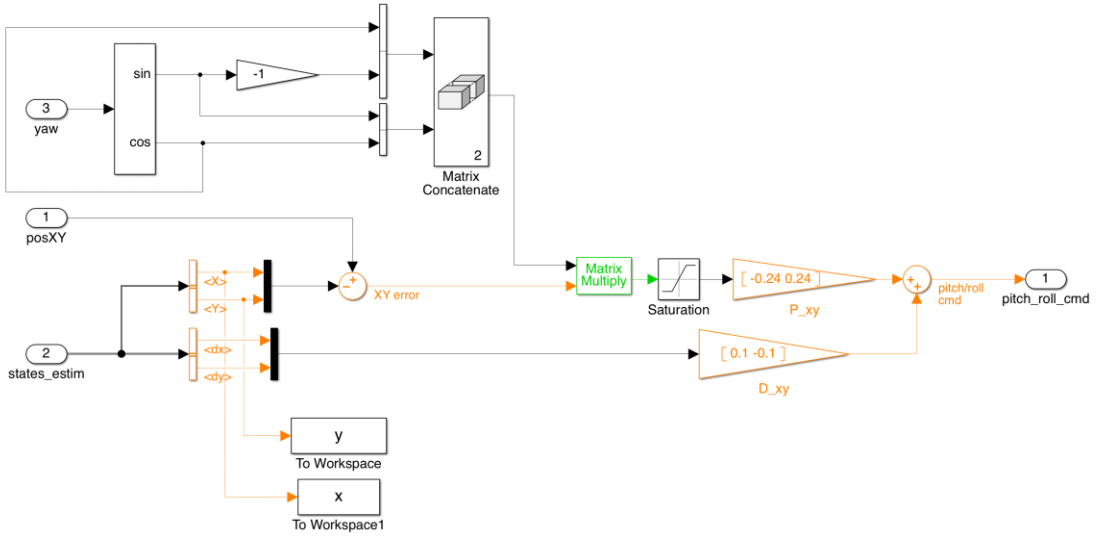


Figure 33: XY-to-reference=orientation block showing P= [-0.24, 0.24] and D= [0.1 -0.1]

The new gains obtained in the previous results helped in better performance of the quadcopter and hence those gains will be used in this test. The controller takeoff gain is set at 0.04. The PD gains inside the gravity feedforward/equilibrium thrust block is set at P= 0.8 and D= 0.41. This can be seen in the following image.

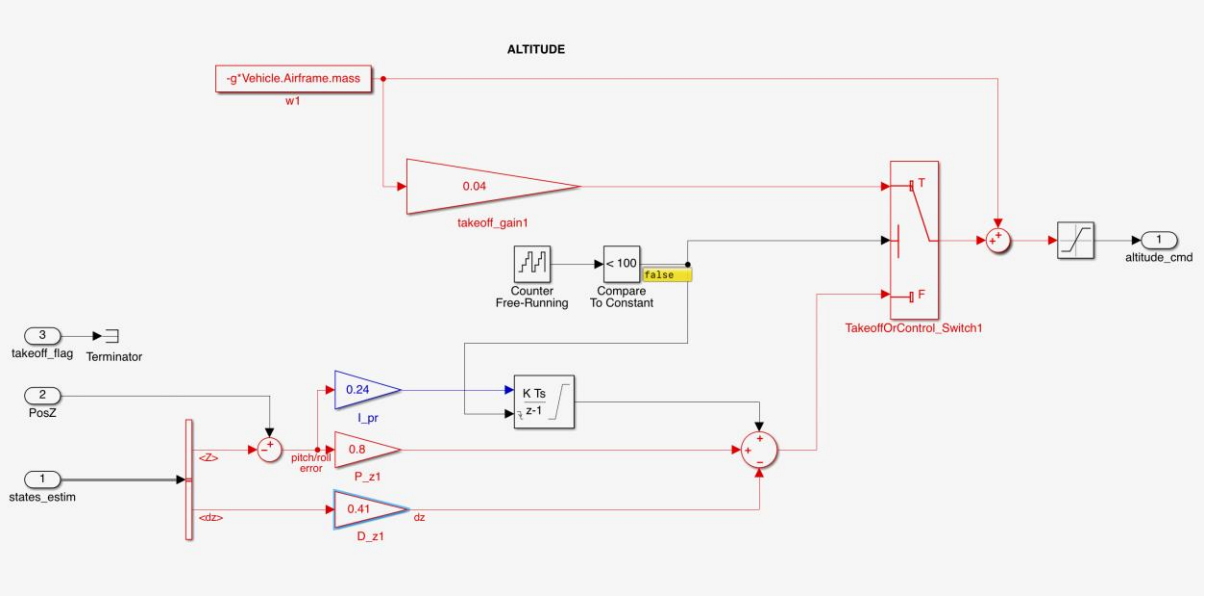


Figure 34: Gravity feedforward/equilibrium thrust block showing P= 0.8 and D= 0.41, controller takeoff gain =0.04

The first test was done with the increase in value of P gain and keeping the D gain constant inside the XY-to-reference orientation block. The P gain used is P= [-0.29, 0.29] and D gain at D= [0.1, -0.1]. This can be seen in the following image.

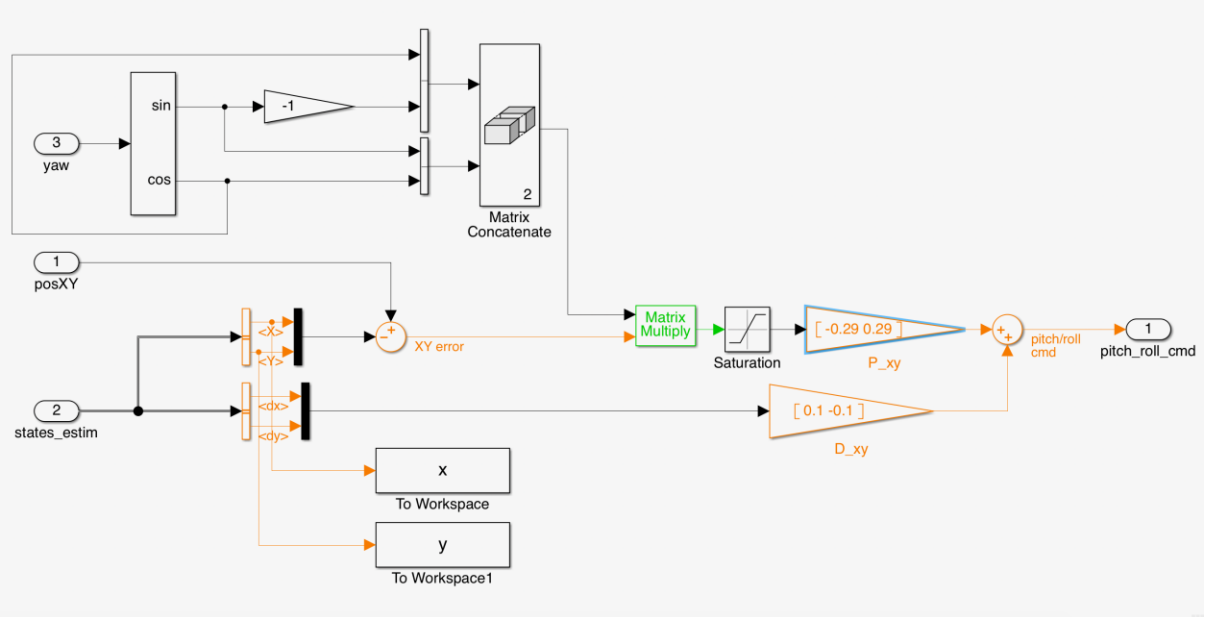
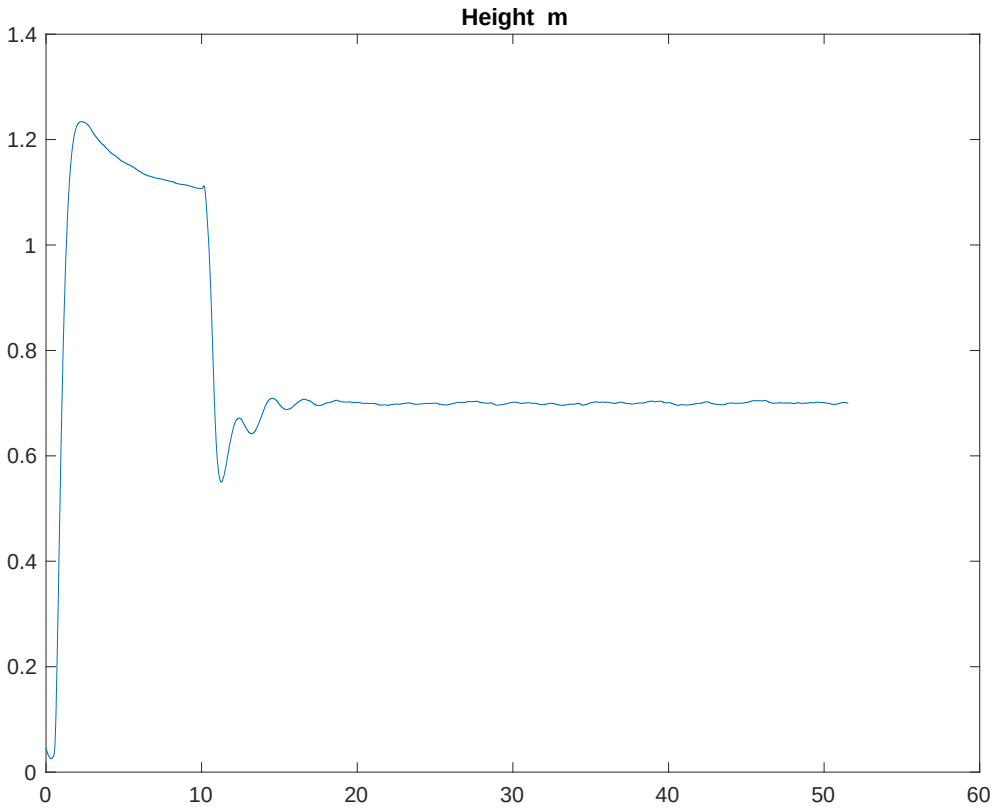
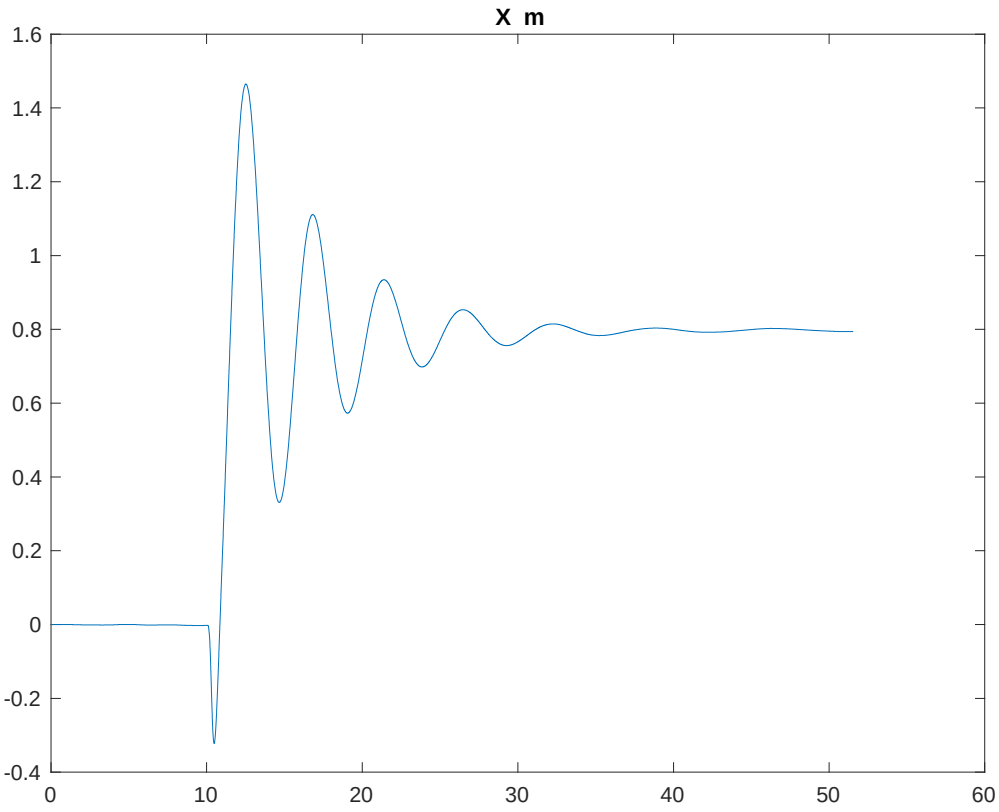


Figure 35: XY-to-reference-orientation block showing P= [-0.29, 0.29] and D= [0.1, -0.1]

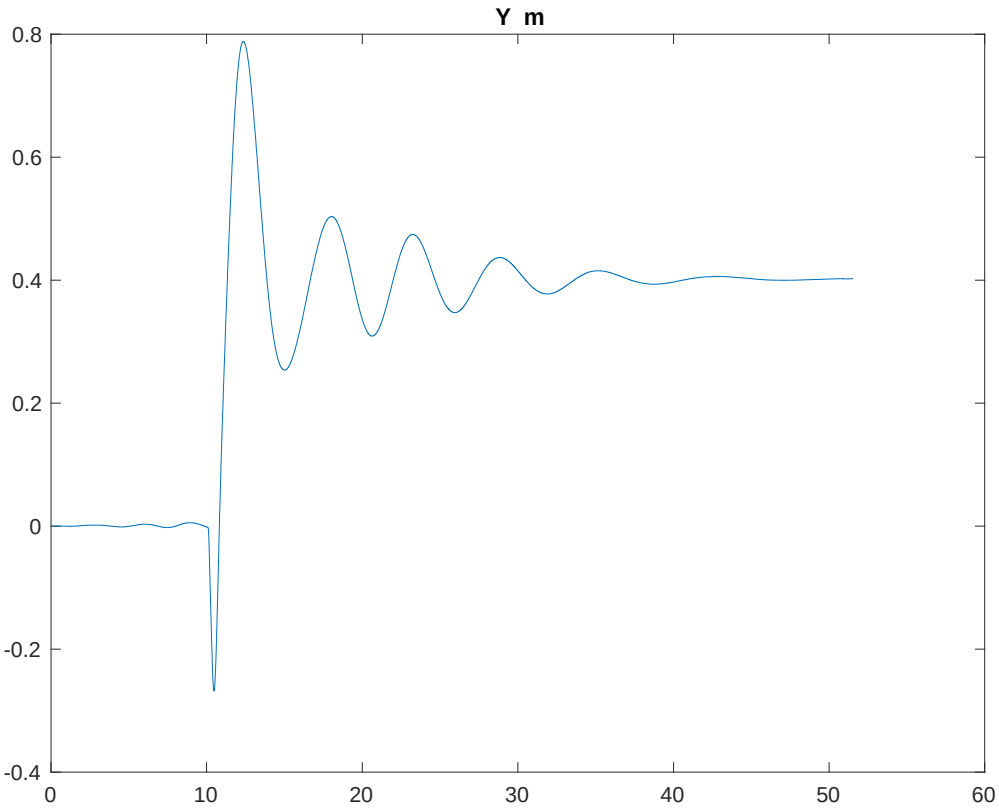
After running the simulation for T=50 the following graphs were obtained.



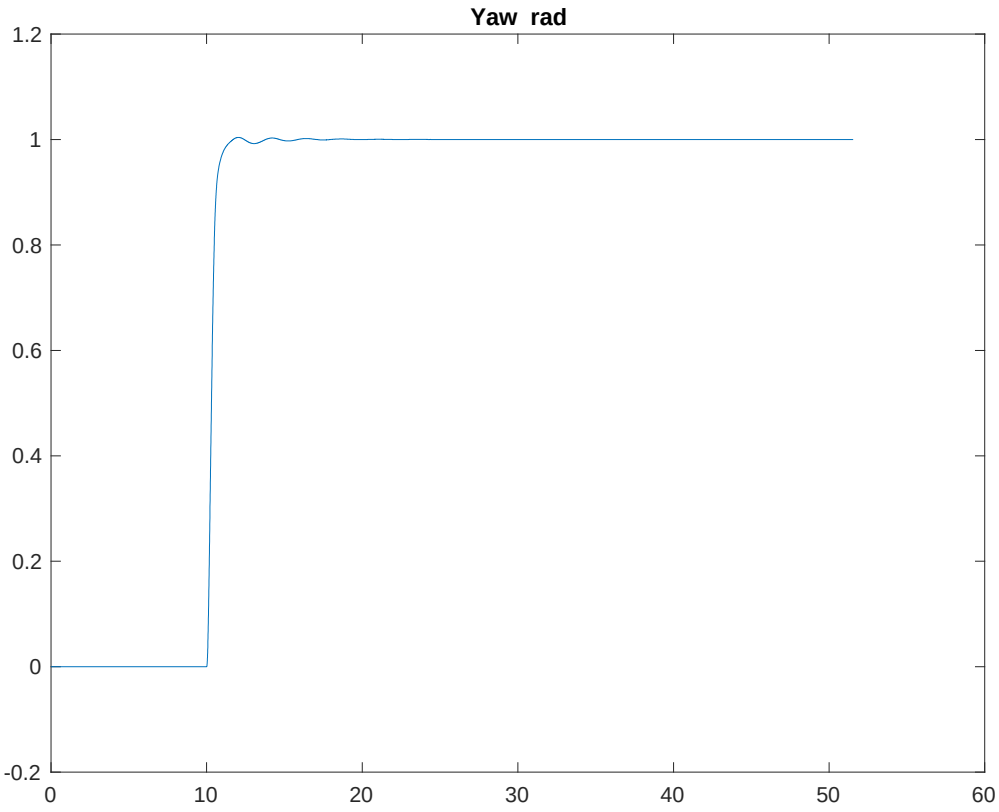
Graph 18: Inspecting the XY-to-reference-orientation PD gain Z axis output against time for P= [-0.29, 0.29] and D= [0.1, -0.1].



Graph 19: Inspecting the XY-to-reference-orientation PD gain X axis output against time for P= [-0.29, 0.29] and D= [0.1, -0.1].



Graph 20: Inspecting the XY-to-reference-orientation PD gain Y axis output against time for P= [-0.29, 0.29] and D= [0.1, -0.1].



Graph 21: Inspecting the XY-to-reference-orientation PD gain yaw output against time for P= [-0.29, 0.29] and D= [0.1, -0.1].

After analysing the Graph 18 we can say that there is not much impact on the overshoot, but the stability decreased a bit. Graph 19 and Graph 20 shows bigger overshoots in X axis and Y axis respectively becoming more stable over time. It is assumed that if the stability in X axis and Y axis increases it will result in better stability in Z axis. Graph 21 shows the yaw output against time for the selected PD gains. The next test was performed by lowering the P gain while keeping the D gain same. The new PD gains were set at P= [-0.08, 0.08] and D= [0.1, -0.1]. This is seen in the following image.

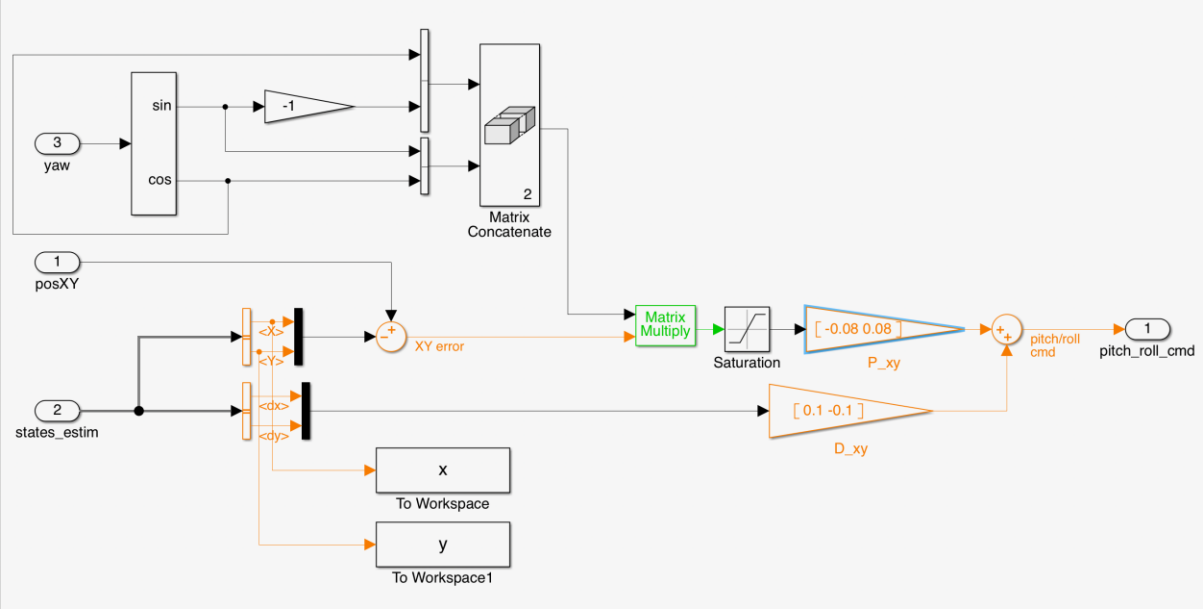
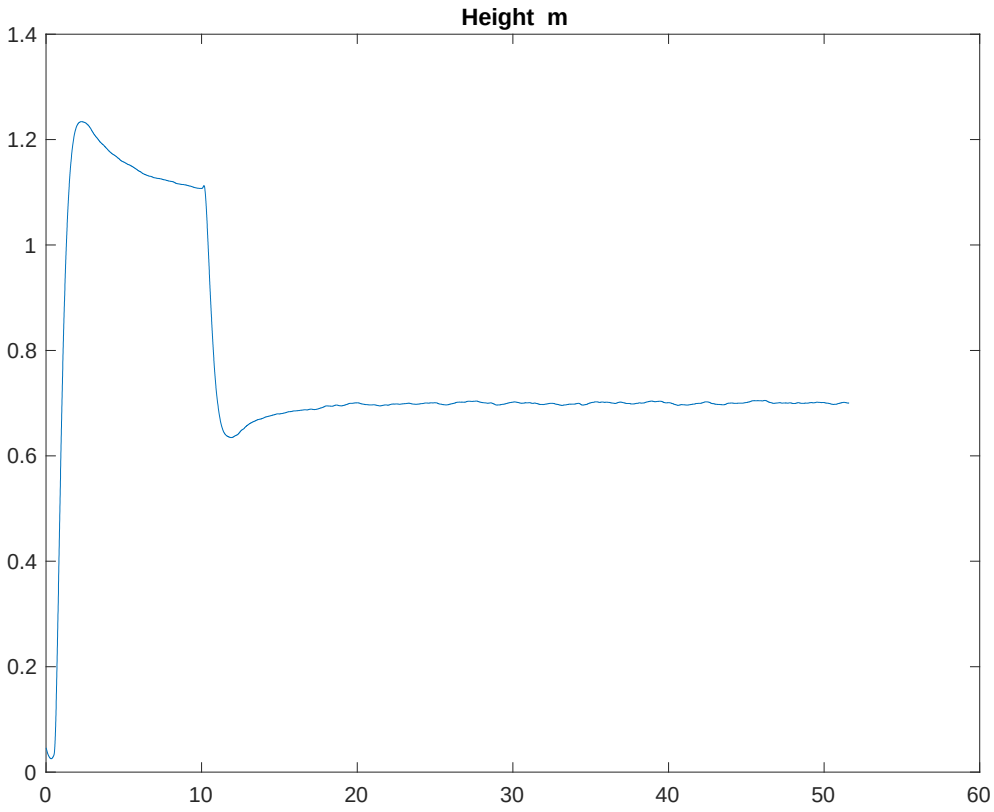
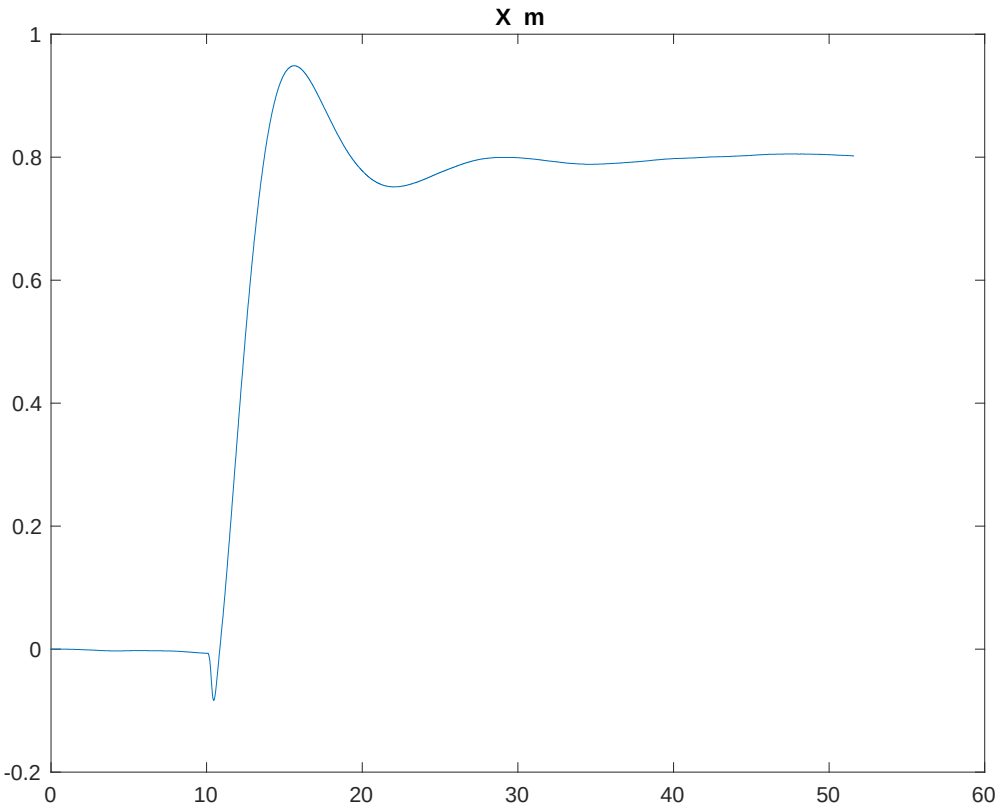


Figure 36: XY-to-reference-orientation block showing P= [-0.08, 0.08] and D= [0.1, -0.1]

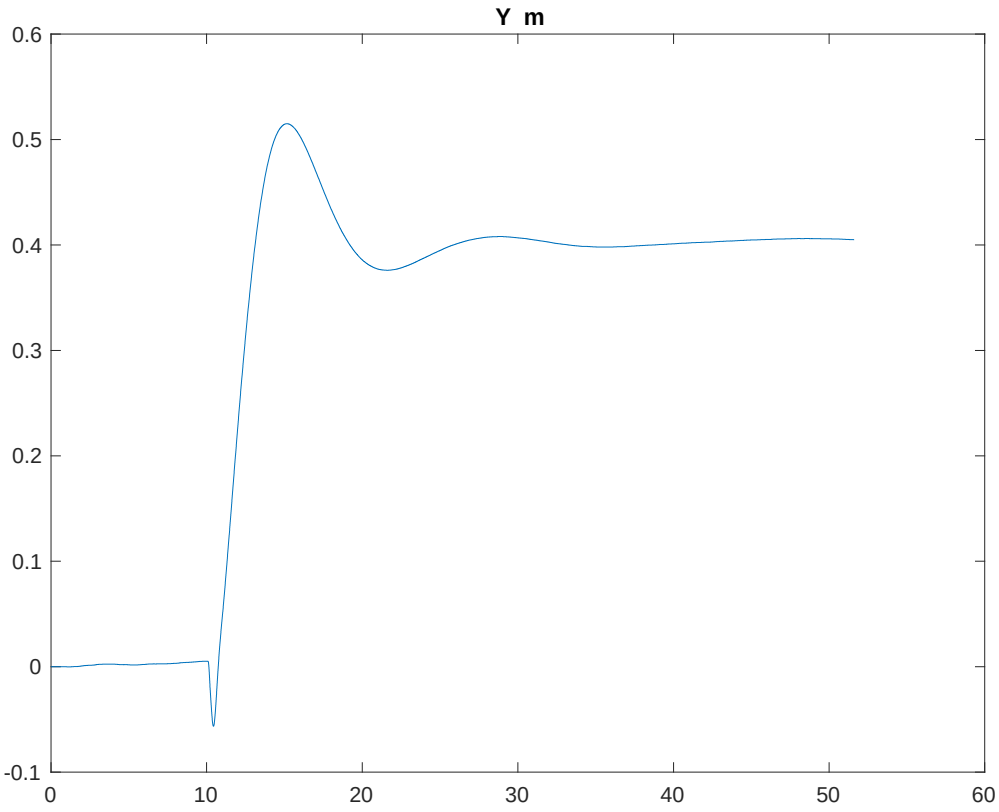
After running the simulation for T=50 the following graphs were obtained.



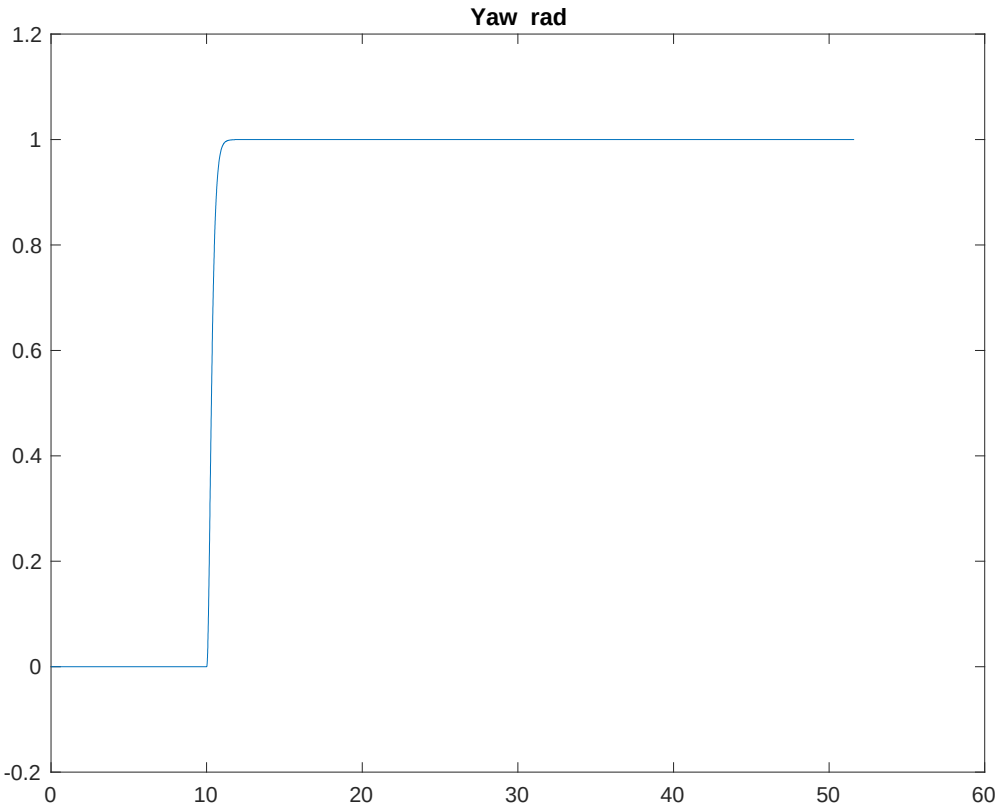
Graph 22: Inspecting the XY-to-reference-orientation PD gain Z axis output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]



Graph 23: Inspecting the XY-to-reference-orientation PD gain X axis output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]



Graph 24: Inspecting the XY-to-reference-orientation PD gain Y axis output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]



Graph 25: Inspecting the XY-to-reference-orientation PD gain yaw output against time for P= [-0.08, 0.08] and D= [0.1, -0.1]

After inspecting Graph 22 we conclude that the performance of the quadcopter has increased significantly. Graph 23 and Graph 24 indicates that the stability in X axis and Y axis has been better for the minidrone. Low value of P gain is found to give better stability in Z axis. Graph 25 shows yaw output against time for the selected PD gains. The next test is performed by increasing both P gain and D gain inside the XY-reference-orientation block by a very small value. The new P gain was set at P= [-0.15, 0.15] and D= [0.18, -0.18]. This is seen in the following image.

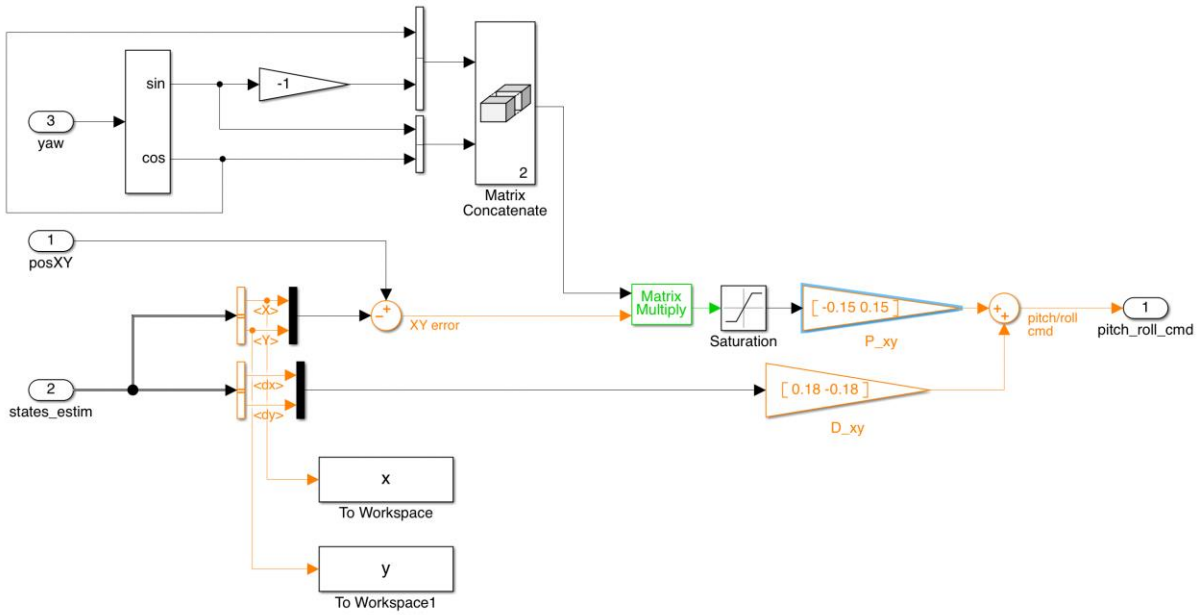
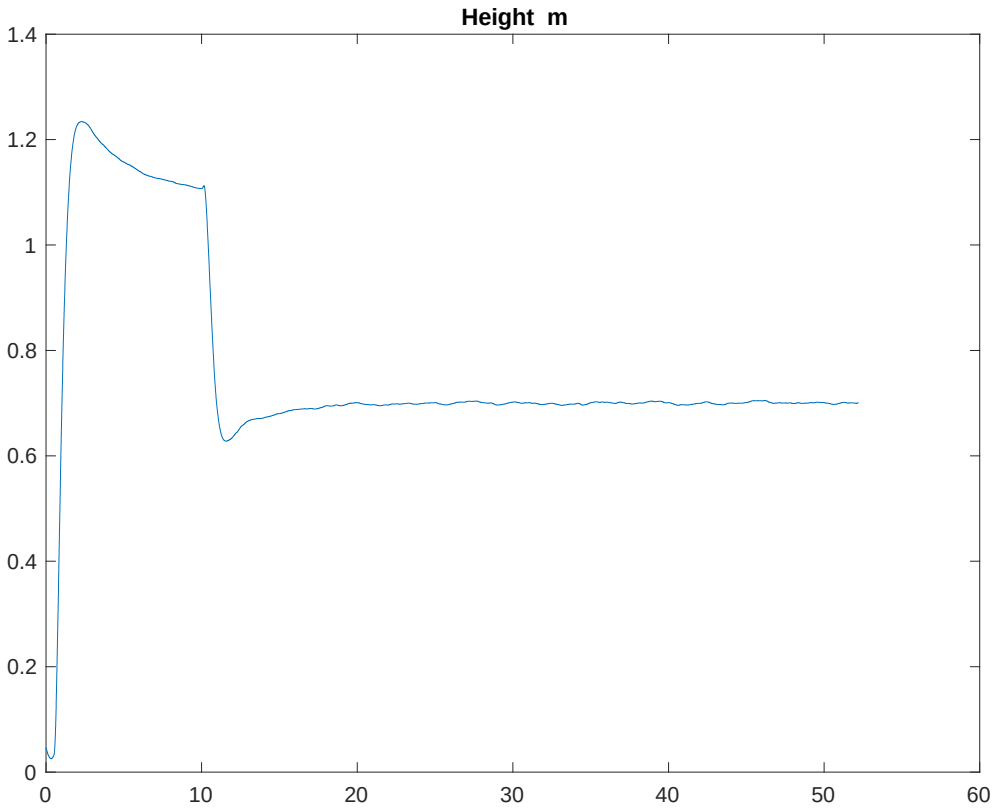
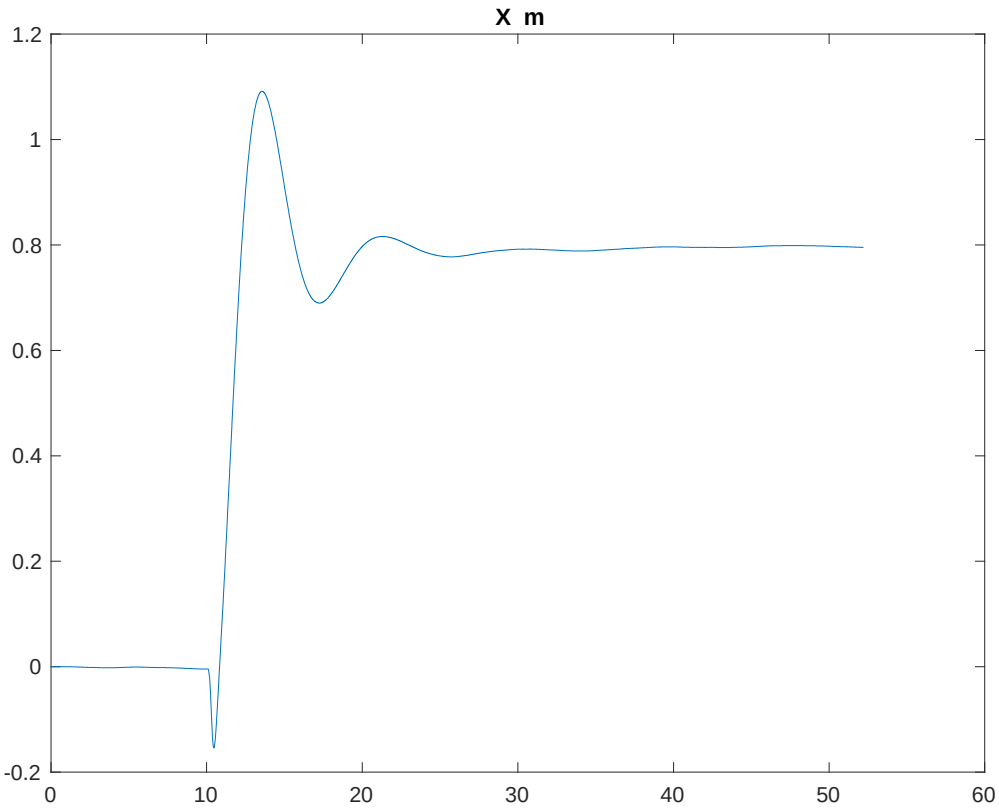


Figure 37: XY-to-reference-orientation block showing P= [-0.15, 0.15] and D= [0.18, -0.18].

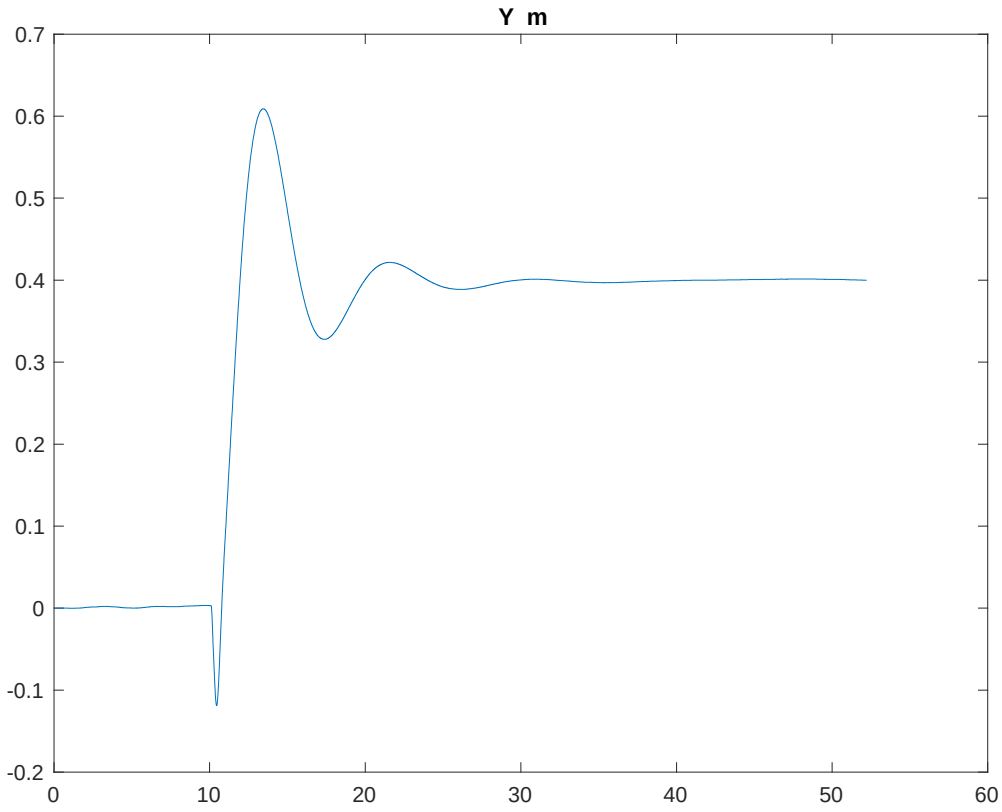
After running the simulation for T=50 the following graphs were obtained.



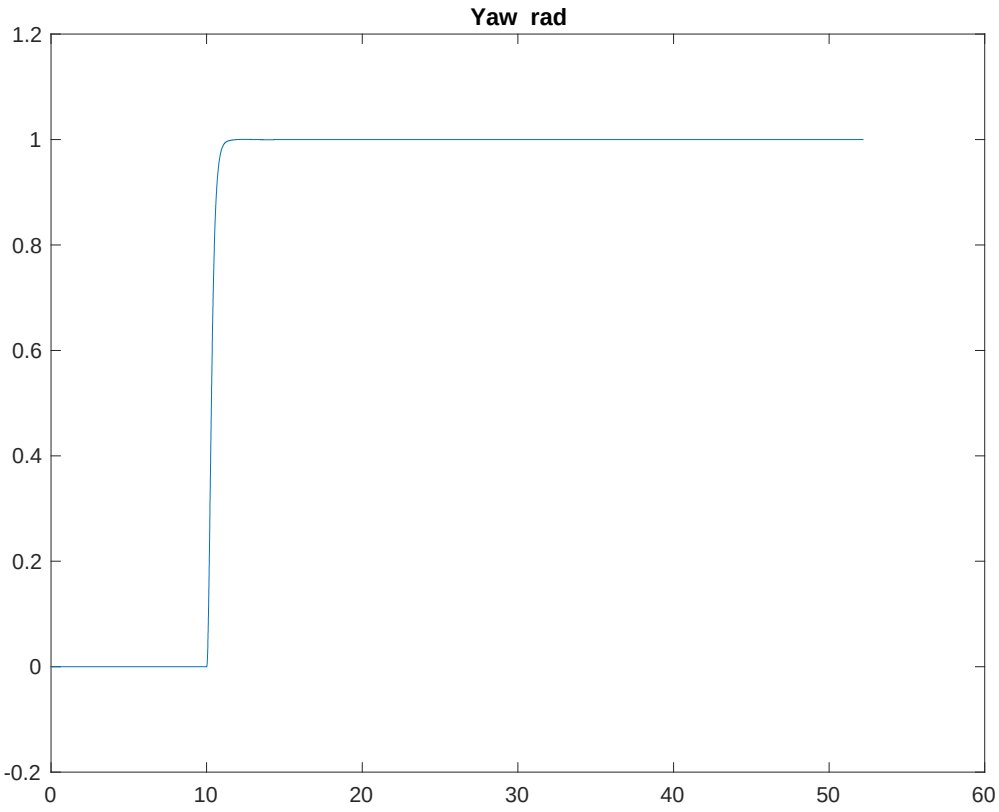
Graph 26: Inspecting the XY-to-reference-orientation PD gain Z axis output against time for P= [-0.15, 0.15] and D= [0.18, -0.18].



Graph 27: Inspecting the XY-to-reference-orientation PD gain X axis output against time for P= [-0.15, 0.15] and D= [0.18, -0.18].



Graph 28: Inspecting the XY-to-reference-orientation PD gain Y axis output against time for P= [-0.15, 0.15] and D= [0.18, -0.18].



Graph 29: Inspecting the XY-to-reference-orientation PD gain yaw output against time for P= [-0.15, 0.15] and D= [0.18, -0.18].

Analysing Graph 26 shows little to no change or we can say very negligible change in performance of quadcopter in Z axis. Graph 27 shows improved stability and settling time for X axis. Graph 28 shows improved stability and settling time for Y axis. Graph 29 shows yaw output against time for the selected PD gains for the quadcopter. We can conclude that these were the best gains found for a good and stable flight.

4.4 Initial Conclusion and Results

After analysing all the effects of different combinations of PD gains and takeoff gains on the simulated quadcopter for initial flight and hover, in the XY-to-reference-orientation block and gravity feedforward/equilibrium thrust block, the PD gains were determined. In gravity feedforward/equilibrium thrust block, the controller takeoff gain was finalised at 0.04, Proportional gain was finalised at P= 0.8 and Derivative gain was finalised at D= 0.41. In the XY-to-reference-orientation block, the Proportional gain was finalised at P= [-0.15, 0.15] and Derivative gain was finalised at D= [0.18, -0.18]. All these gains can be seen in the following images.

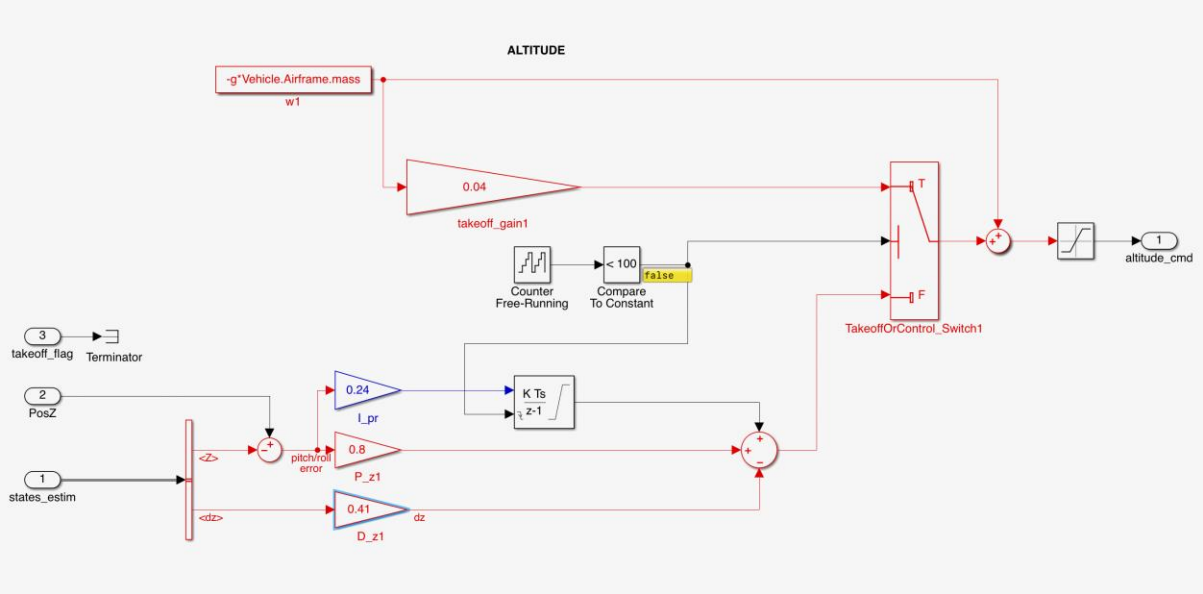


Figure 38: Gravity feedforward/equilibrium thrust block showing P= 0.8 and D= 0.41, controller takeoff gain =0.04

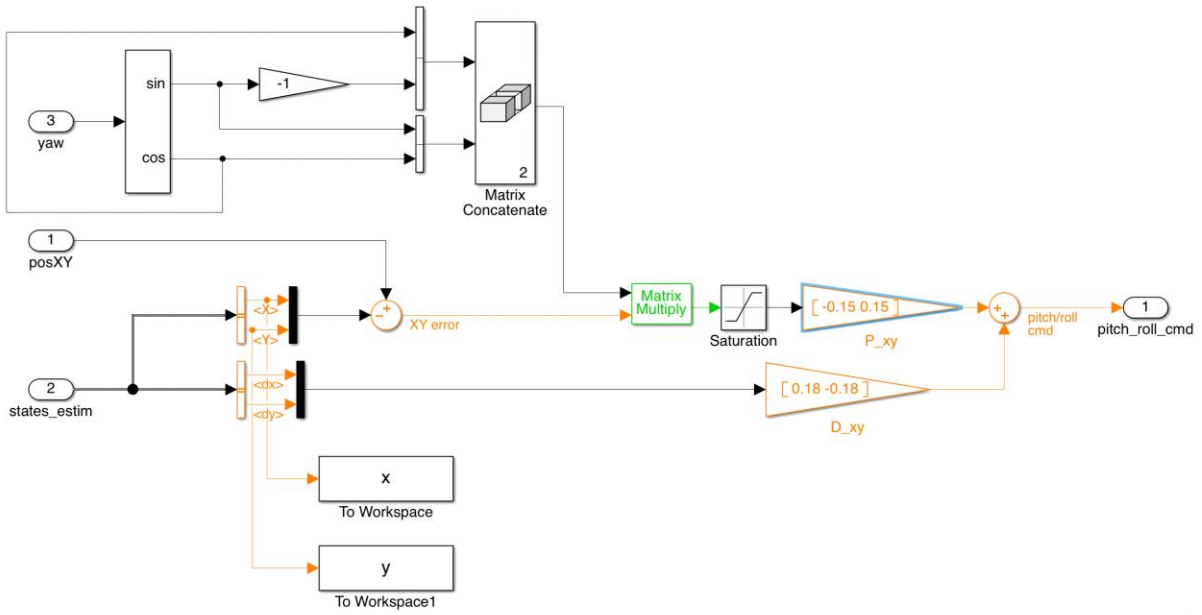


Figure. 39: XY-to-reference-orientation block showing P= [-0.15, 0.15] and D= [0.18, -0.18]

The values for PD gains found in above test may not be the perfect values. There may exist other combinations of PD gains which will give better flight and hover performance of the Parrot Mambo quadcopter. The PD gains selected in the experiments in this report were the best found gains while performing the simulations. Better flight performance with good stability and hover performance could have been achieved if more time was available.

4.5 Inspection of PD gains on different coordinate sets

The PD gains found in previous tests were tested on different sets of coordinates to check the minidrones stability in all the three axes. The tests were performed first with the default controller and PD gains. Then for the same set of coordinates new found PD gains were used. The simulation was ran for set time of T =20 and graphs for X axis against time, Y axis against time, Z axis against time and Yaw against time were plotted.

The first test was done with set point of X= 0.7; Y= 0; and Z= -0.7. This can be seen in the following image of Path Planning block.

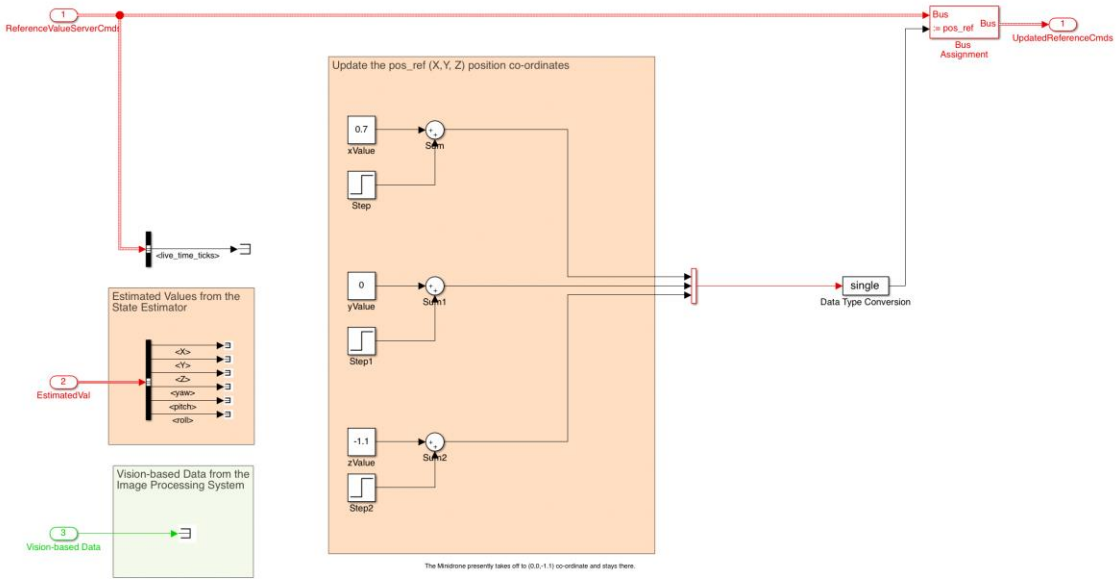
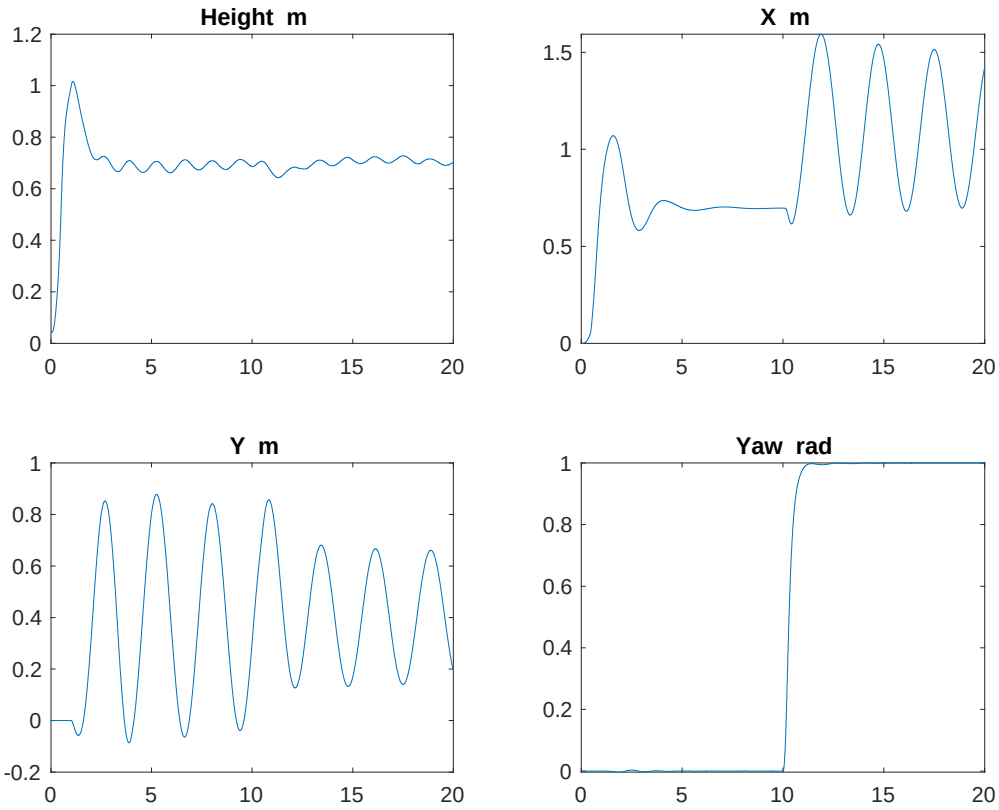
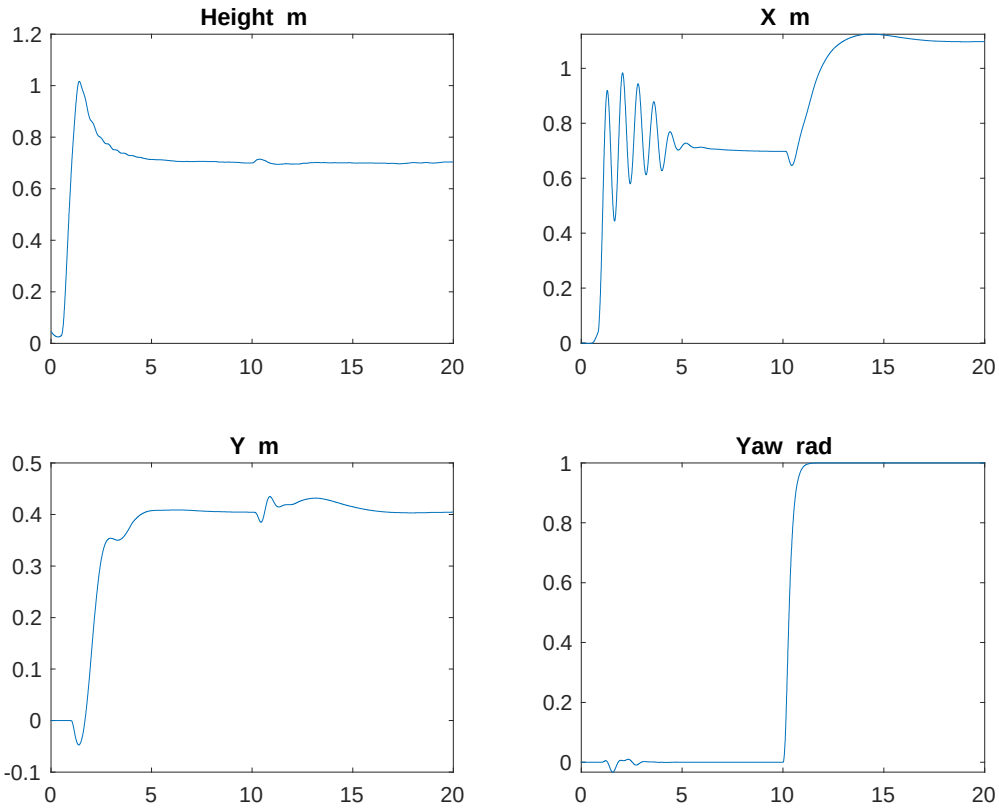


Figure 40: Path Planning block showing set points X= 0.7; Y= 0; and Z= -0.7

After running the simulation for T=20 the following graphs were obtained.



Graph 30: Output plots for default gains of set points X= 0.7; Y= 0; and Z= -0.7.



Graph 31: Output plots for new gains of set points X= 0.7; Y= 0; and Z= -0.7.

The second test was done with set point of X= 0.7; Y= 0.7; and Z= -0.7. This can be seen in the following image of Path Planning block.

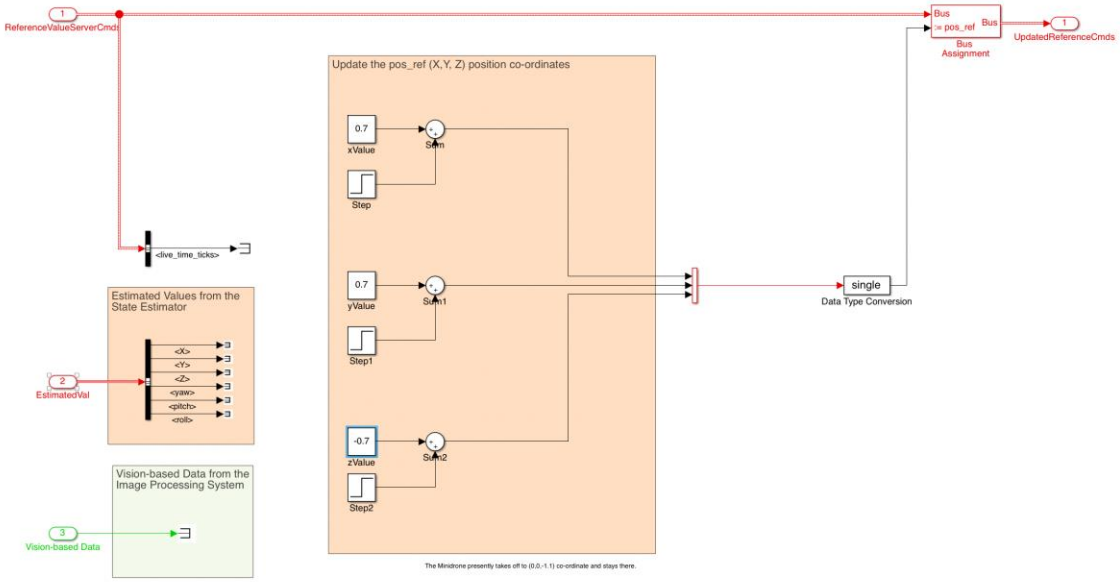
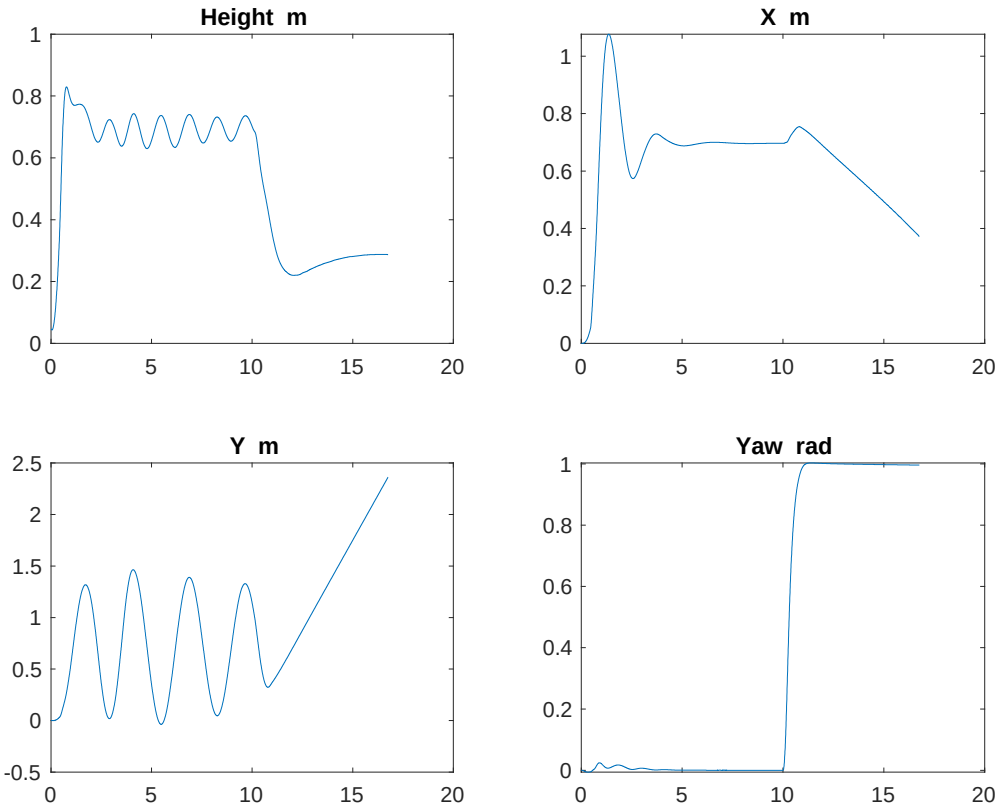
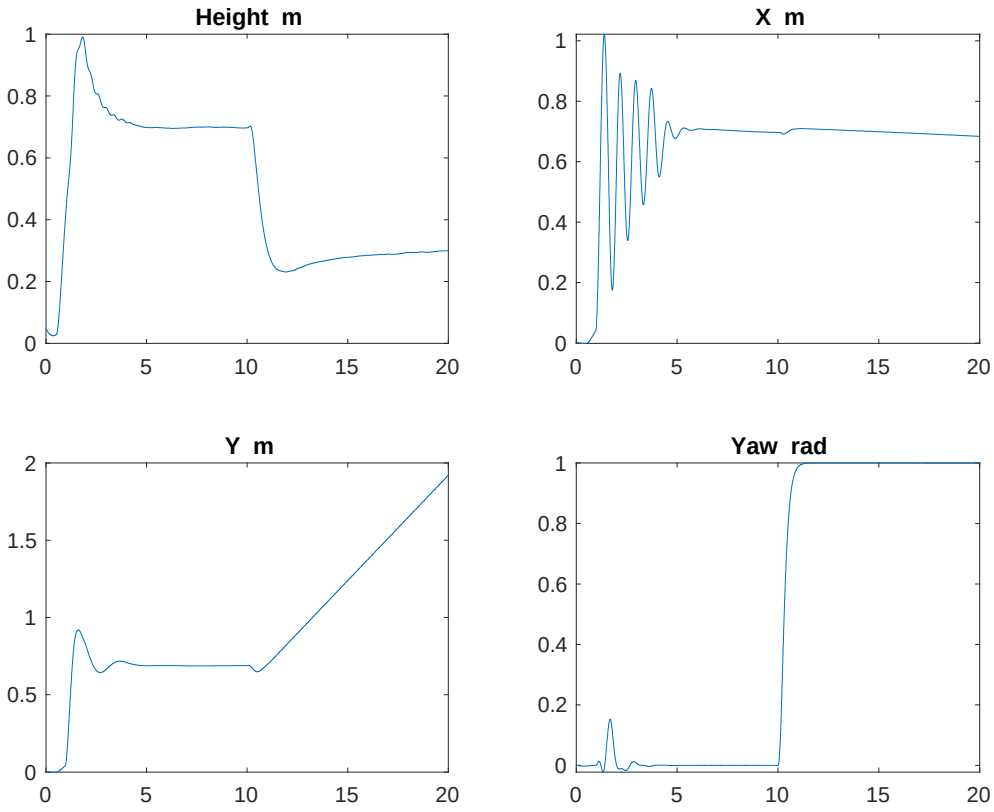


Figure 41: Path Planning block showing set points X= 0.7; Y= 0.7; and Z= -0.7

After running the simulation for T=20 the following graphs were obtained.



Graph 32: Output plots for default gains of set points X= 0.7; Y= 0.7; and Z= -0.7.



Graph 33: Output plots for new gains of set points X= 0.7; Y= 0.7; and Z= -0.7.

The third test was done with set point of X= 0; Y= 1; and Z= -1.1. This can be seen in the following image of Path Planning block.

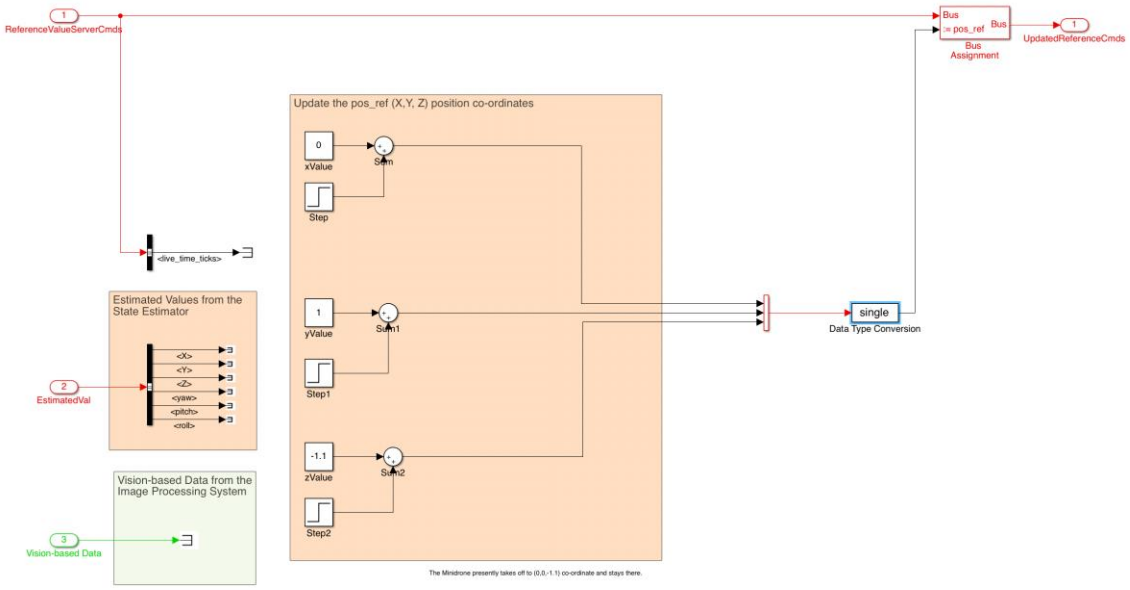
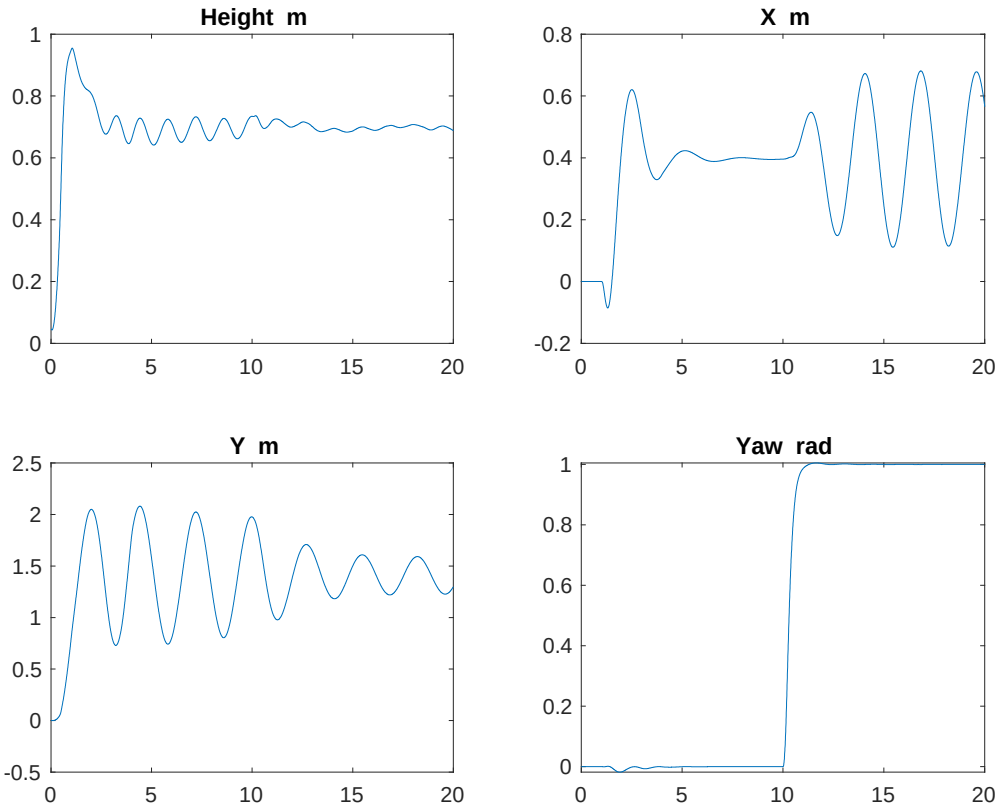
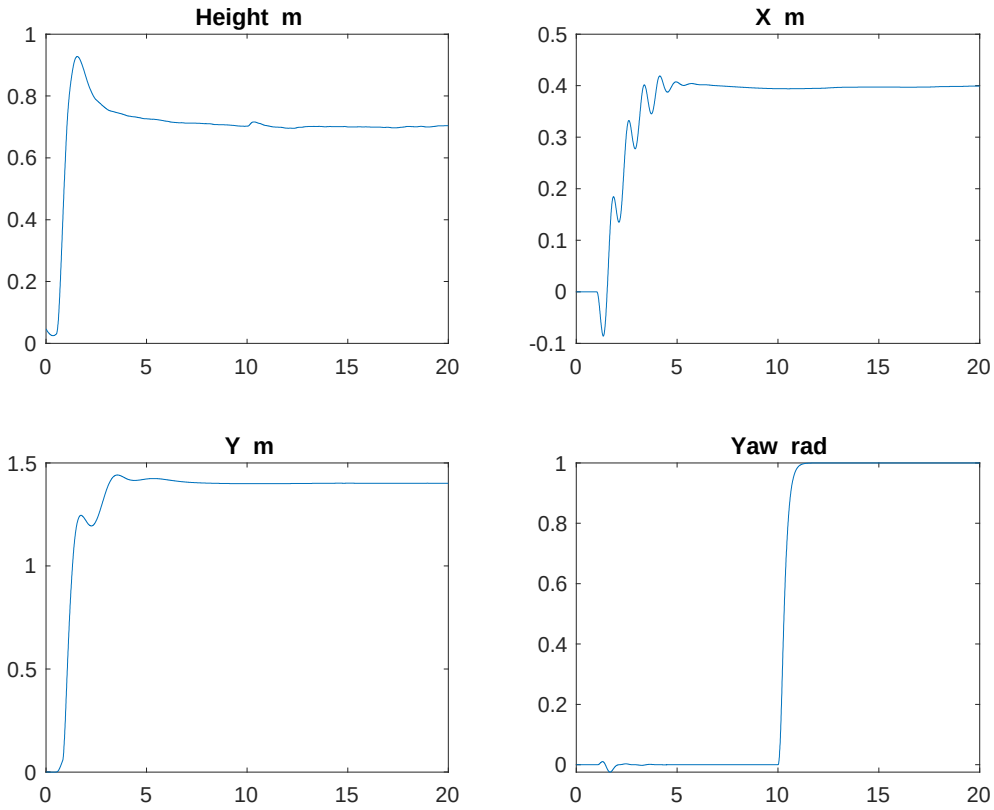


Figure 42: Path Planning block showing set points X= 0; Y=1; and Z= -1.1

After running the simulation for T=20 the following graphs were obtained.



Graph 34: Output plots for default gains of set points X= 0; Y= 1; and Z= -1.1



Graph 35: Output plots for new gains of set points X= 0; Y= 1; and Z= -1.1.

The final test was done with set point of X= 1; Y= 0; and Z= -1.1. This can be seen in the following image of Path Planning block.

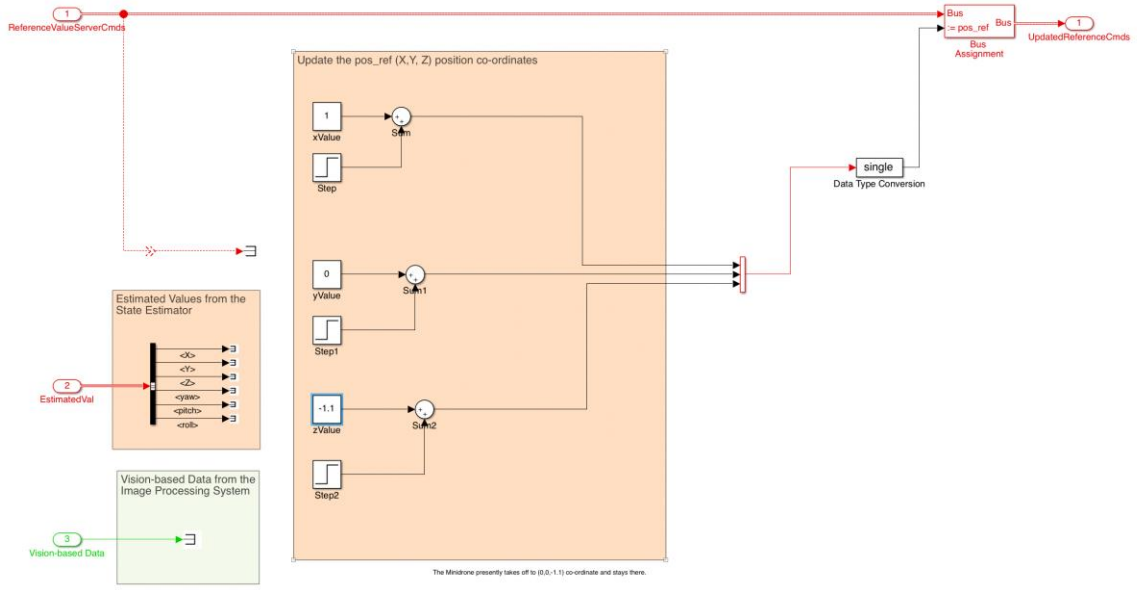
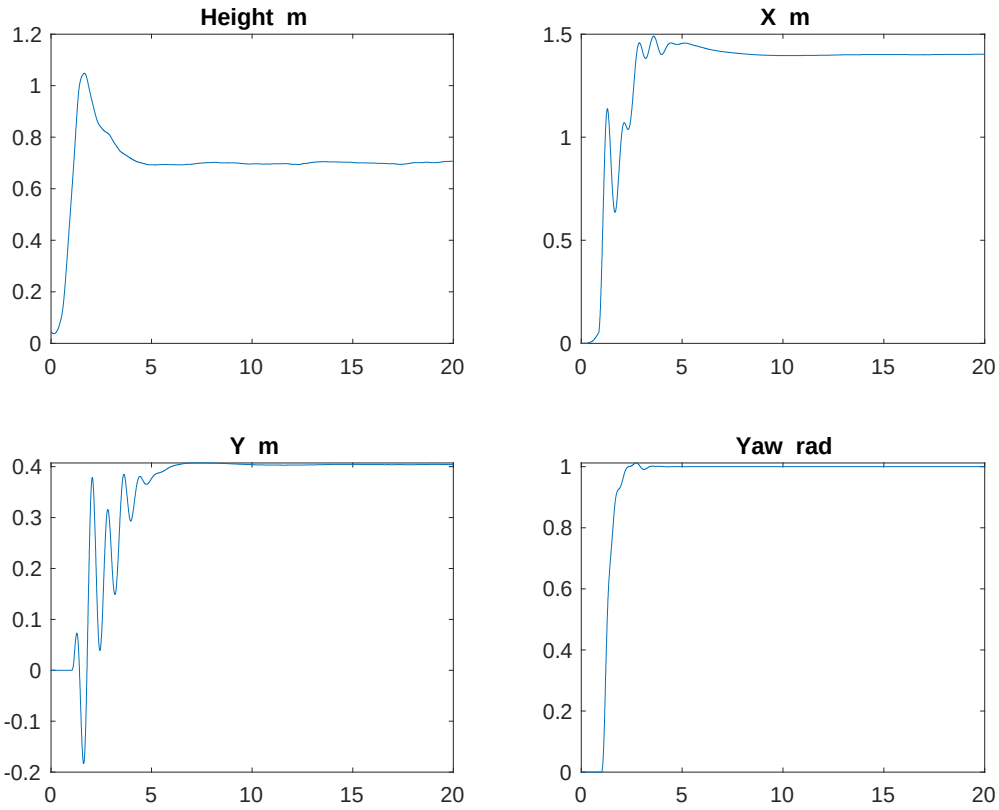


Figure 43: Path Planning block showing set points X= 1; Y=0; and Z= -1.1

After running the simulation for T=20 the following graphs were obtained.



Graph 36: Output plots for X= 0; Y= 1; and Z= -1.1

4.6 Discussions

As seen in the graphs above, we see four plots for each section. The graph on the top left is for Z axis output against time, the graph on the top right is for X axis output against time, the graph on the bottom left is for Y axis output against time and the graph on bottom right is for yaw output against time.

After inspecting all the graphs for different set points, we can conclude that the gains achieved in the tests were not suitable for all the coordinates. For some coordinates the Z axis output and stability were very good but lacked stability in other directions. We can conclude that if the selected gains are not very good but also, they are not that bad at the same time. It does complete the given task of hovering at the selected height. If more time was available, more tests could have been performed so that better stability and hover performance could be achieved.

After getting the quadcopter to hover the next task would be to test it on the real Parrot Mambo quadcopter. This could be done either by take a linear or non-linear approach. The simplest approach would be a linear approach, but its disadvantage would be that every data selected would be an approximation. On the other hand, more accurate results could be achieved by using non-linear model. But first every non-linearity should be identified. There exist a large number of non-linearities and it would be very difficult to establish them. This is one of the issues before testing the code on real minidrone.

Another issue we can face is with the Bluetooth interface. If the connection is weak, this can prevent the code from being downloaded from or to the quadcopter. Another issue relates to the MATLAB support packages used for obtaining the model. As the model we worked on is a linear model it works well with the simulating environment. Although the support package has the option of using non-linear model but to address every non-linearity would be very difficult.

Learning through simulation is a better way of learning things before implementing it on actual quadcopter itself. Learning the control part directly on actual quadcopter would involve lots of risk like damaging the quadcopter or hurting somebody if it runs out of control. It would also consume more time as the code need to be downloaded in the quadcopter every time a new gain is used for test. Hence to learn about MATLAB, Simulink and Flight Control part of the Parrot Mambo quadcopter through simulations was a good idea.

Chapter 5: Conclusion

In this report, a theoretical overview of history of drones and quadcopter along with feedback control systems, PID controllers and tuning of PID controllers has been presented. A theoretical and simulated study on control aspect of Parrot Mambo quadcopter is also presented. MALAB and Simulink were extensively used for achieving all the results. A good use of MATLAB support package for Parrot minidrones and Simulink support package Parrot minidrones is done. These were used to inspect the sensory controls of the quadcopter to determine the effect of changing the Proportional (P) and Derivative (D) gains on the stability and performance of the quadcopter. The main focus was to check the stability on the changing PD gains on Z axis, as the Z axis represented height of the quadcopter.

In short, MATLAB and Simulink were found very useful and highly compatible for this model. After every test was completed the best gains found for the controller takeoff gain was equal to '0.04' with P gain equal to '0.8' and D gain equal to '0.41' inside the gravity feedforward/equilibrium thrust block. The best gains for XY-to-reference-orientation block were found to be P= [-0.15, 0.15] and D= [0.18, -0.18]. There may exist better gains than this which might give better results. But for these gains the results obtained were quite satisfactory.

Chapter 6: Further Work and Project Review

If more time was available and more resources were there this project could have been a lot more. Due to the COVID-19 pandemic very limited amount of resources were available and using those limited resources this project was completed. If more time was given, I would have taken this project to the next level by executing some of my ideas into this project. There were plans of using the image processing sensors to do very interesting stuff like coordinate tracking, colour tracking and line following drone projects. This would also have resulted in better research and better literature review as the Internet was the only source of information available.

After submitting this report, I will make this my top priority to perfect this model and test the non-linear model so that I can use my research on the actual quadcopter. I would also work upon my MATLAB and Simulink skills as I think they are not up to the mark. I will also be doing a deep research on the Aerospace Blockset model so that my control aspect of the flight dynamics is perfected. These are the limitations I discovered in my project.

Over this academic year I have gained lots of knowledge over lots of subjects relating automation and control systems. I have gained a significant amount of knowledge in flight dynamics and control as well as MATLAB and Simulink and PID controllers as I always fascinated aircrafts and wanted to learn more about them. This project helped me develop my confidence in decision making based on limited data that was available to me. This also helped me to develop a detailed timetable and project planning along with time management skills. I conclude by saying that this project was very challenging, educational and fun to work on.

Chapter 7: References

[1] Unmanned Aerial Vehicle Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

[2] MATLAB Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/MATLAB

[3] Simulink Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/Simulink

[4] PID Controllers Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/PID_controller

[5] Andrzej Koszewnik. The parrot UAV controlled by PID controllers. Retrieved from: https://www.researchgate.net/publication/286704568_The_parrot_UAV_controlled_by_PID_controllers

[6] Ali Mahbub. Hexcopter Flight Control. University of Salford.

[7] Shoaib Mansoor and Mana Saedan. Software in the loop simulation of a quadcopter portion for hybrid aircraft control. Retrieved from: https://www.researchgate.net/publication/322995945_Software-in-the-loop_simulation_of_a_quadcopter_portion_for_hybrid_aircraft_control

[8] Tarek Hamel, Robert Mahony, Rogelio Lozano and James Ostrowski. Dynamic modelling and configuration stabilisation for and X4 flyer. Retrieved from: https://www.sciencedirect.com/science/article/pii/S1474667015392697

[9] Guillem Munoz, Cristina Barrado, Ender Cetin and Esther Salami. Deep reinforcement learning for drone delivery. Retrieved from: https://www.researchgate.net/publication/335729548_Deep_Reinforcement_Learning_for_Drone_Delivery

[10] Bara J Emran, Jorge Dias, Lakmal Senevirante and Guowei Cai. Robust adaptive control design for quadcopter payload add and drop applications. Retrieved from: https://www.researchgate.net/publication/305659353_Robust_Adaptive_Control_Design_for_Quadcopter_Payload_Add_and_Drop_Applications

[11] Kael E Wenzel, Andreas Masselli and Andreas Well. Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. Retrieved from: https://www.researchgate.net/publication/220062401_Automatic_Take_Off_Tracking_and_Landing_of_a_Miniature_UAV_on_a_Moving_Carrier_Vehicle

[12] Mucahid Ridvan Kaplan; Abdullah Eraslan; Aykut Beke; Tufan Kumbasar. Altitude and position control of parrot mambo mini drones with PID and fuzzy PID controllers. Retrieved from:

https://www.researchgate.net/publication/339265821_Altitude_and_Position_Control_of_Parrot_Mambo_Minidrone_with_PID_and_Fuzzy_PID_Controllers

[13] Timur Glazkov and Alexey Golubev. Using Simulink support package for parrot mini drones in non-linear control education. Retrieved from: https://www.researchgate.net/publication/338009554_Using_Simulink_Support_Package_for_Parrot_Minidrones_in_nonlinear_control_education

[14] Daniel Lewis. Analysis of the sensory controls of a Mambo Quadcopter with the use of MATLAB and Simulink. University of Salford.

[15] Drone history. Retrieved from: https://www.theflightbay.com/uav/

[16] evolving design of drones. Retrieved from: https://stories.uq.edu.au/eait/ingenuity/a-drone-by-any-other-name/index.html

[17] Working of quadcopters. Retrieved from: https://www.dronezon.com/learn-about-drones-quadcopters/how-a-quadcopter-works-with-propellers-and-motors-direction-design-explained/

[18] Different parts of a quadcopter. Retrieved from: https://dronebotworkshop.com/how-does-a-quadcopter-work/

[19] Roll, Pitch and Yaw. Retrieved from: https://emissarydrones.com/what-is-roll-pitch-and-yaw

[20] MATLAB Tech talks. Retrieved from: https://www.mathworks.com/videos/series/drone-simulation-and-control.html

[21] Frame of reference. Retrieved from: http://zonalandeducation.com/mstm/physics/mechanics/framesOfReference/framesOfReference.html

[22] Feedback control systems. Retrieved from: http://www.cim.mcgill.ca/~ialab/ev/Intro_control1.pdf

[23] Overview of different approaches of PID tuning. Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.407.4606&rep=rep1&type=pdf

[24] Parrot Mambo Drone. Retrieved from: https://www.parrot.com/uk/drones

[25] Parrot drone support from MATLAB. Retrieved from: https://in.mathworks.com/hardware-support/parrot-drone-matlab.html

[26] Parrot Mambo minidrone support from Simulink. Retrieved from: https://in.mathworks.com/hardware-support/parrot-minidrones.html

[27] Image processing algorithms for Parrot minidrones. Retrieved from: https://in.mathworks.com/help/supportpkg/parrot/ref/getting-started-with-parrot-minidrone-vision.html

[28] Spin the motors of a Parrot minidrone without Flying the drone. Retrieved from: https://in.mathworks.com/help/supportpkg/parrot/ref/getting-started-with-simulink-support-package-for-parrot-minidrones.html

[29] Getting started with keyboard control of Parrot minidrones. Retrieved from: https://in.mathworks.com/help/supportpkg/parrot/ref/getting-started-keyboard-control.html

[30] Dr Anthony Jones notes and handouts. University of Salford