ATTITUDE AND ALTITUDE CONTROL OF AN OUTDOOR QUADROTOR

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

ANIL GÜÇLÜ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

MECHATRONICS ENGINEERING

JULY 2012

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof. Dr. İbrahim Akman

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Abdulkadir Erden

Head of Department

This is to certify that we have read the thesis "Attitude and Altitude Control of an Outdoor Quadrotor" submitted by "Anıl Güçlü" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____	_____
Asst. Prof. Dr. Bülent İrfanoğlu	Asst. Prof. Dr. Kutluk Bilge Arıkan
Co-Supervisor	Supervisor

Examining Committee Members

Prof. Dr. Kemal Leblebicioğlu	(METU, E.E. Dept.)	_____
Assoc. Prof. Dr. Fuad Aliew	(ATU, M.E. Dept.)	_____
Asst. Prof. Dr. Ali Emre Turgut	(UTAA, M.E. Dept.)	_____
Asst. Prof. Dr. Bülent İrfanoğlu	(ATU, M.E. Dept.)	_____
Asst. Prof. Dr. Kutluk Bilge Arıkan	(ATU, M.E. Dept.)	_____

Date: 12.07.2012

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Anıl GÜÇLÜ

Signature:

ABSTRACT

ATTITUDE AND ALTITUDE CONTROL OF AN OUTDOOR QUADROTOR

Güçlü, Anıl

M.S., Mechatronics Engineering Department

Supervisor: Asst.Prof.Dr. Kutluk Bilge Arıkan

Co-Supervisor: Asst.Prof.Dr. Bülent İrfanoğlu

July 2012, 76 pages

In this thesis, controller design for the attitude and altitude dynamics of an outdoor quadrotor, which is constructed with low cost actuators and drivers, is aimed. Before designing the controller, the quadrotor is modeled mathematically in Matlab-Simulink environment. To control attitude dynamics, linear quadratic regulator (LQR) based controllers are designed, simulated and applied to the system. Two different proportional-integral-derivative action (PID) controllers are designed to control yaw and altitude dynamics. During the implementation of the designed controllers, different test setups are used. Designed controllers are implemented and tuned on the real system using xPC Target. Tests show that these basic control structures are successful to control the attitude and altitude dynamics.

Keywords: Attitude Controller, Altitude Controller, Quadrotor, PID Controller, LQR Controller

ÖZ

DÖRT PERVANELİ HAVA ARACININ YÜKSEKLİK VE YÖNELİM KONTROLÜ

Güçlü, Anıl
Yüksek Lisans, Mekatronik Mühendisliği Bölümü
Tez Danışmanı: Yrd.Doç.Dr. Kutluk Bilge Arıkan
Yardımcı Tez Danışmanı: Yrd.Doç.Dr. Bülent İrfanoğlu
Temmuz 2012, 76 sayfa

Bu tez çalışmasında, düşük maliyetli eyleyiciler ve sürücülerle oluşturulan dış mekanda uçacak dört rotorlu hava aracının (Quadrotor) yükseklik ve yönelim dinamiklerinin kontrolü amaçlanmıştır. Quadrotor, kontrolcü tasarımından önce Matlab/Simulink ortamında matematiksel olarak modellenmiştir. Yönelim dinamiklerini kontrol etmek amacıyla, LQR tip kontrolcü tasarlanmış, benzetimi yapılmış ve sisteme uygulanmıştır. Sapma ve yükseklik dinamiklerinin kontrolü için iki farklı PID tip kontrolcü tasarlanmıştır. Tasarlanan kontrolcüler değişik test düzenekleri kullanılarak sisteme uygulanmıştır. Denetimciler, xPC Target kullanılarak fiziksel sisteme uygulanmış ve kontrolcü parametre ayarlamaları yapılmıştır. Yapılan testler, bu temel kontrolcü yapıları ile yükseklik ve yönelim dinamikleri kontrolünün, başarılı bir şekilde sağlandığını göstermektedir.

Anahtar Kelimeler: Davranış Kontrolü, Yükseklik Kontrolü, Dört Pervaneli Hava Aracı, PID Kontrolcü, LQR Kontrolcü

To My Father, Mother, and Brother;

ACKNOWLEDGEMENTS

I want to sincerely thank my supervisor Asst.Prof.Dr. Kutluk Bilge Arıkan and my co-supervisor Asst.Prof.Dr. Bülent İrfanoğlu for their guidance during my thesis study. I also appreciate Yılmaz Güçlü for his mechanical information. My mother Ümit Güçlü who every time helps me with her prays. Sinem Doğan who allows me to use her house during the tests. From Roketsan Missile Industry, Birkan Kurşun and Barış Nurver who every time allow me to go the university to study. Meral Aday and Handan Kara who every time welcomes me during the studies at the University. Thanks them for their delicious teas and coffees. Emre Güner, Emre Büyükbayram, Temmuz Güçlü Aras, Kaan Altay Göker, and Bahadır Tanrıverdi for their help during the experiments with their ill-humors. And, Egemen Güçlü who every time in my life I want to be with. I sincerely want to thank him for his endless help.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

DC – Direct Current

ESC – Electronic Speed Controller

FBD – Free Body Dynamic

IMU – Inertial Measurement Unit

IR – Infrared

LED – Light Emitting Diode

LQR – Linear Quadratic Regulator

PID – Proportional Integral Derivative

UAV – Unmanned Air Vehicle

NOMENCLATURE

The followings are the list of variables used in this thesis;

Ω - Angular velocity of the propellers (rad/sec)

ω- Angular velocity vector

θ- Pitch dynamics (rad)

ϕ- Roll dynamics (rad)

ψ- Yaw dynamics (rad)

$b_{1,2,3,4}$ – Thrust coefficient of the propellers

$d_{1,2,3,4}$ – Drag coefficient of the propellers

$F_{1,2,3,4}$ – Thrust forces generated by propellers (N)

I – Inertia Matrix

L – Diagonal length of propeller and center of the system

$M_{x,y,z}$ – Moment about x, y, and z axes

p – Angular velocity of roll dynamics (rad/sec)

q – Angular velocity of pitch dynamics (rad/sec)

r – Angular velocity of yaw dynamics (rad/sec)

$\dot{p}$ – Angular acceleration component of roll axis in body reference frame (rad/sec^2)

$\dot{q}$ – Angular acceleration component of pitch axis in body reference frame (rad/sec^2)

$\dot{r}$ – Angular acceleration component of yaw axis in body reference frame (rad/sec^2)

ω_x – Angular velocity vector component in roll axis (rad/s)

ω_y – Angular velocity vector component in pitch axis (rad/s)

ω_z – Angular velocity vector component in yaw axis (rad/s)

H_c – Angular momentum

$dt_{1,2,3,4}$ – Duty Ratio (%)

z – Altitude (cm)

V – Proximity Sensor Output (V)

CHAPTER 1

INTRODUCTION

In recent years, along with the development of various technologies usage of unmanned air vehicles (UAV) increased noticeably. UAVs can be classified into two main groups, namely, fixed wings and rotary wings. Rotary wing air vehicles have an advantage, which is vertical takeoff and landing capability, compared to fixed-wings UAVs. There are different types of rotary wing air vehicles such as single rotor, twin rotors, tri-rotors, and quad rotors and so on. Usage area of a quadrotor can be separated into three major parts such as: military operations, public applications, and civil applications.

- Military Operations
  - Border Security
  - Security Intelligence
  - Acquisition of Targets
  - Correction of Coordinates
  - Coast Guard
  - Cartography
  - Photography
- Public Applications
  - Search and Rescue
  - Pollution of Seas
  - Security of Petrol Pipe Line
  - Correction of Uncontrolled Trash Areas
  - Forest Fire

- Civil Applications
  - Control of Oil, Fuel and Connection Lines
  - Shooting Movie
  - Analysis of Fire Gas

A quadrotor has four rotors and each rotor has a propeller. With the rotation of the propellers, lift force is obtained. In Figure 1, top view of a propeller is seen. If the propeller rotates counter-clockwise, it creates lift force into the page and a drag moment in clockwise direction [1]. If all motors of the quadrotor rotate in same direction, for example counter-clockwise, a nonzero drag moment is obtained. Then, heading angle of the quadrotor is uncontrollable.

Figure 1: Propeller Rotations and Drag Force

That is why; four motors should be grouped two by two and should be rotated in opposite way in pairs. If motors are grouped as seen in Figure 2, motors numbered as 1 and 3 rotate counter-clockwise and creates drag moment in clock-wise direction, motors numbered 2 and 4 rotate in clockwise direction and creates drag force in counter-clockwise direction. Therefore, all motors create lift force in to the page and drag force created by motors is eliminated.

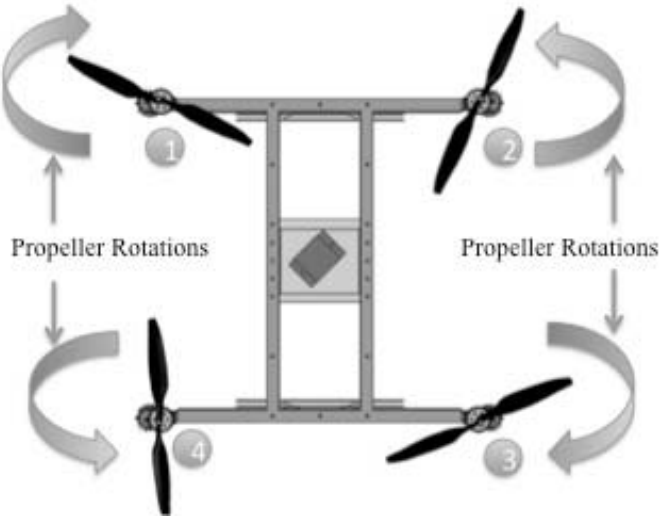


Figure 2: Propellers Rotation Directions

Stability of a quadrotor can be achieved by adjusting the angular velocity of four rotors. These four rotors should be controlled in closed loop. Usually, as a sensor, an inertial measurement unit (IMU) is used. An IMU is able to give different kinds of data types with the help of gyroscopes, accelerometers and magnetometers located inside the IMU. Kalman filter and complementary filter are the two common filtering techniques that are applied in IMUs.

1.1 Aim and Scope

The aim of the thesis is to design control systems to stabilize and control attitude and altitude dynamics of an outdoor quadrotor, which is constructed with low cost actuators and drivers. Controllers are designed using the mathematical model of the quadrotor. Inertial parameters of the physical system are identified experimentally. Propulsion systems are also identified. This study is the initial one in our laboratory that utilizes brushless motors and electronic speed controllers (ESC) as drivers. Therefore it is also aimed to build basic model based design strategies for systems equipped with brushless units. A systematic approach is built and applied on two different platforms successfully.

To control attitude dynamics LQR based controllers are designed, simulated and applied to the system. Two different PID controllers are designed to control yaw and altitude dynamics. During the implementation of the designed controllers, different test setups are used. Designed controllers are implemented and tuned on

the real system using xPC Target real time operating system. Robust and adaptive control algorithms may be implemented in future studies. LQR and PID based integrated and multi-loop control architectures reveal certain level of robustness during the physical implementations.

Quadrotor`s power is supplied from a ground unit by using cables. In addition, control unit is also on the ground. Using embedded power and control units is out of scope of the study. Cables present disturbing inputs and the control system is able to reject the effect of the disturbing effects. Disturbance estimation and rejection algorithms may also be studied in future studies.

CHAPTER 2

LITERATURE SURVEY

Air vehicles can be classified into two categories such as rotary wing and fixed wing aircrafts. Rotary wing air vehicles fly by the thrust force, which is created by propellers. Fixed wing aircrafts creates a thrust force in forward direction. The air, which passes through the wings, obtains a lift force. In recent years, popularity of unmanned air vehicles has been increased. First quadrotor is designed and proposed by De Bothezat and can be seen in Figure 3 [2]. The movement of the air vehicle was slow and at low altitudes. Its horizontal motion was affected by wind more than pilot control.

Figure 3: Older Design of Manned Quadrotor

Different control strategies of rotary wing unmanned air vehicles such as quadrotors have been studied in commercial, academic, and military platforms. Four rotors increase the maneuverability of the vehicle. Having four rotors increases load carrying capacity, on the other hand, constrains it to consume more energy [3]. There are different controller algorithms such as PD-PID controller, inverse control, back stepping control, and sliding mode control that can be applied to the quadrotors [4]. Starmac II, which is shown in Figure 4, can be given as an example of classical type of quadrotor.

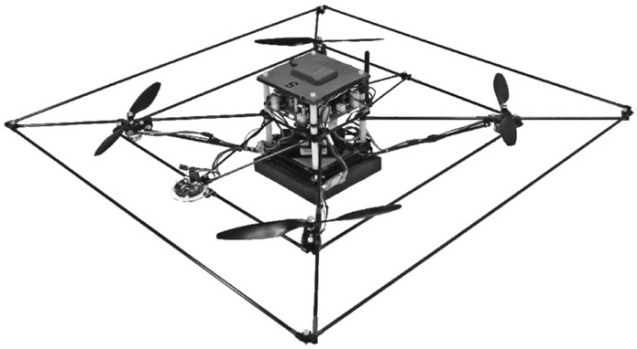


Figure 4: STARMAC II: Quadrotor Helicopter

The control electronics of Starmac II is composed of an electronics interface board and several processors. The low level processor at a fast rate controls attitude and altitude dynamics. Longer computations and position control is done by the low level processors. Atmega 128 microprocessor is used for low level computing. The Atmega 128 processes the IMU data and ultrasonic rangefinder output data. Update rate of the system is 76 Hz. Control algorithm of the Starmac II is written for different types of single board computers. As an IMU, a Microstrain GX1 is used. To measure the distance from the ground, Senscomp Mini-AE ultrasonic rangefinder is used. As actuators, Axi 2208/26 Sensorless Brushless DC Motors are used. To control the motors, Phoenix-25 Electronic Speed Controllers are used [5]. If the aim is to control the orientation and vertical position of the quadrotor, PID controller algorithm is more suitable [6].

Another quadrotor has a structure that consists of a lightweight carbon fiber with four aluminum arms, Figure 5. The arms of the system are 23.2 cm long measured from the center. Total mass of the system is 753 gr. including battery and electronics. Experiments are done while total generated thrust of the system is 960 gr when running at 3500 rpm. As actuators, Razor Micro-Heli v2.0, 3 pole brushless DC motors are used. To produce thrust, EPP1045 type propellers are assembled to the DC motors with a plastic gear train reduction whose gear ratio is 6.6:1. A Castle Creations Phoenix-10 electronic speed controller is used to give power to each motor. To measure the speed of the DC motor, OP550A phototransistor and an IR LED are used. A Parkzone PKZ1030 LiPo battery to the system gives power. To the system, neural-adaptive control, linear-quadratic-regulator and linear parameters adaptive control techniques type controllers are

applied and compared. It is stated that the LQR controller is not good enough to achieve a good performance due to both nonlinearities and payload without an integral term. The LP adaptive control can adjust to payloads, but gives the same performance for more payloads. The neural-adaptive control does not overshoot and gives better results compared to the LQR and the LP type controllers [8].

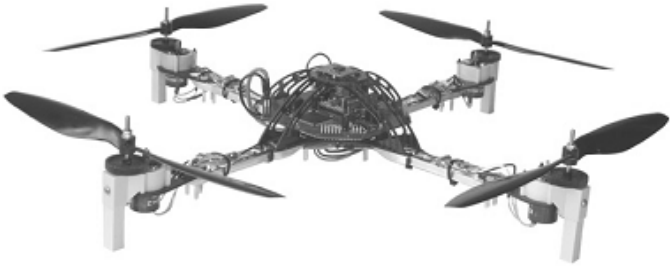


Figure 5: A Prototype Quadrotor

The X-4 Flyer, which is shown in Figure 6, has 4kg of weight and 1kg of payload. Off-the-shelf components are preferred for batteries and motors. The reason for using off-the-shelf components is best performance rotors and motors must be tuned and produced according to the need of the aircraft. By this way, instantaneous thrust changes can be obtained. Since the mass and the inertia increases, when the motor size increases. To increase the bandwidth of the motor, lightweight rotors has to be chosen. A PID type controller is used to control roll and pitch dynamics [7].

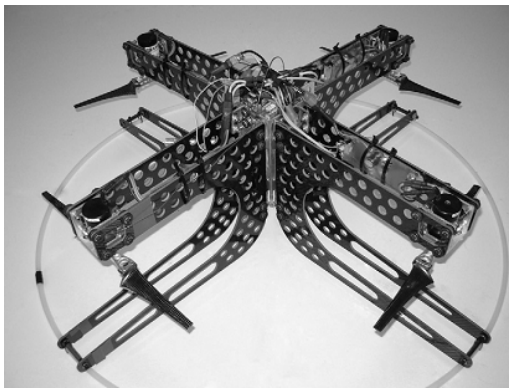


Figure 6: The X-4 Flyer

LQR type controller can be used for controlling the attitude dynamics of the quadrotor [8]. The controller composed of two weighting matrices, which are Q, and R matrices, which contain state variables and input variables, respectively [9].

Using a nonlinear control algorithm, which is based on the super-twisting algorithm, can also do attitude control of the quadrotor. The weight of the system, which is shown in Figure 7, is 0.42 gr. Rotor arm lengths are 20.5 cm. To measure the attitude angles and angular velocities, XSENS MTI-28A53G35 Inertial Measurement Unit is used. IMCS 25 is used for speed control and is driven by PWM signals [10].

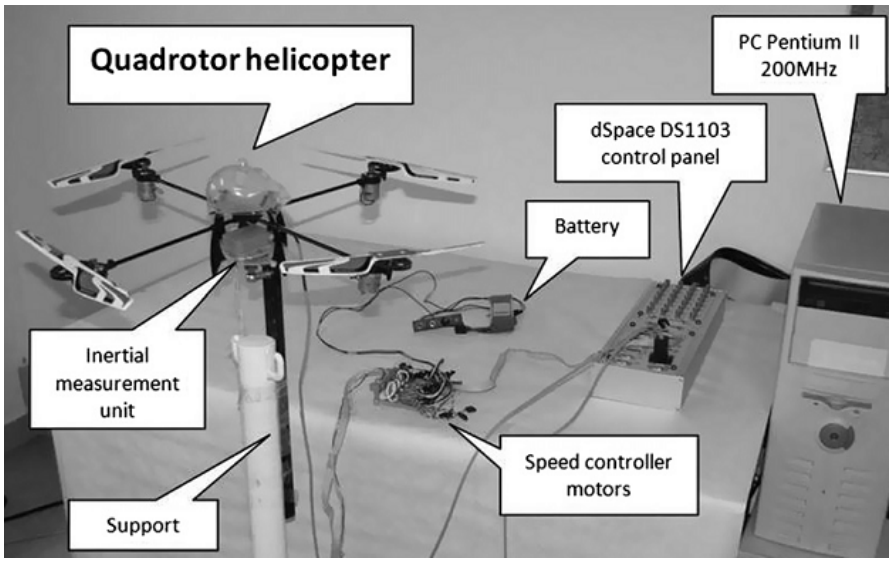


Figure 7: A Quadrotor Test Setup

Floating movement on x-y plane is the most encountered problem since there is no actuator in x-y plane. Adding a GPS or positioning by using external cameras can resolve this problem. However, these components are inadequate in indoor environment. To get rid of the problem, an optical flow sensor may be used. Translational velocity of the quadrotor can be estimated by means of the sensor with using an Extended Kalman Filter [11].

It is very critical and important to get roll, pitch and yaw dynamics of flying systems. Generally, attitude data of quadrotors can be measured by IMUs [11], [12], [13], [14], [15]. IMU is a combination of accelerometers, gyroscopes and magnetometers [16]. Generally, angle measurement of the sensors is not accurate. To have accurate angle measurement, these sensor outputs have to be combined and filtered. By using some complementary and Kalman Filters, angles can be measured accurately [17]. There are different types of IMUs available to measure

roll, pitch and yaw angles of flying system. There are many commercially available IMUs such as CSIRO Eimu [7], Xsens MT9-B [11], Microstrain 3DM-GX1 [15], and Crossbow MNAV100CA [18] that can be used for flying systems.

Different types of controllers such as feedback linearizing control law, feedback proportional-derivative control, predictive control, back stepping, adaptive control techniques are applied to flying objects such as space crafts, micro-satellites, tactical missiles, flexible launch vehicles etc. [19]. Roll, pitch, and yaw dynamics are controlled by increasing or decreasing the speed of four motors [12].

CHAPTER 3

MATHEMATICAL MODELING

In this thesis, attitude and altitude dynamics of a quadrotor are considered. Therefore roll, pitch, yaw, and altitude dynamics are modeled.

3.1 Mathematical Model of Attitude Dynamics

One of the aims of the thesis is to stabilize the quadrotor in the hovering condition. Following assumptions are made [20],

- System structure is supposed to be rigid and symmetrical.
- Earth fixed reference frame is assumed to be inertial.
- The rotors are rigid, i.e. no blade flapping occurs.

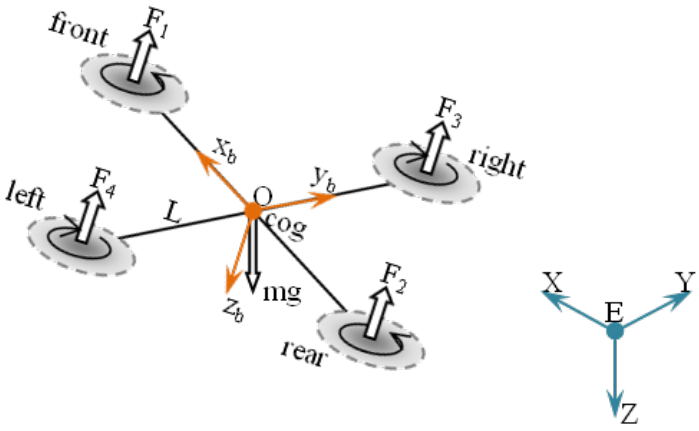


Figure 8: Schematic Representation of the System

Attitude of the quadrotor is defined by roll, pitch and yaw angles; namely φ, θ, and ψ, respectively. Angular velocity components in body reference frame are p, q, and r, respectively [14]. F_1, F_2, F_3, and F_4 are the thrust forces, which are

generated by propellers. Axes and states are represented in Figure 8. It is possible to obtain a rotation matrix with the following equation [21].

$$R = R_1 R_2 R_3 \tag{3.1}$$

$$\psi \text{ Rotation, } R_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \tag{3.2}$$

$$\theta \text{ Rotation, } R_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{3.3}$$

$$\varphi \text{ Rotation, } R_3 = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \tag{3.4}$$

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}, \; R_2 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\theta \end{bmatrix}, \quad R_3 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Substituting R_1, R_2 and R_3 into (3.1), yields the following expression for R,

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \cos\varphi\sin\psi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \sin\varphi\cos\psi \\ -\sin\theta & \sin\varphi\cos\theta & \cos\theta\cos\varphi \end{bmatrix} \tag{3.6}$$

Angular velocity vector, in view of angular velocity components of the quadrotor as illustrated in Figure 8, can be written as.

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.7}$$

It is possible to write state equations of the Euler angles in terms of angular velocity components, p, q, and the r, by using rotation matrices.

$$\dot{\varphi} = p + q\tan\theta\sin\varphi + r\tan\theta\cos\varphi \tag{3.8}$$

$$\dot{\theta} = q\cos\varphi - r\sin\varphi \tag{3.9}$$

$$\dot{\psi} = p + q\sec\theta\sin\varphi + r\sec\theta\cos\varphi \tag{3.10}$$

Angular momentum needs to be found to reach the time derivatives of the angular velocity terms, which are p , q , and r .

$$H_c = I\omega \tag{3.11}$$

Inertia matrix can be written as below.

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3.12}$$

Inertia matrix, I, is assumed to be a diagonal matrix. In other words, the mass products of inertia terms are assumed to be zero. Dynamic equations related with the angular velocities are derived with the help of Newton's Law.

$$\left(\frac{dH_c}{dt}\right)\Big|_{OXYZ} = \left(\frac{dH_c}{dt}\right)\Big|_{OX_BY_BZ_B} + \omega \times H_c = M \tag{3.13}$$

$$I\dot{\omega} + \omega \times (I\omega) = M \tag{3.14}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = Ir^{-1}.\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - I^{-1}.\left\{\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I.\begin{bmatrix} p \\ q \\ r \end{bmatrix}\right\} \tag{3.15}$$

Resultant moments about roll, pitch, and the yaw axes are given below; thrust forces generated by each motor-propeller pair are denoted by F_1, F_2, F_3, and F_4.

$$\sum M_x = (F_3 - F_1)\frac{L}{2} \tag{3.16}$$

$$\sum M_y = (F_2 - F_4)\frac{L}{2} \tag{3.17}$$

$$\sum M_z = (M_1 + M_3) - (M_2 + M_4) \tag{3.18}$$

Resultant moments about roll, pitch, and the yaw axes are given below.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{qr}{I_x}(I_y - I_z) - \frac{L}{2I_x}(F_1 - F_3) \\ \frac{pr}{I_y}(I_y - I_z) - \frac{L}{2I_2}(F_2 - F_4) \\ \frac{pq}{I_z}(I_x - I_y) + \frac{((F_2 + F_4) - (F_1 + F_3))d}{I_z b} \end{bmatrix} \tag{3.19}$$

State vector and state equations considering the yaw, roll, and the pitch dynamics of the platform are expressed below.

$$x = \begin{bmatrix} \varphi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} \tag{3.20}$$

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} p + q\tan\theta\sin\varphi + r\tan\theta\cos\varphi \\ q\cos\varphi - r\sin\varphi \\ p + q\sec\theta\sin\varphi + r\sec\theta\cos\varphi \\ \frac{qr}{I_x}\left(I_y - I_z\right) - \frac{L}{2I_x}\left(F_1 - F_3\right) \\ \frac{pr}{I_y}\left(I_z - I_x\right) - \frac{L}{2I_y}\left(F_2 - F_4\right) \\ \frac{pq}{I_z}\left(I_x - I_y\right) + \frac{\left(\left(F_2 + F_4\right) - \left(F_1 + F_3\right)\right)d}{I_z b} \end{bmatrix} \tag{3.21}$$

Brushless motors are used as the actuators of propellers of the quadrotor. Therefore, force and moment terms in the previous equations should be expressed in terms of duty cycle to the motors.

3.1.1 Motor and Propeller Model

Four brushless motors in the system are driven via Electronic Speed Controllers (ESC) [22]. ESC's converts PWM signals into a three-phased signal, which rotates the motor continuously. Motor and propeller unit models are identified experimentally. These models are algebraic, steady state ones. Dynamic models may be identified in future studies. Use of algebraic models should be considered during the design of closed-loop controllers. It is assumed that actuators have relatively fast dynamics considering the bandwidth of the closed-loop system

Two types of experiments are done to obtain a reliable motor and propeller model. First, propeller angular velocities are measured then thrust force is obtained with mathematical calculations. Second, motor thrust force is measured directly by using a weighing device.

To measure the angular velocity of the propellers, a hand type tachometer is used. The tachometer is seen in Figure 9. Numbers on the display of the tachometer represent the angular velocity of the propeller.

Figure 9: Ty1 Hand Type Tachometer

PWM input signal to the motor driver is carried by 300 Hz. Its duty ratio changes from 5% to 10%. By changing the duty cycle value, angular velocity of the motor is changed. Firstly, duty cycle value is increased from 5% to 10% with 0.1% increments. During these changes, propeller angular velocities are measured with the tachometer. Angular velocity – duty cycle graphics are shown in Figure 10 - Figure 13.

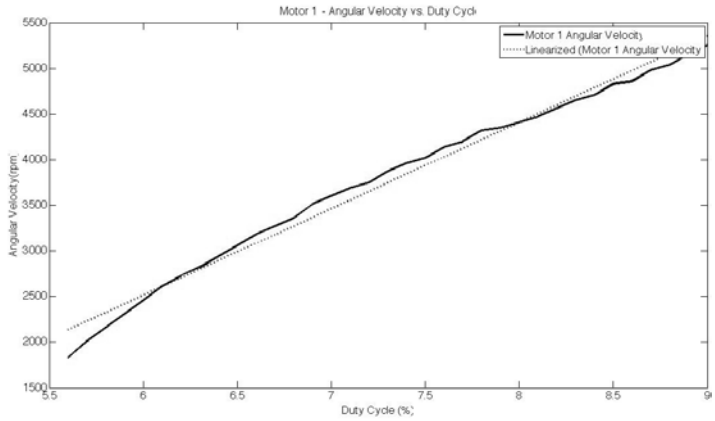


Figure 10: Motor 1 - Angular Velocity vs. Duty Cycle

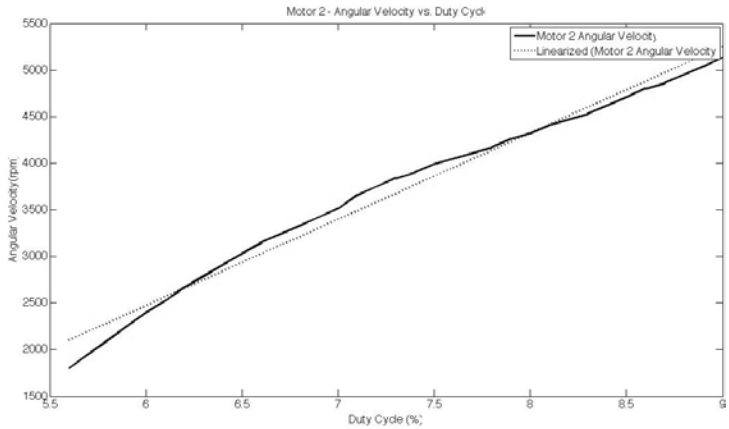


Figure 11: Motor 2 – Angular Velocity vs. Duty Cycle

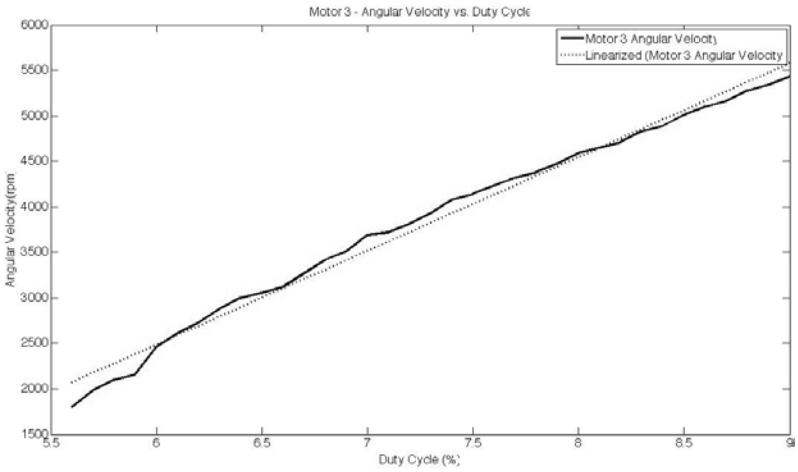


Figure 12: Motor 3 – Angular Velocity vs. Duty Cycle

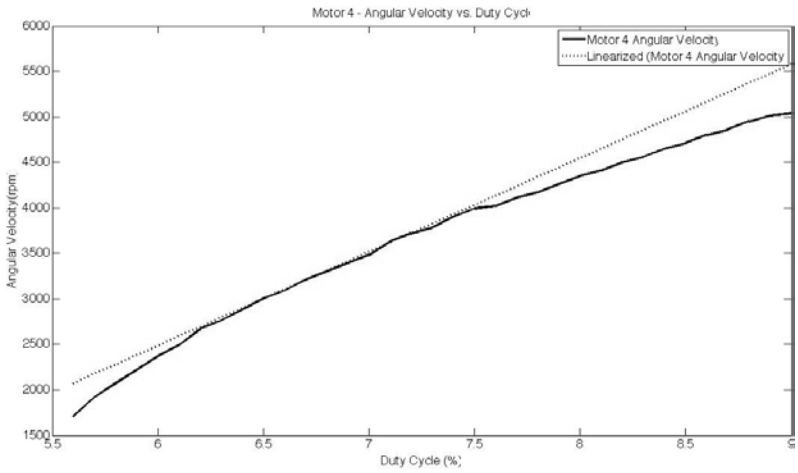


Figure 13: Motor 4 – Angular Velocity vs. Duty Cycle

Using Matlab Curve Fitting Tool, angular velocity data of the four propellers are linearized and shown on the figures as dashed lines.

The linear fit of each motor is given below;

$$\Omega_1 = 11.668(duty_1) + 195.42 \text{ (rad/sec)} \quad (3.22)$$

$$\Omega_2 = 10.860(duty_2) + 186.99 \text{ (rad/sec)} \quad (3.23)$$

$$\Omega_3 = 11.834(duty_3) + 195.59 \text{ (rad/sec)} \quad (3.24)$$

$$\Omega 4 = 11.587(duty_4) + 177.73 \text{ (rad/sec)} \quad (3.25)$$

As a result, thrust force generated due to the rotation of propellers can be calculated with the equation given below;

$$F_z = b\Omega^2 \tag{3.26}$$

Where *b* is thrust factor. In order to calculate this factor, a test setup is employed in the Flying Robotics Laboratory of the Mechatronics Engineering Department at Atılım University. This setup has a bar, which is mounted to the setup with bearings. When the motor rotates, a force generated on the scale device surface is measured. The setup is seen in Figure 14. Motor – propeller unit is fixed on one side of the bar and there is a payload on the other side of the bar.

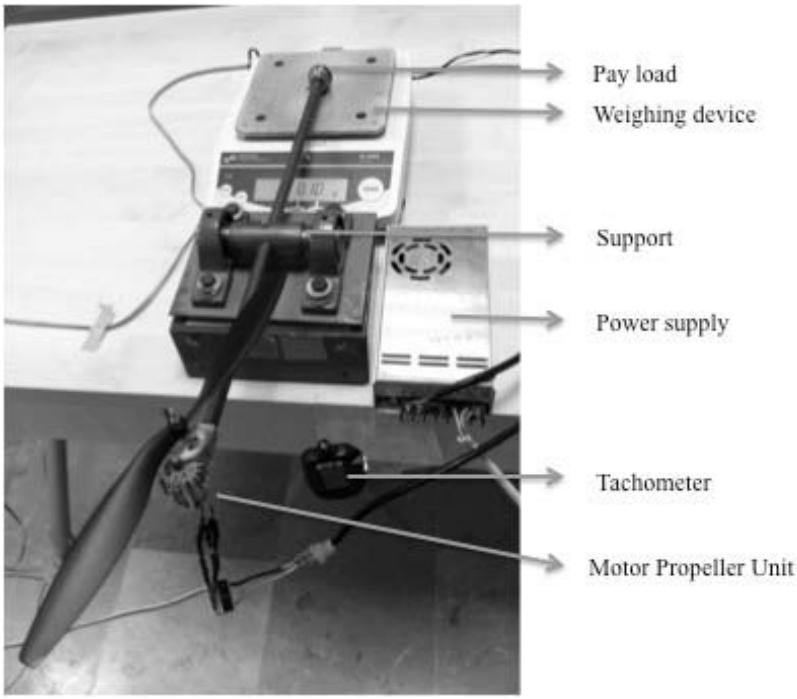


Figure 14: Thrust Measurement Setup

Each motor is placed in the setup and force is measured with the help of the scale device. Although the working range of ESCs is 5% and 10%, motor duty cycle values are incremented from 5.5% to 9% during the tests. The reason is that the motors do not rotate when the duty cycle value is lower then 5%. When the duty cycle values are bigger than 9%, there is no speed change of the motors. At each increment, propeller angular velocity and produced thrust is recorded. Then, thrust

coefficient is calculated with the help of Equation 3.26 at each step. Thrust coefficient values for four motor-propeller units are calculated as seen in Table 1.

Table 1: Thrust Coefficient of Motor – Propeller Units

	Thrust Coefficient (b) $\left[N/(rad/s)^2\right]$
Motor 1	1.89×10^{-5}
Motor 2	1.99×10^{-5}
Motor 3	1.81×10^{-5}
Motor 4	2.03×10^{-5}

Substituting the thrust coefficients into equation 3.30, thrust force equations become;

$$F_1 = 0.186(dt_1) + 0.0947 \text{ [N]} \tag{3.27}$$

$$F_2 = 0.183(dt_2) + 0.0165 \text{ [N]} \tag{3.28}$$

$$F_3 = 0.194(dt_3) - 0.0742 \text{ [N]} \tag{3.29}$$

$$F_4 = 0.192(dt_4) - 0.0552 \text{ [N]} \tag{3.30}$$

The generated drag moments by motor propeller pairs (M_1, M_2, M_3, M_4) are stated as;

$$M_{1,2,3,4} = d(\Omega_{1,2,3,4})^2 \tag{3.31}$$

Equation 3.31 can also be written as follows in terms of thrust forces with using the relationship between thrust and drag coefficient.

$$M_{1,2,3,4} = \frac{d}{b}(F_{1,2,3,4})^2 \tag{3.32}$$

In literature, it is seen that there is a relation between thrust and drag coefficient [23]. This relation is seen in following equation.

$$\frac{b}{d} = \frac{1}{58} \tag{3.33}$$

Substituting all the variables and formulas into equation 3.21 yields equation 3.34.

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} p + q\tan\theta\sin\varphi + r\tan\theta\cos\varphi \\ q\cos\varphi - r\sin\varphi \\ p + q\sec\theta\sin\varphi + r\sec\theta\cos\varphi \\ \frac{qr}{I_x}\left(I_y - I_z\right) - \frac{L\left(0.186\left(dt_1\right) + 0.0947 - 0.194\left(dt_3\right) + 0.0742\right)}{2I_x} \\ \frac{pr}{I_y}\left(I_z - I_z\right)x - \frac{L\left(0.183\left(dt2\right) + 0.0165 - 0.192\left(dt_4\right) + 0.0552\right)}{2I_y} \\ \frac{pq}{I_z}\left(I_x - I_y\right) + \frac{\left(-0.186\left(dt_1\right) + 0.183\left(dt2\right) - 0.194\left(dt_3\right) + 0.192\left(dt_4\right) + 0.1302\right)d}{I_z b} \end{bmatrix} \quad (3.34)$$

In equation 3.34 dt_1, dt_2, dt_3, and dt_4 are the applied duty cycles to the propeller actuators.

3.1.2 Linearization of Nonlinear State Equations for Attitude Dynamics

Equation 3.34 is linearized around the hovering conditions where all Euler angles and angular velocity are accepted as zero. The state space representation is in the form of;

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (3.35)$$

Where,

A is state matrix,

B is input matrix,

C is output matrix,

D is feed forward matrix,

x is state vector,

y is output vector,

u is input vector.

The state and input vector are constructed in the form of;

$$x = \begin{bmatrix} \varphi & \theta & \psi & p & q & r \end{bmatrix}^T \quad (3.36)$$

$$u = \begin{bmatrix} dt_1 & dt_2 & dt_3 & dt_4 \end{bmatrix}^T \tag{3.37}$$

Using following formulas and taking partial derivatives of Equation (3.34) with respect to the state vector and input vector, it is possible to linearize the state space equations [24];

$$\dot{x} = Ax + Bu \tag{3.38}$$

$$A = \frac{\partial f}{\partial x}\Big|_n \tag{3.39}$$

$$B = \frac{\partial f}{\partial u}\Big|_n \tag{3.40}$$

By writing the Equation 3.39 and 3.40 in detail, components of A and B matrix can be derived.

These operations are executed by using Matlab software. An m-file is written which linearize the state space equations. The code is given in Appendix A.

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 & 1 & A_{15} & A_{16} \\ A_{21} & 0 & 0 & 0 & A_{25} & A_{26} \\ A_{31} & 0 & 0 & 0 & A_{35} & A_{36} \\ 0 & 0 & 0 & 0 & A_{45} & A_{46} \\ 0 & 0 & 0 & A_{54} & 0 & A_{56} \\ 0 & 0 & 0 & A_{64} & A_{65} & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ B_{41} & 0 & B_{43} & 0 \\ 0 & B_{52} & 0 & B_{54} \\ B_{61} & B_{62} & B_{63} & B_{64} \end{bmatrix} \begin{bmatrix} dt_1 \\ dt_2 \\ dt_3 \\ dt_4 \end{bmatrix} \tag{3.41}$$

Substituting partial derivatives into A and B matrices, following matrices are obtained;

$$A = \begin{bmatrix} q\cos\varphi\tan\theta - r\sin\varphi\tan\varphi & r\cos\varphi\left(\tan^2\theta + 1\right) + q\sin\varphi\left(\tan^2\theta + 1\right) & \frac{r\cos\varphi\sin\theta}{\cos^2\theta} + \frac{q\sin\varphi\sin\theta}{\cos^2\theta} & 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ -r\cos\varphi - q\sin\varphi & 0 & 0 & 0 & \cos\varphi & -\sin\varphi \\ \frac{q\cos\varphi}{\cos\theta} - \frac{r\sin\varphi}{\cos\theta} & 0 & 0 & 0 & \frac{\sin\varphi}{\cos\theta} & \frac{\sin\varphi}{\cos\theta} \\ 0 & 0 & 0 & 0 & \frac{-I_y r - I_z r}{I_x} & \frac{-I_y q - I_z q}{I_x} \\ 0 & 0 & 0 & \frac{-I_x r - I_z r}{I_y} & o & \frac{-I_y p - I_z p}{I_x} \\ 0 & 0 & 0 & \frac{-I_x q - I_y q}{I_z} & \frac{-I_x p - I_y p}{I_z} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{-0.093L}{I_x} & 0 & \frac{-0.097L}{I_x} & 0 \\ 0 & \frac{-0.0915L}{I_y} & 0 & \frac{-0.096L}{I_y} \\ \frac{-0.186d_1}{I_z b_1} & \frac{-0.183d_2}{I_z b_2} & \frac{-0.194d_3}{I_z b_3} & \frac{-0.192d_4}{I_z b_4} \end{bmatrix}$$

3.2 Mathematical Model of Altitude Dynamics

Another aim of the thesis is to control the altitude dynamic of the quadrotor. Total thrust force and weights are the acting forces in z direction. Translational acceleration of the quadrotor in z-axis can be obtained with using the following formula [25];

$$\ddot{z} = \frac{\sum_{i=1}^{4} F_i}{m} \cos\theta \cos\varphi - g \tag{3.42}$$

This equation reveals position in z-axis and velocity in z-axis as 2 states.

CHAPTER 4

CONTROLLER DESIGN and SIMULATIONS

4.1 Controller Design and Simulation

At first, the quadrotor is modeled mathematically. A PID controller is designed in Matlab/Simulink and applied to the mathematical model. The controller first designed for roll axis. Then, these controller parameters are applied to real system. Outputs of the simulation and real system are compared. The procedure is summarized in Figure 15.

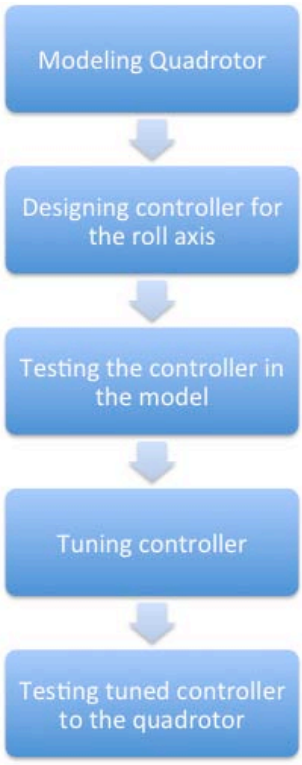


Figure 15: Summary of the Design Procedure

The process flow seen in Figure 15 is also performed for the pitch axis. Then, tuned controllers of the roll and pitch axes are combined and applied to the quadrotor on the roll-pitch experiment setup, which is seen in Figure 44. Same processes are done for LQR controller. Detailed explanation of the controller design, simulations and experiments are given in this chapter.

4.2 PID Controller Design and Simulation

As stated before, the quadrotor is modeled mathematically in Matlab environment. At first, PID type controller is designed in Matlab/Simulink environment.

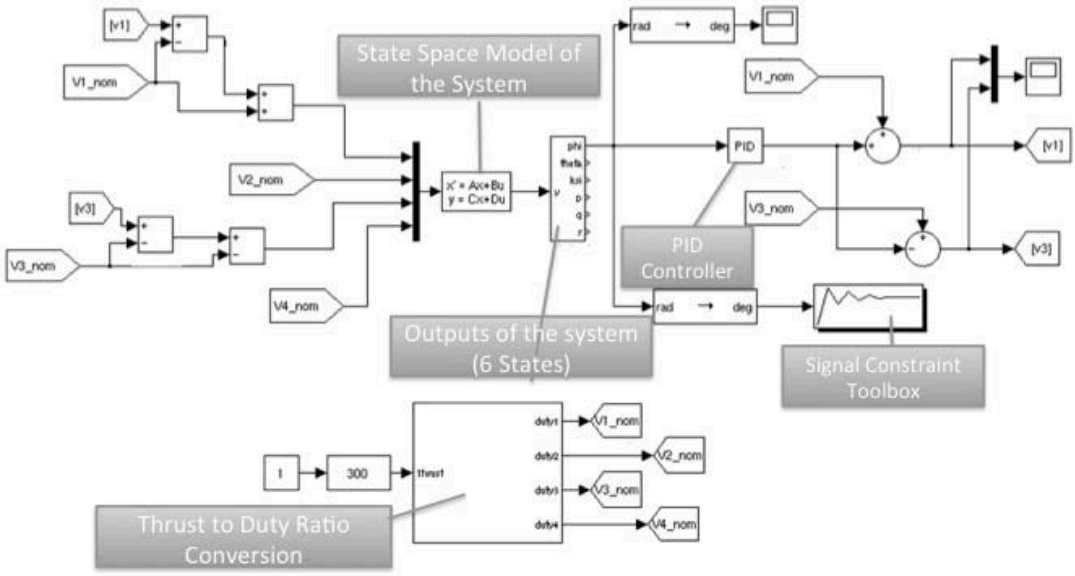


Figure 16: Simulink Model for Roll Axis with PID Controller

In this controller type, initial condition for the roll angle is given as, φ_i, is given as 0°. Whole weight of the quadrotor is approximately 2000 gr., which means each motor has to produce a force to lift approximately 500 gr. in hovering condition. Each motor that is used in the quadrotor is capable of producing a 850-gr. of thrust force at most. In other words, the quadrotor can take off with using 60% of total thrust of the motors. 40% of thrust range of the motors is used to control the quadrotor.

The response of the system with the controller parameters K_p=1, K_i=1, K_d=1 is shown in Figure 17.

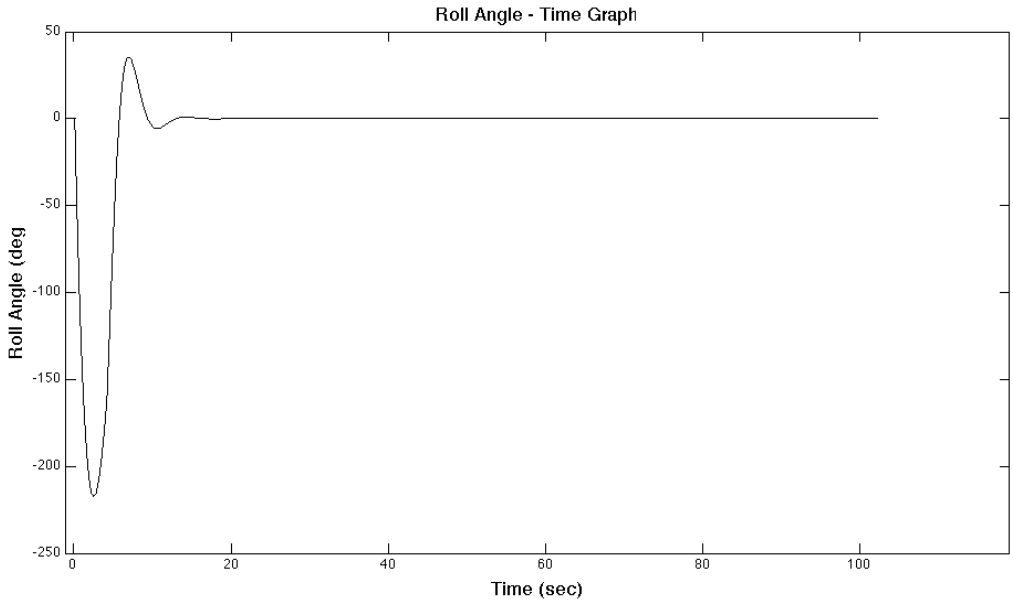


Figure 17: Roll Angle Graph, K_p=1, K_i=1, K_d=1

Response of the system is optimized to meet the given transient response specifications by numerical optimization techniques. This is achieved by using Matlab/Simulink. “Signal Constraint Block” in Figure 16 is responsible from the optimization [26]. There are different types of optimization methods such as gradient descent, pattern search, and simplex search, which can be used during optimization of the parameters. In the model, output of the controller is connected to the Signal Constraint block. First, the model is run with initial controller parameters. After the simulation is completed, Signal Constraint block is opened. In its menu, the parameters, which will be optimized, are chosen. That is also possible to give initial condition, upper and lower limits for the parameters. Then, optimization process is started in the toolbox. As optimization algorithm, gradient descent is selected. The interface of the signal constraint toolbox can be seen in Figure 18.

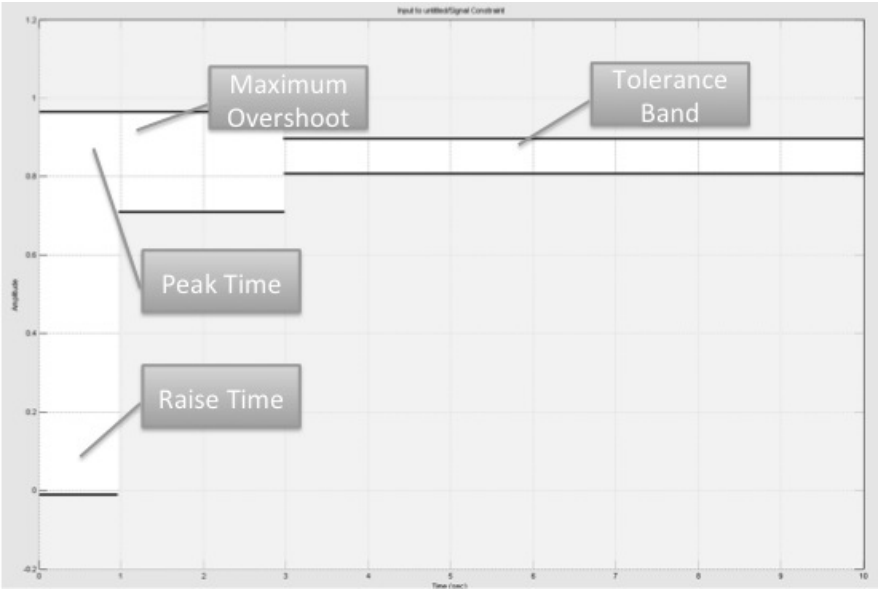


Figure 18: Interface of Signal Constraint Toolbox

After tuning process is completed, response of the system is shown in Figure 19.

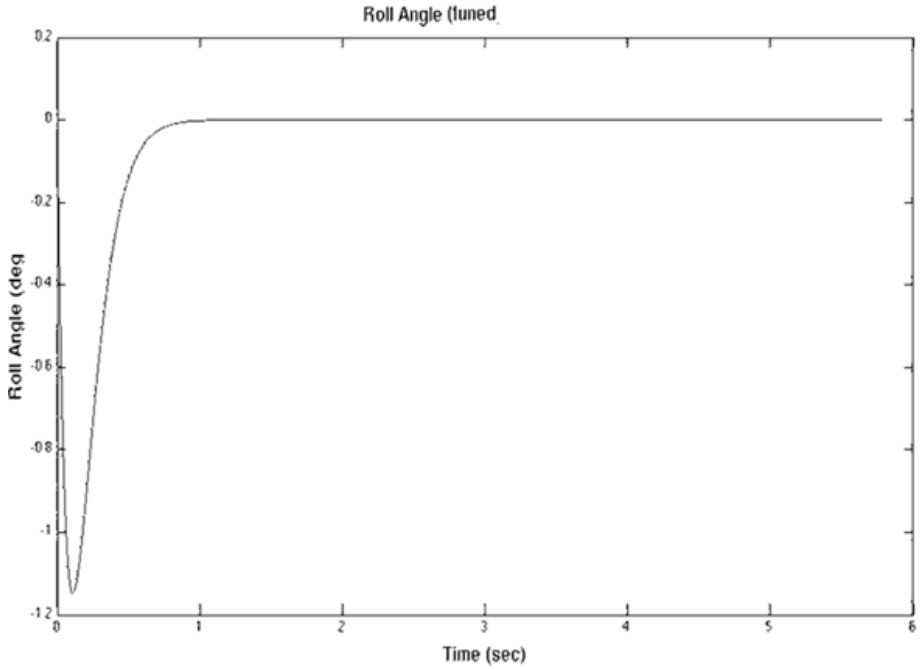


Figure 19: Roll Angle Graph, K_p=25.2, K_i=12.9, K_d=12.9

The Simulink model is modified for design pitch axis controller and shown in Figure 20.

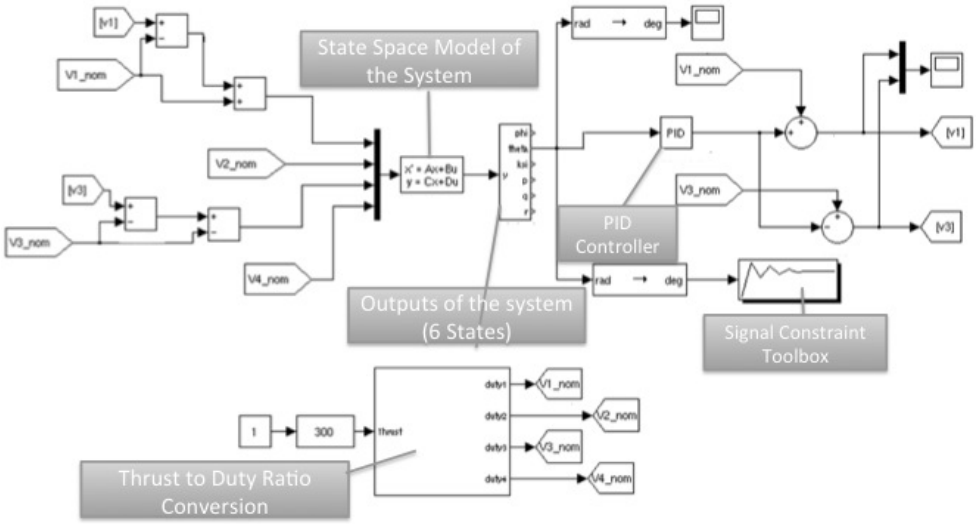


Figure 20: Simulink Model for Pitch Axis with PID Controller

In this controller type, initial condition for the pitch angle is given as, θ_i, given as 0°. The response of the system with the controller parameters K_p=1, K_i=1, K_d=1 is shown in Figure 21.

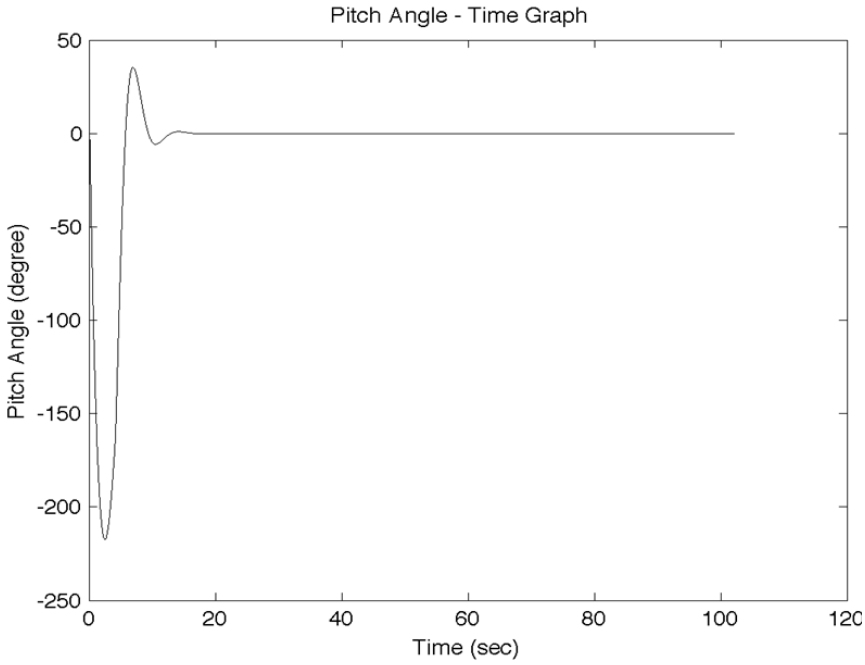


Figure 21: Pitch Angle Graph, K_p=1, K_i=1, K_d=1

As can be seen from the Figure 21, these parameters have to be tuned. For tuning "Signal Constraint" block of the Matlab/Simulink software is used. After tuning process is completed, response of the system is shown in Figure 22.

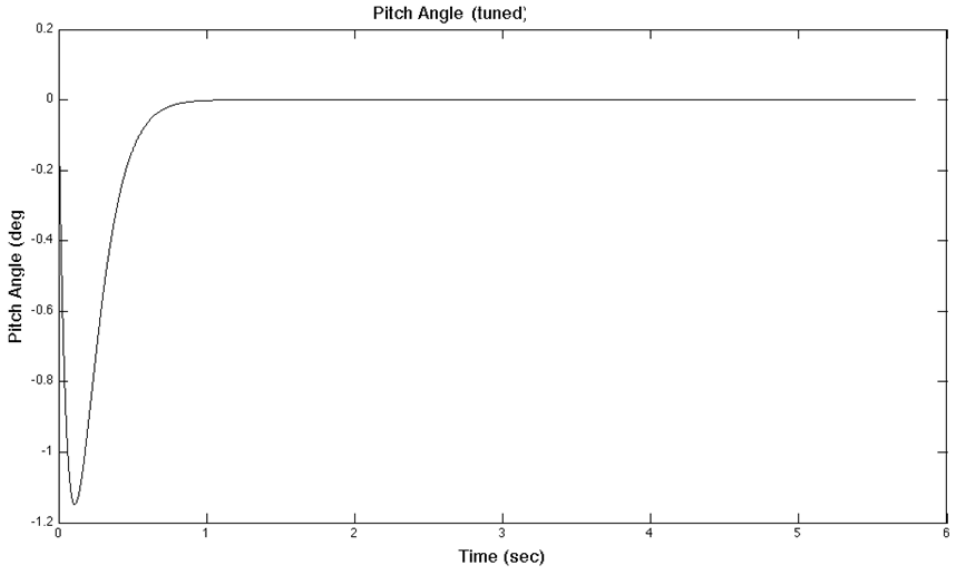


Figure 22: Pitch Angle Graph, K_p=22.4, K_i=12.3, K_d=12.4

Since the mechanical structure of the quadrotor is manufactured symmetrically, simulations results of roll and pitch axes are nearly close. Then, these controller parameters will be tested to the physical system.

4.3 LQR Controller Design and Simulation

Optimal regulator is designed to achieve higher performance. Six attitude states are augmented with two additional error states and a regulator is designed for this new augmented system. Roll and pitch angles are the outputs of the system.

$$y = \begin{bmatrix} \varphi & \theta \end{bmatrix}^T \tag{4.1}$$

Reference angles are

$$R = \begin{bmatrix} R_\varphi & R_\theta \end{bmatrix}^T \tag{4.2}$$

Error is defined as

$$e = R - y \tag{4.3}$$

State equations for new states are given below.

$$\dot{x}_I = e = R - y = R - Cx \tag{4.4}$$

Augmented state vector is given as below.

$$\bar{x} = \begin{bmatrix} x \\ x_I \end{bmatrix} \text{ where } x = \begin{bmatrix} \varphi & \theta & \psi & p & q & r \end{bmatrix}^T \tag{4.5}$$

State equations for the augmented state vector are given below.

$$\dot{\bar{x}} = \begin{bmatrix} \dot{x} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} R \tag{4.6}$$

A and *B* are defined above in the previous chapter. Output matrix *C* is given below.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.7}$$

Control input to the system becomes;

$$u = -\bar{K}\bar{x} = -\begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} \tag{4.8}$$

Cost function to be minimized is given as

$$J = \int_0^\infty \left[\bar{x}^T Q \, \bar{x} + u^T R_{uu} u \right] dt \tag{4.9}$$

Q is positive semi-definite and R_{uu} is positive definite [27].

$$Q = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_4 \end{bmatrix} \tag{4.10}$$

$$R = b.I_m \tag{4.11}$$

Elements of Q matrix affect the states. I_m represents the diagonal matrix and b is the weight on the input [9]. Optimal gain matrix, K, is calculated in Matlab [28]. By using the error states and regulator for the augmented system, effects of steady disturbances are eliminated [29]. This will introduce robustness to a certain level. The system is modeled in Simulink in state-space representation presented in Figure 23.

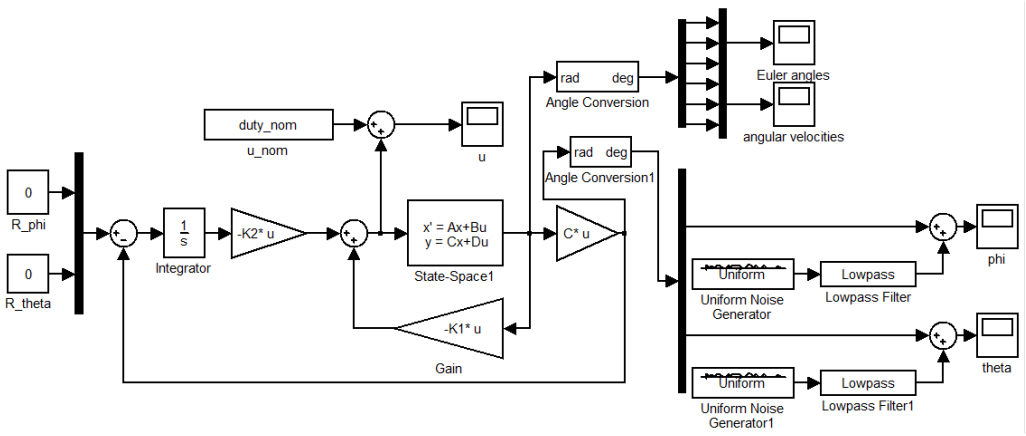


Figure 23: Simulation Model for Roll and Pitch Axis with LQR Controller

To run the model, which is shown in Figure 23, first an m file, which contains all parameters in the model, has to be run. This m file can be seen in Appendix B. The Q matrix is given in below.

$$Q = \begin{bmatrix} 290 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 290 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 70 \end{bmatrix} \quad (4.12)$$

Q(1,1), Q(2,2), Q(3,3), Q(4,4), Q(5,5), Q(6,6) are the gains of attitude dynamics. In Table 2, gains and state relation is given.

Table 2: State and Gain Relation

Q Matrix Index	**Gain**	**Related State**
Q(1,1)	290	Roll angle state
Q(2,2)	290	Pitch angle state
Q(3,3)	0.001	Yaw angle state
Q(4,4)	1	Roll angular velocity state
Q(5,5)	1	Pitch angular velocity state
Q(6,6)	0.005	Yaw angular velocity state

Q(7,7) and Q(8,8) are related with the integral of the errors of roll and pitch states, respectively. Yaw dynamics is not controlled via LQR controller. R_{uu} matrix is the input matrix whose gains are related with the motor signals. R_{uu} matrix is given below.

$$R_{uu} = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix} \quad (4.13)$$

Controller gains, which are K_1 and K_2, are calculated in this m file according to the *Q* and *R* matrices elements and given below.

$$K_1 = \begin{bmatrix} -2.6537 & 0.0118 & -0.1149 & -0.8394 & 0.0908 & -0.3150 \\ 0.0231 & -2.6466 & 0.1137 & 0.0939 & -0.8411 & 0.3115 \\ 2.7693 & -0.0107 & -0.1102 & 0.8725 & -0.0905 & -0.3017 \\ -0.0226 & 2.776 & 0.1083 & -0.0947 & 0.8789 & 0.2966 \end{bmatrix} \quad (4.14)$$

$$K_2 = \begin{bmatrix} 0.4892 & 0.0009 \\ -0.0011 & 0.4877 \\ -0.5106 & -0.0012 \\ 0.0010 & -0.5120 \end{bmatrix} \quad (4.15)$$

Initial angles of roll and pitch angles are given as -1° and 1°, respectively. Reference angles of roll and pitch angles are 0°. Output graphs of roll and pitch dynamics are given in Figure 24 and Figure 25, respectively.

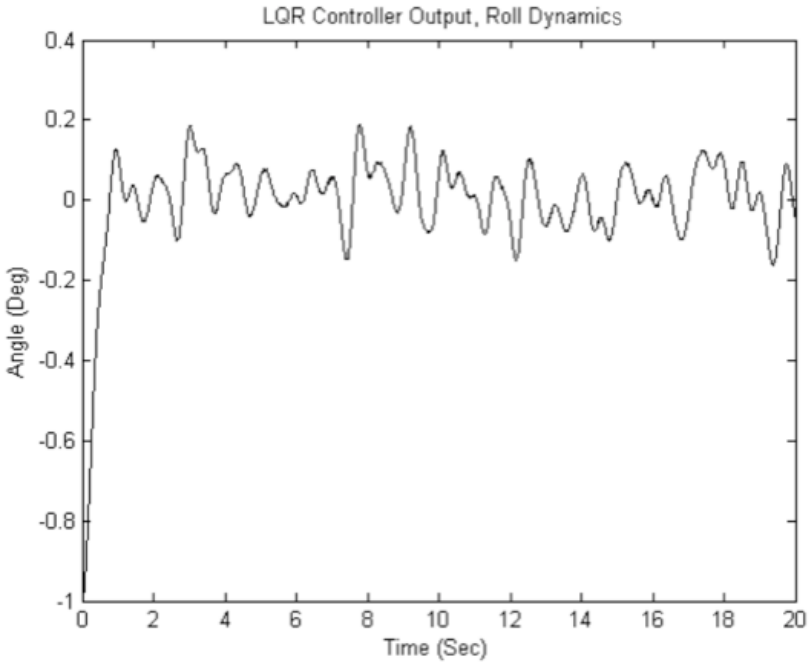


Figure 24: Simulation Model Output of Roll Dynamic with LQR Controller

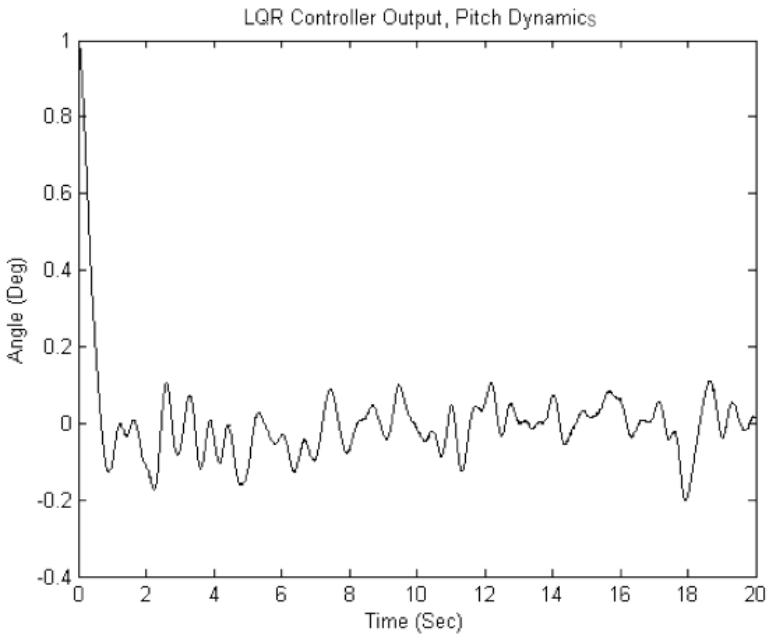


Figure 25: Simulation Model Output of Pitch Dynamic with LQR Controller

4.4 Altitude Controller Design and Simulation

The formula, which is shown in the altitude controller section, is given below and modeled in Simulink environment as seen in Figure 26.

$$\ddot{z} = \frac{\sum_{i=1}^{4} F_i}{m} \cos\theta \cos\varphi - g \tag{4.16}$$

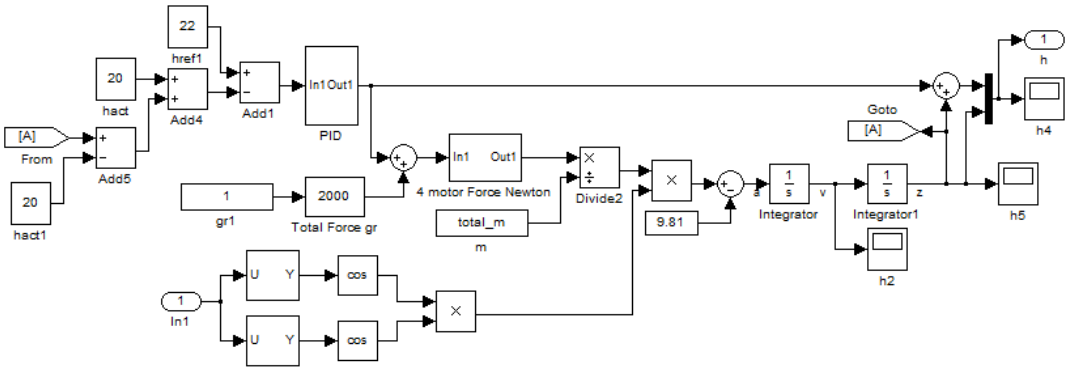


Figure 26: Simulink Model of Altitude Control

CHAPTER 5

PHYSICAL SYSTEM AND REAL TIME CONTROL APPLICATIONS

This chapter describes the physical realization of the quadrotor and real time application of the model based control algorithms on the system. All system components are given in detail in this chapter. The quadrotor is composed of;

- Brushless DC motors and attached propellers
- Electronic speed controllers
- Inertial measurement unit that consist of accelerometers, gyros, magnetometers, and complementary filter
- Proximity sensor
- Power supplies
- Landing gears

The photo of the system can be seen in Figure 27.

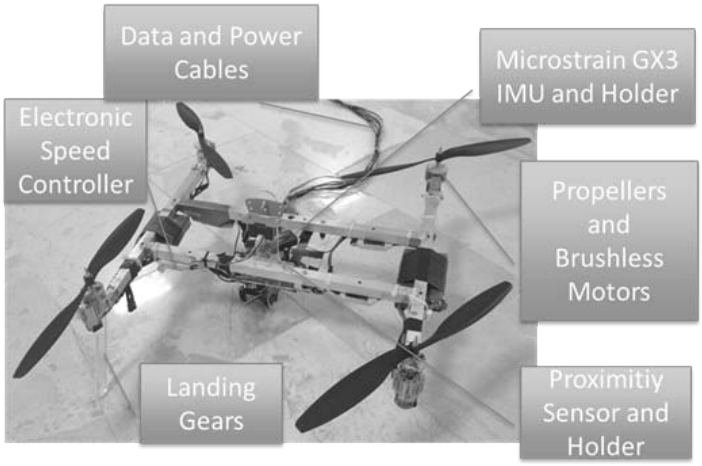


Figure 27: Physical System

The real-time controller is generated in Matlab Simulink by using xPC Target Blockset. Controller hardware is not embedded onto the system. The major aim is to design and develop a quadrotor that has stable and controlled attitude and altitude dynamics. Basic control algorithms are implemented on the system. Further developments and more complex control algorithms can be added to the system; they are not covered in the scope of the thesis. A simple schematic representation of the relation among the components is given in Figure 28.

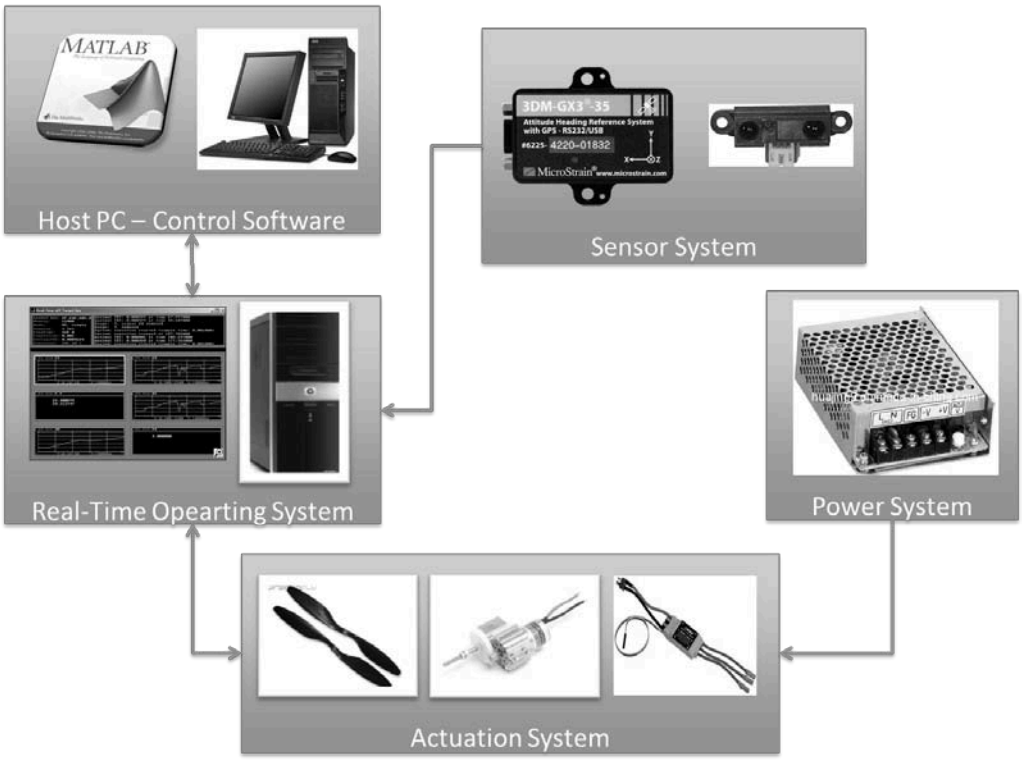


Figure 28: Physical System

Detailed information about the elements of the physical system is described in the following sections, namely, mechanical structure, sensor system, actuator system, controller system parts, and overall system.

5.1 Mechanical Structure

Mechanical structure of the quadrotor is made of aluminum. All components are assembled to the mechanical structure. The mechanical structure can be seen in Figure 29.

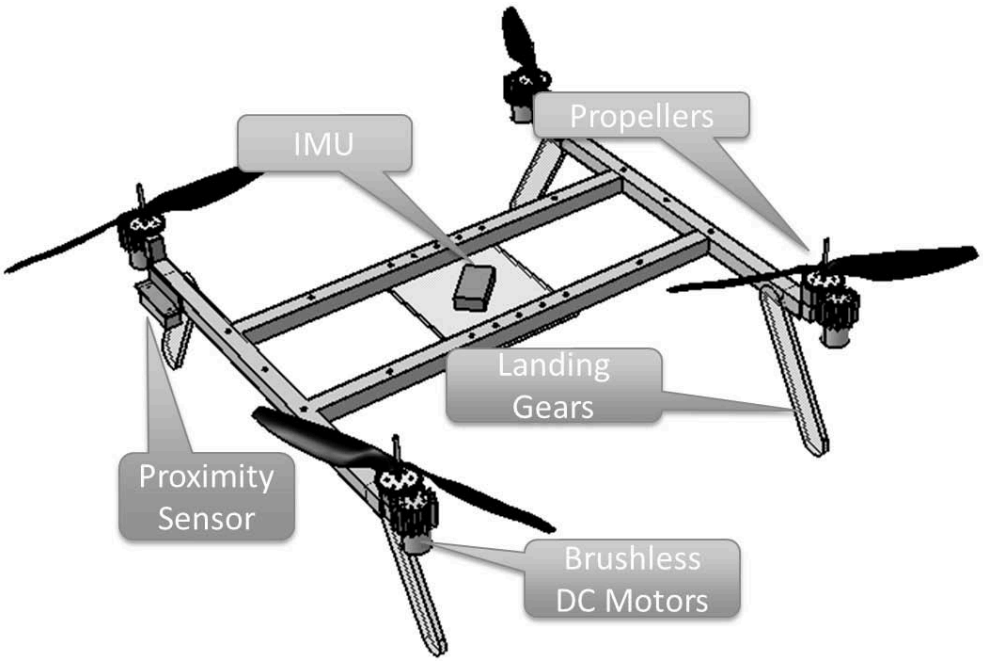


Figure 29: Mechanical Structure

Top view of the chassis can be seen in Figure 30. The chassis is made of aluminum profiles. All the components are assembled to the chassis. To assemble the components, screws are used.

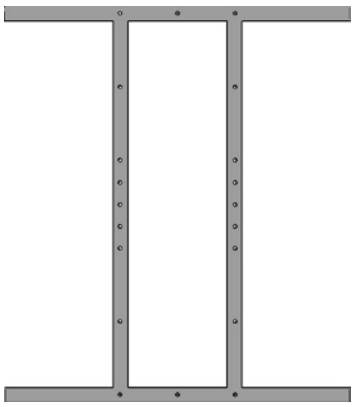


Figure 30: Chassis of the Quadrotor

In the mathematical modeling chapter, the inertia matrix is given in equation 6.1 parametrically.

$$I = \begin{bmatrix} Ixx & -Ixy & -Ixz \\ -Iyx & Iyy & -Iyz \\ -Izx & -Izy & -Izz \end{bmatrix} \tag{5.1}$$

Inertia parameters of the whole system are calculated experimentally. $-I_{xy}$, $-I_{xz}$, $-I_{yx}$, $-I_{yz}$, $-I_{zx}$ $-I_{xy}$, and $-I_{zz}$ $-I_{zy}$ are assumed as zero. To calculate I_{xx}, I_{yy}, and I_{zz} bifilar pendulum experiments are done [30]. According to this experiment, the system is hanged from two sides by ropes. Then, the whole system is rotated by hand and released free. Because of the tension on the ropes, the system starts to swing about the axis that the inertia is to be calculated. Appearance of bifilar pendulum experiment is illustrated in Figure 31 [30].

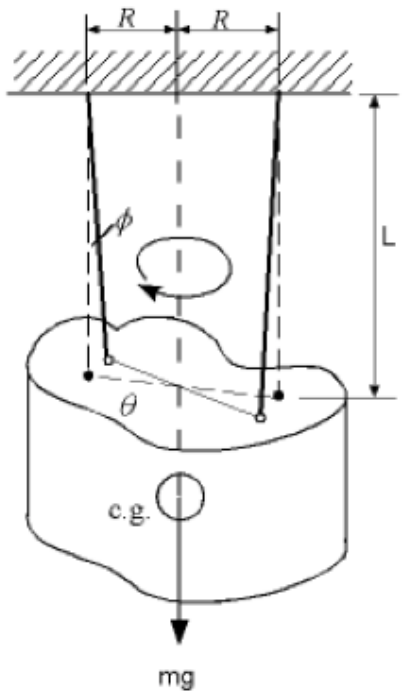


Figure 31: Bifilar Pendulum Experiment [29]

The hanged system during the experiments can be seen in Figure 32.

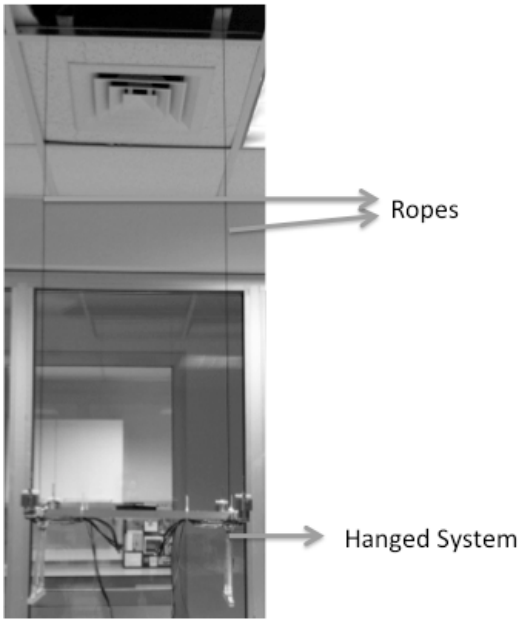


Figure 32: Hanged System

By the help of this experiment, mass moment of inertias about roll, pitch, and yaw axes can be determined. However, since the system is symmetric, the mass moment of inertia about roll and pitch axes are assumed to be same. To determine the mass moment of inertia about rotation axis, the following formula is used.

$$J = \left[\frac{T_n}{2\pi}\right]\frac{mgR^2}{L} \tag{5.2}$$

Where,

T_n= Measured swing period (sec),

m = Mass of the system (kg),

g = Gravitational acceleration (m/s^2)

R =Distance Radius (m),

L = Length of the ropes (m).

The angular rotation data is taken from the IMU, which is placed onto the quadrotor. For yaw mass moment of inertia, I_{zz}, Figure 33 is obtained.

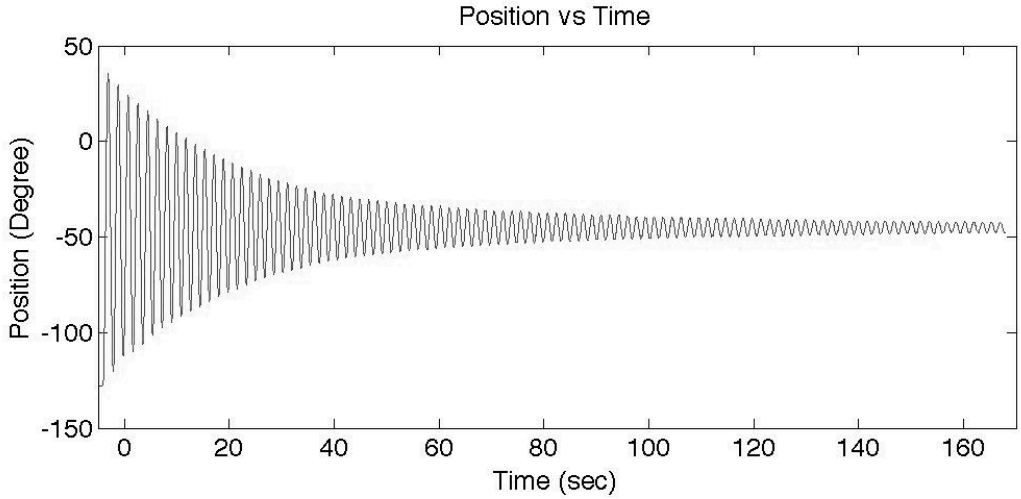


Figure 33: Yaw Swing

Measured swing period for yaw axis is, 1.75 second and other parameters of the equation 5.2 are shown in Table 3;

Table 3: Values for I_{zz}

m, (kg)	1.4
g, (m/s^2)	9.81
R, (m)	0.19
L, (m)	1.2

After substituting the values into the equation 5.2, I_{zz} becomes 0.0321 kgm^2. Same process is done for roll axis and Figure 34 is obtained

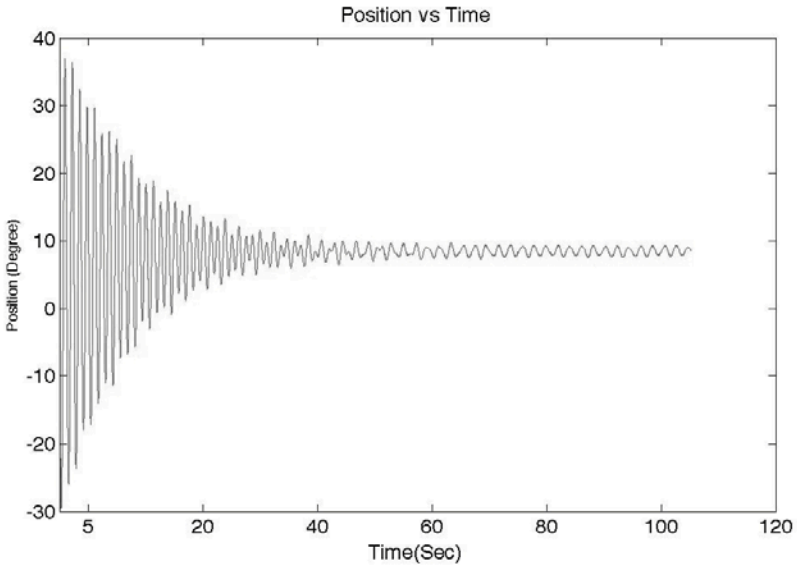


Figure 34: Roll Swing

Measured swing period for roll axis is 1.30 second and other parameters of the equation 5.2 are shown in Table 4;

Table 4: Values for I_{xx}

m, m(Kg)	1.4
g, (kgm/s^2)	9.81
R, (m)	0.25
L, (m)	1.2

After substituting the values into the equation 5.2, I_{xx} becomes 0.0321 kgm^2. As a result, experimentally determined inertia values about rotation axes are given in Table 5;

Table 5: Mass Moment of Inertias

I_{xx}, (kgm^2)	0.0306
I_{yy}, (kgm^2)	0.0306
I_{zz}, (kgm^2)	0.0321

5.2 Sensor System

An IMU and a proximity sensor are the onboard sensors.

5.2.1 Inertial Measurement Unit

An IMU is used in order to measure Euler angles and angular velocities. To measure these parameters MicroStrain 3DM GX3 IMU that is shown in Figure 35 is used. 3DM-GX3 includes a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensor, GPS, and an onboard processor. The communication with controller is done by RS232 protocol.

Figure 35: MicroStrain 3DM GX3

The output of the IMU is a packet; needed data has to be edited according to the information of the technical document of the IMU. Taken data packet information is given in Table 6 [30].

Table 6: IMU Packet Information

Measurement	Byte
Acceleration X	2-5
Acceleration Y	6-9
Acceleration Z	10-13
Angular Rate X	14-17
Angular Rate Y	18-21
Angular Rate Z	22-25
$M_{1,1}$	26-29
$M_{1,2}$	30-33
$M_{1,3}$	34-37
$M_{2,1}$	38-41
$M_{2,2}$	42-45
$M_{2,3}$	46-49
$M_{3,1}$	50-53
$M_{3,2}$	54-57
$M_{3,3}$	58-61

In the thesis, the sampling period of the IMU is adjusted as 300 Hz. To take data from the IMU, it is needed to send a request command. It is possible to take different types of data from the IMU. In the thesis, IMU is used to give Euler angles, angular rates, and accelerations mode.

Each output of the IMU is converted to their values with using IEEE 754 conversion method and Euler angles, angular rates, and accelerations are obtained.

5.2.2 Proximity Sensor

To measure the altitude of the system a Sharp GP2Y0A02YK0F Analog Distance Sensor, which is shown in Figure 36, is used. The sensor has a 20-150 cm measurement range. The sensor gives an analog output according to the position of the object in front of it. Output characteristic of the sensor is given in Figure 37 [32].

Figure 36: GP2Y0A02YK0F Sharp Sensor

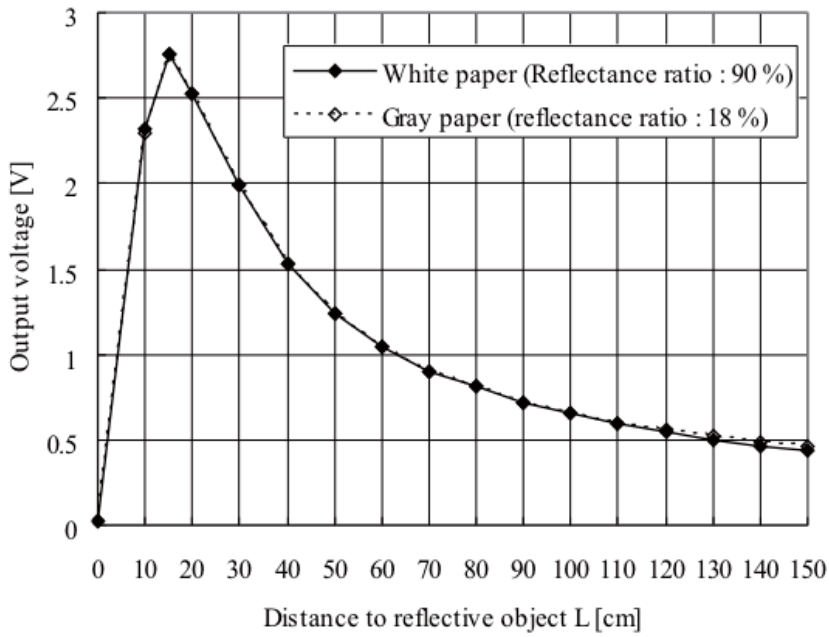


Figure 37: Output Characteristic of the Altitude Sensor [30]

With taking into account the curve in 20 – 150 cm range in Figure 37, a voltage-distance relation is obtained.

$$z = 24.179V^4 - 175V^3 + 472.78V^2 - 585.78V + 326.71 \quad (5.3)$$

Where,

z is altitude in cm,

V is output in Volts.

5.3 Actuation System

To actuate the system, four brushless motors are used. The system has four propellers and each one is actuated independently. That is why; there are four brushless motors on the system. Used brushless motor are Art-Tech B2025-15L, which can be seen in Figure 38. The mass of the motor is 60 grams and there is a 1/5 reduction to the propeller shaft.

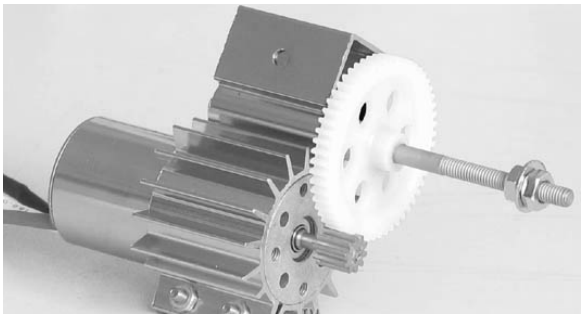


Figure 38: Art-Tech B2025-15L Brushless Motors

These brushless motors have three phases and the output of the controller is one phase PWM signal. That is why; it is needed to use a bridge circuit, which converts a PWM signal into a three-phased signal.

As a bridge circuit, an EMAX 30A Electronic Speed Controller, which is shown in Figure 39, is used.

Figure 39: EMAX Electronic Speed Controllers

To create thrust forces, a propeller is assembled to each brushless motor. EPP 1245 propellers, which is shown in Figure 40, are used.

Figure 40: ECC 12x45 Propellers

A physically assembled motor, ESC, and a propeller actuation unit can be seen in Figure 41.

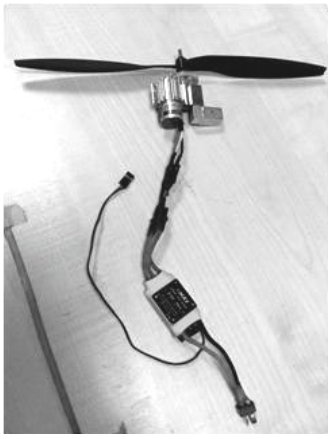


Figure 41: Propellers, Motor, and ESC Actuation Unit

5.4 Controller Hardware

The preferred controller software is Matlab Simulink. During development of the controller, two desktop PCs are used. One desktop PC stands for designing the controller named as host PC. And, the other one used for xPC target to study in real-time. The structure is shown in Figure 42.

Figure 42: System Structure During Development

The IMU is connected to the target PC via RS 232 protocol. Humusoft MF624 DAQ card, which is shown in Figure 43, is used to take data from proximity sensor and to send PWM control signals to the ESC's.

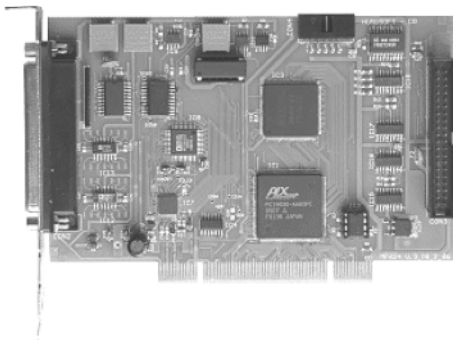


Figure 43: Humusoft MF624 DAQ Card

The MF 624 contains 8 channel fast 14 bit A/D converter with simultaneous sample/hold circuit, 8 independent 14 bit D/A converters, 8 bit digital input port and 8 bit digital output port, 4 quadrature encoder inputs with single-ended or differential interface and 5 timers/counters. Output of the proximity sensor is connected to analog input pin of the card. Four controller output signals are taken from four PWM output pins.

5.5 Controller Software

The controller software is written in Matlab Simulink, with xPC Target toolbox of Matlab. The collected data from the proximity sensor and the IMU is processed in the controller. According to the controller algorithm, controlled output signals goes to the ESC's. Sampling time of the controller software is 300 Hz. At first, PID type controller is used to control the system. After some developments LQR type controller is implemented. During development of the controller, controller software is loaded from host PC to target PC. Then, it runs in target PC.

CHAPTER 6

EXPERIMENTS WITH DESIGNED CONTROLLERS

6.1 Tests on Setup

Firstly, closed-loop attitude dynamics is tested on a special test setup. After mounting the quadrotor on this test setup, it is possible to see the response of the system according to the designed controller parameters. During the experiments nominal voltage applied to motors is sufficient to produce a lift force which is equal to the total weight of the quadrotor. During the tests, disturbances are given to the system and response of the system is observed. The test setup is given in Figure 44.

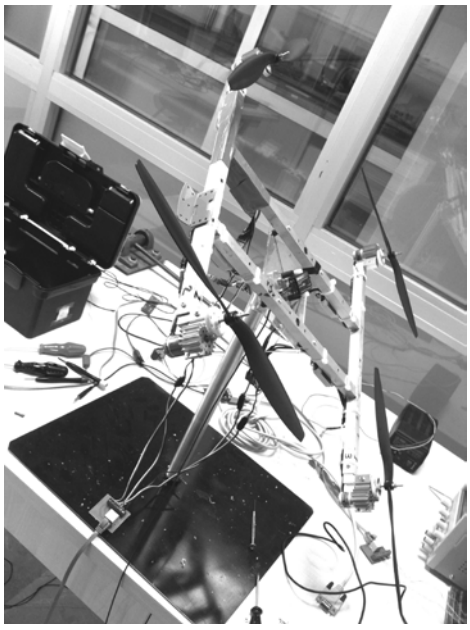


Figure 44: Roll – Pitch Experiment Setup

Quadrotor is mounted to this setup with a spherical joint, which is shown in Figure 45.

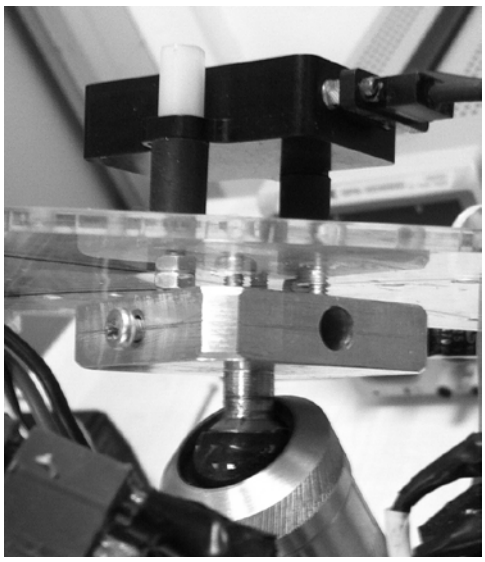


Figure 45: Spherical Joint

6.1.1 Experiments with PID Controller

Controller parameters, which are found during the simulation of the system with the PID controller, are applied to the system and outputs of the controllers are given in Figure 46 and Figure 47.

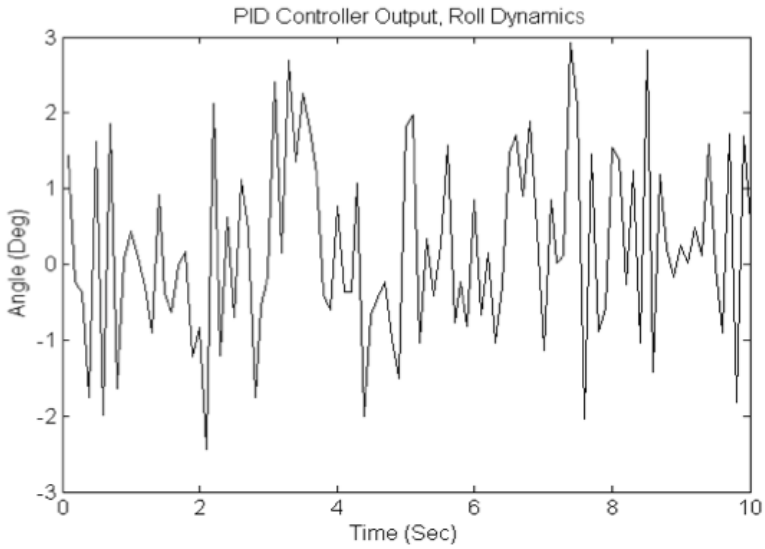


Figure 46: PID Controller, Roll Dynamics Output

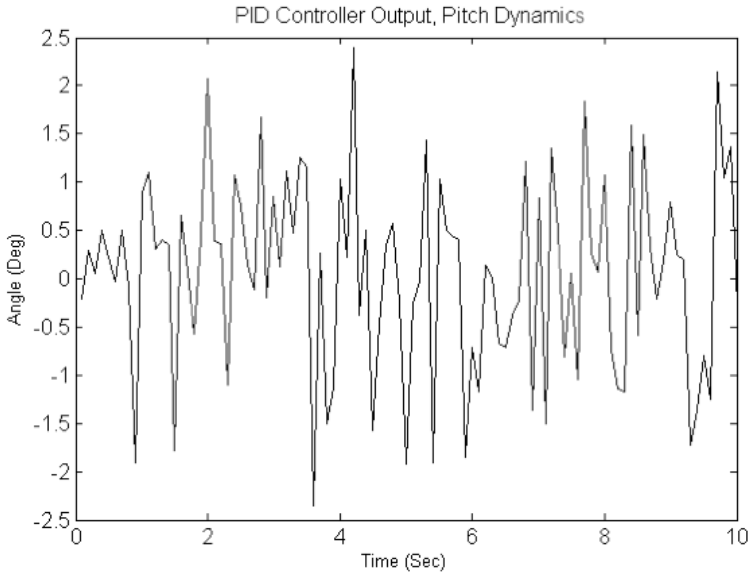


Figure 47: PID Controller, Pitch Dynamics Output

6.1.2 Experiments with LQR Controller

Controller parameters, which are found during the simulation of the system with the LQR controller, are applied to the system and outputs of the controllers are given in Figure 48 and Figure 49.

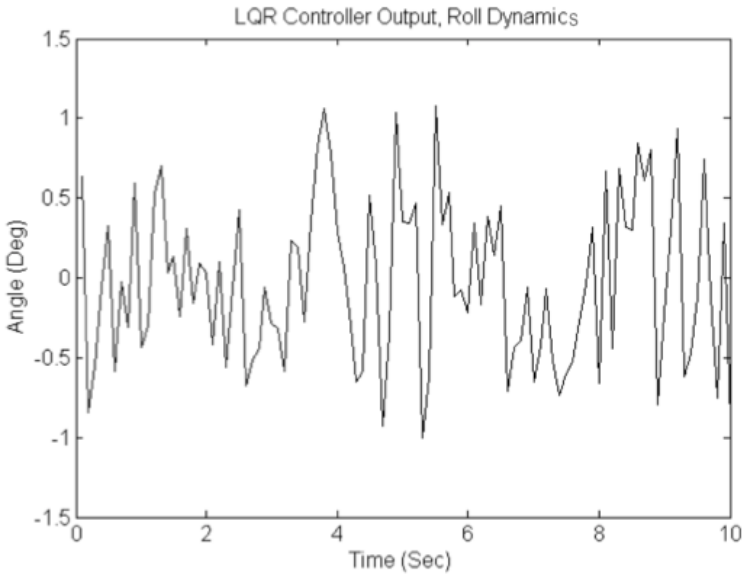


Figure 48: LQR Controller, Roll Dynamics Output

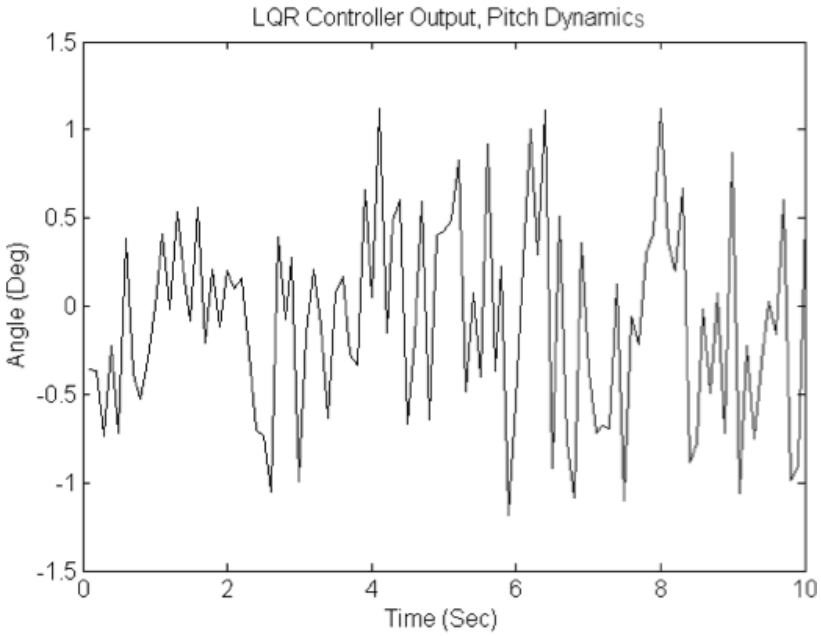


Figure 49: LQR Controller, Pitch Dynamics Output

6.1.3 Comparison of LQR and PID Controllers

During the tests the system is connected to the spherical joint setup. Tuned controller parameters during the simulation of the system with PID and LQR type controllers are applied to the system. During the tests, only controllers are changed and all other parameters remain constant to see the real performance differences between the LQR and PID controllers. As can be seen from Figure 50 and Figure 51, LQR controller gives better results compared to the PID controller. To have this result, integral action of the LQR controller plays an important role. [33].

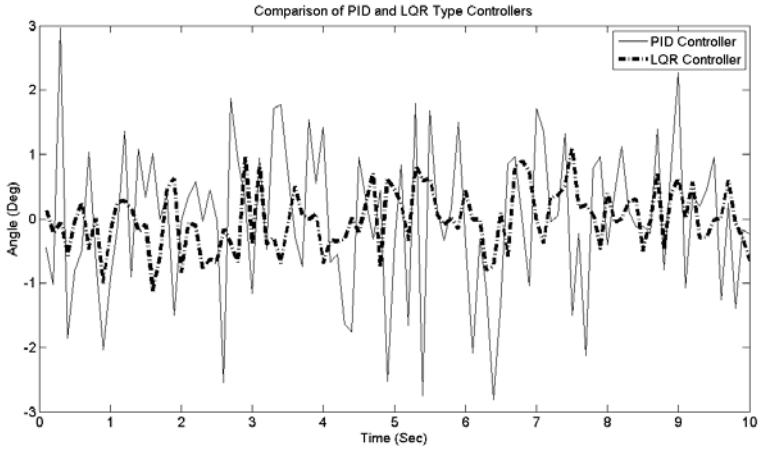


Figure 50: Comparisons of PID and LQR Type Controllers in Roll Dynamics

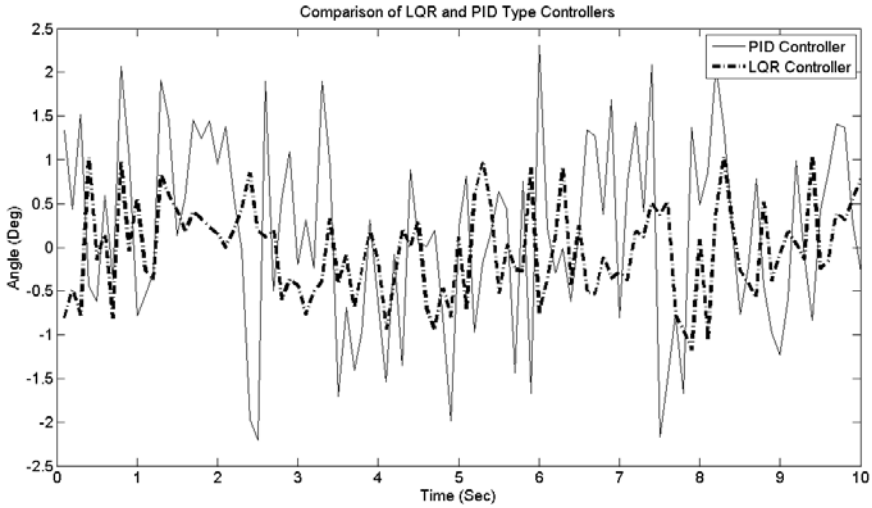


Figure 51: Comparisons of PID and LQR Type Controllers in Pitch Dynamics

6.2 Yaw Controller Experiments

In the system, the motors of the quadrotor are numbered as 1,2,3, and 4, first and third ones rotate counter-clock wise where second and fourth ones rotate clockwise direction [34]. Theoretically, with this counter rotation combination, produced drag moment from the propellers should be zero. However, in practice it is needed to design a controller to make the moment zero. A PID controller is designed for yaw controller. Difference between the measured yaw dynamics and reference, which is the error, goes into the controller. Then, the output of the controller is divided into two parts. In one part, the controller signal is added directly to the motor signals. Another part is multiplied with “-1” then added to the motor signals. By this way, movement in Z direction is controlled. The Simulink model, which is used for the PID controller, can be seen in Figure 52.

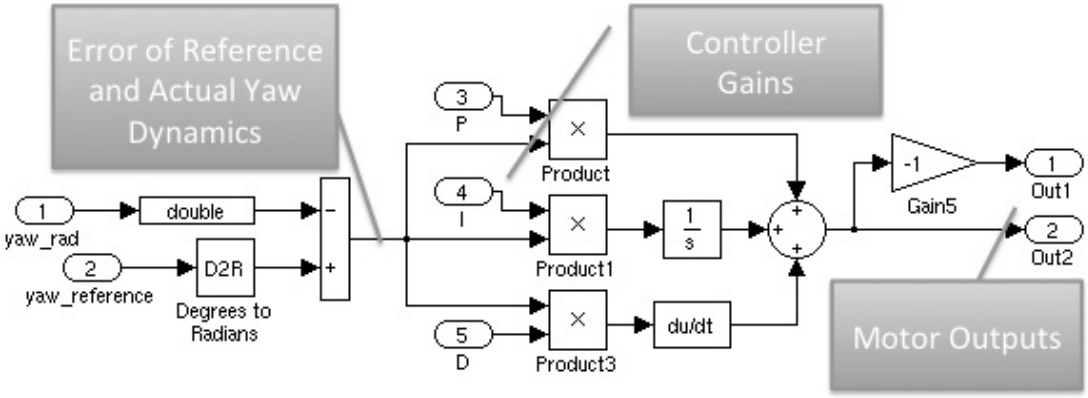


Figure 52: The Simulink Model for Yaw Dynamics

6.3 Ground Tests and Results

To see the performance of the designed controller, the system should be free to move.. The photo of the setup can be seen in Figure 53.

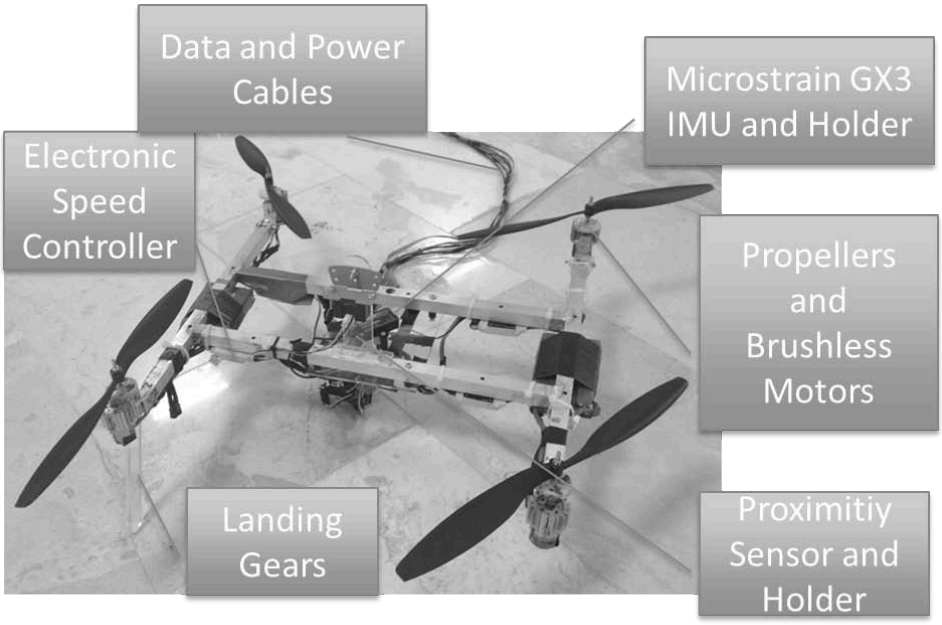


Figure 53: System on the Ground

The code to control system is designed by using a host PC. The code is downloaded to the target PC. In the target PC, xPC Real-Time operating system is running. Data connection between the target PC and the IMU are provided by RS-232 communication protocol. A MF624 Humusoft Data Acquisition card is also

used to take altitude sensor output to the target PC and send the output signals of the controllers to the system. Whole test system can be seen in Figure 54.

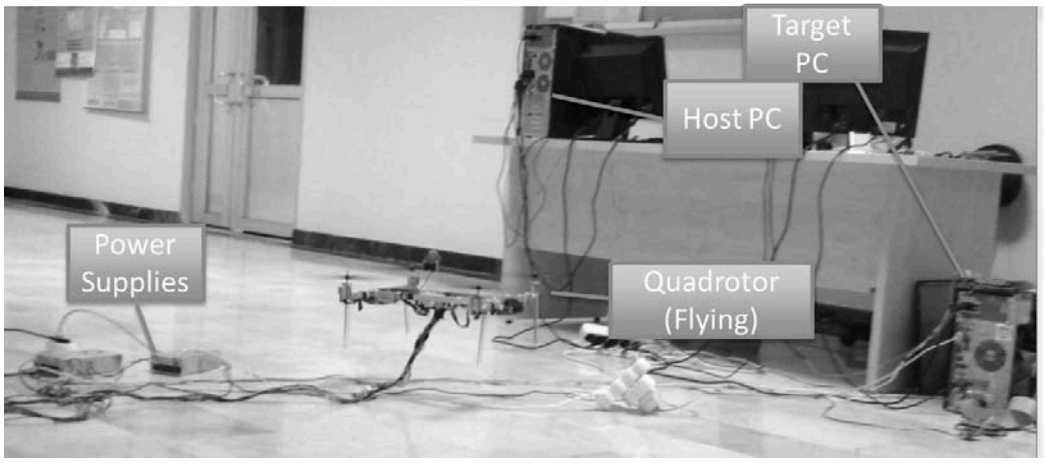


Figure 54: System Test Setup

As stated before, the aim of the project is to control the attitude and altitude dynamics of the quadrotor. After modeling the system and designing PID and LQR controllers, simulations showed that the performance of the LQR controller is better than the PID one. Then, these controllers are tested on the system, which is connected to the roll – pitch test setup. The result is same that LQR controller gives better result compared to the PID controller. That is why, an LQR controller is used to control roll and pitch dynamics. For yaw and altitude dynamics, two PID controllers are used. The taken data from the IMU goes to these three controllers and outputs of the controllers are summed and sent to the actuators. A screenshot of the Simulink model that is used in real time control is given in Figure 55.

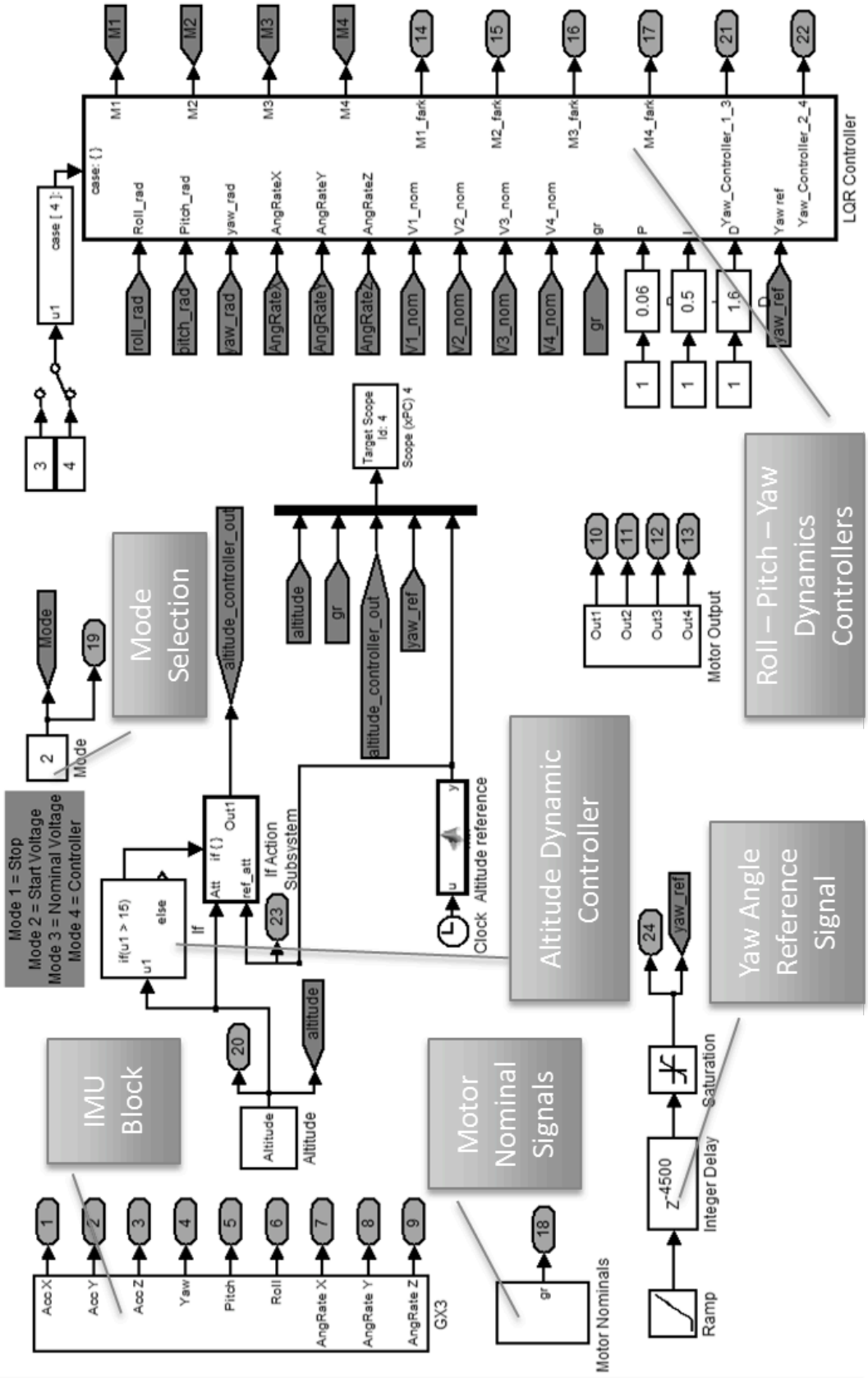


Figure 55: Complete Simulink Model of the System

Some operating modes are defined to perform the tests safely. Model IDs and descriptions can be seen in Table 7.

Table 7: Modes of the Model

Mode ID	Definition
1	Data taken from the IMU and the altitude sensor are collected. There is no signal goes to the motors. Controllers for attitude and altitude dynamics are closed. The aim of design such a mode is to check data connection and whether the sensors work or not.
2	Sending signals to the motors starts in this mode. However, these signals are not powerful to rotate the motors. At this moment, there is a sound, which comes from the each ESC. In this mode, functionality of the ESCs is tested.
3	There is a nominal thrust signal, which is given to the system. The nominal thrust value is converted into the motor signals in the model. Attitude and altitude controllers are opened in this mode. However, they are not sent to the system. In this mode, it is tested whether motors works or not and controller outputs.
4	Attitude and altitude controller outputs are sent to the system. In this mode, the system is able to fly referenced altitude and attitude values.

To perform the experiments, some steps are followed. First, a code, which contains some controller parameters for the system, has to be run. The code is given in Appendix A. Then, the model, which is designed in the host PC, has to be run. During that operation, power supplies, which supplies to the brushless motors have to be off. The data taken from the IMU has to be observed to keep the platform level as much as possible. Necessary adjustments can be done with changing the angles of landing gears. After the adjustments are completed, the program has to be stopped. Then, reference angle values for yaw angle and height of the system have to be defined. Then, the model has to be rebuilt. After

rebuilding process is done, power supplies have to be opened. Then, the system is ready to fly. Initial mode of the system is mode 2. After the simulation starts, in 10 seconds, mode 3 and mode 4 has to be performed by the user. Then the model starts to send the reference signals to the system and it starts to fly. A picture from the flying tests of the system is taken and can be seen in Figure 56 and Figure 57.

Figure 56: System on the Ground

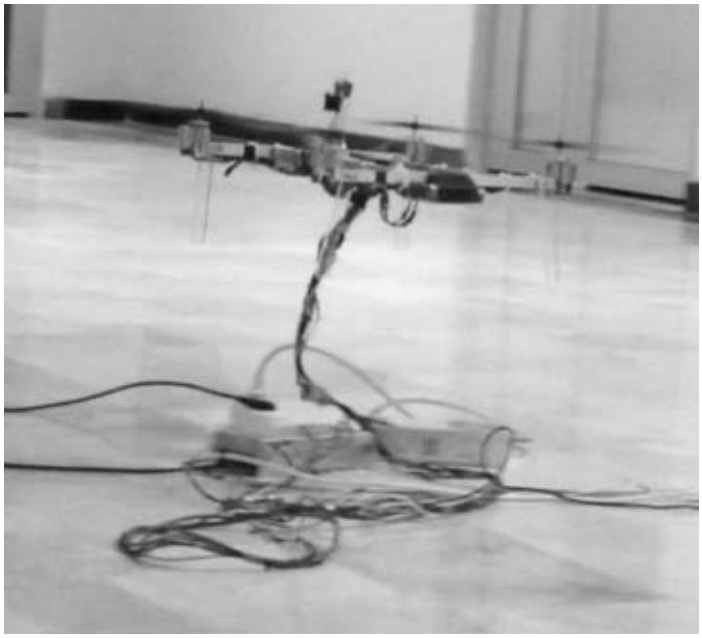


Figure 57: Flying System

During the test, reference values of roll, pitch, and yaw angles are 0° and altitude is 30 cm above ground. After the simulation is done, the model runs a code

automatically. All data, which are reference thrust, motor duty ratios, controller outputs, roll, pitch, yaw and altitude dynamics, during the simulation are logged. By means of the code seen in Appendix D, all data are graphed.

As stated in Section 4.3, outer integral controller is added to eliminate steady state errors. The controller can be seen in Figure 58. This screenshot is just one part of Roll-Pitch-Yaw controller block, which is seen in Figure 55.

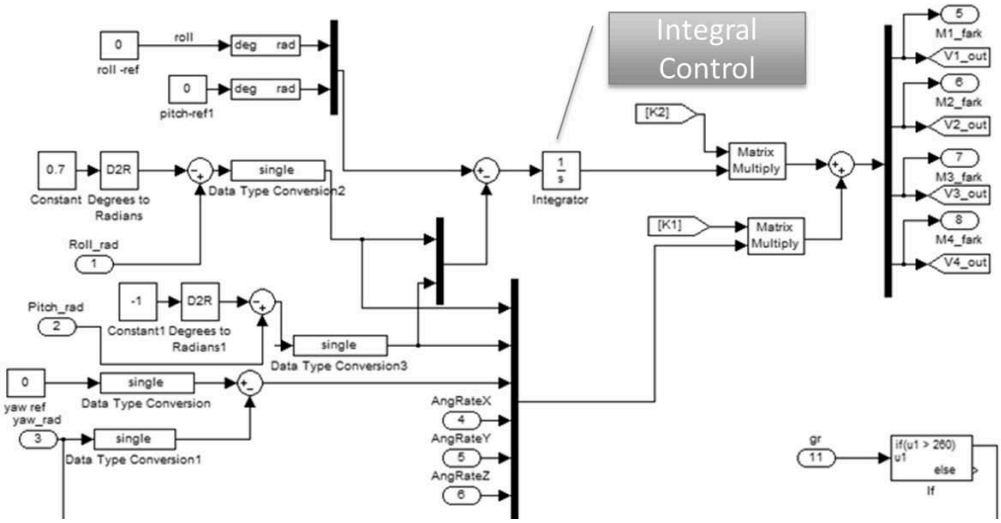


Figure 58: Outer Integral Control

6.3.1 Tests for Roll and Pitch Dynamics

After a test is completed, logged attitude and altitude data are graphed automatically. The output of the IMU for roll dynamic is shown in Figure 59. As can be seen from the Figure 59, there are four regions on the figure. First region represents that the system states stationary on the ground. As stated before, motor starts to rotate at 10^{th} second. The motors start to speed up to produce enough thrust force to lift the system up. This process takes 2-3 seconds. The process can be seen in region 2. Angle changes, which are seen in region 2, are caused by the noise of the vibration. This vibration comes from the rotation of motors. In the third region, the system flies. As can be seen the roll angle changes between ±1° in the third section. Since the system flies with the cables some disturbances occur

during the flying. In section 4, some angle disturbances, which come from the cables, can be seen.

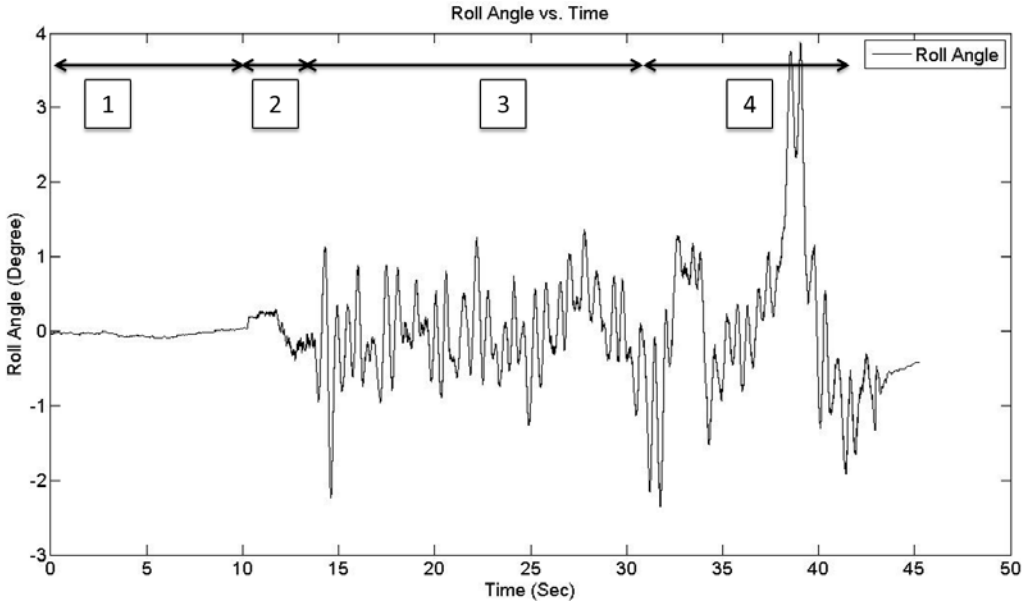


Figure 59: Roll Angle Output

To compare the real system output data and the simulated data, third section of the Figure 59 is taken. Because, in the simulation, it is assumed that the system is flying initially. Comparison of the simulated data and real system output data can be seen in Figure 60.

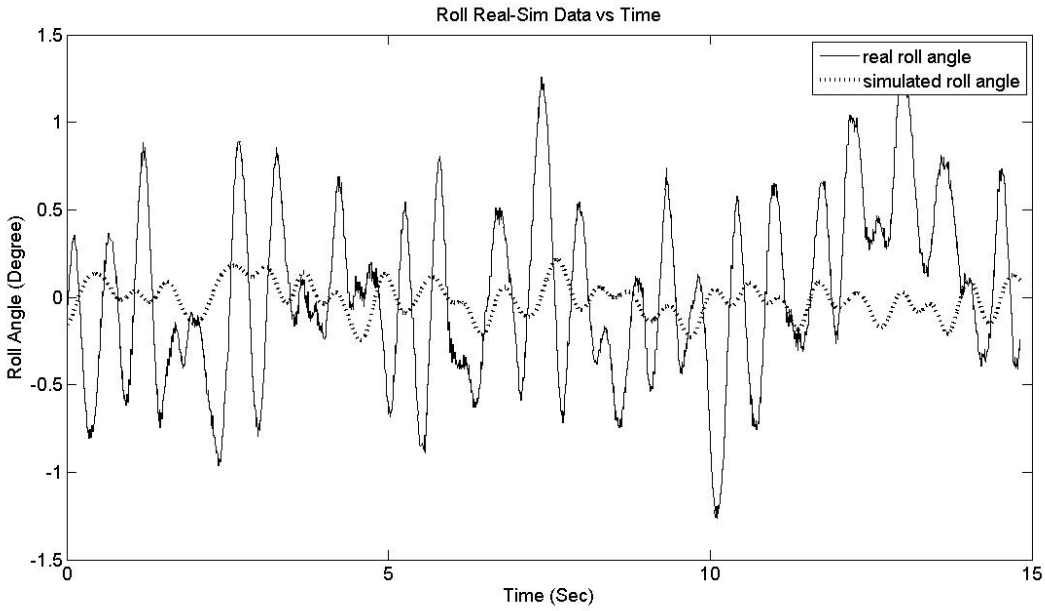


Figure 60: Comparison of Real and Simulated Roll Angles

Real system output of the system for the pitch axis can be seen in Figure 61.

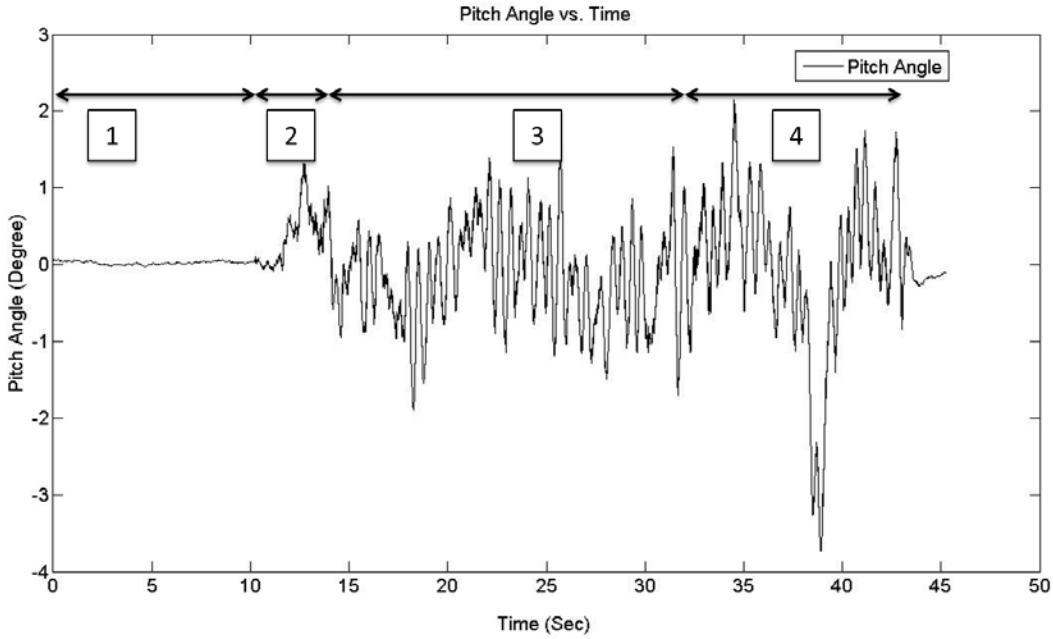


Figure 61: Pitch Angle Output

As stated before, there are four regions on the graph. Explanations, which are done for the roll axis, are valid for also pitch axis. To compare real pitch angle output and the simulation output third region is taken. The comparison graph can be seen in Figure 62.

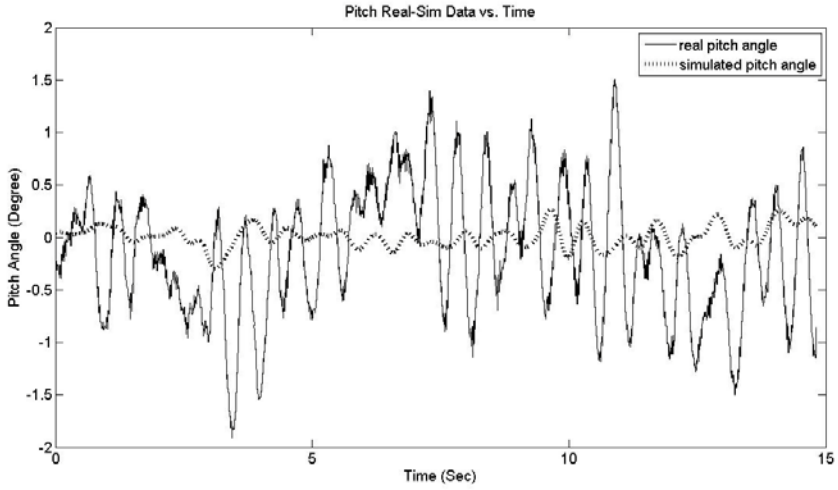


Figure 62: Comparison of Real and Simulated Pitch Angles

6.3.2 Tests for Yaw Dynamics

For yaw dynamics, first the system is flown without the yaw controller. Since the resultant moment about body z-axis is not zero, the system rotates about z-axis. The output of yaw dynamics without the controller is given in Figure 63.

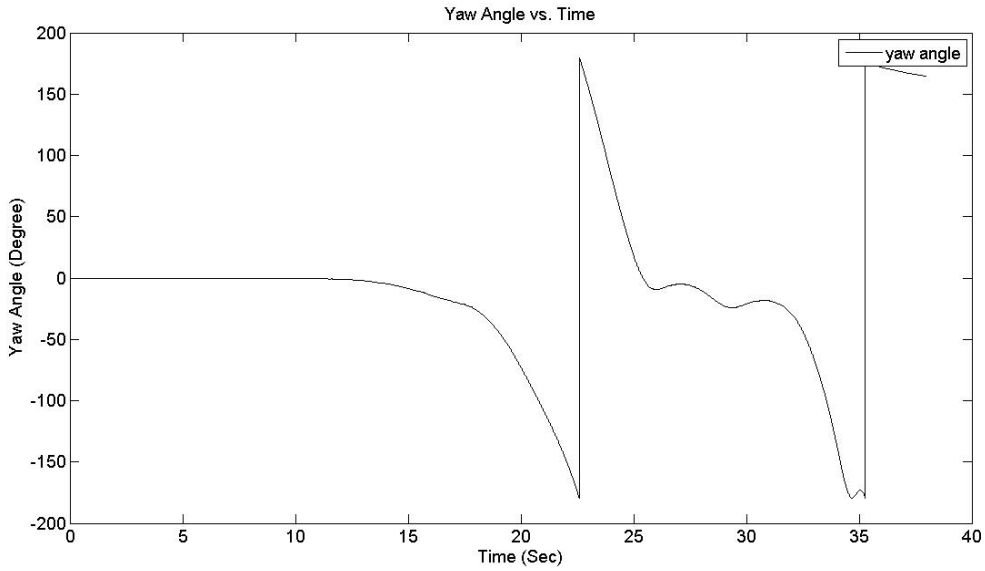


Figure 63: Yaw Dynamics without Controller

As can be seen in the Figure 63, after the system is lifting up, it starts to rotate counter-clock direction. In 22th second, yaw angle goes from -179° to +179°. This is caused by the IMU output while the system rotates counter-clockwise continuously. After the controller is designed (K_p=0.06, K_i=0.5, K_d=1.6), the output of the system is given in Figure 64.

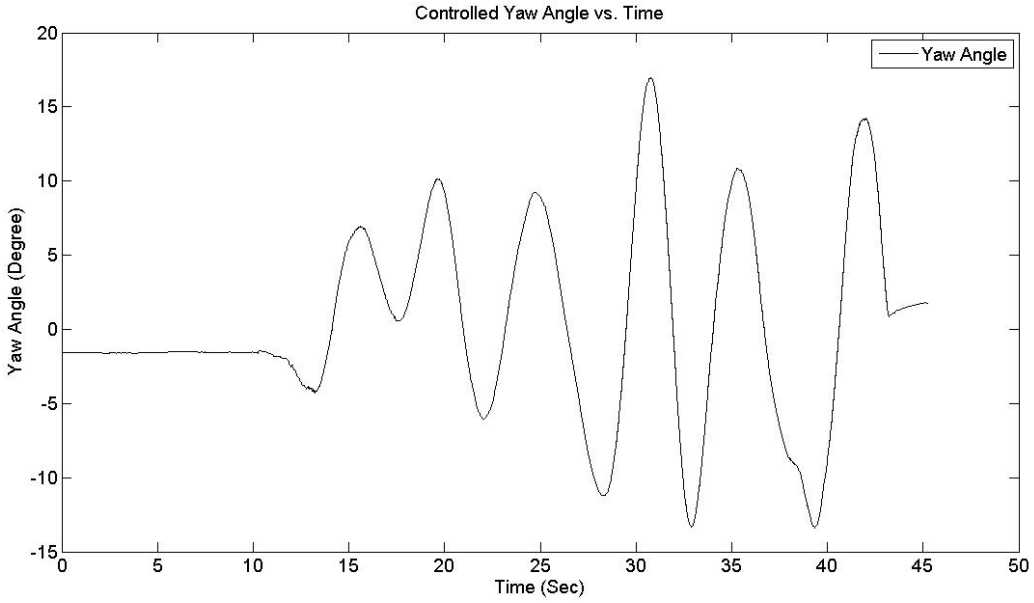


Figure 64: Yaw Dynamics with Controller

In another experiment, a smooth step input is given as a reference signal for yaw dynamics. The output of the system can be seen in Figure 65.

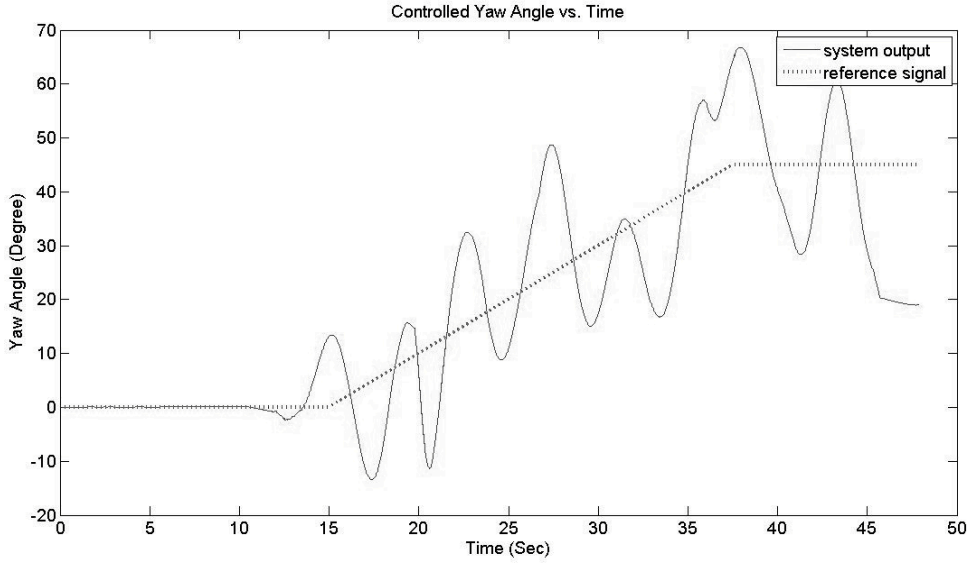


Figure 65: Yaw Dynamics – Reference Tracking

6.3.3 Tests for Altitude Dynamics

As stated before, mode 3 of the model increases the nominal thrust force. This mode increases the thrust force up to the system weight. Initially, the system height is 14 cm. After the nominal signal reaches its upper limit, altitude of the

system gets higher than 15 cm. Altitude controller is initially passive. After the altitude of the system is higher than 15 cm, the altitude controller is activated automatically. Then output of the altitude controller is added to the nominal signal. By this way, altitude controller is obtained. Another PID type controller is tuned as K_p=0.06, K_i=0.5, K_d=1.6. A screenshot of the altitude controller is given in Figure 66.

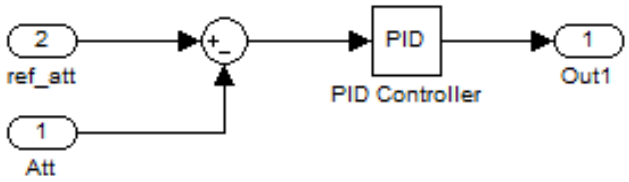


Figure 66: Altitude Controller

Outputs of the proximity sensor are converted to a distance in centimeters and are given in Figure 67.

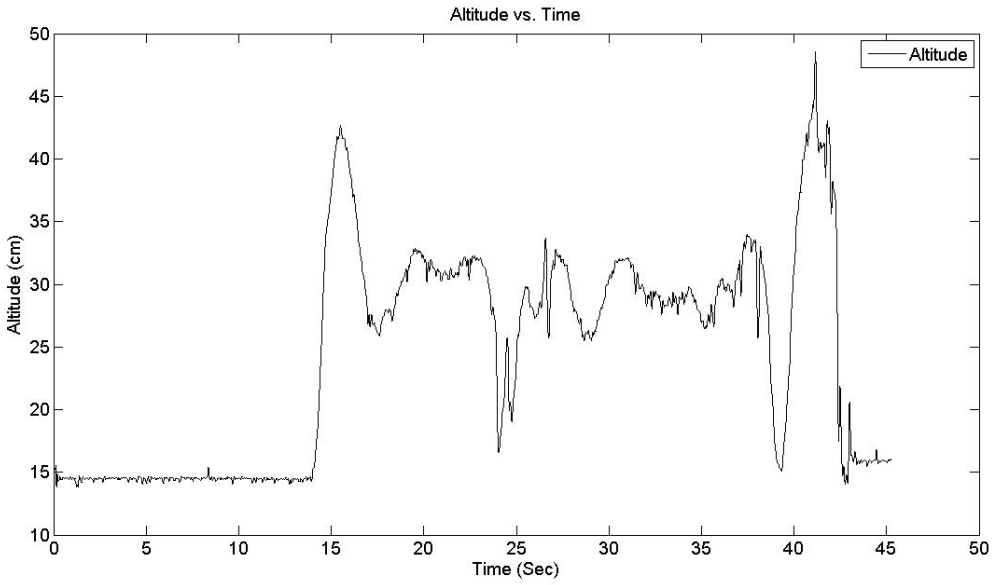


Figure 67: Altitude Dynamics Outputs

The reference altitude is given as 30 cm. when the time is approximately 25 seconds cables cross the line between the sensor and the ground. That is why, at that time altitude goes to the 20 cm. When the time is nearly 40 seconds, same situation occurs. If there were no cables going down from the system, the altitude controller will work better.

CHAPTER 7

DISCUSSION AND CONCLUSION

In this thesis, aim was to design controllers for attitude and altitude dynamics of a quadrotor. The system is modeled at first. Physical measurements are achieved to identify the actuator dynamics and inertial parameters. The system is composed of off-the-shelf and cheap brushless DC motors and speed controllers. After modeling the system, controllers are designed for the attitude and altitude dynamics. PID and LQR type controllers with integral action are designed for roll and pitch dynamics in the simulation. Simulation outputs are compared and it is seen that LQR type controller with integral action gives better result compared to the PID type controller. With the help of the integral action, steady state errors and disturbances are rejected. Then, both controllers are applied to the system on a spherical joint test setup. Results show that, LQR type controller with integral action gives better result compared to the PID type controller for roll and pitch dynamics. To control the yaw and altitude dynamics, two different PID type controllers are designed and applied to the system. All these model-based controllers are tuned on the real system and applied successfully.

In addition to the reported design and applications on the system, a 2 dof robot arm, is also placed on the quadrotor. It will be used to apply known disturbances to study disturbance rejection algorithms. It will be used as a steering system and stability augmenting system also. Robot arm is controlled manually by remote control in initial tests. However, it will be coupled to the control structure and be controlled for the steering and/or stabilizing purposes.

Cables give disturbances to the system when it is flying. These disturbances affect the altitude, attitude dynamics, and the movement in the x-y plane. Although this is not in the scope of the thesis, if the system is powered with batteries, and the controller software is embedded to the system, these disturbances may be reduced. With the fewer disturbances, the performance of the designed controllers could be better. A single board computer is ready to be placed on the quadrotor. Propellers can produce enough thrust to carry the total system with battery pack and control hardware.

To conclude, with using multi-loop control architectures, attitude and altitude controllers are designed based on the mathematical models and applied to the real system and the dynamics are controlled successfully. In addition nonlinear control architectures will be implemented on the system. Integral action gives robustness to the system. However, it may be necessary to improve the robustness for outdoor conditions. Adaptive and robust controllers together with disturbance rejection algorithms will be studied also.

REFERENCES

[1] Y. Amir, V. Abbass, 2008, "Modeling of Quadrotor Helicopter Dynamics", *International Conference on Smart Manufacturing Application*, pp. 100-105.

[2] J. W. R. Taylor, 1997, " Jane's Book of Remotely Piloted Vehicles", Collier Books, 1977.

[3] B. Erginer, E. Altug. 2007, "Modeling and PD Control of a Quadrotor VTOL Vehicle", *2007 IEEE Intelligent Vehicles Symposium*, pp. 894-899, Istanbul, Turkey.

[4] İ. C. Dikmen, A. Arısoy, H. Temeltaş., 2009, "Attitude Control of a Quadrotor", *Recent Advances in Space Technologies*, pp. 722-727, Istanbul, Turkey.

[5] G. M.Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin., 2011, "Precision Flight Control for a Multi-Vehicle Quadrotor Helicopter Testbed", *Control Engineering Practice*, pp. 1023-1036, Stanford, U.S.A.

[6] S. G. Vazquez, J. M. Valenzuela, 2010, "A New Nonlinear PD/PID controller for Quadrotor Posture Regulation", *2010 Electronics, Robotics and Automative Mechanics Conference*, pp. 642-647, Tijuana, Mexico.

[7] P. Pounds, R. Mahony, P. Corke, 2010, "Modeling and Control of A Large Quadrotor Robot", *Control Engineering Practice*, pp. 691-699, Australia.

[8] C. Nicol,, C.J.B. Macnab, A. Ramirez-Serrano, 2011, "Robust Adaptive Control of A Quadrotor Helicopter", *Control Engineering Practice*, pp. 927-938, Alberta, Canada.

[9] J. Zhang, W. Zhang, 2012, "LQR Self-adjusting Based Control for the Planar Double Inverted Pendulum", *2012 International Conference on Applied Physics and Industrial Engineering*, pp. 1669-1676, Henan Polytechnic University, China.

[10] L. Derafa, A. Benallegue, L. Fridman, 2011, "Super Twisting Control Algorithm for the Attitude Tracking of a Four Rotors", *Journal of The Franklin University, UAV*, pp. 685-699, Algeria.

[11] K. Alexis, C. Papachristos, G. Nikolakopoulos, A. Tzes, 2011, "Model Predictive Quadrotor Indoor Position Control", *19th Mediterranean Conference on Control and Automation*, Corfu, Greece.

[12] A. Zul Azfar, D. Hazry, 2011,"A Simple Approach on Implementing IMU Sensor Fusion in PID Controller for Stabilizing Quadrotor Flight Control", *7th International Colloquium on Signal Processing and its Applications*, 2011, pp. 28-32.

[13] P. Pounds, R. Mahony, 2009, "Design Principles of Large Quadrotors for Practical Applications", *IEEE International Conference on Robotics and Automation*, pp. 3265 – 3270.

[14] S. Bouabdallah, P. Murrieri, R. Siegwart, 2004, "Design an Control of an Indoor Micro Quadrotor", *IEEE International Conference Robotics and Automation*, pp. 4393 – 4398.

[15] S. Bouabdallah, R. Siegwart, 2007, "Full Control of a Quadrotor", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 153-158.

[16] N. Abas, A. Legowo, R. Akmeliawati, 2011, "Parameter Identification of an Autonomous Quadrotor", *4th International Conference on Mechatronics*, pp. 1-8.

[17] P. Bauer, G. Ritzinger, A. Soumelidis and J. Bokor, 2008, "LQ Servo control design with Kalman filter for a quadrotor UAV", *Periodica Polytechnica Transportation Engineering*, Vol. 33 1-2, pp. 9-14.

[18] L. Kis, G. Regula, B. Lantos, 2008, "Design and Hardware-in-the-Loop Test of the Embedded Control System of an Indoor Quadrotor Helicopter", *International Workshop on Intelligent Solutions in Embedded Systems*, pp. 1-10.

[19] J.F. Guerrero-Castellanos, N. Marchand, A. Hably, S. Lesecq, J. Delamare, 2011, "Bounded Attitude Control of Rigid Bodies: Real-Time Experimentation to A Quadrotor Mini-helicopter", *Control Engineering Practice*, pp. 790-797, France.

[20] B. Min, J. Hong, E. Matson, 2011, "Adaptive Robust Control (ARC) for an Altitude Control of a Quadrotor Type UAV Carrying an Unknown

Payloads", 11th International Conference on Control, Automation and Systems, pp. 1147 – 1151.

[21] S. Lee, S. Kang, "Trajectory Tracking Control of Quadrotor UAV", *11th International Conference on Control, Automation and Systems (ICCAS),* pp. 281 – 285.

[22] P. Sai Dinesh, K. Aditya, 2010, "Low Cost and Real Time Electronic Speed Controller of Position Sensorless Brushless DC Motor", *5th Internaional Conference on Information and Automation for Sustainability*, pp. 329 – 334.

[23] Doğanç , Küçlük. Design Of Two Wheeled Twin Rotored Hybrid Robotic Platform. MS thesis. Atılım University, 2010. Ankara: Print.

[24] Local Linearization of Nonlinear Systems, Retrieved from http://reference.wolfram.com/legacy/applications/control/NonlinearControlSystems/11.1.html

[25] A. Patel, M. Patel, and D Vyas., 2012, "Modeling and Analysis of Quadrotor using Sliding Mode Control", *44th IEEE Southeastern Symposium on System Theory*, University of North Florida, Jacksonville, 2012 .

[26] Ogata, Katsuhiko. *Modern Control Engineering*. 3rd. Edition, University of Minnesota, pp. 151 – 152.

[27] H. Wang, H. Dong, 2011, "Design and Simulation of LQR Controller with the Linear Inverted Pendulum", *International Conference on Electrical and Control Engineering (ICECE)*, pp.699-702.

[28] C. Chitu, H. Waser, 2011, "A Robust and Optimal LQR Controller Design for Electric Power Steering System", *Joint 3rd International Workshop on Nonlinear Dynamics and Synchronization (INDS)* , pp.1-5.

[29] Y.M. Al-Younes, M. A. Al-Jarrah, A.A. Jhemi, "Linear vs. Nonlinear Control Technique for a Quadrotor", *7th International Symposium on Mechatronics and its Applications,* American University of Sharjah, Sharjah, UAE, 2010.

[30] Pook, L.P. *Understanding Pendulum: A Brief Introduction*, 2011,Vol. 12, pp. 33

[31] "3DM-GX3 ® -25 Data Communications Protocol." . N.p., 2012. Web. 2 Jul 2012. http://files.microstrain.com/3DM-GX3-Data-Communications-Protocol.pdf

[32] "GP2Y0A02YK0F." Sharp. N.p., n.d. Web. 2 Jul 2012. http://www.sharpsma.com/webfm_send/1487.

[33] L. Prasad, B. Tyagi, H. Gupta, 2011, "Optimal Control of Nonlinear Inverted Pendulum Dynamical System with Disturbance Input using PID Controller & LQR", *2011 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp.540 – 545.

[34] J.li, Y.Li, 2011, "Dynamic Analysis and PID Control for a Quadrotor", Mechatronics and Automation (ICMA), pp. 573 – 578.

[35] A. Tayebi, S. McGilvray, 2006, "Attitude Stabilization of a Four-rotor Aerial Robot", *IEEE International Conference Robotics and Automation*, pp. 4393-4398.

APPENDICES

APPENDIX A

Matlab code to design controllers.

```
% x axis = roll axis
% y axis = pitch axis
% z axis = yaw axis
% phi    = roll
% theta  = pitch
% ksi    = yaw
% p      = roll ang rate
% q      = pitch ang rate
% r      = yaw ang rate
% Ix     = roll moment of inertia
% Iy     = pitch moment of inertia
% Iz     = yaw moment of inertia
% x      = state matrix
% v1     = motor 1 voltage
% v2     = motor 2 voltage
% v3     = motor 3 voltage
% v4     = motor 4 voltage
% F1     = motor 1 thrust
% F2     = motor 2 thrust
% F3     = motor 3 thrust
% F4     = motor 4 thrust
% L      = length between motor rotation axis
% b      = thrust factor (EPP1245 propeller)
% d      = drag coefficient  (EPP1245 propeller)

clc;
clear all;
Ts=1/300;
syms Mx My Mz Ixx Ixy Ixz Iyx Iyy Iyz Izx Izy Izz p q r phi theta
ksi dphi dtheta dksi dp dq dr duty1 duty2 duty3 duty4 v1 v2 v3 v4
F1 F2 F3 F4 L b d M1 M2 M3 M4 M d b gr1 gr2 gr3 gr4

I=[Ixx -Ixy -Ixz; -Iyx Iyy -Iyz; -Izx -Izy Izz];
w=[p; q; r];

gr1=189.6306*duty1-1014.3;
gr2=186.5251*duty2-1008.9;
gr3=198.0064*duty3-1076.8;
gr4=195.8662*duty4-1063.3;

F1=(gr1/1000)*9.81;
F2=(gr2/1000)*9.81;
F3=(gr3/1000)*9.81;
F4=(gr4/1000)*9.81;

M1=(d/b)*F1;
M2=(d/b)*F2;
M3=(d/b)*F3;
M4=(d/b)*F4;
```

```
Mx=(F3-F1)*L/2;
My=(F4-F2)*L/2;
Mz=(M2+M4)-(M1+M3);
M=[Mx; My; Mz];

dphi=[p+q*tan(theta)*sin(phi)+r*tan(theta)*cos(phi)];
dtheta=[q*cos(phi)-r*sin(phi)];
dksi=[q*sec(theta)*sin(phi)+r*sec(theta)*cos(phi)];

dw=inv(I)*(-(cross(w,I*w))+M);

dp=dw(1);
dq=dw(2);
dr=dw(3);

x=[phi; theta; ksi; p; q; r];
u=[duty1; duty2; duty3; duty4];
dx=[dphi; dtheta; dksi; dp; dq; dr];
dxnn=subs(dx);

As=jacobian(dxnn,x);
Bs=jacobian(dxnn,u);

%%
%Mathematical Calculations

gr1=300;
gr2=300;
gr3=300;
gr4=300;

duty1=0.0052*gr1+5.3613;
duty2=0.0053*gr2+5.4226;
duty3=0.0050*gr3+5.4609;
duty4=0.0051*gr4+5.4416;

b=1.86*10^-5;
d=3.8*10^-7;

phi=0*pi/180;
theta=0*pi/180;
ksi=60*pi/180;
p=0*pi/180;
q=0*pi/180;
r=0*pi/180;

L=0.19;

Ixx=0.032676779;  %kgm2
Ixy=0.010291232;  %kgm2
Ixz=0.000008295;  %kgm2
Iyx=Ixy;          %kgm2
Iyy=0.032687945;  %kgm2
Iyz=-0.000000489; %kgm2
Izx=Ixz;          %kgm2
Izy=Iyz;          %kgm2
```

```
Izz=0.064609760;  %kgm2

% Inertia Matrix
%  Ixx -Ixy -Ixz
% -Iyx  Iyy -Iyz
% -Izx -Izy  Izz

As=subs(As);
A=double(As)

Bs=subs(Bs);
B=double(Bs)

% C=eye(6);
% D=[zeros(6,4)];

C=[1 zeros(1,5);0 1 zeros(1,4)]% 0 0 0 1 0 0; 0 0 0 0 1 0];
D=zeros(2,4)

SYS=ss(A,B,C,D)
P=pole(SYS)

Co=ctrb(SYS);
Ob=obsv(SYS);

disp('Controllability and Observability Matrix RANKS')
Controllability=rank(Co)
Observability=rank(Ob)

duty_nom=[duty1; duty2; duty3; duty4]

Ab=[A zeros(6,2); -C zeros(2,2)]
Bb=[B;zeros(2,4)]
Sb=ss(Ab,Bb,eye(8,8),zeros(8,4))

Q=eye(8);
Q(1,1)=290;
Q(2,2)=290;
Q(3,3)=0.001;
Q(4,4)=1;
Q(5,5)=1;
Q(6,6)=0.005;
Q(7,7)=70;
Q(8,8)=70;

R=eye(4);
R(1,1)=22;
R(2,2)=22;
R(3,3)=22;
R(4,4)=22;
[K,S,E] = lqr(Sb,Q,R)
K1=K(:,1:6)
K2=K(:,7:8)
Q
R
E
```

APPENDIX B

Matlab Code to get real time data from target PC and plot sensor and controller outputs.

```
%% DATA
log=tg.OutputLog;
log(1:10,:)=0;
time=tg.timelog;
sample=size(tg.timelog);
sample=sample(1,1);
accx=log(1:sample,1);
accy=log(1:sample,2);
accz=log(1:sample,3);
yaw=log(1:sample,4);
pitch=log(1:sample,5);
roll=log(1:sample,6);
arx=log(1:sample,7);
ary=log(1:sample,8);
arz=log(1:sample,9);
m1=log(1:sample,10);
m2=log(1:sample,11);
m3=log(1:sample,12);
m4=log(1:sample,13);
v1fark=log(1:sample,14);
v2fark=log(1:sample,15);
v3fark=log(1:sample,16);
v4fark=log(1:sample,17);
gr=log(1:sample,18);
mode=log(1:sample,19);
cm=log(1:sample,20);
yaw_controller_1_3=log(1:sample,21);
yaw_controller_2_4=log(1:sample,22);
cm_ref=log(1:sample,23);
yaw_ref=log(1:sample,24);

accx(1:3,1)=0;
accy(1:3,1)=0;
accz(1:3,1)=0;
yaw(1:3,1)=0;
pitch(1:3,1)=0;
roll(1:3,1)=0;
arx(1:3,1)=0;
ary(1:3,1)=0;
arz(1:3,1)=0;
m1(1:3,1)=0;
m2(1:3,1)=0;
m3(1:3,1)=0;
m4(1:3,1)=0;
v1fark(1:3,1)=0;
v2fark(1:3,1)=0;
v3fark(1:3,1)=0;
v4fark(1:3,1)=0;
gr(1:3,1)=0;
mode(1:3,1)=0;
ref(1:3,1)=0;
```

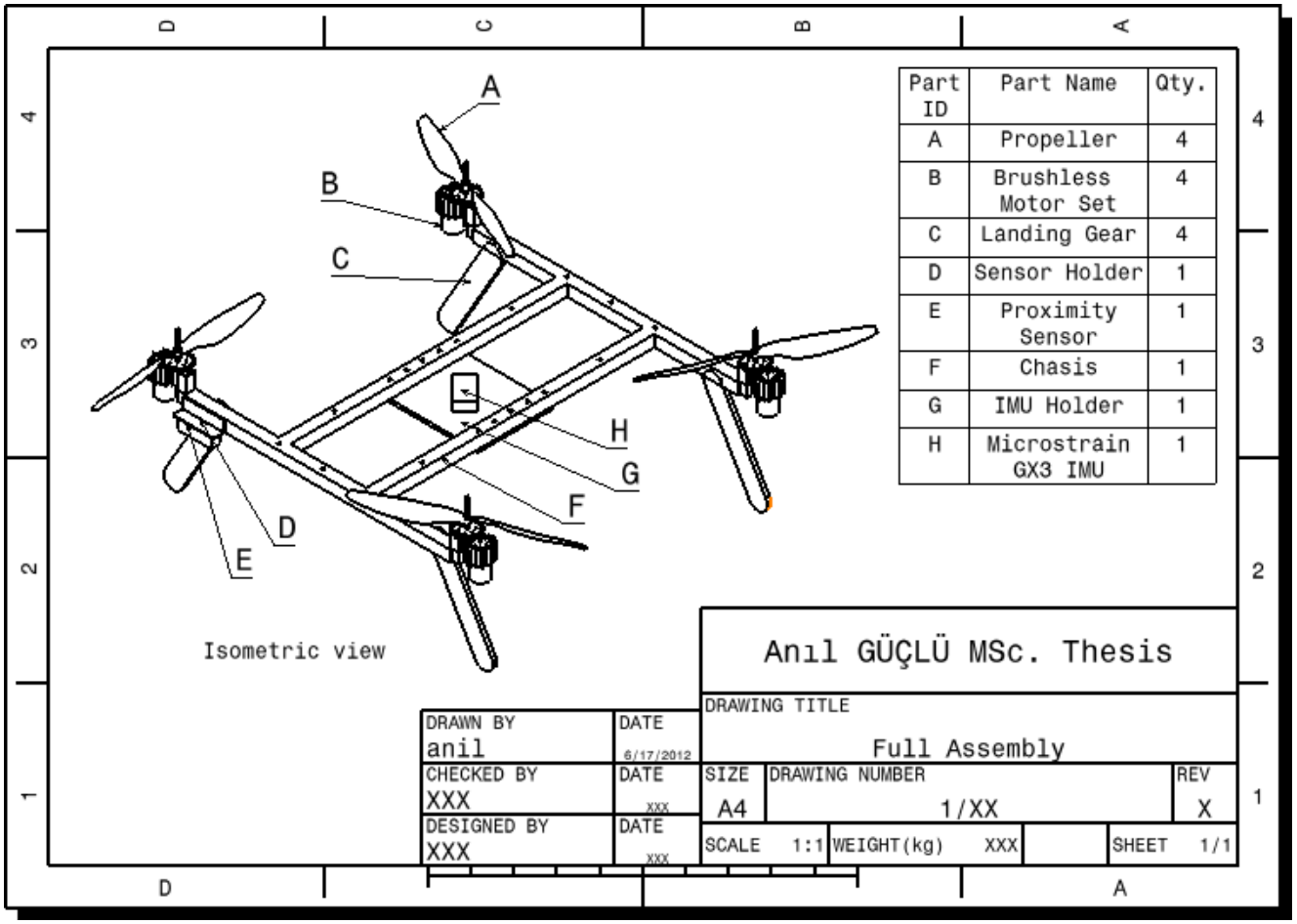
```
%% mode

figure('NumberTitle','off','Name','Mode vs Time')
plot(time, mode, 'DisplayName', 'mode', 'XDataSource', 'time',
'YDataSource', 'mode'); hold off; figure(gcf)
legend('2:Stop 3:Nominal 4:Controller');
ylabel('Mode');
title('Mode vs Time');
xlabel('Time (sec)');


%% gr

figure('NumberTitle','off','Name','gr vs Time')
plot(time, gr, 'DisplayName', 'gr', 'XDataSource', 'time',
'YDataSource', 'gr'); hold off; figure(gcf)
legend('gram')
ylabel('Gram');
title('Gr vs Time');
xlabel('Time (sec)');


%% Acceleration Plot

figure('NumberTitle','off','Name','Acc. vs Time')
plot(time, accx, 'DisplayName', 'accx', 'XDataSource', 'time',
'YDataSource', 'accx'); hold all; plot(time, accy, 'DisplayName',
'accy', 'XDataSource', 'time', 'YDataSource', 'accy'); plot(time,
accz, 'DisplayName', 'accz', 'XDataSource', 'time', 'YDataSource',
'accz');  hold off; figure(gcf)
legend('accx','accy','accz')
ylabel('Acceleration (m/s2)');
title('Acceleration vs Time');
xlabel('Time (sec)');

%% Euler Plot

figure('NumberTitle','off','Name','Euler vs Time')
plot(time, yaw, 'DisplayName', 'yaw', 'XDataSource', 'time',
'YDataSource', 'yaw'); hold all; plot(time, pitch, 'DisplayName',
'pitch', 'XDataSource', 'time', 'YDataSource', 'pitch');
plot(time, roll, 'DisplayName', 'roll', 'XDataSource', 'time',
'YDataSource', 'roll');  hold off; figure(gcf)
legend('yaw','pitch','roll')
ylabel('Position (Degree)');
title('Position vs Time');
xlabel('Time (sec)');

%% Ang. Rate Plot

figure('NumberTitle','off','Name','Ang. Rate vs Time')
plot(time, arx, 'DisplayName', 'arx', 'XDataSource', 'time',
'YDataSource', 'arx'); hold all; plot(time, ary, 'DisplayName',
'ary', 'XDataSource', 'time', 'YDataSource', 'ary'); plot(time,
arz, 'DisplayName', 'arz', 'XDataSource', 'time', 'YDataSource',
'arz');  hold off; figure(gcf)
legend('AngRateX','AngRateY','AngRateZ')
```

```
ylabel('Rotational Velocity (Degree/sec)');
title('Rotational Velocity vs Time');
xlabel('Time (sec)');


%% Motor Duty Cycle

figure('NumberTitle','off','Name','Duty vs Time')
plot(time, m1, 'DisplayName', 'm1', 'XDataSource', 'time',
'YDataSource', 'm1'); hold all; plot(time, m2, 'DisplayName',
'm2', 'XDataSource', 'time', 'YDataSource', 'm2'); plot(time, m3,
'DisplayName', 'm3', 'XDataSource', 'time', 'YDataSource', 'm3');
plot(time, m4, 'DisplayName', 'm4', 'XDataSource', 'time',
'YDataSource', 'm4'); hold off; figure(gcf)
legend('m1','m2','m3','m4')
ylabel('Duty Cycle (%)');
title('Duty Cycle vs Time');
xlabel('Time (sec)');

%% Fark

figure('NumberTitle','off','Name','Fark')
plot(time, v1fark, 'DisplayName', 'v1fark', 'XDataSource', 'time',
'YDataSource', 'v1fark'); hold all; plot(time, v2fark,
'DisplayName', 'v2fark', 'XDataSource', 'time', 'YDataSource',
'v2fark'); plot(time, v3fark, 'DisplayName', 'v3fark',
'XDataSource', 'time', 'YDataSource', 'v3fark'); plot(time,
v4fark, 'DisplayName', 'v4fark', 'XDataSource', 'time',
'YDataSource', 'v4fark'); hold off; figure(gcf)
legend('v1fark','v2fark','v3fark','v4fark')
ylabel('Vfark');
title('Vfark vs Time');
xlabel('Time (sec)');

%% LQR -Q

figure('NumberTitle','off','Name','Q Matrix')
QQ=[Q(1,1) Q(2,2) Q(3,3) Q(4,4) Q(5,5) Q(6,6) Q(7,7) Q(8,8)];
LLQ=[1 2 3 4 5 6 7 8];
stem(LLQ, QQ, 'DisplayName', 'Q parameters', 'XDataSource', 'L',
'YDataSource', 'K'); figure(gcf)
QQstr1=num2str(QQ(1));
QQstr2=num2str(QQ(2));
QQstr3=num2str(QQ(3));
QQstr4=num2str(QQ(4));
QQstr5=num2str(QQ(5));
QQstr6=num2str(QQ(6));
QQstr7=num2str(QQ(7));
QQstr8=num2str(QQ(8));
text(LLQ(1)+0.1,QQ(1)+0.1,['Q(1,1)']);
text(LLQ(2)+0.1,QQ(2)+0.1,['Q(2,2)']);
text(LLQ(3)+0.1,QQ(3)+0.1,['Q(3,3)']);
text(LLQ(4)+0.1,QQ(4)+0.1,['Q(4,4)']);
text(LLQ(5)+0.1,QQ(5)+0.1,['Q(5,5)']);
text(LLQ(6)+0.1,QQ(6)+0.1,['Q(6,6)']);
text(LLQ(7)+0.1,QQ(7)+0.1,['Q(7,7)']);
text(LLQ(8)+0.1,QQ(8)+0.1,['Q(8,8)']);
```

```
title('Q Matrix');
xlabel('Q Matrix Index');
ylabel('Q Matrix Gain');

%% LQR -R

figure('NumberTitle','off','Name','R Matrix')
RR=[R(1,1) R(2,2) R(3,3) R(4,4)];
LLR=[1 2 3 4];
stem(LLR, RR, 'DisplayName', 'R parameters', 'XDataSource', 'L',
'YDataSource', 'K'); figure(gcf)

RRstr5=num2str(RR(1));
RRstr6=num2str(RR(2));
RRstr7=num2str(RR(3));
RRstr8=num2str(RR(4));
text(LLR(1)+0.1,RR(1)+0.1,['R(1,1)']);
text(LLR(2)+0.1,RR(2)+0.1,['R(2,2)']);
text(LLR(3)+0.1,RR(3)+0.1,['R(3,3)']);
text(LLR(4)+0.1,RR(4)+0.1,['R(4,4)']);

title('R Matrix');
xlabel('R Matrix Index');
ylabel('R Matrix Gain');

figure
subplot(3,1,1), plot(time, yaw, 'DisplayName', 'yaw',
'XDataSource', 'time', 'YDataSource', 'yaw'); hold all; plot(time,
pitch, 'DisplayName', 'pitch', 'XDataSource', 'time',
'YDataSource', 'pitch'); plot(time, roll, 'DisplayName', 'roll',
'XDataSource', 'time', 'YDataSource', 'roll');  hold off;
figure(gcf)
legend('yaw','pitch','roll')
ylabel('Position (Degree)');
title('Position vs Time');
xlabel('Time (sec)');
subplot(3,1,2), plot(time, v1fark, 'DisplayName', 'v1fark',
'XDataSource', 'time', 'YDataSource', 'v1fark'); hold all;
plot(time, v2fark, 'DisplayName', 'v2fark', 'XDataSource', 'time',
'YDataSource', 'v2fark'); plot(time, v3fark, 'DisplayName',
'v3fark', 'XDataSource', 'time', 'YDataSource', 'v3fark');
plot(time, v4fark, 'DisplayName', 'v4fark', 'XDataSource', 'time',
'YDataSource', 'v4fark'); hold off; figure(gcf)
legend('v1fark','v2fark','v3fark','v4fark')
ylabel('Vfark');
title('Vfark vs Time');
xlabel('Time (sec)');
subplot(3,1,3), plot(time, m1, 'DisplayName', 'm1', 'XDataSource',
'time', 'YDataSource', 'm1'); hold all; plot(time, m2,
'DisplayName', 'm2', 'XDataSource', 'time', 'YDataSource', 'm2');
plot(time, m3, 'DisplayName', 'm3', 'XDataSource', 'time',
'YDataSource', 'm3'); plot(time, m4, 'DisplayName', 'm4',
'XDataSource', 'time', 'YDataSource', 'm4'); hold off; figure(gcf)
legend('m1','m2','m3','m4')
ylabel('Duty Cycle (%)');
title('Duty Cycle vs Time');
xlabel('Time (sec)');
```

```
%% cm

figure('NumberTitle','off','Name','Altitute vs Time')
plot(time, cm, 'DisplayName', 'cm', 'XDataSource', 'time',
'YDataSource', 'cm');
hold all
plot(time, cm_ref, 'DisplayName', 'cm_ref', 'XDataSource', 'time',
'YDataSource', 'cm_ref');
legend('Altitude','Reference')
ylabel('Altitude (cm)');
title('Altitude vs Time');
xlabel('Time (sec)');

%% Yaw Controller

figure('NumberTitle','off','Name','Yaw_Controller vs Time')
plot(time, yaw_controller_1_3, 'DisplayName', '1_3',
'XDataSource', 'time', 'YDataSource', 'controller');
hold all
plot(time, yaw_controller_2_4, 'DisplayName', '2_4',
'XDataSource', 'time', 'YDataSource', 'controller'); figure(gcf)
legend('1_3','2_4')
ylabel('Controller Signal');
title('Yaw Controller vs Time');
xlabel('Time (sec)');

%% Yaw angle

figure('NumberTitle','off','Name','Yaw vs Time')
plot(time, yaw, 'DisplayName', 'yaw_actual', 'XDataSource',
'time', 'YDataSource', 'controller');
hold all
plot(time, yaw_ref, 'DisplayName', 'yaw_ref', 'XDataSource',
'time', 'YDataSource', 'controller'); figure(gcf)
legend('yaw_act','yaw_ref')
ylabel('Controller Signal');
title('Yaw Controller vs Time');
xlabel('Time (sec)');
```

APPENDIX C



Part ID	Part Name	Qty.
A	Propeller	4
B	Brushless Motor Set	4
C	Landing Gear	4
D	Sensor Holder	1
E	Proximity Sensor	1
F	Chasis	1
G	IMU Holder	1
H	Microstrain GX3 IMU	1

Anıl GÜÇLÜ MSc. Thesis

DRAWN BY: anil	DATE: 6/17/2012	DRAWING TITLE: Full Assembly	
CHECKED BY: XXX	DATE: XXX	SIZE: A4	DRAWING NUMBER: 1/XX — REV: X
DESIGNED BY: XXX	DATE: XXX	SCALE: 1:1	WEIGHT(kg): XXX — SHEET: 1/1