IMPROVEMENT OF A RC QUADROTOR PLATFORM

TO A FLYING ROBOT FOR TARGET TRACKING

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

FIRAT TANSU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF MECHATRONICS ENGINEERING

JULY 2013

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

_____

Prof. Dr. K. 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 "Thesis Name" submitted by "Candidates Name" 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

Asst. Prof. Dr. Zühal Erden	_____
Asst. Prof. Dr. Ali Emre Turgut	_____
Asst. Prof. Dr. Kutluk Bilge Arıkan	_____
Asst. Prof. Dr. Bülent İrfanoğlu	_____
Instr. H. Orhan Yıldıran	_____

Date: (19.07.2013)

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: Fırat Tansu

Signature:

ABSTRACT

IMPROVEMENT OF A RC QUADROTOR PLATFORM

TO A FLYING ROBOT FOR TARGET TRACKING

Tansu, Fırat

M.S., Mechatronics Engineering Department

Supervisor: Asst. Prof. Dr. Kutluk Bilge Arıkan

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

July 2013, 54 pages

In this study, transformation of Draganfly Vi Ti Pro, a remote controlled (RC) Quadrotor platform, into a target tracking flying robot is aimed. This thesis examines controller design for the attitude and altitude dynamics and target tracking. The major aim of this work is to design a normalized linear quadratic regulator (LQR) for attitude and altitude dynamics and achieve target tracking by PID controllers on a moving object. Before designing the controller, a mathematical model has been developed using the Euler-Lagrange method. Simulations and real-time applications have been performed on MATLAB/Simulink environment. For target detection algorithm, *ArUco,* which is a C++ code for basic augmented reality applications, has been used. Designed controllers are implemented on the physical system and satisfactory results are achieved. Comparison of simulation with the real system has been done and discussed.

Keywords: Quadrotor, Linear Quadratic Regulator, Draganfly, PID, ArUco, Target Tracking

ÖZ

UZAKTAN KUMANDALI DÖRT ROTORLU BİR PLATFORMUN HEDEF TAKİP EDEBİLEN UÇAN BİR ROBOTA DÖNÜŞTÜRÜLMESİ

Tansu, Fırat

Yüksek Lisans, Mekatronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Kutluk Bilge Arıkan

Ortak Tez Yöneticisi: Yrd. Doç. Dr. Bülent İrfanoğlu

Temmuz 2013, 54 sayfa

Bu çalışmada uzaktan kumandalı Draganfly Vi Ti Pro model dört rotorlu platformun, hedef takip edebilen bir robota dönüştürülmesi amaçlanmaktadır. Dört rotorlu insansız hava aracının (İHA), yönelim ve yükseklik dinamikleri ve hedef takibi için denetleyici tasarımı konularını incelemektedir. Bu araştırmanın temel amacı yönelim ve yükseklik dinamikleri için normalize edilmiş bir doğrusal kuadratik regülatör (DKR) tasarlamak ve hedef takibini PID deneticisi ile sağlamaktır. Kontrolcü tasarımına başlamadan önce, matematiksel model Euler-Lagrange yöntemi kullanılarak elde edilmiştir. Simülasyonlar ve gerçek zamanlı uygulamalar MATLAB / Simulink ortamında gerçekleştirilmiştir. Hedef tespit algoritmasında, artırılmış gerçeklik uygulamaları için geliştirilmiş C++ kodu olan, ArUco, kullanılmıştır. Tasarlanan denetimciler fiziksel sistem üzerinde uygulanmış ve tatmin edici sonuçlar elde edilmiştir. Gerçek sistem ve benzetim karşılaştırılması yapılmış ve sonuçları tartışılmıştır.

Anahtar Kelimeler: Quadrotor, Doğrusal Kuadratik Regülatör, PID, Draganfly, Aruco, Hedef Belirleme

To My Family and Friends

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Asst. Prof. Dr. Kutluk Bilge Arıkan for his guidance and insight throughout the research. Thanks also go to my co-supervisor Asst. Prof. Dr. Bülent İrfanoğlu for his guidance. The technical and mental assistance of Alp Kaçar, Ayça Göçmen, Cahit Gürel, Emre Kara, Hassan Golmohammadzadeh, Emre Güner, Meral Aday, Handan Kara and Azizhan Tekin are gratefully acknowledged. I would like to offer sincere thanks for my parents, Haluk and Nil Tansu, and my girlfriend, Aybüke Kendirligil, for their continuous support and patience during this period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AR – Augmented Reality

COG – Center of Gravity

DAQ – Data Acquistion Card

DC – Direct Current

FST- Frenet-Serret Theory

IMU – Inertial Measurement Unit

IR – Infra-Red

LQ – Linear Quadratic

LQR – Linear Quadratic Regulator

MECE- Mechatronics Engineering

MBPC - Model Based Predictive Control

PID – Proportional Derivative Integral

RC – Remote Control

RTWT –Real Time Windows Target

UAV – Unmanned Air Vehicle

UDP- User Datagram Protocol

NOMENCLATURE

ϕ - Roll Angular Position (rad)

$\dot{\phi}$ - Roll Angular Velocity (rad/s)

$\ddot{\phi}$ - Roll Angular Acceleration (rad/s^2)

θ - Pitch Angular Position (rad)

$\dot{\theta}$ - Pitch Angular Velocity (rad/s)

$\ddot{\theta}$ - Pitch Angular Acceleration (rad/s^2)

ψ - Yaw Angular Position (rad)

$\dot{\psi}$ - Yaw Angular Velocity (rad/s)

$\ddot{\psi}$ - Yaw Angular Acceleration (rad/s^2)

x - Linear Position on x axis (m)

$\dot{x}$ - Linear Velocity on x axis (m/s)

$\ddot{x}$ - Linear Acceleration on x axis (m/s^2)

y - Linear Position on y axis (m)

$\dot{y}$ - Linear Velocity on y axis (m/s)

$\ddot{y}$ - Linear Acceleration on y *a*xis (m/s^2)

z - Altitude along z axis (m)

$\dot{z}$ - Altitude velocity along z axis (m/s)

$\ddot{z}$ - Altitude acceleration along z axis (m/s^2)

J – Inertia Values

T – Period

m – Mass of the System (kg)

g – Gravitational Force (m/s^2)

L – Length of the Ropes (m)

R- Radius to COG (m)

ξ – Position Vector

I – Inertia Matrix

$\tilde{\tau}_{\phi}$ – Rolling Moment along x axis (Nm)

$\tilde{\tau}_{\theta}$ – Rolling Moment along y axis (Nm)

$\tilde{\tau}_{\psi}$ – Rolling Moment along z axis (Nm)

Ω - Angular Velocity of Propellers (rad/s)

$V_{1,2,3,4}$ - Voltage Inputs (V)

Ex - Error in x displacement (m)

$E\dot{x}$ - Change in x error (m/s)

Ey -Error in y displacement (m)

$E\dot{y}$ - Change in y error (m/s)

CHAPTER 1

INTRODUCTION

In recent years, there is an increasing amount of investigation and research done on UAV's (Unmanned Aerial Vehicle). These studies have proved that, in the modern world UAV's are becoming more useful in research platform, military, law enforcement and commercial use. Quadrotors are one of the most popular UAVs in recent years. Quadrotor which can be also called quadrotor helicopter, quadcopter or quadrocopter is a type of copter that has four rotors. These copters are classified as rotorcrafts which uses lift generated by wings called rotor blades opposed to fixed-wing aircrafts. These vehicles generally use symmetrically placed wings to generate torque load and thrust which can be controlled by altering pitch and rotation rate of one or more rotor discs. The Figure 1 illustrates the main structure of a simple quadrotor, together with the free body diagram itself.

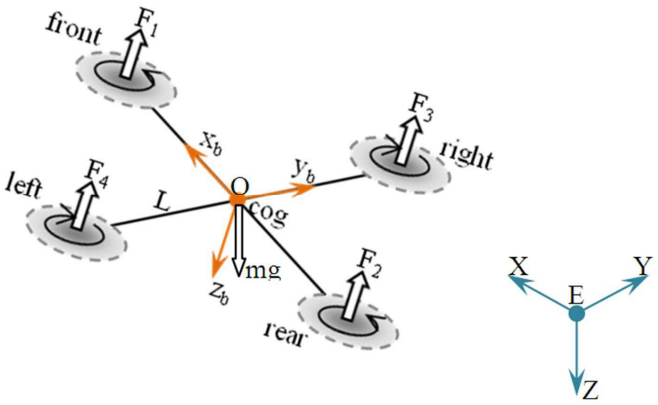

Figure 1: Body and Earth fixed reference frames

There are several advantages using quadrotors compared to other copters;

- Quadrotors does not require any mechanical linkage to change the rotor's pitch angle compared to helicopters
- Usage of four rotors allows possessing less kinetic energy during the entire flight.
- Construction and control processes are easier than other helicopters.
- Minimized structures help the user to get closer approach to specified location or target
- As compared to other copters entire mechanical system gives more agility because of the four individual rotors (in case of wing crash)

In this thesis, Draganflyer's VI Ti Pro model[1] has been re-assembled and LQR (Linear Quadratic Regulator) controller approach has been applied to the system to control the attitude and altitude dynamics. Proportional-Integral-Derivative (PID) type controller is employed to perform target tracking.

The rest of the thesis is organized as follows: In chapter two, researches done about quadrotors are explained. In this chapter, previous works related with quadrotors will be explained and investigated. Usage area, main system structures, sensor components and control algorithms are categorized.

In chapter three, physical system is introduced. In this part, hardware and software components are explained. Chapter three also discusses the construction of the main structure.

Chapter four introduces the mathematical model of the system. Lagrange equations, system matrixes and dynamic equations are shown in detail.

Chapter five presents the controller design. In this chapter the control algorithm of the project is explained. Tuning process and simulations done on the system are also discussed. Encounters that are faced in the design process are explained and the desired scenario is shown.

Chapter six examines the results of the real time experiments done on the system. This section provides detailed information about controller outputs applied on the actual system.

Finally, discussion, conclusions and future works are given in Chapter 7.

1.1 Aim and Scope of the Thesis

It is important to express that aim of the thesis is to improve the RC platform to a flying robot by controlling the attitude and altitude dynamics of the quadrotor while tracking the target on the ground. Prior approach at start on the quadrotor system was maintaining the altitude control by a transmitter unit. The model has been specified as Draganflyer VI Ti Pro's quadrotor structure. At this step, the aim was to generate the required voltage value that is going to be applied to the Transmitter unit with the help of National Instrument's DAQ (Data Acquisitions) card and terminal. Experiments and tests show that, usage of this system will not be sufficient on the control system; so that, Transmitter unit has been cancelled and original controller board has been disassembled. The basic aims of the thesis have stated below:

-Obtaining a new test bench using the Draganflyer's chassis

Main structure has been saved as it is in the original state. Battery and controller board have been removed but the motors and propellers have been kept. In order to control the motors, selected motor drivers have been assembled on a plate alongside with IMU unit and the plate has been inserted at the point where the controller board is placed.

-Mathematical modeling, calculation of the physical parameters and controller design

Attitude and altitude dynamics are modeled and controlled at first. Controller designs were performed over linearized model. LQR controller has been applied in order to stabilize dynamics with specified reference points. While performing these applications Matlab alongside with data acquisition board for Real Time Windows Target has been used. LQR based control architecture is extended to track the target on ground, Figure 2. PID controllers are utilized for target tracking also. Propeller parameters have been acquired from the literature survey and motor dynamic equation has been obtained from previous studies done in the Flying Robotics Laboratory. Moment of inertia values have been obtained experimentally after assembling the entire system.

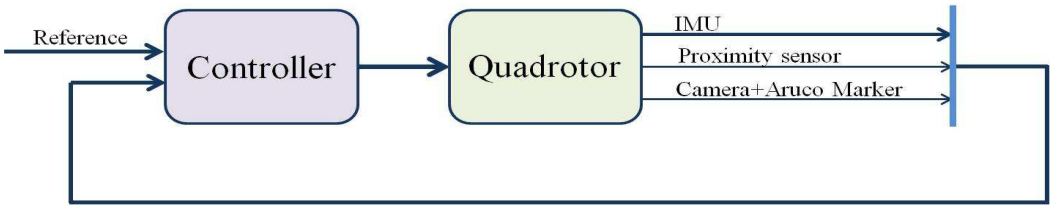

Figure 2: Theory of Operation

- *Physical implementation*

It is desired to detect and track a target on the ground. The target is called *Peri* which is a four wheeled unmanned ground vehicle produced by Aselsan Company. In this application, xPc Target application of Matlab Software has been used. The entire system has been re-programmed with Simulink package of Matlab to drive the system autonomously in a given reference signal. After achieving a target which can move on a desired path, detection of the target process has been started. A C Code which generates augmented reality called Aruco has been modified for the current system. The main feature of Aruco is to detect simple markers on the ground by a camera and send the relative position of the marker according the location of the camera origin. Embedded control is out of the scope of the thesis. However, initial studies to deploy the entire Simulink model to an embedded controller unit are started. Texas Instrument's C2000 card has been chosen for this application. Again, using Embedded Target of Matlab software, created models have been deployed to this card partially and connection between Matlab and C2000 has been established for fast compiling simulation trials.

CHAPTER 2

LITERATURE SURVEY

Quadrotors are becoming more and more useful every year. It has become a very powerful research and test tool for universities in many countries in the world. With this, new ideas that will result in the new concept and controller designs by selecting different types of sensors, equipment and tools have been formed. In earlier studies, Etienne Oehmichen studied his designs in the 1920s.[2] He studied six different designs. Among all of them, his second design had four rotors and eight propellers, and all were driven by a single engine. This design used a steel-tube frame, with two-bladed rotors at the ends of the four arms. It was possible to vary the angle of blades by warping. Five of the propellers, spinning in the horizontal plane, stabilized the machine laterally. Another propeller was mounted at the nose for steering. The remaining pair of propellers was for forward propulsion. The aircraft exhibited a considerable degree of stability and controllability for its time, and made more than a thousand test flights [2]. Nevertheless such designs now have four rotors mounted on each corner and driven by four specific motors. In many practical applications on quadrotors, altitude and attitude control strategies were driven by a RC controller. These controllers help the user to achieve any moment action on *x, y,* or *z* axis. However, for autonomous quadrotors, there are different control strategies. These strategies have been studied in different platforms currently. The most adapted approach might be to decouple the system dynamics by using the Non-Linear Inverse Dynamics method [3], [4]. However, this approach can only be used if and only if the parameters of the plant model and all external disturbances are known precisely. Apart from this, there are also such controllers like Proportional-Integral-Derivate (PID), Linear Quadratic (LQ), H2, and $H\infty$ methods. According to [5] PID and LQ controller have been compared and the result was an autonomous hover flight. After

these processes backstepping and sliding-mode techniques have been tested. These tests have shown good disturbance rejection but lack of stabilization at hovering. Tests benches that used in these experiments have been show in Figure 3.

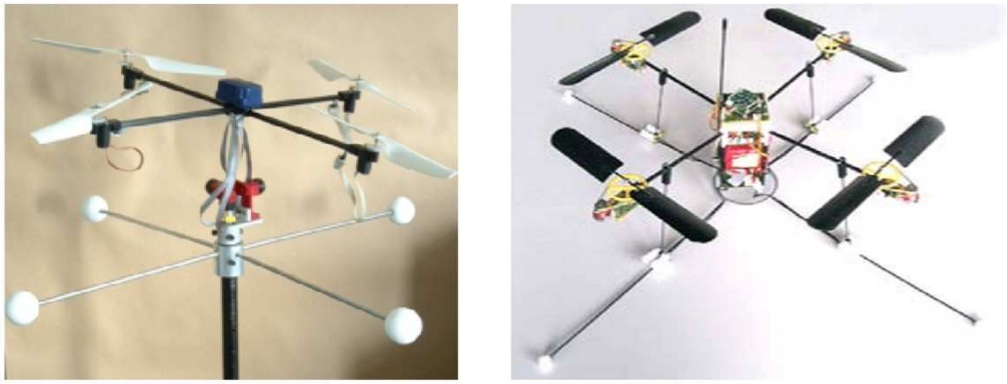

Figure 3 – Two Test Benches for Controller Design [4]

According to [6], the normalized form has a number of advantages compared with the standard form. It is stated that components of the system states are dimensionless and restricted to the standard value. This enables better understanding of the problem. In this work [6] system is normalized for input matrix; as an advantage it is stated that all physical parameters are removed from the matrix during the normalization process. This concluded with simplified calculations. Another approach according to [7] was backstepping methodology. According to the results, high speed maneuvering the backstepping performance (in relation to error) tracking is about 2.5x times better than using PID technique. Also it is stated that "The PID delays to reject the disturbance whereas the Backstepping+FST immediately compensates angular position based on velocity change rate, which consequently improves on the tracking error (in X-Y position). This improvement was basically achieved by introducing a desired angular acceleration command (as a function of the maneuvering velocity) that quickly responds to abrupt angular rate change, making the attitude stabilization more reliable" [7]. Another technique - the so-called LQR - used by [8], where the controller was succeeded but control performance has been observed to be degrading at higher thrust levels due to vibrations. As a solution to this issue, cost functions on the attitude deviations have been lowered however, this results in decrease in tracking performance. In [9], $H\infty$ method has been studied. In this study, it is possible to observe robust, better

reference tracking and disturbance rejection on nonlinear simulation model. This kind of architecture allows decoupling of commands and tracking control with measurement control [10]. After observing the success in [9], [11] investigated the Model Based Predictive Control (MBPC) where the control system has been considered as two different loops. Inner loop stabilizes the pitch and roll angles, yaw rate and the speed along z axis while the outer loops controls the longitudinal and lateral speed, altitude and yaw angle. At the end, outer loop closes with the MBPC controller. Disturbance and different input-output limitations have also been tested and gave satisfactory results. According to [12] and [13], in modern control theory two main groups of optimization problems have been popular; H2 and $H\infty$. The goal of H2 is to minimize the mean value of error signal over all frequencies. In contrast, $H\infty$ is called as the worst case formulation in which cost function defined the worst possible value of the error signal over all frequencies is reasonable. It is possible to combine different types of sensory equipment while building a quadrotor. If the topic is sensory equipment for quadrotors the heart will be the IMU. As it is observed in [14], [15], [16], [17] IMU usage is a must for each UAV. It is important to measure translational and rotational acceleration while working on such systems. Despite the advantages of these systems, they do have handicaps while outputting the requested data. One of them is the noisy output. Generally, a complementary filter is chosen in the flight control industry because of its simple technique [18]. The complementary filter users do not consider any statistical description for the noise corrupting signals, and their filter is obtained by a simple analysis in the frequency domain [19]. Considering both Kalman and complementary filtering techniques, complementary filter is much simpler [19,20].

Another topic in this research process is to obtain information about localization problems. In literature, there are various types of target detection algorithm, localization techniques and reference tracking studies. One of the studies, [21], has underlined deploying large numbers of self-powered sensors which are able to receive and process data. The aim in that work is to create a procedure for calibrating the locations and orientations of the network of this sensor group. Another localization method has been studied on soccer robots, [22], where the position knowledge is acquired from distinctive features in the surroundings with a stereo vision system and comparing it to a model of the world. Another work for

localization in University of Tubingen was implementing Wii remote for tracking a pattern of IR (infrared) lights [23]. However it is also stated that implementation of this technique might be impractical in outdoor environments. Chemnitz University sought new approaches where GPS is not available while tracking a localized point. After this, researches had come up with the idea to select the target to be a unique but also simple enough to be tracked at high frame rate. They created a series of white rings on a black background as in the Figure 4 so that the target would be recognized even all rings are not visible. [24]

Figure 4: Quadrotor Target

However, without any additional hearing from target, quadrotor should always hover parallel to the ground. Each diversion in any axis will affect the system dramatically. Institute of Automatic Control Engineering researchers have developed a project that uses onboard vision system alongside IMU. This method uses five different shapes and sizes to determine the altitude of UAV and yaw angle [25]. Sample target has been shown in Figure 5.

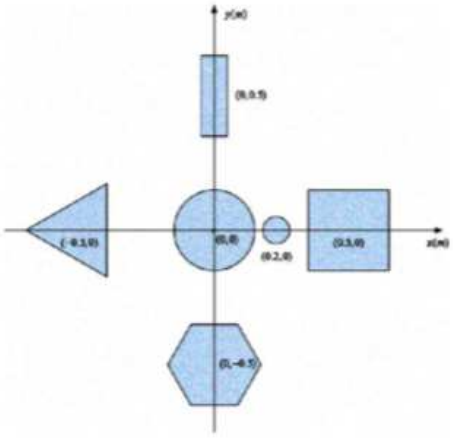

Figure 5: Shape Based Tag [25]

In [26], software called Aruco has been studied. In this work, this technique has been used for detection and avoidance. It is also stated that, this algorithm has better response than the color detection and is not affected by illumination. Aruco mainly used the improved Hamming Code with error detection. This means that, it is possible to extract the center of the marker as x and y coordinates and track them. Because of using AR (Augmented Reality), there is no limitation whether the camera is parallel to the ground or not. Figure 6 shows the detection of the target.

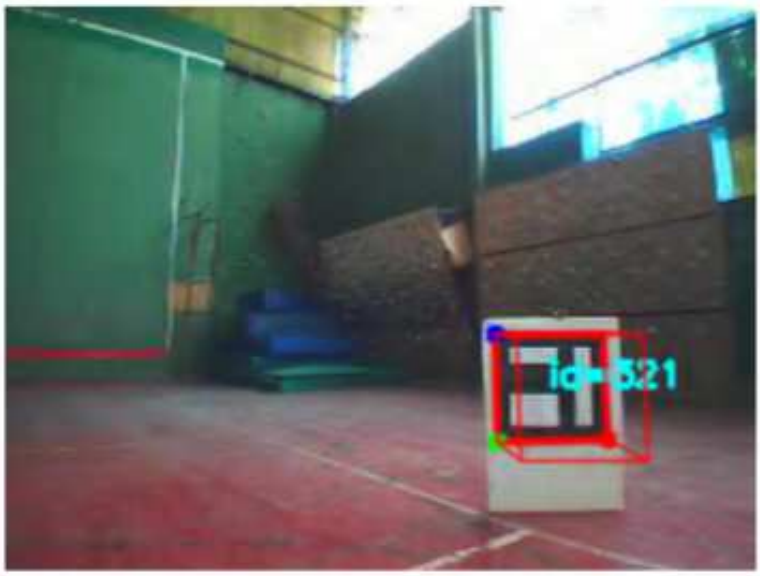

Figure 6: Detection of marker Using Aruco [26]

CHAPTER 3

SYSTEM ARCHITECTURE

This chapter is going to introduce the components of the entire quadrotor system shown in Figure 7. Each electronic part that is used in the structure will be explained in detail with its specification. Electronics components in the system are listed below:

- Brushed high speed electric motors and propellers
- Pololu High-Power Motor Driver 18v15
- Microstrain's GX-2 IMU(Inertial Measurement Unit)
- Sharp IR Distance Sensor (GP2Y0A02YK0F)
- Thermaltake Thoughpower Power Supply
- Logitech HD Webcam C270
- Humusoft MF614 DAQ Card

Figure 7: Physical Structure

The entire controller algorithm has been completed on Matlab ©, Simulink. Real Time Windows Target (RTWT) blockset of this software has been used for controller real time control implementations. Developed controller runs on a computer where, connected with Humusoft MF614 Data Acquisition Board to generate required signals and acquisitions of the sensory components. It is also possible to embed this software on a microcontroller such as Texas Instrument's C2000 series, for which studies have been started.

3.1 Mechanical Structure

Main frame of the quadrotor system consists of carbon fiber tubes which surrounds the motors. To avoid vibration generated by motors, extra carbon fiber tubes have been attached to the junction of each motor made of ABS material (Acrylonitrile Butadiene Styrene). Inertia values for the quadrotor system have been calculated experimentally by using bifilar pendulum experiment. In this experiment it is aimed to calculate inertial changes around *x, y,* and *z* axes. The system was hung from two sides by using ropes. Next step was to give an external input to the system that will generate a rotational movement around the specified axis. With this, tension generated across the ropes will drive the system to a swinging action where inertia values will be calculated. The experimental setup is illustrated in Figure 8. [27]

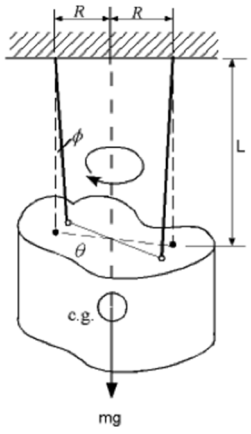

Figure 8: Bifilar Pendulum Experiment [27]

By using this setup, all of the inertial values have been obtained for each axis. Even though the system is symmetric, inertial values for x and y axes have been also calculated. Equation (3.1) has been used to calculate inertia

$$J = \left[\frac{T_n}{2\pi}\right]\frac{mgR^2}{L} \tag{3.1}$$

Where T denotes the period in seconds, m is the mass, g is the gravitational acceleration, R is distance radius and L is the length of ropes. Rotational displacement changes have been recorded by using IMU. Figure 9-11 show the displacement change according to explanation give above.

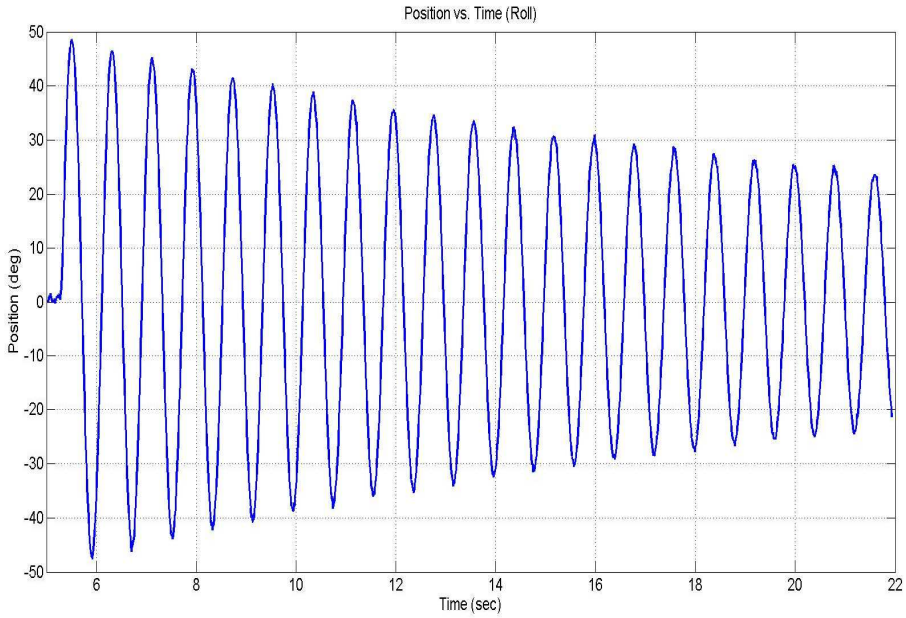

Figure 9: Position vs. Time (Roll)

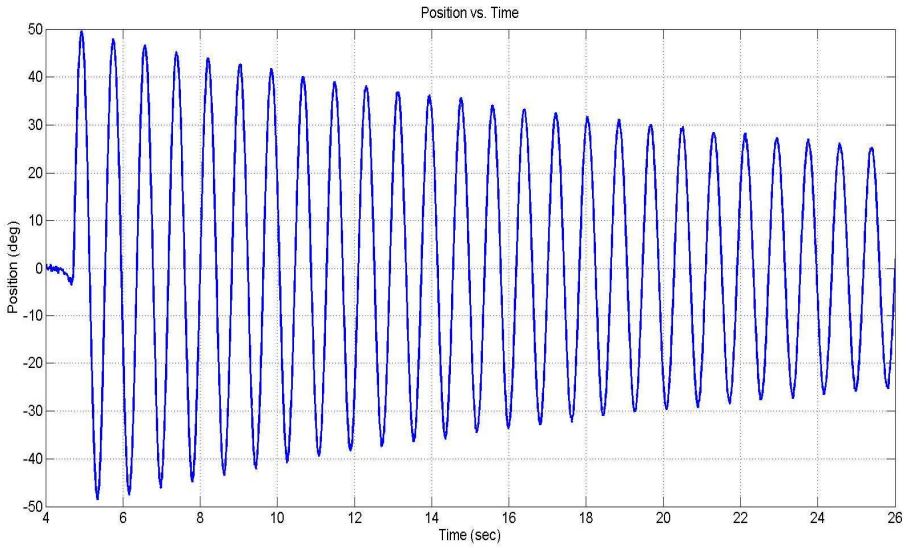

Figure 10: Position vs. Time (Pitch)

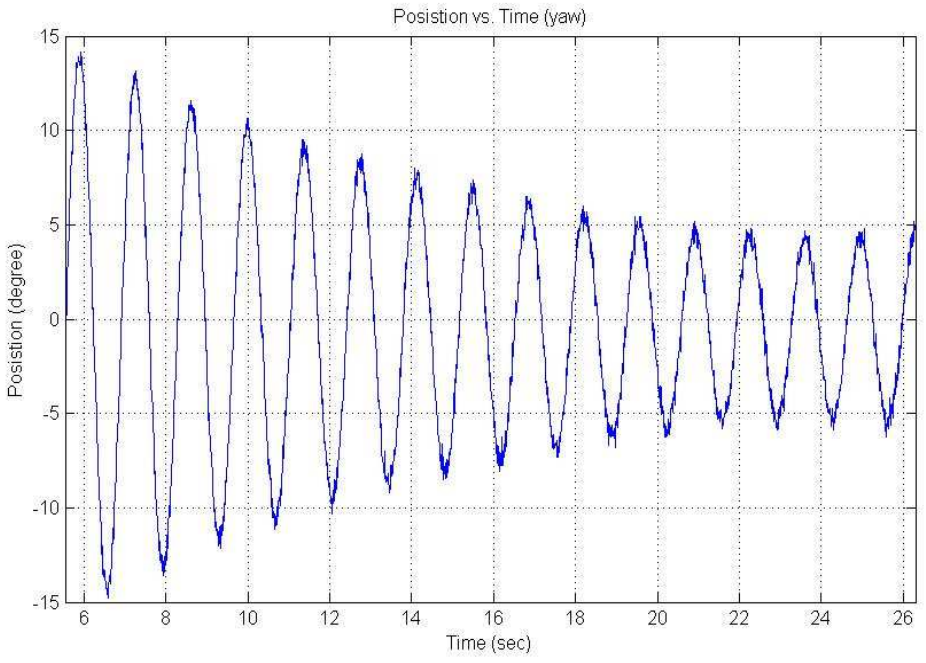

Figure 11: Position vs. Time (yaw)

Values for the setup used during the tests are shown in the Table 1.

Table 1: Values for Inertia Test

m (kg)	0.4913
$g\left(\frac{m}{s^2}\right)$	9.81
R (m)	0.18
L_{roll} (m)	0.35
L_{pitch} (m)	0.35
L_{yaw} (m)	0.51
T_{roll} (sec)	0.8040
T_{pitch}(sec)	0.8183
T_{yaw} (sec)	1.3657
$J_{roll}\ \left(kgm^2\right)$	0.0073
$J_{pitch}\left(kgm^2\right)$	0.0076
$J_{yaw}\left(kgm^2\right)$	0.0145

3.2 Sensory and Hardware Structure

An IMU, a proximity sensor, and a camera form the sensor set on the quadrotor platform. Motor drivers, brushed dc motors, propellers, Aruco markers, and the data acquisition board are the critical hardware that are utilized within the physical system and control computer. All these components are described briefly in the following sections.

3.2.1 Microstrain's 3DM-GX-2 Inertial Measurement Unit

An IMU, which is an electronics device, measures and outputs velocity, orientation and gravitational aceeleration by means of gyroscopes, acceleration and magnetometers. 3DM-GX2 combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm. Data communication between IMU and PC is achieved by using RS-232 protocol with a standard serial cable. 3DM offers various

information stored on its embedded card. The main working principle relies on the request and send system, where, user prompts the specified code to sensor and sensor replies the related orientation matrix. In this thesis, Euler Angles and Euler Rates have been prompted from sensor which the code stands as *0xCF*. Sampling period has been set to 300 Hertz. Sensor replies a matrix where each byte carries a specific data for decode.

Decoding has been completed on PC side with the help of Matlab by using IEEE 754 conversion method. Table 2 shows the byte representation of the response. [28]

Table 2: IMU Response to *0xCF*

Command Byte:	0xCF
Command Data:	None
Response:	31 bytes defined as follows
Byte 1	Header = 0xCF
Byte 2-5	Roll
Byte 6-9	Pitch
Byte 10-13	Yaw
Byte 14-17	AngRate(x)
Byte 18-21	AngRate(y)
Byte 22-25	AngRate(z)
Byte 26-29	Timer
Byte 30-31	Checksum

3.2.2 Sharp IR Proximity Sensor

As previously stated GP2Y0A02 Sharp Sensor has been used for the altitude control for the quadrotor. This sensor has the detection range between 20 cm to 150 cm. Main reason for choosing this sensor is that there is no need for an external circuit. In today's world the Sharp sensor has become very useful in various applications for range and obstacle detection. This is because the Sharp sensor can provide good results for specified ranges for any mechanism. Also, Sharp sensor is less influenced on the colors of reflected objects and their reflectivity, due to optical triangle measuring methods.

The main characteristics of this sensor can be observed on the Figure 12;

(T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Distance measuring range	ΔL	*2 *3	20	–	150	cm
Output terminal voltage	V_O	*2 L=150cm	0.25	0.4	0.55	V
Difference of output voltage	ΔV_O	*2 Output change at L=150cm to 20cm	1.8	2.05	2.3	V
Average dissipation current	I_{CC}	–	–	33	50	mA

Note) L:Distance to reflective object
*2 Using reflective object:White paper (Made by Kodak Co. Ltd. gray cards R-27 · white face, reflective ratio:90%)
*3 Distance measuring range of the optical sensor system

Figure 12: Characteristics of the Sharp Sensor [29]

Table 3 shows feature summary of the Sharp sensor.

Table 3: Feature Summary of Sharp Sensor [29]

Operation Voltage	4.5V to 5.5V
Average Current Consumption	33mA
Distance Measuring Range	20 cm to 150 cm
Output Type	Analog Voltage
Output Voltage Differential Over Distance Range	2V
Response Time	38 ms
Package size	29.5x13x21.5mm
Weight	4.8g

Linearization has been acquired from the datasheet however, while working on previous works on the quadrotor where stabilizing altitude over the RC, linearization process has been done on Matlab by recording the data. After the recording, data have been fitted on a curve to find a relation between voltage and altitude. Figure 13 examines the relation between voltage and altitude which is used in the quadrotor system.

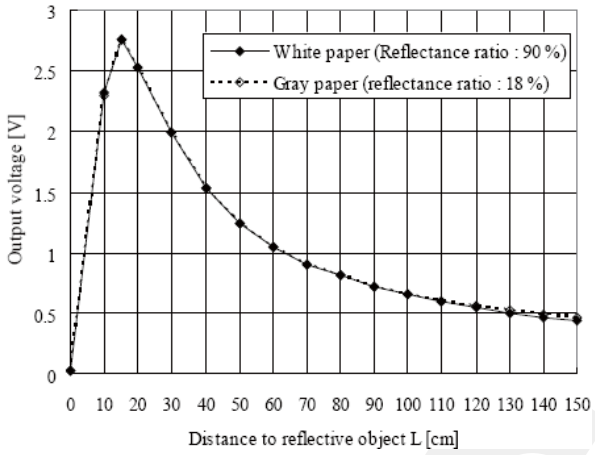

Figure 13: Relation between Altitude and Applied Voltage [29]

3.2.3 Logitech C270 Webcam

Webcam has been used for target detection algorithm in the quadrotor system. The specifications have been listed below.

- HD video calling (1280 x 720 pixels) with recommended system
- Video capture: Up to 1280 x 720 pixels
- Logitech Fluid Crystal™ Technology
- Photos: Up to 3.0 megapixels (software enhanced)
- Built-in mic with noise reduction
- Hi-Speed USB 2.0 certified (recommended)
- Universal clip fits laptops, LCD or CRT monitors

Target detection algorithm has been worked with C code called Aruco where it runs from a different computer with the Linux operating system. The main features of Aruco are:

- Detect markers with a single line of C++ code
- Detection of AR boards (markers composed by several markers)
- Requires only OpenCv (>=2.1)
- Up to 1024 different markers
- Trivial integration with OpenGL and OGRE
- Fast, reliable and cross-platform because relies on OpenCv
- Examples that will help to get running AR application in less than 5 minutes

Simply, Aruco is a marker detection system and allows the user to create its own markers with specified co-ordinate system or only one marker to relative position. With this, it is possible to extract error in *x* and *y* axis. A sample of markers created by the software has been shown in Figure 14.

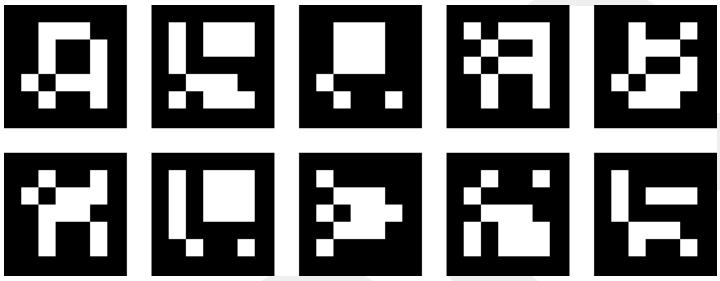

Figure 14: Sample Aruco Markers

Aruco works same with augmented reality, which is, a live, direct, indirect, view of a physical, real-word environment whose elements are supplemented by computer generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified by a computer. As a result, the technology functions by enhancing one's current perception of reality [30]. By contrast, virtual reality replaces the real world with simulation [31], [32]. Augmentation is conventionally in real-time and in sematic context with environmental elements. With the help of advanced AR technology (e.g. adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and digitally manipulable. Artificial information about the environment and its objects can be overlaid on the real world. [33], [34], [35], [36].

3.2.4 Pololu High-Power Motor Driver 18v11

Brushed motors have been driven by using of these motor drivers. The reason behind choosing these drivers is the property of delivering continuous 15 A (ampere) without a heat sink. The Pololu high-power motor driver is a discrete MOSFET H-bridge designed to drive large DC brushed motors. The H-bridge is made up of one N-channel MOSFET per leg, and most of the board's performance is determined by

these MOSFETs (the rest of the board contains the circuitry to take user inputs and control the MOSFETs). The MOSFETs have an absolute maximum voltage rating of 30 V, and higher voltages can permanently destroy the motor driver. Under normal operating conditions, ripple voltage on the supply line can raise the maximum voltage to more than the average or intended voltage, so a safe maximum voltage is approximately 24 V. Each individual drive can only drive only one motor. Directions, PWM (Pulse-Width-Modulation), 5 V-out (Voltage) inputs have been helpful during the assembly process [37]. Input scheme is illustrated in Figure 15.

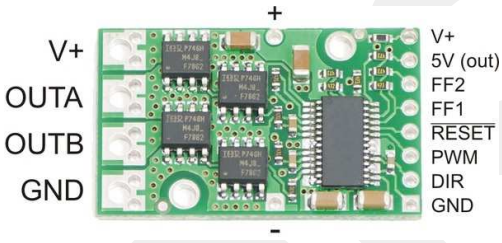

Figure 15: Pololu 18v15 Motor Driver

Table 4 shows the general specifications of the motor driver.

Table 4: General Specifications of 18v15 [37]

Motor Channels	1
Minimum Operating Voltage	5.5V
Maximum Operating Voltage	30V
Continuous Output Current per Channel	15A
Maximum PWM Frequency	40kHz
Maximum Logic Voltage	5.5V
MOSFET on-resistance	3.5mΩ
Reverse Voltage Protection	No

3.2.5 Brushed Motors

Draganflyer's brushed DC electric motor with capacitor and pinion gear has been used in this work. This is the stock motor for the Draganflyer's III, IV and V Ti RC helicopters. The motor attaches to the Draganflyer's motor mount using existing bolts. Soldering is required for installation of the power wires. Characteristics of the motor has been shown in Table 5.

Table 5: Specification Table for RC-280SA [38]

MODEL		VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL		
		OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE		OUTPUT	TORQUE		CURRENT
		V		r/min	A	r/min	A	mN·m	g·cm	W	mN·m	g·cm	A
RC-280SA	2865	4.5 - 9.0	6	14000	0.28	11910	1.60	4.46	45.5	5.56	29.9	305	9.10
	20120	4.5 - 12.0	9	11300	0.14	9540	0.76	4.06	41.3	4.04	26.0	265	4.10
	2485	4.5 - 9.6	6	10800	0.18	9130	0.99	3.71	37.8	3.54	24.0	245	5.40

Table 6 illustrates results done on the motor with different PWM and frequency modes. According to results on each mode, best result has been chosen and applied to the real time model. Each result is compared with each other by considering the current flow maximum frequency.

Table 6: Motor PWM Tests

%50 Duty Ratio			**%80 Duty Ratio**		
Frequency (Hz)	Current (A)	Tachometer	Frequency (Hz)	Current (A)	Tachometer
100	1,39	78	100	1,78	147
200	1,32	92	200	1,682	153
300	1,256	101	300	1,636	156
500	1,122	116	500	1,575	162
600	0,988	120	600	1,56	165
1000	0,826	126	1000	1,49	168
2000	0,7	130	2000	1,442	171
5000	0,649	130	5000	1,428	171
%75 Duty Ratio			**%90 Duty Ratio**		
Frequency (Hz)	Current (A)	Tachometer	Frequency (Hz)	Current (A)	Tachometer
100	1,767	133	100	1,842	168
200	1,645	141	200	1,773	171
300	1,583	147	300	1,761	174
500	1,52	153	500	1,74	174
600	1,495	156	600	1,734	174
1000	1,4	162	1000	1,722	177
2000	1,332	165	2000	1,709	177
5000	1,314	165	5000	1,7	177

3.2.6 Propellers

Stock propellers of the Draganflyer's V Ti Pro model have been used in the actual quadrotor system. Parameters of these propellers have been taken in literature survey while modeling and these parameters have been accepted as valid for the entire controller design. Propeller dimension is 38 x 10 x 3 cm and each propeller has weight of 45g each.

3.2.7 Humusoft MF614 DAQ Card

To communicate between the PC and quadrotor, Humusoft's MF614 DAQ card has been used. This card allows user to communicate with any desired system with MATLAB/Simulink. Hence, all motor inputs and sensor outputs (except the camera and IMU) has been connected to MF614. Specifications of the DAQ card has been given below.

- Eight single-ended 12-bit analog input channels
- Four 12-bit analog output channels
- Sampling rate up to 100 kHz
- 8 digital inputs, 8 digital outputs
- Programmable A/D ranges
- Four quadrature encoder inputs (differential)
- Five counters/timers
- Low power consumption

CHAPTER 4

MATHEMATICAL MODELLING

The quadrotor mathematical model has been acquired by representing this vehicle as a rigid body in a three dimensional space. Following assumptions have been made.

- System structure is rigid and symmetrical.
- No vibration and no blade flapping occur.
- Gyroscopic moments are neglected.
- Earth fixed frame is assumed to be inertial.

Let the generalized coordinates of the quadrotor be expressed as;

$$q = (\phi, \theta, \psi, X, Y, Z) \in \Re^6 \tag{4.1}$$

Where $\xi = (x, y, z \in \mathbb{R}^3)$ implies the position vector of the COG (Center of Gravity) relative to the fixed inertial frame. Euler angles of the quadrotor have been expressed as $\eta = (\phi, \theta, \psi) \in \mathbb{R}^3$, where, ϕ is the roll angle around the x-axis, θ is the pitch angle around y-axis, and ψ is the yaw angle around the z-axis, Figure 16.

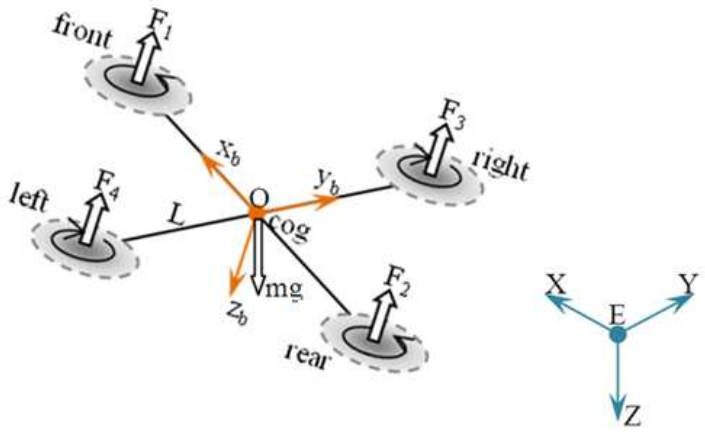

Figure 16: Body and Inertial Reference Frames

Let us define the Lagrangian equation by [39]

$$L(q,\dot{q}) = T_{trans} - T_{rot} - U \tag{4.2}$$

Where $T_{trans} = \frac{m}{2}\dot{\xi}^T\dot{\xi}$ is the translational kinetic energy, $T_{rot} = \frac{1}{2}\Omega^T \mathrm{I}\Omega$ is the rotational kinetic energy, $U = -mgz$ is the potential energy of the quadrotor z is negative of the altitude, m is the mass of the entire system, Ω is the vector of angular velocity, I is the inertia matrix, and g denotes gravitational acceleration. The angular velocity vector ω resolved in the body-fixed frame is related to the generalized velocities $\dot{\eta}$ (in the region where the Euler angles are valid) by means of the standard kinematic relationship. [39]

$$\Omega = W_\eta \dot{\eta} \tag{4.3}$$

Where

$$W_\eta = \begin{bmatrix} -\sin\theta & 0 & 1 \\ \cos\theta\sin\phi & \cos\phi & 0 \\ \cos\theta\cos\phi & -\sin\phi & 0 \end{bmatrix} \tag{4.4}$$

Then

$$\Omega = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \\ \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{bmatrix} \tag{4.5}$$

Let us define,

$$J = J(\eta) = W_\eta^T I W_\eta \tag{4.6}$$

where,

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{4.7}$$

So that,

$$T_{rot} = \frac{1}{2}\dot{\eta}^T J \dot{\eta} \tag{4.8}$$

With this $J = J(\eta)$ will be transformed into the inertia matrix for the full rotational kinetic energy of quadrotor which is expressed directly in terms of generalized coordinates η. Entire model of the quadrotor system has been acquired using *Euler-Lagrange* method.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = \begin{bmatrix} F_\xi \\ \tau \end{bmatrix} \tag{4.9}$$

Where $F_\xi = R\widehat{F} \in \mathbb{R}^3$ is the translational force applied to quadrotor due to main thrust, $\tau \in \mathbb{R}^3$ represents the yaw, pitch, and roll moments and R stands for the rotational matrix. $R(\phi, \theta, \psi) \in SO(3)$ represents the orientation of the quadrotor relative to a fixed frame:[39]

$$R = \begin{bmatrix} -c\theta c\psi & c\psi s\theta s\phi & s\phi s\psi + c\phi c\psi s\theta \\ c\theta s\psi & c\phi c\psi + s\theta s\phi s\psi & \text{c}\phi\text{s}\theta\text{s}\psi - \text{c}\psi\text{s}\phi \\ -\text{s}\theta & \text{c}\theta\text{s}\phi & \text{c}\theta\text{c}\phi \end{bmatrix} \tag{4.10}$$

**c stands for cos and s stands for sin functions.*

$$\hat{F} = \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} \tag{4.11}$$

Where u represents the main thrust that is acting on the quadrotor which is going to be expressed as

$$u = \sum_{i=1}^{4} f_i \tag{4.12}$$

For each i values varies from 1 to 4, f_i represents the generated force produced by the i th motor. In general, $f_i = k\omega_i^2$ where k is a constant and ω is the angular speed of the corresponding motor. However, in this thesis experimentally identified motor model is used. Setup in Figure 17 is configured to perform identification experiment. Thrust generated by the motor-propeller unit is identified as a function of applied voltage to the motor, i.e. $f_i = kV$. k denotes the relation between force and voltage of the motor. In other words, it shows the generated force per voltage.

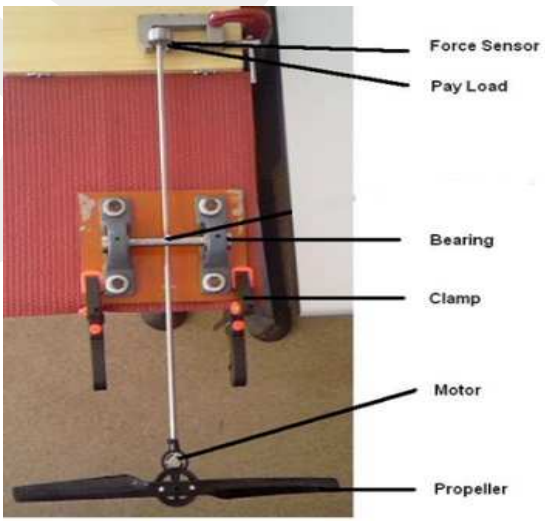

Figure 17: Thrust Measurement Setup

Thus, generalized torque values can be represented as:

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \hat{=} \begin{bmatrix} (f_4 - f_3)L \\ (f_1 - f_2)L \\ \sum_{i=1}^{4} = \tau_{Mi} \end{bmatrix} \tag{4.13}$$

Where L indicates the distance between propellers and COG. τ_{Mi} is the drag moment on each propeller around the body z axis of the quadrotor.

Since the Lagrangian contains no cross coupling terms in the kinetic energy combining $\dot{\xi}$ with $\dot{\eta}$, the Euler–Lagrange equation can be partitioned into dynamics for ξ coordinates and η coordinates. The Euler–Lagrange equation for the translational motion is [39]:

$$\frac{d}{dt}\left[\frac{\partial L_{trans}}{d\dot{\xi}}\right] - \frac{dL_{trans}}{\partial \xi} = F_\xi \tag{4.14}$$

then

$$m\ddot{\xi} + mgE_z = F_\xi \tag{4.15}$$

As for the η coordinates, it can be written

$$\frac{d}{dt}\left[\frac{\partial L_{rot}}{\partial \dot{\eta}}\right] - \frac{\partial L_{rot}}{\partial \eta} = \tau \tag{4.16}$$

or

$$\frac{d}{dt}\left[\dot{\eta}^T J \frac{\partial \dot{\eta}}{\partial \eta}\right] - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}^T J \dot{\eta}) = \tau \tag{4.17}$$

With this it is possible to obtain:

$$J\ddot{\eta} + \dot{J}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}^T J \dot{\eta}) \tag{4.18}$$

By defining the *Coriolis-Centripetal* Vector

$$\bar{V}(\eta,\dot{\eta}) = \dot{J}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial\eta}(\dot{\eta}^T J\dot{\eta}) \tag{4.19}$$

One may write

$$J\ddot{\eta} + \bar{V}(\eta,\dot{\eta}) = \tau \tag{4.20}$$

But $\bar{V}(\eta,\dot{\eta})$ can be expressed as

$$\begin{aligned}\bar{V}(\eta,\dot{\eta}) &= (\dot{J} - \frac{1}{2}\frac{\partial}{\partial\eta}(\dot{\eta}^T J))\dot{\eta} \\ &= C(\eta,\dot{\eta})\dot{\eta}\end{aligned} \tag{4.21}$$

where $C(\eta,\dot{\eta})$ is referred to as *Coriolis* term and contains the gyroscopic and centrifugal terms associated with η dependence of J. This yield

$$m\ddot{\xi} + mgE_z = F_\xi \tag{4.22}$$

$$J\ddot{\eta} = \tau - C(\eta,\dot{\eta})\dot{\eta} \tag{4.23}$$

To simplify it is possible to take

$$\tilde{\tau} = \begin{pmatrix}\tilde{\tau}_\phi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\psi\end{pmatrix} = J^{-1} - C(\eta,\dot{\eta})\dot{\eta} \tag{4.24}$$

Finally it is possible to extract

$$\ddot{X} = \frac{u}{m}(\sin\ \phi\ \sin\ \psi\ +\ \cos\ \phi\ \cos\ \psi\ \sin\ \theta\ ) \tag{4.25}$$

$$\ddot{Y} = \frac{u}{m}(\cos\ \phi\ \sin\ \theta\ \sin\ \psi\ -\ \cos\ \psi\ \sin\ \phi\ ) \tag{4.26}$$

$$\ddot{Z} = \frac{u}{m} cos\ \theta\ cos\ \phi - g \tag{4.27}$$

$$\ddot{\phi} = \tilde{\tau}_{\phi} \tag{4.28}$$

$$\ddot{\theta} = \tilde{\tau}_{\theta} \tag{4.29}$$

$$\ddot{\psi} = \tilde{\tau}_{\psi} \tag{4.30}$$

Where X and Y stands for the coordinates in the horizontal plane and Z for the negative of altitude (vertical position). Also one may write Equations 4.28, 4.29 and 4.30 as follows:

$$\ddot{\phi} = \frac{\dot{\theta}\dot{\psi}\,(I_{yy} - I_{zz})}{I_{xx}} + \frac{\tau_{\phi}}{I_{xx}} \tag{4.31}$$

$$\ddot{\theta} = \frac{\dot{\phi}\dot{\psi}\,(I_{zz} - I_{xx})}{I_{yy}} + \frac{\tau_{\theta}}{I_{yy}} \tag{4.32}$$

$$\ddot{\psi} = \frac{\dot{\theta}\dot{\phi}\,(I_{xx} - I_{yy})}{I_{zz}} + \frac{\tau_{\psi}}{I_{zz}} \tag{4.33}$$

CHAPTER 5

CONTROLLER DESIGN AND SIMULATIONS

In the process of controller design the following steps have been implemented;

- Quadrotor model has been obtained mathematically
- Linearization of the mathematical model has been completed
- LQR controller has been designed
- Designed controller has been applied to nonlinear system model
- Gains are tuned
- Outputs of the simulation and real system have been compared and analyzed together

5.1 Linearization of Nonlinear State Equations

State variables are modified so as to design controllers to track a target on ground, Figure 18 illustrates the representation.

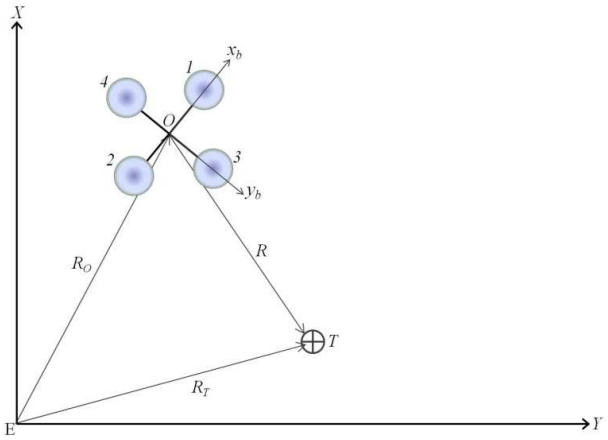

Figure 18: Target Tracking Representation

Following vectors give the relevant positions. $\vec{R}_O$ is the position vector of the COG of the quadrotor, $\vec{R}_T$ represents the position vector of the target on XY plane.

$$\vec{R}_O = X\vec{I} + Y\vec{J} \tag{5.1}$$

$$\vec{R}_T = X_T\vec{I} + Y_T\vec{J} \tag{5.2}$$

Relative position of the target with respect to the quadrotor on X-Y plane is given as below.

$$\vec{R} = \vec{R}_T - \vec{R}_O = (X_T - X)\vec{I} + (Y_T - Y)\vec{J} \tag{5.3}$$

Let $E_X = X_T - X$ and $E_Y = Y_T - Y$. It is assumed that target has a very low velocity and an acceleration. Also yaw angle of the quadrotor is zero. Under these assumptions following equations can be written for the error dynamics.

$$\dot{E}_X \cong -\dot{X} \tag{5.4}$$

$$\ddot{E}_X \cong -\ddot{X} \tag{5.5}$$

$$\dot{E}_Y \cong -\dot{Y} \tag{5.6}$$

$$\ddot{E}_Y \cong -\ddot{Y} \tag{5.7}$$

Mathematical model is modified by replacing the state variables related with the position and velocity in X-Y plane by the relevant error terms, E_X, $\dot{E}_X$, E_Y, and the $\dot{E}_Y$ State variable vector is given as below.

$$x = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & E_X & \dot{E}_X & E_Y & \dot{E}_Y & Z & \dot{Z} \end{bmatrix}^T \tag{5.8}$$

$$u = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 \end{bmatrix}^T \tag{5.9}$$

Quadrotor tracks the slowly moving target on the ground by regulating the state variables. Nonlinear state equations are derived using the equations in the previous chapter considering the modified state variables. Nonlinear state equations are linearized around the hovering condition and zero tracking error condition by using

Taylor Series expansion. The linear time invariant state-space representation of the system is obtained in the following form.

$$\dot{x} = Ax + Bu \tag{5.10}$$

$$y = Cx + Du \tag{5.11}$$

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{E}_X \\ \ddot{E}_X \\ \dot{E}_Y \\ \ddot{E}_Y \\ \dot{Z} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ E_X \\ \dot{E}_X \\ E_Y \\ \dot{E}_Y \\ Z \\ \dot{Z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -6.57 & -6.57 \\ 0 & 0 & 0 & 0 \\ 6.84 & -6.84 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.27 & 0.27 & -0.27 & -0.27 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.25 & -0.25 & -0.25 & -0.25 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \tag{5.12}$$

All state variables are measured by using inertial measurement unit, proximity sensor and Aruco markers detection system.

5.2 Controllability Analysis

Controllability check is required before designing a state feedback controller. The state space representation of linear time invariant systems has been given in Equation 5.10 as;

$$\dot{x} = Ax + Bu \tag{5.13}$$

Where;

A is the 12 x 12 “state matrix”,

B is the 12 x 4 “input matrix”,

Controllability matrix given in Equation 5.14 has full rank revealing the full state controllability [40];

$$Co = \begin{bmatrix} B & BA & BA^2 & \dots & BA^{n-1} \end{bmatrix} \tag{5.14}$$

5.3 LQR Design

The settings of a controller that is applied to quadrotor are found by using a mathematical algorithm that minimizes a cost function with weighting factors supplied by the designer. In this work, LQR controller has been designed with the help of MATLAB. For a continuous-time linear system described by

$$\dot{x} = Ax + Bu \tag{5.15}$$

With a cost-function defined as;

$$\Im = \int_0^\infty \left(x^T Q x + u^T R u\right) dt \tag{5.16}$$

The feedback control law that minimizes the value of the cost is

$$u = -Kx \tag{5.17}$$

Where K is given by;

$$K = R^{-1} B^T P \tag{5.18}$$

And P is found by solving the continuous time algebraic Riccati equation

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \tag{5.19}$$

Where R and Q matrices are the weighting matrices. Regulator design is extended to track the reference inputs for the given output vector, $[\phi, \theta, \psi, E_X, E_Y, Z]^T$. This has been achieved by creating new augmented states where they declare the integrals of the errors in between the reference inputs and outputs in real time. Therefore while designing LQR controller for this system, six additional states have been considered. System outputs are defined as

$$y = [\phi, \theta, \psi, E_X, E_Y, Z]^T \tag{5.20}$$

Where the reference inputs are;

$$R_y = \lfloor R_\phi, R_\theta, R_\psi, R_{eX}, R_{eY}, R_z \rfloor \tag{5.21}$$

Let the error defined as;

$$e = R_y - y \tag{5.22}$$

New state equations now can be defined as

$$\dot{x}_I = R_y - Cx \tag{5.23}$$

Augmented state vector can be defined as;

$$\bar{x} = \begin{bmatrix} x \\ x_I \end{bmatrix} \tag{5.24}$$

Where x is the state vector; given in Equation 5.8

State equations for the augmented state vector can be defined as;

$$\dot{\bar{x}} = \begin{bmatrix} \dot{x} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} R_y \tag{5.25}$$

C matrix is;

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.26}$$

Control input is now;

$$u = -\bar{K}\bar{x} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} \tag{5.27}$$

While designing LQR controller for the system, each element of Q matrix will affect the corresponding state as stated before where the Q matrix is a diagonal matrix. R matrix will define energy consumption for the controller effect. It is aimed to eliminate the effects of steady disturbances by using the error states. Below Figures 19 and 20 show the simulation model that is used for controller design.

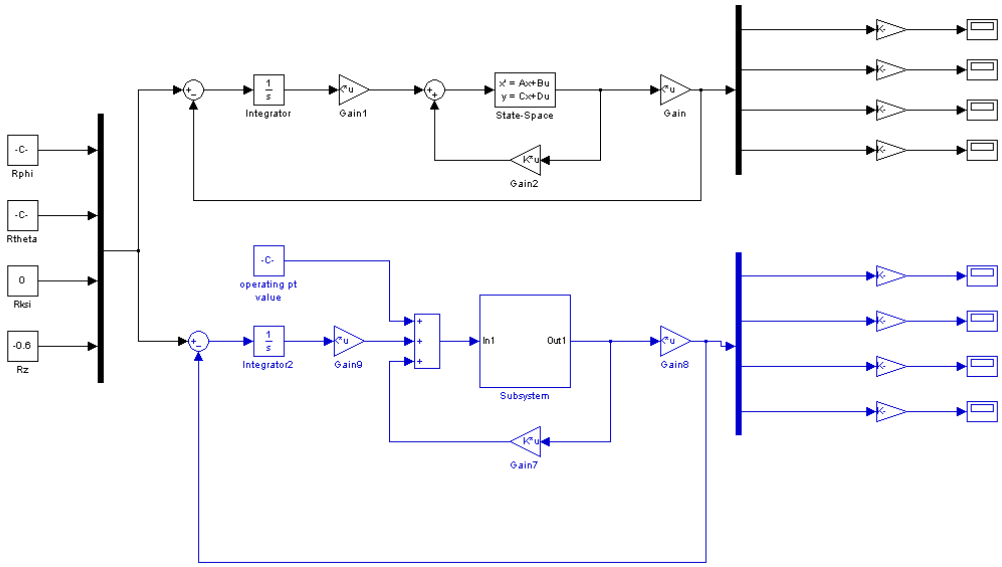

Figure 19: Linear (Black) and Nonlinear (Blue) Models for Attitude and Altitude Dynamics

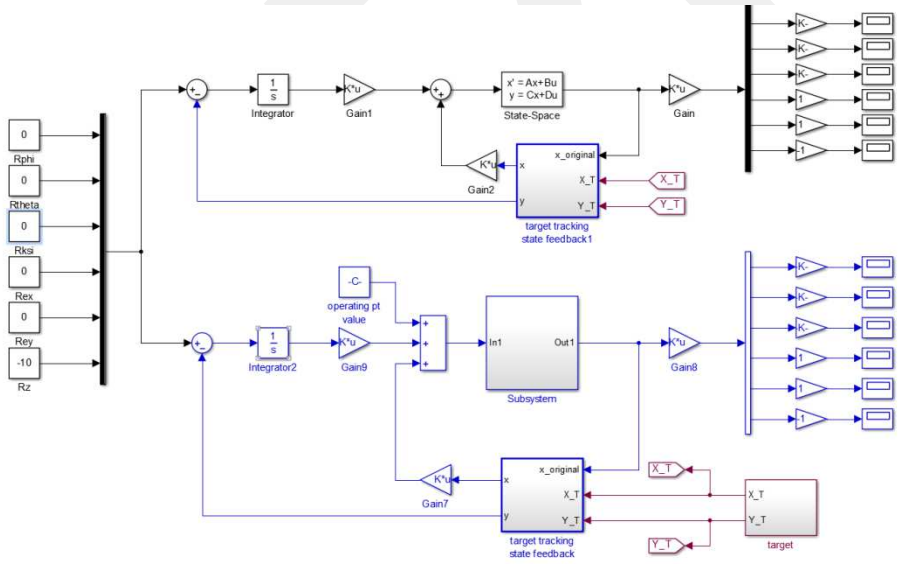

Figure 20: Linear (Black) and Nonlinear (Blue) Models for Attitude, Altitude and Target Tracking Controller Design

As it is stated before normalized LQR approach has been used while designing LQR controller. Normalized weighting matrices Q and R are used:

$$Q = \begin{bmatrix} \frac{\alpha_1^2}{(x_1)_{max}^2} & & & \\ & \frac{\alpha_2^2}{(x_2)_{max}^2} & & \\ & & \ddots & \\ & & & \frac{\alpha_n^2}{(x_n)_{max}^2} \end{bmatrix} \tag{5.28}$$

$$R = \rho \begin{bmatrix} \frac{\beta_1^2}{(u_1)_{max}^2} & & & \\ & \frac{\beta_2^2}{(u_2)_{max}^2} & & \\ & & \ddots & \\ & & & \frac{\beta_m^2}{(u_m)_{max}^2} \end{bmatrix} \tag{5.29}$$

The $(x_i)_{max}$ and $(u_i)_{max}$ represents the largest desired amplitude/control input for that component of the state/actuator signal. Following constraints are set for the relecant coefficients.

$\sum_i \alpha_i^2 = 1$ and $\sum_i \beta_i^2 = 1$

ρ is used as the last relative weighting between the control and state penalties. Therefore it gives a relatively concrete way to discuss the relative size of Q and R and their ratio.

5.4 Controller Simulations

Before applying designed controllers to the actual system, simulations have been done on the system. Firstly, attitude and altitude dynamics are controlled by normalized LQR. $\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, z, \dot{z}$ states have been controlled in this simulation. Architecture of the controller design has been shown in Figure 21.

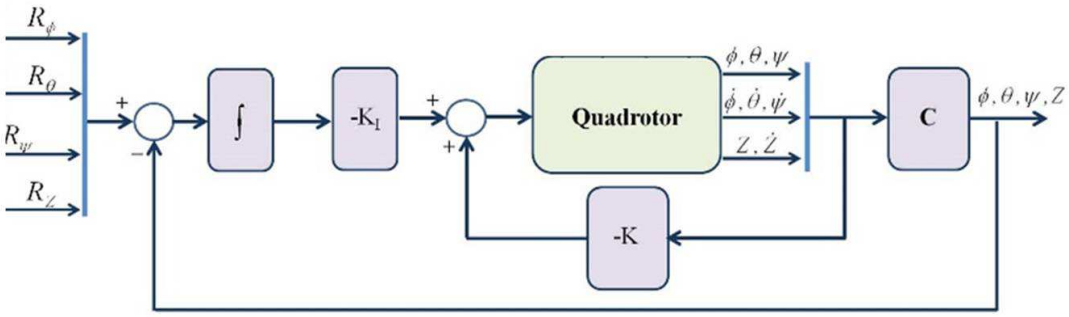

Figure 21: Theory of LQR Controller for Attitude and Altitude Dynamics

Figure 22 illustrates the simulation results for attitude dynamics ϕ, θ, ψ. It is important to remember that simulations have been completed by using non-zero initial conditions.

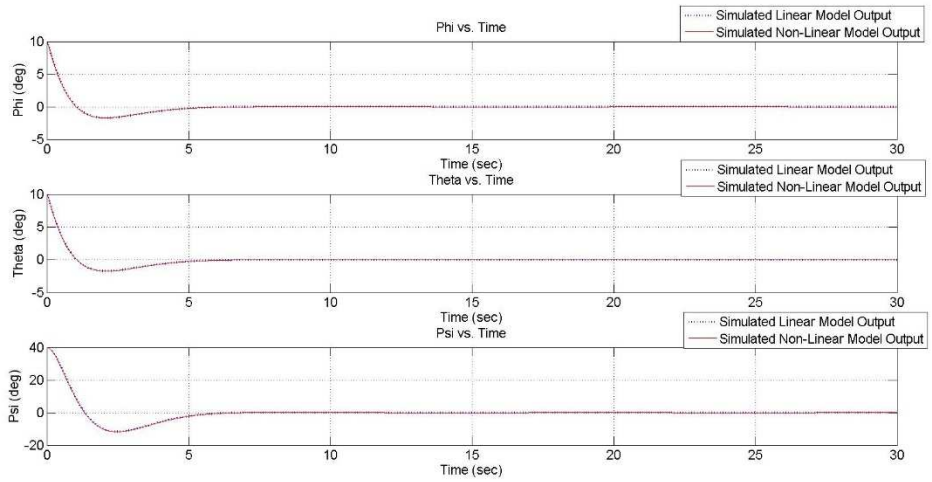

Figure 22: Attitude Dynamics

Figure 23 illustrates the case that nonzero reference angles are used and tracking performance on nonlinear model.

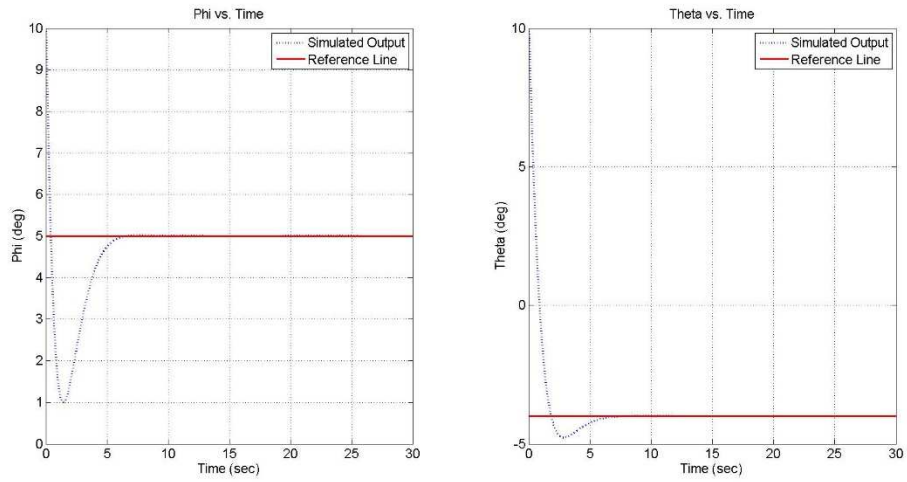

Figure 23: Roll and Pitch Tracking with Varying References

After achieving success on attitude dynamics, altitude dynamics ($z, \dot{z}$) have been added to the same simulation. Simulations of the altitude control on linear and nonlinear model have been shown in Figure 24.

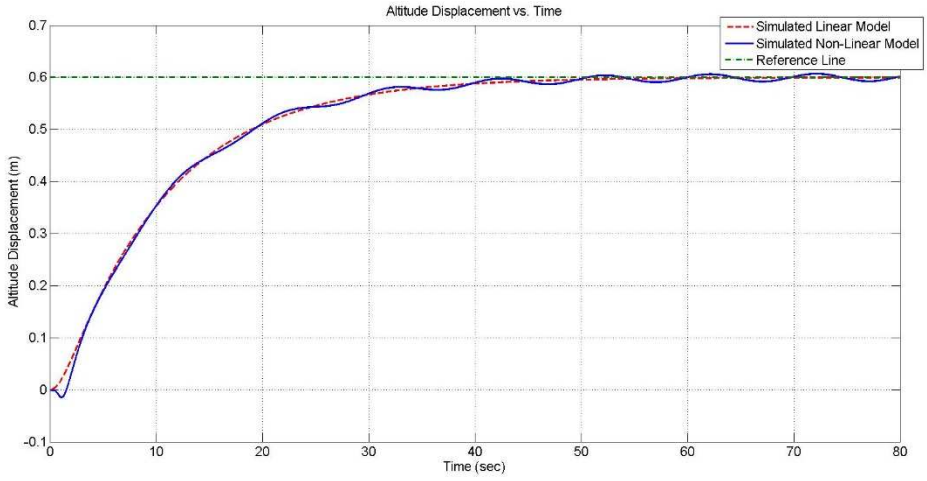

Figure 24: Altitude Displacement vs. Time

While designing LQR controller Q matrix has been obtained by using normalized LQR method where each state gain has been obtained with a specified range of change in each state.

$$Q = \begin{bmatrix} 60.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 40.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20.3 \end{bmatrix} \tag{5.30}$$

$$R = \begin{bmatrix} 8.3333 & 0 & 0 & 0 \\ 0 & 8.3333 & 0 & 0 \\ 0 & 0 & 8.3333 & 0 \\ 0 & 0 & 0 & 8.3333 \end{bmatrix} \tag{5.31}$$

After achieving success on attitude and altitude controller $E_x, \dot{E}_x, E_y \dot{E}_y$ states have been added to the same model for observing target tracking performance. Simulations of the designed LQR for target tracking are shown in Figure 25 and Figure 26.

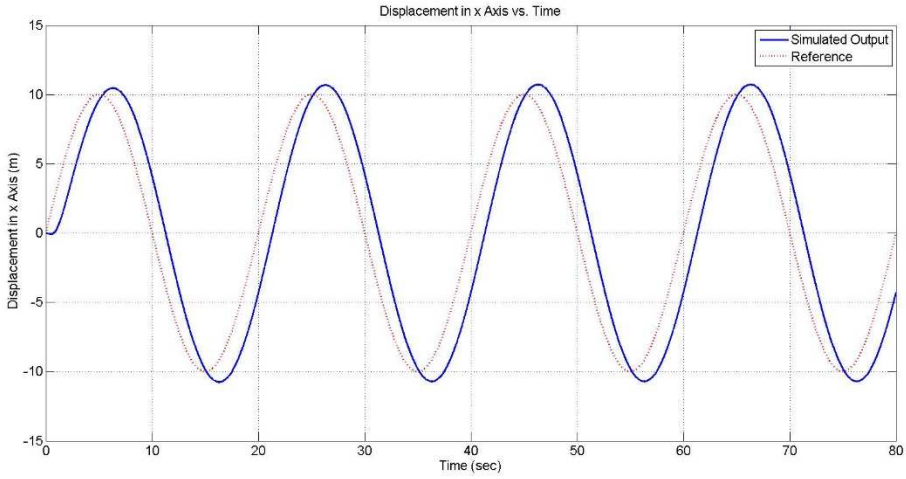

Figure 25: Target Tracking on *x* Axis Referenced to a Sinusoidal Input

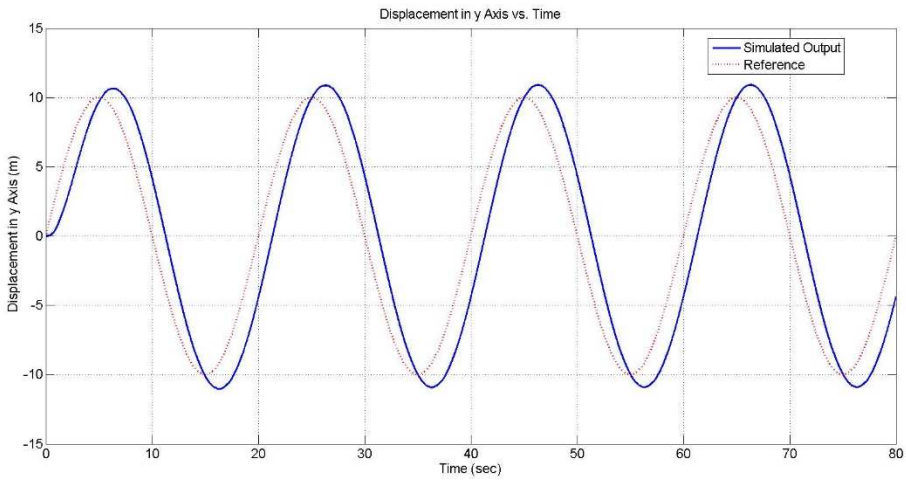

Figure 26: Target Tracking on *y* Axis

Target has been assumed to move in a sinusoidal path in both X and Y axes. According to references shown in Figure 25 and Figure 26 roll and pitch changes while tracking the target can be observed in Figure 27.

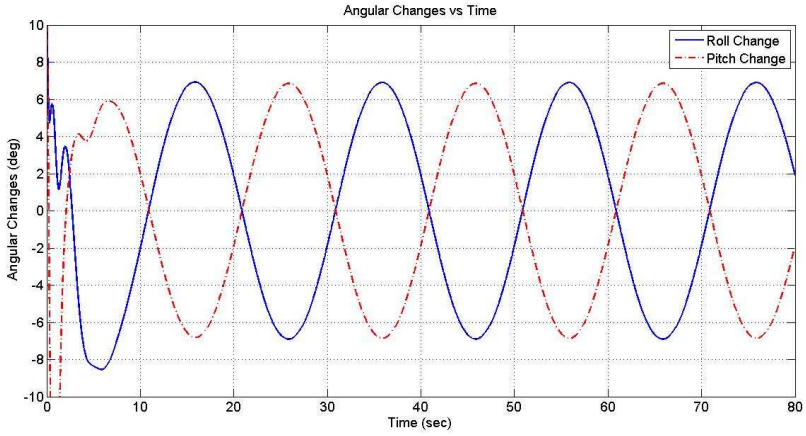

Figure 27: Angular Changes for Tracking Sine Input

This time target has been assumed to move in a square path. Results have been shown in Figure 28 and Figure 29.

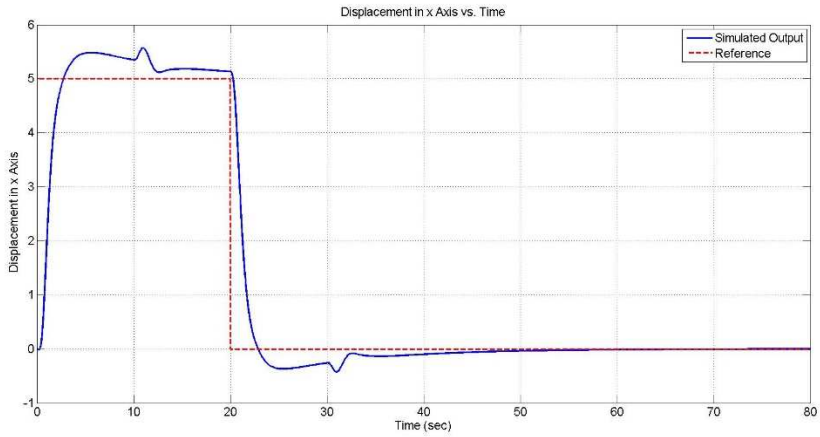

Figure 28: Target Tracking on x Axis Referenced to a Square Input

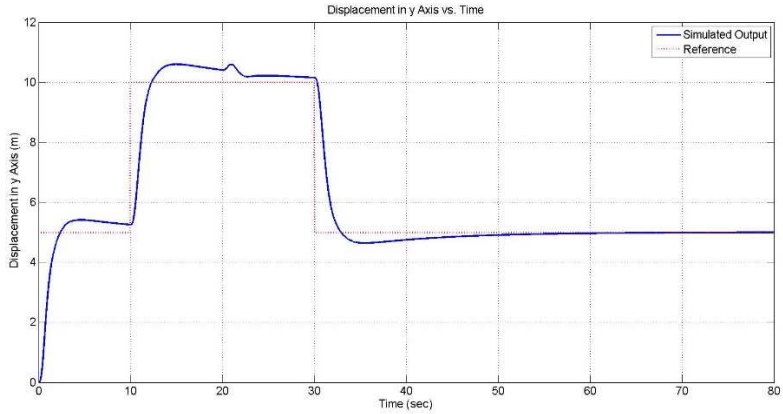

Figure 29: Target Tracking on y Axis Referenced to a Square Input

Q matrix is further extended to handle the new state vector of the system. Q and R matrices are also tuned to reach the desired performance. After observing the results, controller has been checked for generated voltage throughout the entire operation. *R* matrix has been tuned to avoid exceeded voltage generation. With this, controller performance has been verified with low-power consumption. Controller output has been illustrated in Figure 30.

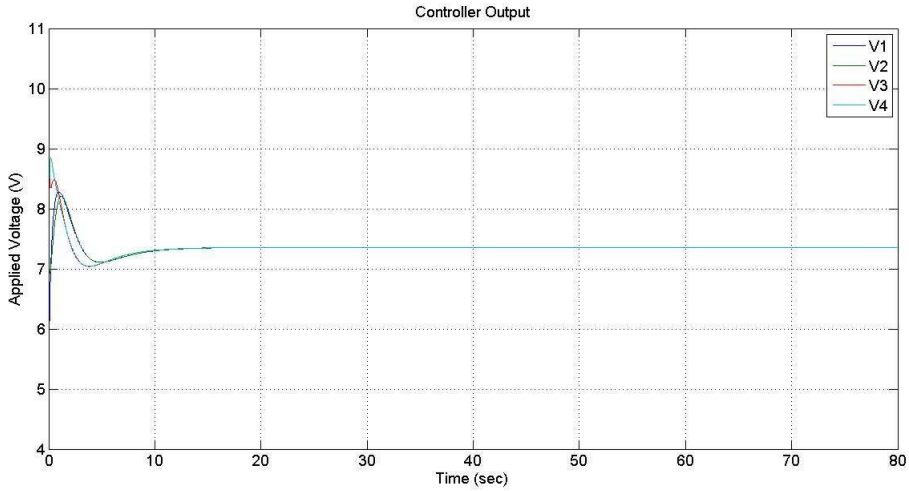

Figure 30: Controller Output (With Nominal Voltage)

Increase in R will provide power save on controller effect. It is important to remember that each R gain value affects the voltage that is applied to corresponding motor. Calculations and approaches have been also done on the same software.

$$R = \begin{bmatrix} 10.3333 & 0 & 0 & 0 \\ 0 & 10.3333 & 0 & 0 \\ 0 & 0 & 10.3333 & 0 \\ 0 & 0 & 0 & 10.333 \end{bmatrix} \tag{5.33}$$

For target tracking, PD controller has also been tested. For attitude and altitude dynamics again LQR controller has been used. Aim is to give reference points to roll and pitch where reference declares the error in corresponding axes. Figure 31 shows the theory of the PD+LQR controller design for target tracking.

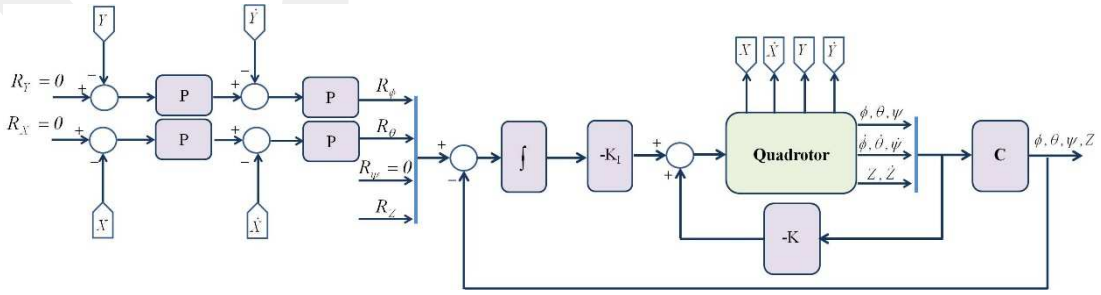

Figure 31: Theory of PID+LQR Controller

In Figure 32 Simulink model used for PD controller has been shown.

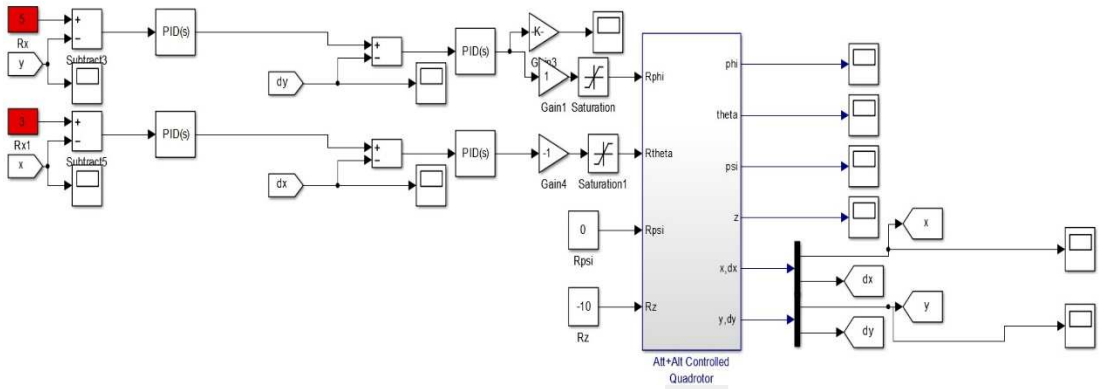

Figure 32: PID Simulation Model

The red boxes declare the reference points to *x* and *y* axes for target tracking. To observe the controller output these values have been taken non-zero values. In order to track the target in *y* axis corresponding reference has been given to roll movement, for *x* axis reference has been given to pitch movement. Figure 33 shows the simulation results.

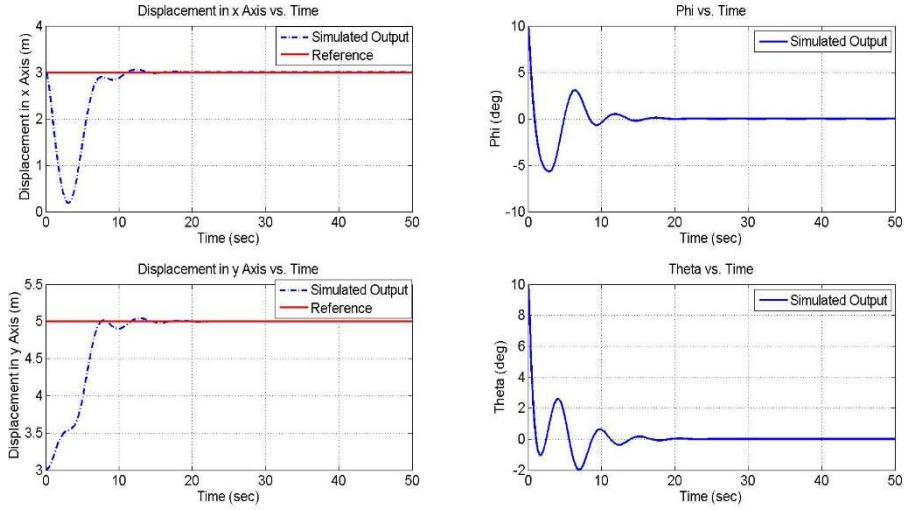

Figure 33: PID Simulation Results for Target Tracking

CHAPTER 6

REAL-TIME EXPERIMENTS ON THE SETUP

6.1 Real Time Test

After simulations have been completed, real time controller algorithm has been applied on the system by Simulink Real Time Windows Target Blockset. Analog input, frequency output, and packet input blocks in RTWT are used for data input/ output. LQR gains on attitude and altitude dynamics in simulation have been used for the real time experiment. Gains of target tracking states have been tuned by trial and error method.

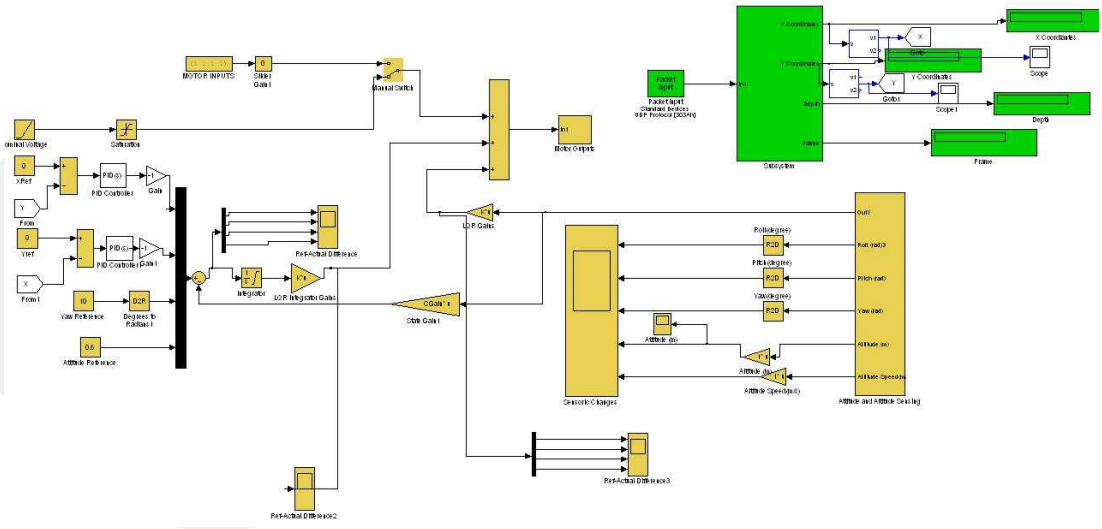

Figure 34: Simulink Model of the Quadrotor System

In Figure 34, it is possible to observe the sensor blocks which are painted to yellow. Left part of the model declares the reference inputs to the system and green boxes are representing the camera output. Conversions of the data that are received from the all

sensors have been done by MATLAB also. For GX-2 IEEE method, for sharp sensor polynomial fit and for camera ASCII conversion scripts have been used.

6.1.1 Attitude and Altitude Control Experiment

Aim of the first experiment is to regulate roll and pitch dynamics at origin and keep the yaw angle at 10 deg. reference as shown in Figure 35 and Figure 36.

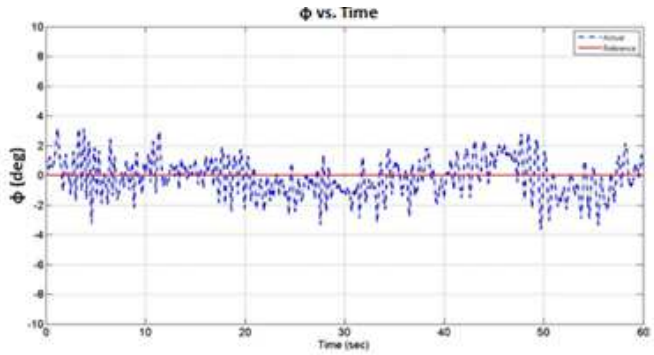

Figure 35: Roll Angle

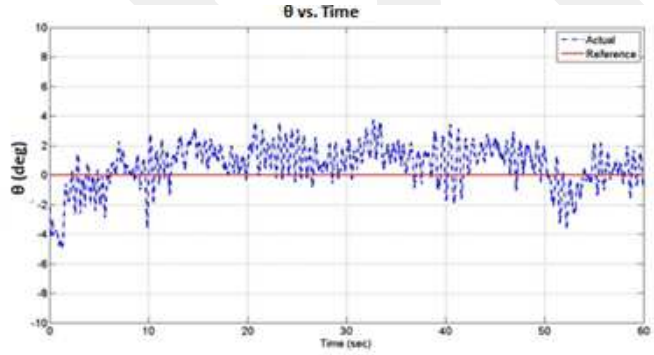

Figure 36: Pitch Angle

After stabilizing roll and pitch axis, yaw angle and its angular rate have been controlled. Yaw axis stabilization graph seen in Figure 37.

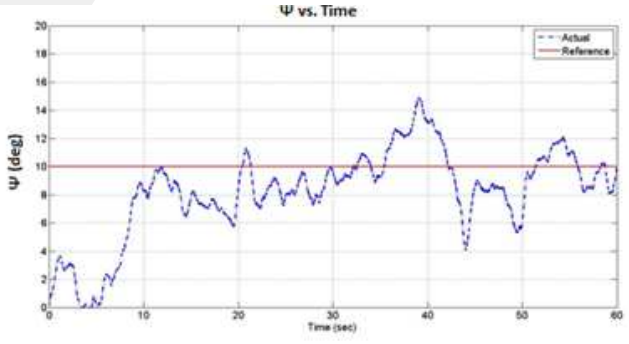

Figure 37: Yaw Angle

After stabilizing the attitude dynamics of the system, altitude tracking has been studied. To be able to detect the targets on floor, the system should be able to fly between 0.6 to 1 meters. This experiment shows the results on altitude tracking referenced to 0.6 meters in Figure 38.

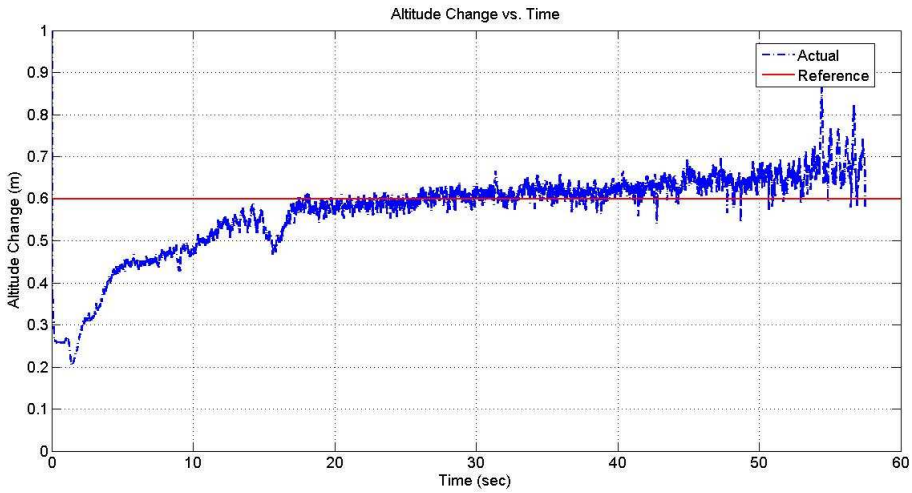

Figure 38: Altitude Tracking referenced to 0.6 meters

System also has been tested with varying reference points. Figure 39 illustrates shots taken on flight test with altitude and attitude control active together.

Figure 39: Shots taken during Flight Tests

It is important to note that system experiments have been done in two different phases. The first one was to maintain attitude and altitude control together, and in the second phase target tracking control has been added to system with LQR controller. In the Figure 40 controller output has been shown.

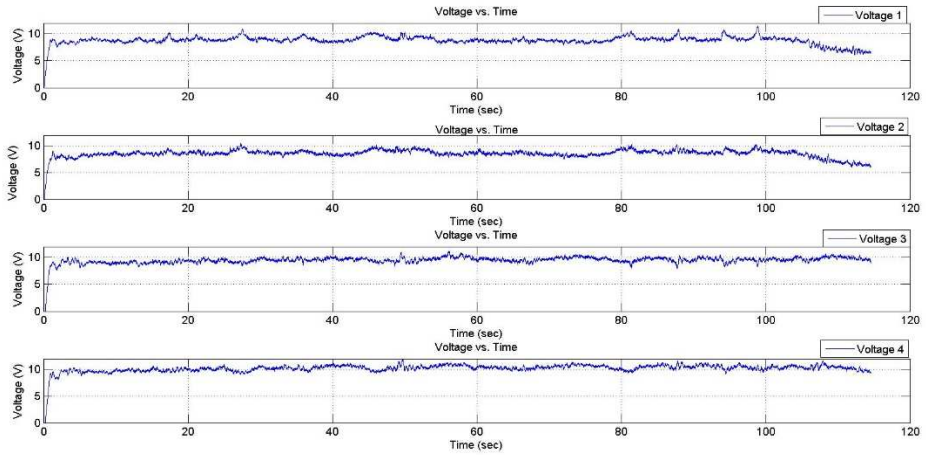

Figure 40: Controller Output

6.1.2 Target Detection and Tracking Control Experiment

As previously stated, for target detection algorithm Aruco has been used. C Code explanation and information about Aruco has been given in previous chapters. Aim is to detect the marker on the ground. Theoretically, it is aimed to detect and track this marker that gives position information about quadrotor's position relative to it. Data that is received from camera has been collected to an another PC which runs Linux operating system and the communication between this PC with control PC has achieved by using UDP protocol. Figure 41 shows how the processing has been completed in Linux operating system. Left part of the figure shows the collected information. On the other hand, right part shows what quadrotor sees at hovering.

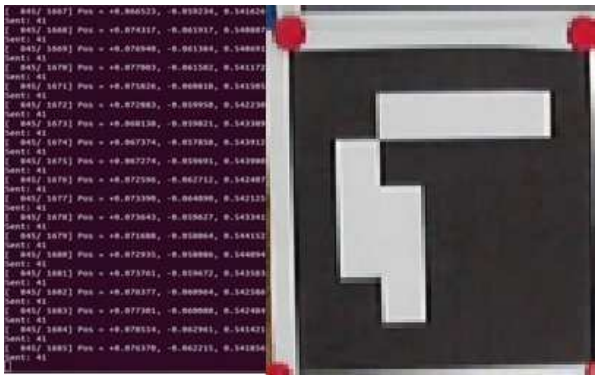

Figure 41: Information Process using Aurco

Test results have indicated that LQR controller needs great effort to be tuned. Therefore, PID controller has been applied to system. Figure 42 illustrates the experimental results of target tracking where error in *x* and *y* axis referenced to zero.

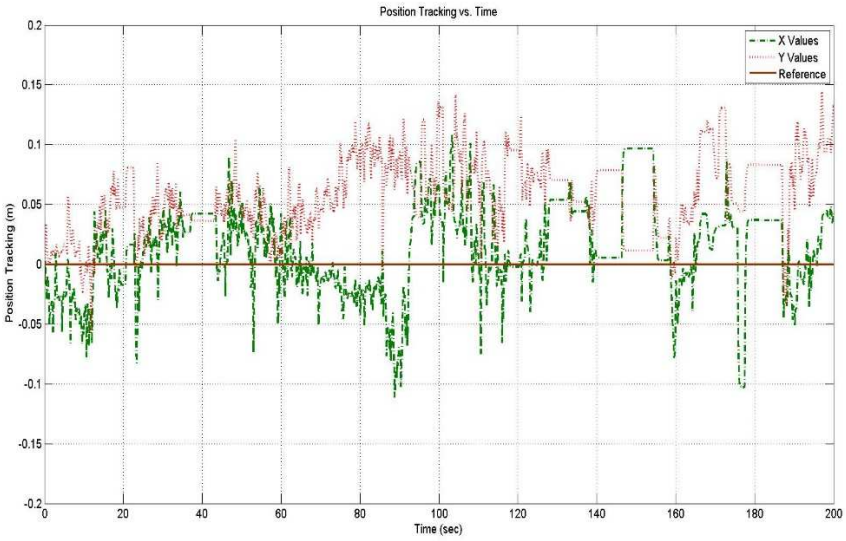

Figure 42: Position Tracking on x and y Axis

As a result, LQR controller was successful at attitude and altitude dynamics. Normalized values for the problem help to achieve better results. However for target tracking, LQR controller was unable to achieve success. PID controller results indicate that system can track a moving object while it tries the keep error to the center of target as zero. However, position change with respect to time has not been used in real time application due to problems for acquiring velocity data. Theory of the PID controller in real time application has been shown in Figure 43. As a future work LQR controller for target tracking could be worked on.

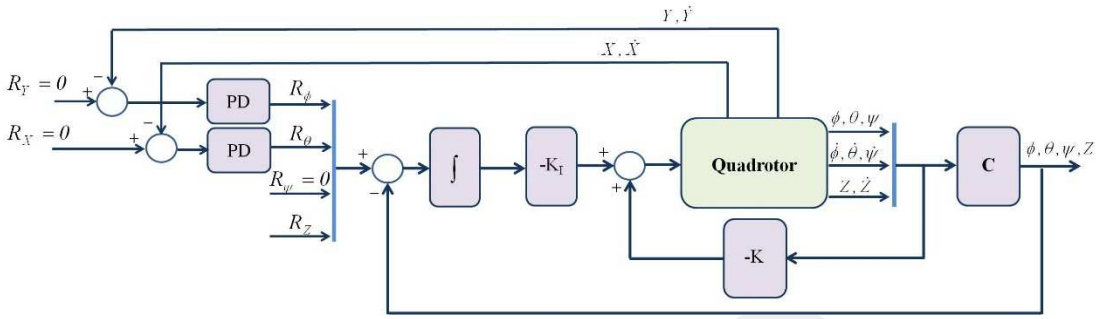

Figure 43: Theory of PID Controller in Real Time Application

CHAPTER 7

DISCUSSIONS AND CONCLUSIONS

In this thesis, RC platform Draganfly, is improved to track a target on the ground. Only the chassis and actuator units of the platform are used. Attitude and altitude controllers are designed by using LQRs. Target tracking problem is also solved by extending the LQR approach. However, tuning part required extreme effort. Instead, target tracking is achieved by providing reference angles for roll and pitch dynamics by an outer PID loop which is comparing the relative position of the quadrotor with respect to the marker.

Attitude and altitude controller is designed using normalized weighting matrices to minimize the tuning period. Designed controllers are implemented on the physical system and similar response with the simulations is achieved with minor tunings. Use of augmented error states makes the system robust to external disturbances such as cables.

In addition to reported design, cause of disturbances and delays has also been studied. It has been observed that the system crashes, cable weights, and lack of an embedded controller adversely affect the system stability and robustness. Due to system crashes, structures of the chassis and parameters have varied with time. This also degrades the IMU data.

As an embedded controller, Texas Instrument's C2000 microcontroller has been considered at this time. Aim was to embed the entire controller, into this card. However, MATLAB's limited access to this card causes observability problems in real-time tests. Nevertheless, controller algorithm that covers IMU information and

sensor reading except camera has been successfully embedded to C2000 in discrete time. Using an embedded control unit will be studied in a future project.

Finally, target tracking algorithm has been tuned for the system. Aruco marker is placed on the ground and given quasi-static motions. It will be placed on the ground vehicle, Peri, and quadrotor will track the moving marker and ground vehicle. Marker may perform more dynamic motions with Peri. Therefore, control architecture should be improved to perform an agile tracking performance. Estimaiton algorithms regarding the dynamics of the marker will be used for this purpose. In this thesis it is just presented that Aruco Markers can be utilized as targets to be tracked by a quadrotor using a simple control algorithm.

REFERENCES

1. “Draganflyer V Ti PRO RC Gyro Stabilized Electric Helicopter with High Res Color Video Camera System”, http://www.rctoys.com/rc-toys-and-parts/DF-VTIPRO/RC-HELICOPTERS.html

2. “A Successful French Helicopter”, Flight Magazine, 24 January 1924 p.47

3. G. Balas, W. Garrard, and J. Reiner “Robust dynamic inversion for control of highly maneuverable aircraft”, Journal of Guidance Control & Dynamics, vol. 18, no. 1, 1995, pp. 18-24.

4. S. H. Lane and R. F. Stengel, “Flight control design using non-linear inverse dynamics”, Automatica, vol. 24, 1988, pp. 471-483.

5. S. Bouabdallah, “Design and Control of Quadrotors With Application to Autonomous Flying”, Université Aboubekr Belkaid, 2007, Thesis

6. A. C. Y. Zhang, “Flatness Based Trajectory Planning/Replaning for a Quadrotor Unmanned Aerial Vehicle”, IEEE Paper, 2012

7. J. Colorado, A. Barrientos, Senior Member, IEEE, A. Martinez, B. Lafaverges, and J. Valente, “Mini-quadrotor Attitude Control based on Hybrid Backstepping & Frenet-Serret Theory”, Robotics and Automation (ICRA), 2010 IEEE International Conference, 3-7 May 2010

8. Hoffmann, G. M., Rajnarayan, D. G., Waslander, S. L., Dostal, D., Jang, J. S. and Tomlin, C. J., “The stanford testbed of autonomous rotorcraft for multi agent control”, Digital Avionics Systems Conference, Vol. 2, 2004, pp. 12.

9. Chen, M. and Huzmezan, M. (2003), “A Simulation Model and $H\infty$ Loop shaping Control of a Quad Rotor Unmanned Air Vehicle”, Proceedings of MS03 Conference, 2003, Palm Springs, California.

10. Balas, C. (2007) “Modeling and Linear Control of a Quadrotor”, Cranfield University, Thesis.

11. Chen, M. and Huzmezan, M. (2003), “A combined MBPC/2 DOF $H\infty$ controller for a quad rotor UAV”, AIAA Atmospheric Flight Mechanics Conference, 2003, Austin, Texas.

12. S.Skogestad and I.Postlethwaite, “Multivariable feedback control analysis and design”, New York, Automatic Control, IEEE Transactions (Volume: 26, Issue: 1), 1996.

13. L.Guzzella, “Discrete-Time Control Systems”, ETH Zurich, 2009, Course Notes.

14. R.Andreas, “Dynamics Identification & Validation, and Position Control for a Quadrotor”, ETH Zuric, 2011, Thesis.

15. J.How, E.King, and Y.Kuwata, “Flight Demonstrations of Cooperative Control for UAV Teams”, Massachusetts Institute of Technology, Paper, 2004.

16. Syed Ali Raza and Wail Gueaieb, “Intelligent Flight Control of an Autonomous Quadrotor”, University of Ottawa, Thesis, 2010, pp.246-264.

17. T.Bresciani, “Modelling, Identification and Control of a Quadrotor Helicopter”, Department of Automatic Control Lund University, Thesis, 2008.

18. S.S. Osder, W.E. Rouse, and L.S. Young, “Navigation, guidance and control systems for V/STOL aircraft”, Sperry Tech. vol. 1, no. 3, 1973.

19. W.T.Higgins JR., “A Comparison of Complementary and Kalman Filtering”, Arizona State University, Paper, 1975.

20. R.G. Brown, “Integrated navigation systems and Kalman filtering: a perspective, Navigation”, J. Inst. Navigation, vol. 19, no. 4, pp. 355-362, Winter 1972-73.

21. R.L.Moses, D.Krishnamurthy, and R.Patterson, “A Self-Localization Method for Wireless Sensor Networks”, Department of Electrical Engineering, Paper, 2002.

22. Abdul Bais, Tobias Deutsch, Gregor Novak, “Comparison of Self-Localization Methods for Soccer Robots”, Industrial Informatics, 2007 5th IEEE International Conference on (Volume:1).

23. A Masselli, S Scherer, and K Wenzel, (2010, May), “Autonomous Flying Robots” , http://www.ra.cs.unituebingen.de/forschung/flyingRobots/welcome_e.html

24. S Lange, P Protzel, and N Sunderhauf, "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments”, Advanced Robotics, 2009, pp. 1-6, 22-26.

25. M.Achtelik, K.Kuhnlenz, Tianguang Zhang, and Ye Kang, “Autonomous hovering of a vision/IMU guided quadrotor”, Mechatronics and Automation, pp. 2870-2875, 9-12, 2009.

26. M.A.O. Méndez, “Soft-Computing Based Visual Control for Unmanned Vehicles”, Universidad Politécnica de Madrid, Thesis, 2013

27. A. Güçlü, “Attitude and Altitude Control of an Outdoor Quadrotor”, MS thesis, Atılım University, Thesis, 2010.

28. “3DM-GX2® -25 Data Communications Protocol”, Datasheet

29. “GP2Y0A02 Sharp IR Sensor”, Datasheet

30. Graham, M., Zook, M., and Boulton, A. “Augmented reality in urban places: contested content and the duplicity of code.” Transactions of the Institute of British Geographers, DOI: 10.1111/j.1475-5661.2012.00539.x 2012.

31. Steuer, Jonathan, “Defining Virtual Reality: Dimensions Determining Telepresence”, Department of Communication, Stanford University, Journal of Communication Volume 42, Issue:4, 15 October 1993.

32. “Introducing Virtual Environments National Center for Supercomputing Applications”, University of Illinois.

33. Chen, Brian X. “If You’re Not Seeing Data, You’re Not Seeing”, Paper, 2009.

34. Maxwell, Kerry. “Augmented Reality, Macmillan Dictionary Buzzword.”, Paper

35. “Augmented reality-Everything about AR, Augmented Reality On.”, Paper

36. Azuma, Ronald. A Survey of Augmented Reality Presence: “Teleoperators and Virtual Environments”, pp. 355–385, August 1997

37. “Pololu High-Power Motor Driver 18v15”, Datasheet

38. “280SA Motor Specifications”, Datasheet

39. G.Carrillo, L.R., D.López, A.E., Lozano, R., Pégard, C., “Quad Rotorcraft Control”, Printed Book, 2013, pp.23-33.

40. D. Küçük, Design of Two Wheeled Twin Rotored Hybrid Robotic Platform, MS thesis, Atılım University, Thesis, 2010.