

AUTOPILOT DESIGN FOR A QUADROTOR

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET SAMI BÜYÜKSARIKULAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2014



Approval of the thesis:

**AUTOPILOT DESIGN FOR A QUADROTOR**

submitted by **MEHMET SAMI BÜYÜKSARIKULAK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. Gönül Turhan Sayan  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Prof. Dr. Kemal Leblebicioğlu  
Supervisor, **Electrical Electrical Engineering Dept., METU** \_\_\_\_\_

**Examining Committee Members:**

Assoc. Prof. Dr. Çağatay Candan  
Electrical Electrical Engineering Department, METU \_\_\_\_\_

Prof. Dr. Kemal Leblebicioğlu  
Electrical Electrical Engineering Department, METU \_\_\_\_\_

Assoc. Prof. Dr. İlkey Yavrucuk  
Aerospace Engineering Department, METU \_\_\_\_\_

Assoc. Prof. Dr. Umut Orguner  
Electrical Electrical Engineering Department, METU \_\_\_\_\_

Dr. Oğuzhan Çifdalöz  
Aselsan INC. \_\_\_\_\_

**Date:** \_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: MEHMET SAMI BÜYÜKSARIKULAK

Signature :

# ABSTRACT

## AUTOPILOT DESIGN FOR A QUADROTOR

Büyüksarıkulak, Mehmet Sami

M.S., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Kemal Leblebicioğlu

December 2014, 96 pages

Nowadays, UAVs (Unmanned Aerial Systems) are important study areas for military and academic world. These systems can play crucial roles in some applications, especially in hazardous environments due to the fact that no human pilot will undergo danger.

Quadrotor is an underactuated UAV system which has four fixed rotors and four degrees of freedom. Roll, pitch and yaw rotations and elevation/declination can be changed by differences of propeller velocity of rotors with respect to each other. Horizontal motion is depended on roll and pitch angle behaviours and can not be controlled independently.

In this thesis, first aim is modelling of a quadrotor which is based on Newton-Euler equations. Different control strategies will be developed which are proportional-integral (PI), linear quadratic regulator (LQR) and sliding mode controller. After that a line of sight (LOS) guidance algorithms is designed to provide the quadrotor to move autonomously using the way points along the desired path. All these proce-

dures are first implemented on a Matlab/Simulink platform. Later on they have been applied on an autopilot board to realize the real system.

Keywords: Quadrotor, System Modelling, PID, LQR, Sliding Mode Control, Guidance, Autopilot

# ÖZ

## KUADROTOR İÇİN OTOPILOT TASARIMI

Büyüksarıkulak, Mehmet Sami  
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü  
Tez Yöneticisi : Prof. Dr. Kemal Leblebicioğlu

Aralık 2014 , 96 sayfa

Günümüzde insansız hava araçları (İHA lar) hem akademik hem de askeri dünyada önemli bir araştırma konusudur. Bu sistemler insan pilot içermediği için, özellikle tehlikeli ortamlarda, kritik roller üstlenmektedirler.

Kuadro rotor dört sabit rotora sahip bir insansız hava aracıdır. Yalpa, yunuslama ve yönelme açıları değişimi ile yükselme-alçalma hareketleri sabit rotorların hızlarını birbirine göre değişimiyle düzenlenebilmekte yatay eksenindeki hareketleri ise yalpa ve yunuslama açılarının değişime bağlı olarak meydana gelmektedir.

Bu tezde ilk amaç kuadro rotorun Newton-Euler denklemleri ile detaylı bir modelinin çıkarılmasıdır. Ardından uçuş için gerekli oransal integral (PI), doğrusal ikinci dereceden regülatör (LQR) ve kayma modlu tipinde kontrolcüler tasarlanacaktır. Daha sonrasında görüş hattı (LOS) yapısında bir güdüm algoritması ile sistem otonom hale getirilecektir. Yapılan bu işlemler öncelikle Matlab/Simulink ortamında gerçekleştirilecek, sonrasında otopilot kartında gerçek sistem üzerine uygulanacaktır.

Anahtar Kelimeler: Quadrorotor, Sistem Modelleme, Doğrusal Kontrolörler, PID, LQR,  
Kayan Mod Kontrolcü, Güzüm, Otopilot



*To my family*

## ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Kemal Leblebiciođlu for his constant support. He also influenced me highly in scientific context. I also would like to thank Doctor Ođuzhan iftdalöz for his support and guidance.

This work is funded by ASELSAN Inc. M. Özgür Ateşođlu supplied a lot of important materials to construct the system, Serkan Avgören helped me during the thesis and provided test setups. Talha Korkmaz's support made it possible that in a short time I were able to build the frame of this work.

Aydın Pehlivanlı, Ersan Aktaş, Faruk Yurtsever and Hasan Hamzaebi spent their valuable time with me to software implementations. I am grateful to both of them.

Burak Seymen and Burak Krk discussed with me during the study, they influenced and supported this work scientifically and their contribution were most valuable for me.

There are a lot of people that were with me during this thesis. I want to thank to them individually; Mustafa Karakurt, Ömer Nuzumlali, Eren Kahraman, ađrı Tepe, Bora Eser Kart, Ufuk Irmak, ađlar Uzun, Ahmet Sayman, and Onur Ertin.

# TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vii
ACKNOWLEDGMENTS . . . . .	x
TABLE OF CONTENTS . . . . .	xi
LIST OF TABLES . . . . .	xv
LIST OF FIGURES . . . . .	xvi
LIST OF ABBREVIATIONS . . . . .	xx
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Unmanned Vehicles (UVs) . . . . .	1
1.2 Autopilot Systems . . . . .	2
1.3 Quadrotors . . . . .	3
1.4 History of Quadrotor . . . . .	5
1.5 Motivation . . . . .	7
1.6 Aim and Scope . . . . .	8
1.7 Literature Review . . . . .	8

1.8	Outline of the Thesis . . . . .	9
2	MODELLING OF A QUADROTOR . . . . .	11
2.1	Introduction . . . . .	11
2.2	Reference Frames . . . . .	11
2.2.1	Body Frame . . . . .	12
2.2.2	Earth Frame (NED) . . . . .	12
2.2.3	Geodetic and ECEF Coordinate System . . . . .	13
2.3	Quadrotor Kinematics . . . . .	13
2.4	Quadrotor Dynamics . . . . .	17
2.4.1	Newton-Euler Equations . . . . .	17
2.4.2	The Forces and Torques . . . . .	19
2.4.2.1	Gravity Effect . . . . .	20
2.4.2.2	Thrust Force . . . . .	20
2.4.2.3	Roll and Pitch Moment . . . . .	22
2.4.2.4	Yaw Moment . . . . .	23
2.4.2.5	Propeller Gyro Effect . . . . .	25
2.4.2.6	Other Aerodynamical Effects . . . . .	26
2.5	State Variables and Equations . . . . .	27
2.6	Linearization of the System . . . . .	29
2.6.1	Equilibrium Points (Trim States) . . . . .	29
2.6.2	Obtaining the State Matrices . . . . .	34

3	CONTROLLER DESIGN . . . . .	37
3.1	Introduction . . . . .	37
3.2	PI Controller . . . . .	37
3.2.1	PI Controller of the Quadrotor . . . . .	38
3.2.1.1	Roll & Pitch Rate Control . . . . .	39
3.2.1.2	Roll & Pitch Control . . . . .	41
3.2.1.3	Horizontal Velocities Control . . . . .	42
3.2.1.4	Vertical Velocity Control . . . . .	43
3.2.1.5	Yaw Rate and Yaw Angle Control . . . . .	44
3.3	Linear Quadratic Regulator (LQR) . . . . .	47
3.3.1	LQR Design for Quadrotor . . . . .	49
3.4	Sliding Mode Control . . . . .	53
3.4.1	Sliding Mode Control of Quadrotor . . . . .	58
3.5	Comparison of the Designed Controllers . . . . .	62
4	GUIDANCE . . . . .	65
4.1	Introduction . . . . .	65
4.2	Line of Sight (LOS) Guidance . . . . .	66
4.3	LOS Guidance of a Quadrotor . . . . .	67
5	SYSTEM HARDWARE & SOFTWARE . . . . .	73
5.1	Introduction . . . . .	73
5.2	Hardware Components . . . . .	73

5.2.1	Auto Pilot Board . . . . .	73
5.2.2	Quadrotor Frame . . . . .	74
5.2.3	Magnetometer (Compass) . . . . .	75
5.2.4	GPS . . . . .	75
5.2.5	Motors & Propellers . . . . .	76
5.2.6	RC Transmitter & Receiver . . . . .	77
5.2.7	Electronic Speed Controller . . . . .	77
5.2.8	Wireless Telemetry . . . . .	78
5.2.9	Battery . . . . .	78
5.2.10	Other Components . . . . .	79
5.3	Software . . . . .	80
6	HARDWARE IMPLEMENTATIONS . . . . .	81
7	CONCLUSION . . . . .	89
	REFERENCES . . . . .	93

## **LIST OF TABLES**

Table 3.1	Comparison of the Controllers with Different Performance Indices	. 64
-----------	--	------

## LIST OF FIGURES

### FIGURES

Figure 1.1	A Scheme of NGC System . . . . .	3
Figure 1.2	Quadrotor . . . . .	4
Figure 1.3	+ and X Quadrotor Configurations . . . . .	5
Figure 1.4	Richet's Gyroplane n:01 . . . . .	6
Figure 1.5	Convertawings Model A . . . . .	6
Figure 2.1	Earth and Body Frame of Quadrotor . . . . .	12
Figure 2.2	Euler Angles . . . . .	16
Figure 2.3	Effectuated Forces and Torques on the Quadrotor . . . . .	22
Figure 2.4	Pitch Moment Generation . . . . .	23
Figure 2.5	Yaw Moment . . . . .	24
Figure 2.6	Propeller Gyro Effect . . . . .	25
Figure 2.7	Pitch Angle for Different Trimmed States . . . . .	31
Figure 2.8	Roll Angle for Different Trimmed States . . . . .	32
Figure 2.9	Thrust for Different Trimmed States . . . . .	32
Figure 2.10	Roll Moment for Different Trimmed States . . . . .	33
Figure 2.11	Pitch Moment for Different Trimmed States . . . . .	33



Figure 3.1	PI+PreFilter Control Structure . . . . .	38
Figure 3.2	Roll Rate Step Response . . . . .	40
Figure 3.3	Closed Loop Roll Rate Bode Plot . . . . .	41
Figure 3.4	Roll Dynamics of the System . . . . .	41
Figure 3.5	Roll Angle Step Response . . . . .	42
Figure 3.6	$v$ Step Response . . . . .	43
Figure 3.7	Vertical Speed Closed Loop Structure . . . . .	44
Figure 3.8	$w$ Step Response . . . . .	44
Figure 3.9	Yaw Rate( $r$ ) Step Response . . . . .	45
Figure 3.10	Yaw Angle Step Response . . . . .	45
Figure 3.11	PI Cascade Loops for Quadrotor Autopilot . . . . .	46
Figure 3.12	LQR Design for Tracking(Type 1) . . . . .	49
Figure 3.13	$v$ Step Response . . . . .	52
Figure 3.14	$w$ Step Response . . . . .	52
Figure 3.15	Yaw Angle Step Response . . . . .	52
Figure 3.16	State Trajectories and Sliding Surface . . . . .	54
Figure 3.17	Chattering . . . . .	57
Figure 3.18	Signum & Saturation Function . . . . .	57
Figure 3.19	One Dimension Phase Portrait (Phase Line) . . . . .	59
Figure 3.20	$v$ Step Response . . . . .	61
Figure 3.21	$w$ Step Response . . . . .	61
Figure 3.22	$\psi$ Step Response . . . . .	61

Figure 3.23 $U_2$ Control Signal . . . . .	63
Figure 3.24 $U_1$ Control Signal . . . . .	63
Figure 3.25 $U_4$ Control Signal . . . . .	63
Figure 4.1 Way Point Guidance . . . . .	66
Figure 4.2 LOS Guidance of a Quadrotor . . . . .	67
Figure 4.3 Velocity Vector and Reference Heading ( $\psi$ ) Angle . . . . .	69
Figure 4.4 Yaw Angle Correction . . . . .	70
Figure 4.5 Delta Yaw vs. Forward Speed ( $\kappa = 0.5, u = 3m/s$ ) . . . . .	70
Figure 4.6 Guidance of Quadrotor . . . . .	71
Figure 4.7 Simulink Model of Autonomous System . . . . .	72
Figure 5.1 APM 2.6 . . . . .	74
Figure 5.2 Three Axis Magnetometer . . . . .	75
Figure 5.3 Connection of GPS to APM 2.6 . . . . .	76
Figure 5.4 RC Transmitter & Receiver . . . . .	77
Figure 5.5 Connection of Wireless Telemetry Kit to APM 2.6 . . . . .	78
Figure 5.6 Fully Assembled Quadrotor System . . . . .	79
Figure 5.7 Arduino IDE . . . . .	80
Figure 6.1 Test Setup for Attitude Control . . . . .	82
Figure 6.2 Motor Inversion Block of the Autopilot . . . . .	83
Figure 6.3 PID Implementation of the Autopilot . . . . .	85
Figure 6.4 $\phi = 0$ Response of the Real System . . . . .	86

Figure 6.5 $\theta = 0$ Response of the Real System . . . . .	86
Figure 6.6 $\dot{\psi} = 0$ Response of the Real System . . . . .	86
Figure 6.7 Disturbance Rejection of the Real System . . . . .	87
Figure 6.8 Roll Angle Tracking of the Real System . . . . .	87
Figure 6.9 Pitch Angle Tracking of the Real System . . . . .	88
Figure 6.10 Yaw Rate Tracking of the Real System . . . . .	88

## LIST OF ABBREVIATIONS

$x$	Earth Frame x Axis Position
$y$	Earth Frame y Axis Position
$z$	Earth Frame z Axis Position
$u$	Body Frame x Axis Velocity
$v$	Body Frame y Axis Velocity
$w$	Body Frame z Axis Velocity
$\phi$	Euler Roll Angle
$\theta$	Euler Pitch Angle
$\psi$	Euler Yaw Angle
$p$	Body Frame x Angular Rate
$q$	Body Frame y Angular Rate
$r$	Body Frame z Angular Rate
$R$	Rotation Matrix (Direction Cosine Matrix)
$p$	Momentum Vector of Body
$h$	Angular Momentum Vector of Body
$F$	Force on Body Frame
$t$	Time
$G$	Weight
$g$	Earth Gravity Constant
$m$	Mass
$l$	Length of Arm
$T$	Thrust Vector
$D$	Drag moment
$C_D$	Drag Coefficient
$J_{xx}, J_{yy}, J_{zz}$	Inertial Components
$J_r$	Rotor Inertia
$\omega$	Propeller Rotation Speed
$R$	Propeller Diameter

$d$	Drag Factor
$b$	Thrust Factor
$F_D$	Air Friction
$C_D$	Drag Coefficient
$A_w$	Frontal Area (Wetted Area)
$\rho$	Air density
$\mu$	Advance Ratio
$\lambda$	Inflow Ratio
$\sigma$	Rotor Solidity
$a$	lift scope
$C_T$	Thrust Coefficient
$C_Q$	Drag Coefficient
$A$	Propeller Sweeping Area
$T_{IGE}$	Thrust in Ground Effect Region
$T_{OGE}$	Thrust out of Ground Effect Region
$U_1$	Total Thrust
$U_2$	Roll Moment
$U_3$	Pitch Moment
$U_4$	Yaw Moment
$A$	State Matrix
$B$	Input Matrix
$C$	Output Matrix
$D$	Feedforward Matrix
$PID$	Proportional-Integral-Derivative
$PI$	Proportional-Integral
$\omega_n$	Natural Frequency
$\zeta$	Damping Ratio
$B_\omega$	Bandwidth
$K_I$	Integral Gain
$K_P$	Proportional Gain
$LQR$	Linear Quadratic Regulator
$K$	State Feedback Vector
$J$	Cost Function
ARE	Algebraic Riccati Equation

SMC	Sliding Mode Control
VSC	Variable Structure Control
LOS	Line of Sight
NGC	Navigation Guidance Control
WP	Way Point
$R_{acc}$	Region of Acceptance
APM	Ardu Pilot Mega
UV	Unmanned Vehicle
UAV	Unmanned Aerial Vehicle
GPS	Global Positionin System
ESC	Electronic Speed Controller
IMU	Inertial Measurement Unit
MEMS	Micro Electro Mechanical System
ECEF	Earth Centered Earth Fixed Coordinate System
NED	North East Down Coordinate System
BLDC	Brushless Direct Current
DC	Direct Current
RPM	Rotation Per Minute
BLDC	Brushless DC Motor
BEC	Battery Eliminator Circuit
RC	Radio Controlled
PWM	Pulse Width Modulation
LiPo	Lithium Polymer
NiCad	Nickel Cadmium
NiHM	Nickel Methal Hydride

# CHAPTER 1

## INTRODUCTION

### 1.1 Unmanned Vehicles (UVs)

An unmanned vehicle (UV) is a vehicle which does not carry a human pilot on board. It may be remote controlled, semi-autonomous or autonomous. The autonomous system has a guidance-navigation-control system (GNC) so that it senses the environment and steers on its own. A pre-defined mission can be completed by the vehicle without human intervention.

Unmanned vehicles are categorized in several group depending on the types of application such as unmanned surface vehicles (USVs), unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs) and unmanned underwater vehicles (UUVs).

The application areas of UVs have been increasing for the last few decades. Although, military applications have led to UV developments, today they are used for many purposes. The primary purpose of UVs are investigation of hazardous places like battleship, forest fires, nuclear/chemical leakage areas etc. Governments use UAVs for power and pipeline security, custom/border patrol, rescue operation and social event observations. Moreover, the scientific research, inspection of space and ocean/ sea floor, logistics and transportation are other usage areas of UVs. Recently UAVs have become popular among civilians to be used for aerial photography and recreational activities.

UAVs are the one of the most significant type of UV system. They do not need a human pilot. Therefore, a cockpit and various life support units (pressure, temperature

controllers, launch seats) are not found on board. All these equipments and systems mean extra weight and expense for the system. So that, UAV are more efficient and inexpensive than pilot controlled system. Furthermore, g effects on the pilot limit the system's maneuverability, however UAVs perform acrobatics without considering human physiology.

The small size UAV systems have the electric motors that work silently, when compared to the internal combustion engines. This feature makes them suitable for surveillance systems because they are barely noticed by the foe.

The UAV systems are a growing family, there exists many types which are mainly classified with respect to flight ranges and altitudes [1].

## **1.2 Autopilot Systems**

An autopilot that can be considered as the brain of the vehicle carries out all autonomous mission requirements. During the flight, the autopilot decides on all actions on his own in order to achieve system objectives. The autopilot ensures that the UAVs can autonomously take off and land. Furthermore, the autopilot can find optimal path and avoid from the obstacles.

The components of an autopilot system are given in Figure 1.1. The flight management is usually managed by people before or during the operation which contains mission objectives, routes etc. The navigation system informs about the system's attitude and orientation with the use of sensors such as GPS, IMU, pitot, altimeter, compass and navigation/ filtering algorithm. The guidance gets information from the navigation system and determines the vehicle trajectory that should be followed. The control system performs according to the guidance system commands. It drives the system actuator for a proper motion to follow the desired trajectory.



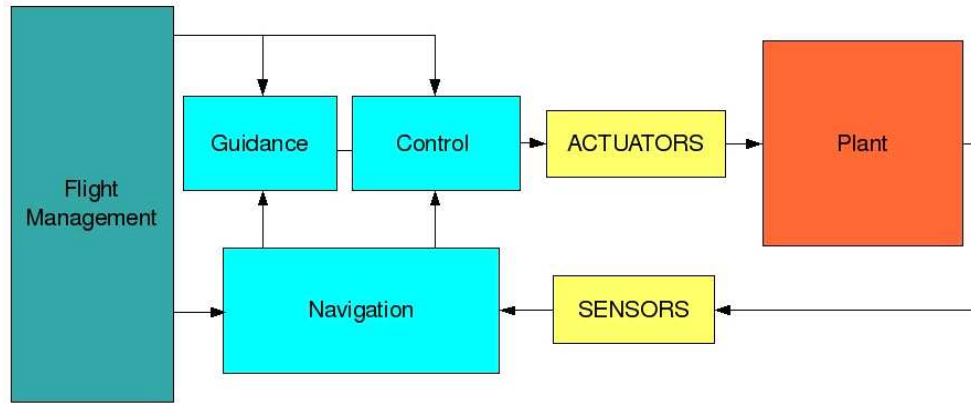


Figure 1.1: A Scheme of NGC System

### 1.3 Quadrotors

UAVs can be categorized into two groups; fixed wings and rotor craft systems. A fixed wing vehicle has long range and can fly a very long time (relatively). Moreover, they have high payload capacity. On the other hand, the rotor craft systems have good manoeuvre capabilities; they can hover, land and take off vertically.

Quadrotor is a fixed rotor craft system which has four identical propellers. It does not have swash plate mechanism to change the angle of attack to move into three dimensional spaces. The quadrotor can change only rotational speed of propellers to control thrust and torque.

Quadrotors have some advantages over helicopters; They do not contain complex mechanical linkages (swash plate) to change the pitch angle of rotor. Their individual propellers are smaller than equivalent helicopter propellers. This decreases the damage if the rotor hit something. They can fly with lower risk at challenging environments.

Main components of a quadrotor are frame, motors and propellers. The frame carries controller, sensor, power source and communication gadgets. Motor and propeller are located on the tip of the frame arm.

In the quadrotor, the fixed rotors generate thrust vector which is upward through the ground. When the quadrotor increases the same amount of the rotational speed of propellers, the vehicle elevates. If the thrust vector is equal to the vehicle weight, it preserves its own altitude.

The attitude behaviour of the system is controlled by the roll, pitch and yaw moments. A positive roll angle is obtained when the left side propellers' speeds are increased and right side propellers' speeds are decreased. It causes the roll moment which makes the system turn around  $X$  axis. A pitch angle is generated by speed differences between front and backside propellers and the pitch moment is obtained with same manner as well. In a quadrotor, each successive propeller rotates in opposite direction to remove their drag moment. The two sets of counter rotating blades balance the system along  $Z$  axis. The quadrotor can generate an imbalance to obtain a yaw angle with change of propellers' rotational speed.

The system is an underactuated system, which refers to the fact that the number of the degrees of freedom is more than the inputs of the system. The quadrotor has six degrees of freedom but have only four rotors. Therefore, two degrees of freedom depends on the others. When the tilt angles (the roll and pitch angle) are changed, a horizontal component of the thrust vector is obtained, which makes it possible to move the system through the  $X - Y$  plane. As a result, the motion along  $X$  direction depends on the pitch angle and the motion along  $Y$  axis depends on the roll angle.



Figure 1.2: Quadrotor

For a quadrotor with four rotors, two design configurations are available, which are the plus (+) and X configuration. The thrust and yaw dynamics are similar for both whereas the roll and pitch dynamics are different. In other words, in plus configuration, only the two associated motors are responsible for generating roll and pitch moment but all four motors are responsible for generating tilt moments in X configuration,. The X frame is more convenient for mounting a camera than + frame. Because, the camera is usually placed to see the front of the vehicle and camera's field of view may be curtained by front propeller. Thus, X frame becomes more convenient for aerial photography and observation.

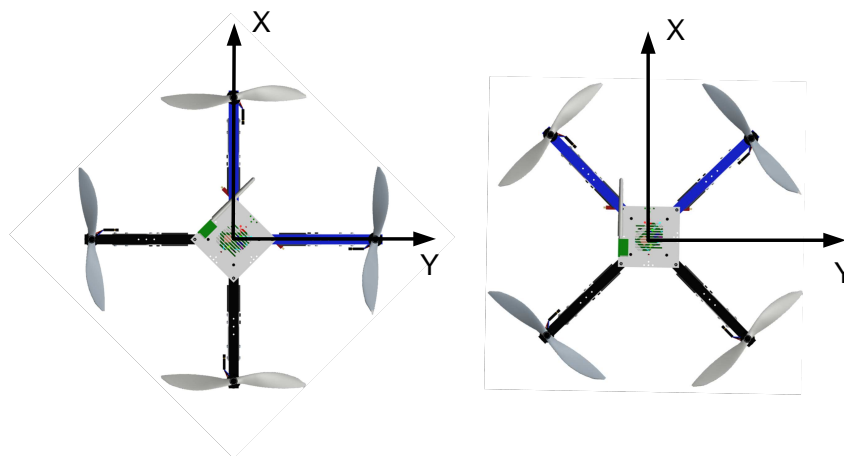


Figure 1.3: + and X Quadrotor Configurations

#### 1.4 History of Quadrotor

The history of quadrotor starts with the beginning of the 20<sup>th</sup> century. French scientist, Charles Richet, built an unmanned helicopter which never succeeded in flying. Loius Breguet, Richet's student, carried on his study and constructed first quadrotor "Gyroplane n:01" in 1907. It has 8.1 meters propellers and 578 kilograms. He used one 50 Hp internal combustion motor to drive four propellers. The system elevated 1.5 meters and it is controlled by people on the ground to stabilize it [2].

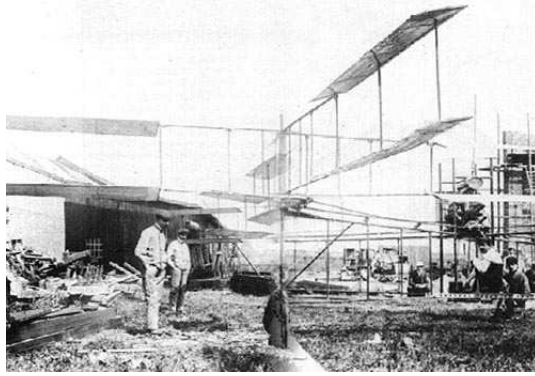


Figure 1.4: Richet's Gyroplane n:01

Etienne Oemichen is another engineer who targeted the vertical flight. He designed several machines in 1920s. In the 14<sup>th</sup> of May 1924, his machine flew for fourteen minutes and it took more than a mile [3]. At the same time, Dr. George de Bothezat and Ivan Jerome studied on a VTOL vehicle in the USA. Their work was finalized with the result of a quadrotor with 1678 kilograms, 9 meters arms and four 8.1 meters six-blade rotors . They managed over one hundred flights but the vehicle only lifted 5 meters in height [6].

Prototype of "Convertawings Model A" was completed in 1956. Despite successful testing and development, the military support for the prototype ceased after cutbacks in defense [7].

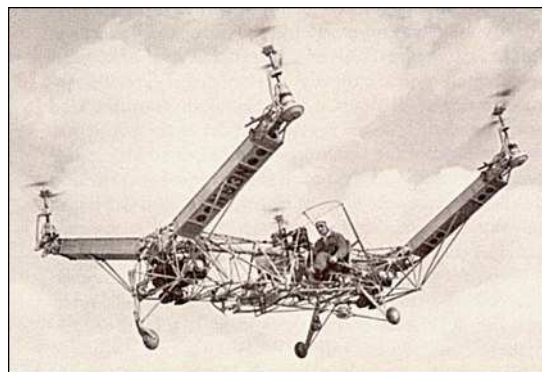


Figure 1.5: Convertawings Model A

Bell Boeing V-22 Osprey has tilt rotors which are combined with both functionality of helicopter and plane. It can vertically take off and land, carry heavy payloads and fly with high velocity [8].

Recently, the small size quadrotors have become a subject for the UAVs. With their small size and agile maneuverability, these vehicles can fly indoors and outdoors.

Nowadays, the researches have focused on multi vehicle communication and environmental exploration. Due to small payload ability of quadrotor, cooperating multi quadrotors will be expected to achieve many tasks with their higher maneuverability.

AeroQuad and Arducopter are open source systems and give the change to people develop their own autopilots. Even amateurs can construct their own quadrotor owing to cheap electronic and light design materials.

## **1.5 Motivation**

The UAVs have been the center of attraction for the last few decades. Great progresses have been achieved in this period and today they have different capabilities and sizes depending on various applications which are described at earlier parts of this section. However, their potential has not been fully reached yet, much needs to be done to advance the UAV technology. Low cost MEMS sensors have to be improved which are less accurate when they are compared to conventional sensors. It is obvious that in the future, MEMS technology will provide more reliable and cheap sensors. The sensors with low level bias and noise will increase system accuracy and capability.

The quadrotor is a hand-held size UAV system. The cost and size smallness and mechanical simplicity of it give a chance to operate the system in a laboratory. In addition, the power requirement of them is relatively low and the test equipments can be designed with an affordable budget. All of them makes the quadrotor a useful platform to develop new ideas for flight control, indoor- outdoor navigation etc. The designed algorithms can be experimented easily and quickly.

APM 2.6 which is open-source autopilot board which contains all necessary sensor equipments. Around the world, numerous users verify the hardware's reliability. Fur-

thermore, owing to the ready-to-fly kits, construction of the system does not take too much time. The system can be compounded easily and all mechanical manufacturing procedures are eliminated. Therefore, the researchers have much more time to focus on autopilot implementations.

## **1.6 Aim and Scope**

The objective of this thesis is an autopilot design for a quadrotor. First of all, the mathematical model of the system is derived in detail. After the linearization procedure, several control methods are designed and simulated in the MATLAB/ Simulink platform. Then, a guidance algorithm is constructed which ensures that the quadrotor flies automatically. Later on, the designed controllers are implemented on the quadrotor hardware. A ready-to-fly kit is used to build the setup and APM 2.6 is selected as an autopilot. The autopilot codes are written with help of Arduino IDE. The attitude and altitude controllers are designed initially. So, the system can be remote controlled by a human operator. Then, GPS and inertial navigation system outputs are fused together and the linear speed controllers are designed. Finally, the guidance algorithm is implemented on the real system. Hence, the vehicle becomes autonomous and is able to fly between way points.

## **1.7 Literature Review**

In the last decade, the quadrotor surveys have always been a special attraction point for many researchers. In the literature, the modelling of the system and aerodynamic effects are usually represented with simple dynamical equations, because the quadrotor is relatively a small aircraft and its linear velocities are not so many. The control strategies are divided into many methods such as classic control (PI, PD or PID) [42], [34], [26], [33]; linear quadratic regulator [43], [49], [27], [19]; feedback linearization [17]; backstepping [40], [16] and sliding mode control [50], [53], [25], [24].

Some papers mainly focus on the rotor performance and the thrust-rotation speed relations. These studies investigate the thrust behavior for the forward and vertical flight

[38], [46], [44]. Furthermore, Mahony and Bangura expressed the drag like effects on the quadrotor. They presented a dynamic and aerodynamic model for a quadrotor which makes agile and complex manoeuvres [20]. Fay revealed aerodynamic effects in a mathematical way at Mesicopter project [36].

Wierama proposed to combine three gyroscopes, three accelerometers and three magnetometers of an IMU with infrared sensors to achieve accurate state estimation for indoor application. The nonlinear measurement and state equations were obtained and an Extended Kalman filter were used to estimate the states [52].

Cutler studied on a variable pitch control quadrotor. His study is rare in the quadrotor researches, because most studies focus on fixed pitch structure. He revealed their agilities and explained that how the variable pitch quadrotor overcame the system limitation of the fixed pitch quadrotor [32].

## 1.8 Outline of the Thesis

The contents of the next chapters are summarized here

- **Chapter 2.** First of all, a brief information about coordinate frames will be given. Secondly, an extended quadrotor model will be introduced and the working principles of the system will be mentioned. The model equation will be linearized around the hover condition in order to design the controller. Finally, the other trimmed states will be acquired and compared with the hover condition states.
- **Chapter 3.** This chapter will explain the design steps of the controllers. Two linear (PI, LQR) and one nonlinear (sliding mode ) controller will be proposed. Later on, all the simulation results will be given and the controller performance will be compared with each others.
- **Chapter 4.** The main concept of line of sight guidance will be introduced and an algorithm will be formed for the autopilot. The guidance algorithm will be adjusted to achieve more smooth motion by defining some additional features.

- **Chapter 5.** The technical specifications about the hardware and software components will be given.
- **Chapter 6.** Design procedures of the hardware implementation will be explained and the test results will be given and commented on
- **Chapter 7.** In the conclusion part, the contribution of the thesis will be given. There will be comments about simulation and test results. Also, the future works will be listed.



## CHAPTER 2

### MODELLING OF A QUADROTOR

#### 2.1 Introduction

This chapter presents all the modelling steps of the system. First of all, a brief summary will be given about coordinate frames. Secondly the kinematic and dynamic equations will be derived with the use of Newton-Euler method. After that, the state equations of the system will be written and linearization procedure will be applied to design linear controllers. The behaviour of the system will be investigated under different trim conditions.

#### 2.2 Reference Frames

The quadrotor has many sensors and mission requirements. Each mission and sensor output is based on different coordinate frames. Therefore, one coordinate frame is not sufficient to represent the quadrotor's motion. For example, an accelerator and gyroscope give outputs according to the body frame, but a magnetometer gives output with respect to the Earth frame. Furthermore, GPS's outputs are based on the geodetic frame. The reference speeds are commanded with respect to the body frame. However, the way points are defined with respect to the Earth frame.

All equations must be expressed in the same frame in the mathematical model, so transformations must be performed between the coordinate frames [21].

### 2.2.1 Body Frame

This body frame is linked to the quadrotor and the origin is located at the center of gravity of the vehicle. It is previously mentioned that the quadrotor is designed according to  $X$  configuration. Therefore, the  $X$  axis of body frame extends to forward direction of quadrotor between the first and third arms, the  $Z$  axis points downward, which is oriented to gravity.  $Y$  axis is formed within a right handed orthogonal set, right side of the system.

### 2.2.2 Earth Frame (NED)

North-East-Down coordinate system is used to represent the Earth frame that is commonly preferred in aviation. The origin of the frame can be chosen arbitrary according to any constant point on the earth, however vehicle's landing point is generally selected as the origin. The  $Z$  axis is oriented through direction of the gravitational force. The  $X$  axis is directed north,  $Y$  axis stretches out to east. Due to the system's low speed, the Earth is assumed as fixed (non-rotating) and flat [22].

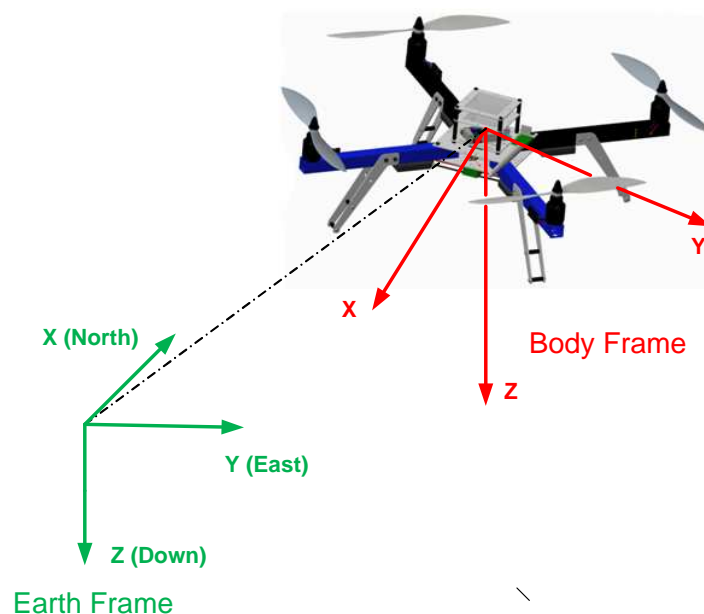


Figure 2.1: Earth and Body Frame of Quadrotor

### 2.2.3 Geodetic and ECEF Coordinate System

The geodetic coordinate system is used in GPS based navigation system. It is not a usual Cartesian system, which has not orthogonal components; instead longitude, latitude and height (altitude) are used to express a coordinate.

In the Earth Centered Earth Fixed (ECEF) coordinate system, the origin of the system is located at the center of earth, the  $Z$  axis points to north pole, the  $X$  axis intersects the earth surface at the equator and  $0^\circ$  longitude. The frame rotates together with the earth.

ECEF coordinates are used as the intermediate step to convert a position from the geodetic coordinates to the Earth frame coordinates. The outputs of GPS are in the geodetic coordinate system, so they are transformed into ECEF frame according to the world shape model which is based on WGS 84 (world geodetic system 84). Thereafter, the ECEF coordinates are converted to earth frame coordinates. It is necessary for the navigation algorithm of the autopilot system. Further information can be found in [28].

### 2.3 Quadrotor Kinematics

Kinematics is a fundamental subject which studies the motion of objects without considering the forces and moments acting on them. The transformation of coordinate systems is the interest of kinematics. There are two coordinate frames for the model of quadrotor and these are the earth frame and the body frame.

Before the dynamic equations are written, the states are introduced. There are twelve states to express the quadrotor motion. The components of the state vector are linear position and velocity, angular rates and each Euler angle defined at different frames. They are listed according to the associated frames.

- Quadrotor position with respect to earth frame is expressed as  $[x \ y \ z]^T$
- Linear velocities are given with respect to body frame as  $[u \ v \ w]^T$ .
- Angular rates of the body frame are  $[p \ q \ r]^T$

- Euler angles which are called roll, pitch and yaw angle are shown with  $[\phi \ \theta \ \psi]^T$

Assume that  $\vec{V}^b$  is a vector which is expressed in the body frame. When  $\vec{V}^b$  is multiplied with a rotation matrix ( $R_b^e$ ) which is also known as direction cosine matrix (DCM), the result of  $\vec{V}^e$  is the same vector with  $\vec{V}^b$  but it is expressed in the earth frame.

$$\vec{V}^e = R_b^e \vec{V}^b \quad (2.1)$$

The inverse of the transformation matrix is equal to its transpose. So, a vector in the earth frame can be expressed in the body frame with the use of  $R_b^e$ 's transpose.

$$(R_b^e)^{-1} = (R_b^e)^T = (R_e^b) \quad (2.2)$$

To express the relationship between the earth and the body frame, three elemental rotation matrices are used in a sequence. The orientation of the quadrotor is defined by Tait-Bryan angles. They are mostly confused with Euler angles, however, when Euler angles are used, the first and third rotations are made along the same axes, but in Tait-Bryan angle, each axis rotates once [22]. Tait-Byran angles notation is used but the angles are called as Euler angles in this paper.

In order to transform the body frame to the Earth frame, first of all, the body frame is rotated about  $X$  axis by an amount of  $\phi$  (roll) angle and the resultant frame is called  $v_1$  which has the same  $X$  axis, but its  $Y$  and  $Z$  axes are rotated by an amount of roll angle. Then,  $v_1$  frame is rotated about its own  $Y$  axis by an amount of  $\theta$  (pitch) angle and the resultant frame is  $v_2$ . At the last step,  $v_2$  frame is rotated about its own  $Z$  axis by an amount of  $\psi$  (yaw) angle and the resultant frame is the Earth frame.

The rotation matrices;

$$R_b^{v_1}(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$R_{v_1}^{v_2}(y, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.3)$$

$$R_{v_2}^e(z, \psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

After three sequential rotations, the transformation between the body frame to the Earth frame is given by;

$$R_b^e(\phi, \theta, \psi) = R_{v_2}^e(z, \psi)R_{v_1}^{v_2}(y, \theta)R_b^{v_1}(x, \phi)$$

$$R_b^e(\phi, \theta, \psi) = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \quad (2.4)$$

where  $s(\phi) \triangleq \sin(\phi)$  and  $c(\phi) \triangleq \cos(\phi)$

The relationship between the linear velocity of the body frame and the linear positions with respect to the Earth frame is expressed by rotation matrix in the following:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_b^e(\phi, \theta, \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.5)$$

Euler angles ( $[\phi \ \theta \ \psi]^T$ ) are defined in different frames which are  $v_1$ ,  $v_2$  and earth frame respectively. Therefore, the procedure that transforms angular rate vector  $[p \ q \ r]^T$  to Euler angles are little bit complicated. The  $T(\phi, \theta, \psi)$  matrix is used to explain this relationship. Consider the angular rate defined in body frame, which is expressed in the Earth frame by multiplying with the inverse of  $T(\phi, \theta, \psi)$ .

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T^{-1}(\phi, \theta, \psi) \frac{d}{dt} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (2.6)$$

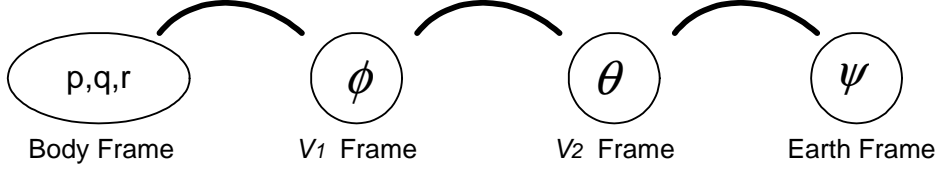


Figure 2.2: Euler Angles

Each angle should be transformed frame by frame to the body frame. In order to define the roll angle rate ( $\dot{\phi}$ ) into the body frame, only one rotation is sufficient. For the pitch angle rate, two successive rotations, and for yaw angle rate three successive rotations should be made (Figure 2.2). The transformation is expressed mathematically as follows;

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_{v_1}^b(x, \dot{\phi}) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{v_1}^b(x, \phi) R_{v_2}^{v_1}(y, \dot{\theta}) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{v_1}^b(x, \phi) R_{v_2}^{v_1}(y, \theta) R_e^{v_2}(z, \dot{\psi}) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.7)$$

and assume that  $\dot{\phi}$ ,  $\dot{\theta}$ ,  $\dot{\psi}$  are so small then;

$$R_{v_1}^b(x, \dot{\phi}) = R_{v_2}^{v_1}(y, \dot{\theta}) = R_e^{v_2}(z, \dot{\psi}) = I \quad (3 \times 3 \text{ identity matrix})$$

$T^{-1}(\phi, \theta, \psi)$  will be;

$$T^{-1}(\phi, \theta, \psi) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (2.8)$$

$T(\phi, \theta, \psi)$  is the inverse of  $T^{-1}(\phi, \theta, \psi)$  matrix,

$$T_b^e(\phi, \theta, \psi) = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) \sec(\theta) & \cos(\phi) \sec(\theta) \end{bmatrix} \quad (2.9)$$

After obtaining the  $R_b^e(\phi, \theta, \psi)$  and  $T(\phi, \theta, \psi)$ , all kinematic equations are combined with a matrix form.

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} R_b^e(\phi, \theta, \psi) & 0_{3 \times 3} \\ 0_{3 \times 3} & T(\phi, \theta, \psi) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \quad (2.10)$$

## 2.4 Quadrotor Dynamics

### 2.4.1 Newton-Euler Equations

A dynamic equation expresses the vehicle physical motion as a function of time while taking force, torque and energy into consideration. In the quadrotor, the generated forces and torques cause the linear and angular motion. The Newton-Euler method is chosen to obtain dynamic equations. All dynamic equations are written with respect to the body frame.

Some assumptions are made in order to make the calculations easy [23];

- The structure is supposed as rigid,
- The structure is supposed as symmetrical,
- The center of mass and the body fixed frame origin are assumed to coincide with each other,
- The propellers are supposed as rigid,

- Thrust and drag are proportional to the square of propeller's speed
- The inertia matrix is time-invariant.

According to Newton-Euler method [37];

$$\frac{d\vec{p}}{dt} \triangleq \vec{F}, \quad \frac{d\vec{h}}{dt} \triangleq \vec{\tau} \quad (2.11)$$

where  $\vec{p}$  is momentum vector,  $\vec{h}$  is angular momentum vector,  $\vec{F}$  is force vector,  $\vec{\tau}$  is torque vector.

Before the equations are derived, it is necessary to mention about the coriolis effect. If the frame  $B$  is rotated with respect to frame  $A$ , the orientation and the rotation between the frames are changed. Consider the case,  $\omega_{B/A}$  which is the angular velocity vector of frame  $B$  with respect to frame  $A$ . Suppose that the vector  $\vec{S}$  is moving in the frame  $B$  while the frame  $B$  is also rotating and translating with respect to frame  $A$ . Derivation of a vector is meaningful if the projection frame is defined. So, the derivation of  $\vec{S}$  in frame  $A$  with respect to the time  $t$  is equal to the derivation of  $\vec{S}$  in the frame  $B$  with respect to the time  $t$  plus the cross product of the angular velocity  $\omega_{B/A}$  vector with the vector  $\vec{S}$  [22]. This cross product is called as the coriolis effect.

$$\frac{d\vec{S}}{dt}|_A = \frac{d\vec{S}}{dt}|_B + \omega_{B/A} \times \vec{S} \quad (2.12)$$

From Equation 2.11 and Equation 2.12, one can derive that;

$$\begin{aligned} \frac{d\vec{p}}{dt}|_e &= m\left(\frac{d\vec{V}}{dt}|_e\right) = \vec{F} \\ \vec{F} &= m\left(\frac{d\vec{V}}{dt}|_e\right) = m\left(\frac{d\vec{V}}{dt}|_b + \vec{\omega} \times \vec{V}\right) \end{aligned} \quad (2.13)$$

where they are  $\vec{F} \triangleq [F_x \ F_y \ F_z]^T$ ,  $\vec{\omega} \triangleq [p \ q \ r]^T$  and  $\vec{V} \triangleq [u \ v \ w]^T$ . In detail, they can



be written as follows;

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{\vec{F}}{m} - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (2.14)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{pmatrix} F_x/m \\ F_y/m \\ F_z/m \end{pmatrix} - \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix}$$

A rotation motion for a rigid body is given as [55];

$$\tau = \frac{d\vec{h}}{dt}|_e = \left(\frac{d\vec{h}}{dt}\right)|_b + \vec{\omega} \times \vec{h} \quad (2.15)$$

Here  $\vec{\tau} \triangleq [\tau_x \quad \tau_y \quad \tau_z]^T$  and  $\vec{h} = J\vec{\omega}$  where  $J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}$ .

When the angular rate is differentiated, Equation 2.15 becomes;

$$\begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.16)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_x/J_{xx} \\ \tau_y/J_{yy} \\ \tau_z/J_{zz} \end{bmatrix} + \begin{bmatrix} (J_{yy} - J_{zz})/J_{xx} & qr \\ (J_{zz} - J_{xx})/J_{yy} & pr \\ (J_{xx} - J_{yy})/J_{zz} & pq \end{bmatrix}$$

In conclusion, the dynamic equations of the system are produced by Newton- Euler equations. The next step is the determination of the force and torque components of the system.

## 2.4.2 The Forces and Torques

This section introduces the main forces and torques which affect the system dynamics. Then main forces are the gravity, the propeller thrust and the air frictions. The roll,

pitch, yaw moment, propeller gyro effect are the main torques which influence the vehicle.

#### 2.4.2.1 Gravity Effect

Quadrotor body is directly exposed to the gravitation force along the positive  $Z$  axis with respect to the Earth frame. If the vertical component of the propeller's thrust is equal to the gravitation force, the body stays at a hovering condition and secures its own altitude. The force with respect to body frame ( $\vec{G}^b$ ) is found by multiplying with matrix  $R_e^b$ .

$\vec{G}^e$  is the gravitation force on the quadrotor which is expressed in the Earth frame as  $[0 \ 0 \ mg]^T$ . It is transformed into the body frame as follows;

$$\vec{G}^b = R_e^b \vec{G}^e$$

$$\vec{G}^b = \begin{bmatrix} -mg \sin(\theta) \\ mg \cos(\theta) \sin(\phi) \\ mg \cos(\theta) \cos(\phi) \end{bmatrix} \quad (2.17)$$

#### 2.4.2.2 Thrust Force

The motion of the quadrotor depends on the thrust force which are generated by four propellers. The vertical component of the thrust vector with respect to the Earth frame causes the elevation or declination. The direction of the thrust is always upward and it does not depend on propeller rotation direction due to special airfoil design profile.

Thrust value of a propeller is derived with the help of the blade element theory. The blade element theory is used to model the aerodynamic effects on the blade element. All forces and torques are integrated along the blade element from the rotor shaft to the rotor tip. The thrust value is computed by Equation 2.18.

$$T = C_T \rho \pi \omega^2 R^4 \quad (2.18)$$

where  $\rho$  is the air density,  $\omega$  is the rotor speed and,  $R$  is the rotor radius. They can be considered as time invariant. However, thrust coefficient ( $C_T$ ) is a time variant

parameter and it is obtained from the blade element theory [47].

$$C_T = \frac{1}{2}\sigma a \left( \frac{1}{3}\theta_{0.75} \left( 1 + \frac{3}{2}\mu^2 \right) - \frac{1}{2}\lambda \right) \quad (2.19)$$

where  $\sigma$  is rotor solidity, which is the ratio between the area of rotor blade over rotor sweeping area.  $a$  is lift slope and equals to  $2\pi$ . The inflow ratio ( $\lambda$ ) defines the relationship of induced velocity with the rotor tip speed ( $V_{tip} = \omega R$ ). The advance ratio ( $\mu$ ) is the division of the horizontal velocity ( $V = \sqrt{u^2 + v^2}$ ) over the rotor tip speed. The inflow and advance ratios are the key parameters for the thrust coefficient value.

$$\lambda = \frac{v_{induced} - w}{\omega R} \quad (2.20)$$

$$\mu = \frac{V_{forward}}{\omega R}$$

At the hover condition, the quadrotor has no forward and vertical velocity. Therefore,  $\mu$  will be zero and  $\lambda$  can be assumed as constant [36]. In this case, the thrust equation is simplified fairly and only the rotational speed of propeller affects the thrust value. If  $T_i$  is the thrust value of the  $i^{th}$  rotor, then the thrust is given as;

$$T_i = b\omega_i^2 \quad (2.21)$$

The quadrotor has four rotors, so the total thrust  $U_1$  will be the summation of all individual propeller thrusts.

$$U_1 = \sum_{i=1}^4 T_i = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (2.22)$$

$U_1$  is coincident to the  $Z$  axis of body frame and it is shown in the vector form ( $\vec{U}^b$ ) as  $[0 \ 0 \ -U_1]^T$

When  $U_1$  is equal to quadrotor weight ( $mg$ ), the system preserves its own altitude, which is called the hover condition. The nominal rotational speed of the system is found at the hover condition.

$$U_1 = mg$$

$$4b\omega_{nom}^2 = mg \quad (2.23)$$

$$\omega_{nom} = \sqrt{\frac{mg}{4b}}$$

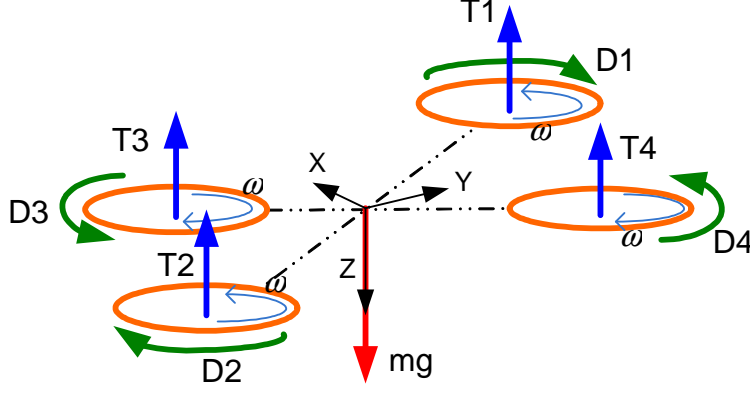


Figure 2.3: Effected Forces and Torques on the Quadrotor

### 2.4.2.3 Roll and Pitch Moment

The quadrotor is an underactuated system; therefore two degrees of freedom are coupled with the others. The thrust vector always points upward while hovering. Due to fixed rotors, the vehicle should change the tilt angle to move horizontally.

The roll and pitch moments are generated by the speed change of the propellers. When quadrotor rotates along only  $X$  axis (changing roll angle), the propeller  $p_1$  and  $p_4$  decrease their own speeds while  $p_2$  and  $p_3$  do the opposite. The left hand side of the quadrotor has more thrust than the right hand side have. Consequently, this thrust inequity is multiplied by the moment arm and generates the roll moment ( $\hat{=} U_2$ ). If the speed of the propellers  $p_1$  and  $p_3$  increase, while the propellers  $p_2$  and  $p_4$  decrease their own speeds, the pitch moment ( $\hat{=} U_3$ ) is generated along the quadrotor's  $Y$  axis.

The quadrotor's arms are perpendicular to each other and there is  $45^\circ$  between the body frame axes and the arms. Therefore, the moment arm is equal to  $1/\sqrt{2}$  times of the quadrotor's arm.

Roll and pitch moments are computed as follows;

$$U_2 = b \frac{l}{\sqrt{2}} (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (2.24)$$

$$U_3 = b \frac{l}{\sqrt{2}} (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

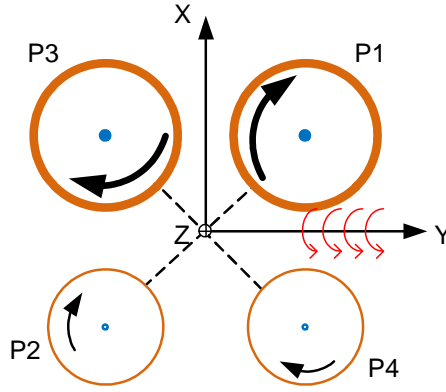


Figure 2.4: Pitch Moment Generation

#### 2.4.2.4 Yaw Moment

The yaw moment is produced by the drag moment of the propellers. The drag moment is the horizontal moment acting about the spinning rotor shaft and the direction of moment is the opposite of the rotation. Each propeller has a drag moment depending on the rotation speed and direction.

Drag moment is given as [36];

$$D = C_D \rho A (\omega R)^2 R \quad (2.25)$$

Due to  $A = \pi R^2$ , the equation can be written;

$$D = C_D \rho \pi \omega^2 R^5 \quad (2.26)$$

The expression is simplified as  $C_D$  and  $\rho$  are constant. Therefore, the equation 2.25 is given as;

$$D = d \omega_i^2 \quad (2.27)$$

where  $d$  is defined as drag factor and  $d \triangleq C_D \rho \pi R^5$ .

The neighboring rotors should revolve at the same speed but different directions. In that way, the drag moment of a propeller is balanced by the neighboring propellers and the system stabilizes itself along the Z axis. With change among the rotor speeds,

an imbalance occurs at the yaw moment which makes it possible to change the yaw (heading) angle.

The total yaw moment ( $U_4$ ) is calculated as;

$$U_4 = \sum_{i=1}^4 D_i = d(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \quad (2.28)$$

where  $D_i$  is the yaw moment of the  $i^{th}$  propeller.

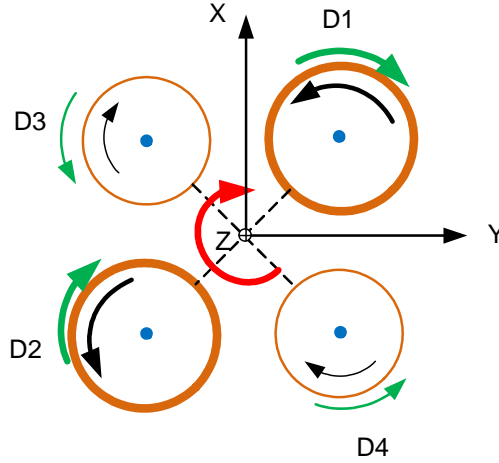


Figure 2.5: Yaw Moment

The relationship between the propeller speeds and the generated force and torques of the system can be given in a matrix form with the combination of Equation 2.22, 2.24 and 2.28.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ -l\frac{b}{\sqrt{2}} & l\frac{b}{\sqrt{2}} & l\frac{b}{\sqrt{2}} & -l\frac{b}{\sqrt{2}} \\ l\frac{b}{\sqrt{2}} & -l\frac{b}{\sqrt{2}} & l\frac{b}{\sqrt{2}} & -l\frac{b}{\sqrt{2}} \\ d & d & -d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.29)$$

$[U_1 \ U_2 \ U_3 \ U_4]^T$  will be the controller outputs which will be shown at the next chapter. If the matrix is inverted, the rotational speeds of the propellers can be computed from the controller outputs. The matrix distributes the controls inputs to the motors.

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \sqrt{\begin{bmatrix} b & b & b & b \\ -l\frac{b}{\sqrt{2}} & l\frac{b}{\sqrt{2}} & l\frac{b}{\sqrt{2}} & -l\frac{b}{\sqrt{2}} \\ l\frac{b}{\sqrt{2}} & -l\frac{b}{\sqrt{2}} & l\frac{b}{\sqrt{2}} & -l\frac{b}{\sqrt{2}} \\ d & d & -d & -d \end{bmatrix}^{-1}} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (2.30)$$

#### 2.4.2.5 Propeller Gyro Effect

Consider that a coordinate frame is fixed on the rotor shaft, and the Z axis of this frame shows the spinning direction of the propeller. While the propeller rotates with the speed  $\omega$ , if the frame turns along the X or Y axis, a torque emerges along the other axes. This phenomena is called as gyro effect [26], [20].

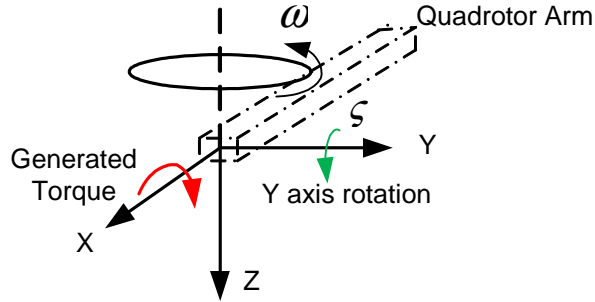


Figure 2.6: Propeller Gyro Effect

It is seen in Figure 2.6, that a propeller has its own local frame. The arm of quadrotor and the X axis of propeller is coincident and the X axis direction lays outside of the quadrotor. The Z axis is directed as downward and the Y axis is formed by the right hand rule. The generated torque ( $\vec{\tau}_i^p$ ), due to gyro effect, is calculated with respect to the propeller local frame as;

$$\vec{\tau}_i^p = J_p \begin{bmatrix} p_i \\ q_i \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ \omega_i \end{bmatrix} \quad (2.31)$$

where  $\tau_i^p$  is the  $i^{th}$  propeller's torque due to the gyro effect.

The resultant torque is expressed in the local frame, so it should be transformed to the quadrotor body frame. This action is to be repeated for all propellers.

The rotation matrix ( $R_p^b$ ) from the propeller local frame to the quadrotor body frame contains only yaw rotation. The yaw angle value is  $-45^\circ$  for the first propeller,  $135^\circ$  for the second propeller,  $45^\circ$  for the third propeller and  $-135^\circ$  for the fourth propeller.

$$\vec{\tau}_i^b = R_p^b \vec{\tau}_i^p, \quad \text{where } R_p^b = \begin{bmatrix} \cos(\psi_i) & \sin(\psi_i) & 0 \\ -\sin(\psi_i) & \cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

After the rotation, to obtain the total propeller gyro effect ( $\tau_{gyro}$ ), the individual torques are summed.

$$\tau_{gyro}^{\rightarrow} = \sum_{i=1}^4 \tau_i^b \quad (2.33)$$

#### 2.4.2.6 Other Aerodynamical Effects

- **Ground Effect:** When a rotor is close to the ground, its performance changes due to altering air inflow through the rotor. The rotor slipstream tends to expand rapidly as it approaches the surface. This phenomena is called as the ground effect.

Cheesman & Bennett examined the effect analytically, and they expressed the ground effect on the rotor thrust at the constant power with Equation 2.34 [31].

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - (\frac{R}{4z})^2} \quad (2.34)$$

where  $T_{IGE}$  is the rotor thrust in ground effect,  $T_{OGE}$  is the rotor thrust out ground effect,  $z$  is the rotor height and  $R$  is rotor radius.

This effect should be taken into consideration while designing controllers for landing and take off. Due to the small size of the quadrotor, one can consider



that the quadrotor is always out of ground effect ( $R/4z < 0.25$ ). Therefore, ground effect is neglected for modelling.

- **Parasite Drag**

The parasite drag is the air friction of non rotating parts of the quadrotor, which limits the forward and performance of the vehicle. The direction is always opposite to the quadrotor motion and it is proportional to the air density, drag coefficient (depends on the vehicle's geometry), wetted (projected) area and square of quadrotor linear speed. The air friction force ( $F_D^b$ ) can be expressed for  $X$ ,  $Y$  and  $Z$  axes of the body as follows;

$$\begin{aligned} F_{Dx}^b &= \frac{1}{2} \rho C_{Dx} A_{wX} u|u| \\ F_{Dy}^b &= \frac{1}{2} \rho C_{Dy} A_{wY} v|v| \\ F_{Dz}^b &= \frac{1}{2} \rho C_{Dz} A_{wZ} w|w| \end{aligned} \quad (2.35)$$

- **Hub Force and Rolling Moment**

The hub force and rolling moment are the other aerodynamical effects, which are generated by the rotors. They affect the system when a horizontal motion exists. They are ignored for modelling, because their effects are relatively so small when they are compared with the other effects [18].

## 2.5 State Variables and Equations

There are twelve state variables of the quadrotor system. The six of them come from the kinematic equations, the other six are obtained from the dynamic equations. The states ( $\Gamma$ ) are listed as follows;

$$\Gamma = [x \ y \ z \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r]^T$$

The equation 2.14 is rewritten to obtain linear and angular accelerations.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} F_x/m \\ F_y/m \\ F_z/m \end{bmatrix} - \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} \quad (2.36)$$

Total amount of the thrust ( $\vec{U}^b$ ), gravity ( $\vec{G}^b$ ) and air friction ( $\vec{F}_D^b$ ) are equal to the total force on the system ( $\vec{F}$ ). So Equation 2.36 will be;

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -g \sin(\theta) + rv - qw - \frac{1}{2m}\rho A_{wX} C_{Dx} u |u| \\ g \cos(\theta) \sin(\phi) + pw - ru - \frac{1}{2m}\rho A_{wY} C_{Dy} v |v| \\ -U_1/m + g \cos(\theta) \cos(\phi) + qu - pv - \frac{1}{2m}\rho A_{wZ} C_{Dz} w |w| \end{bmatrix} \quad (2.37)$$

The rotational dynamics are written with use of Equation 2.16. The total torques ( $\vec{\tau}$ ) of the system are combined with roll moment ( $U_2$ ), pitch moment ( $U_3$ ), yaw moment ( $U_4$ ) and the torque which is generated by propeller gyro effect ( $\tau_{gyro}$ ).

$$\tau_{gyro} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \vec{\tau} \quad (2.38)$$

As a result, Equation 2.5 is expressed as below;

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (U_2 + (J_{yy} - J_{zz})qr + \tau_{gyroX})/J_{xx} \\ (U_3 + (J_{zz} - J_{xx})rp + \tau_{gyroY})/J_{yy} \\ (U_4 + (J_{xx} - J_{yy})pq)/J_{zz} \end{bmatrix} \quad (2.39)$$

The kinematic relations are obtained previously at Equation 2.5 and 2.9. They are rewritten to show all state equations as a whole.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_b^e(\phi, \theta, \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.40)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T(\phi, \theta, \psi) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.41)$$

The three of the states (2.40) are related to the guidance/ navigation solution. The aim of the control action is to keep the Euler angles and vertical velocity at the desired values. So, while designing controllers,  $x$ ,  $y$  and  $z$  states become redundant. It will be seen that in the next chapter, all the controllers will be designed by Equations 2.37, 2.39, 2.41.

## 2.6 Linearization of the System

The model of the quadrotor is nonlinear like almost all physical systems. However, linear controllers can be designed for the linear systems. Therefore, the nonlinear system must be linearized around the determined operating points which are called as equilibrium or steady state points.

### 2.6.1 Equilibrium Points (Trim States)

Consider that  $\dot{x} = f(x, u)$  is a nonlinear system, In a dynamic equation, equilibrium points/ trim states ( $\bar{x}$ ) and equilibrium inputs ( $\bar{u}$ ) satisfy  $f(\bar{x}, \bar{u}) = 0$  for all  $t$  [27]. It means that in the trimmed condition, net applied force and torque become zero. The dynamic equations of the quadrotor which are given in Equation 2.37 and 2.39 should be solved within this manner.

Previously, it is mentioned that Equation 2.40 is related to the navigation; therefore, they are redundant for the trim states formation [41]. Furthermore, the yaw angle ( $\psi$ ) represents the orientation of the system and has no effect on the dynamic equations. Indeed, it is only related to the initial orientation of the system. Therefore, the yaw angle is also redundant to calculate the trim states. After all, there are eight unknown states and four unknown inputs, but only nine equations are available. Therefore, some constraints or initial conditions must be added to solve the equations. The constraints are not selected at random, they should be selected with respect to expected quadrotor's motions and scenarios. In this paper, trimmed states are computed as the quadrotor moves with low speed and does not make acrobatics. The trim conditions will be computed iteratively for different  $u, w, \dot{\psi}$  values.  $u$  and  $w$  value varies between  $-2 \text{ m/sec}$  and  $2 \text{ m/sec}$  with increase of  $0.5 \text{ m/sec}$  and  $\dot{\psi}$  is changed from  $-90^\circ/\text{sec}$  to  $90^\circ/\text{sec}$  with increase of  $10^\circ/\text{sec}$ .  $v$  is taken as zero because it will be seen in Chapter 4 that in autonomous mission the quadrotor always holds the lateral velocity at zero ( $v = 0 \text{ m/sec}$ ).

$$u = -2 : 0.5 : 2 \text{ m/sec}$$

$$w = -2 : 0.5 : 2 \text{ m/sec}$$

$$v = 0 \text{ m/s}$$

$$\dot{\psi} = -90 : 10 : 90^\circ/\text{sec}$$

Solving the equations analytically are not easy, so Newton- Raphson method is used.  $\Gamma$  is the state vector and  $\bar{\Gamma}$  is the trimmed state.  $f(\Gamma)$  and  $J(\Gamma_k)$  represent the state equations and Jacobian of  $f(\Gamma)$  respectively. At the trim condition ( $f(\bar{\Gamma}) = 0$ ), first, proper initial states ( $\Gamma_0$ ) are given for the solution. After sufficient number of iterations ( $k$ :iteration number) are performed, the states converge a set of values. The iteration runs until  $\varepsilon = \Gamma_{k+1} - \Gamma_k$  is fairly small.

$$\Gamma_{k+1} = \Gamma_k - J(\Gamma_k)^{-1} f(\Gamma_k) \quad (2.42)$$

From Figure 2.7 to Figure 2.11, trimmed pitch ( $\theta$ ), roll ( $\phi$ ), thrust ( $U_1$ ), roll moment ( $U_2$ ) and pitch moment ( $U_3$ ) are shown with respect to different forward velocity ( $u$ ), vertical velocity ( $w$ ) and rotation/ yaw rate ( $\dot{\psi}$ ).

The pitch angle varies between  $-5^\circ$  and  $5^\circ$  which mainly depends on the forward velocity. The roll angle varies from  $-20^\circ$  to  $20^\circ$  and it is proportional to the yaw rate and forward velocity. Because, the roll angle is generated to balance drifting/ coriolis effect while the system is rotating with a constant  $\dot{\psi} = c$ . The horizontal component of the thrust holds the coriolis effect, consequently net force is equal to zero.

The trimmed thrust is the resistant force against the linear air friction and coriolis effect. The coriolis effect may be negligible because it is fairly smaller than the air friction.  $U_1$  increases to oppose the air friction when the vertical speed rises. So, net force becomes zero. It is seen in Figure 2.7,  $U_1$  is varied maximum 10% from the quadrotor's weight. Therefore, it can be said that it will be always around the system nominal thrust ( $U_{1nom}$ ).

The roll and pitch moments are very small values about zero. So, one can assume that they are equal to zero. Furthermore, yaw moment is always zero, because at the equation 2.39,  $J_{xx} - J_{yy} = 0$  which makes the equation  $\dot{r} = U_4/J_{zz}$ . At the trim condition  $\dot{r} = 0$  so  $U_4 = 0$  for all  $t$  (It is not plotted here).

In conclusion, these results reveal that a single trim state may be sufficient to controller design. So, the hover condition is selected as the trim state. The system is linearized around the hover condition and then controller is constructed based on this point. It is expected that designed controller will properly work for all operating regions.

Trim states and inputs for hover condition ( $u = v = w = \dot{\psi} = 0$ ) are given as;

$$\begin{aligned} \bar{U} &= [mg \ 0 \ 0 \ 0]^T \\ [\bar{p} \ \bar{q} \ \bar{r} \ \bar{\phi} \ \bar{\theta}] &= 0_{5 \times 1} \end{aligned}$$

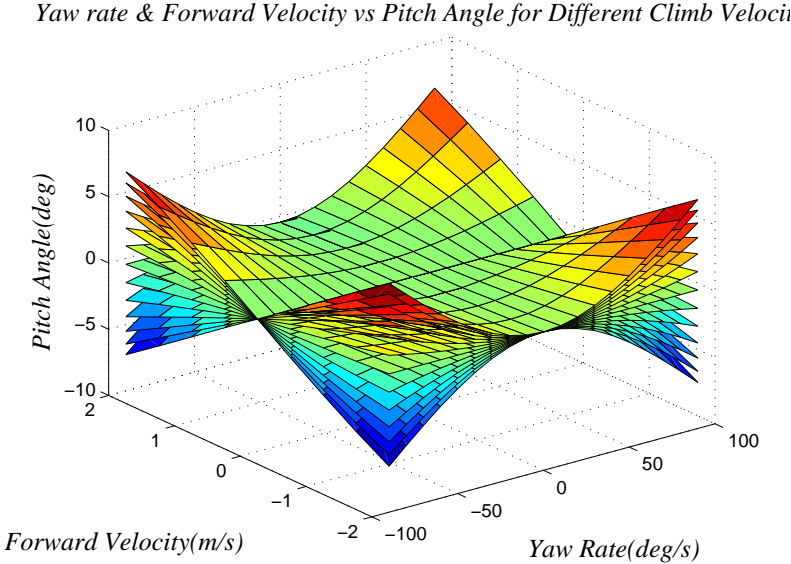


Figure 2.7: Pitch Angle for Different Trimmed States

*Yaw rate & Forward Velocity vs Roll for Different Climb Velocity*

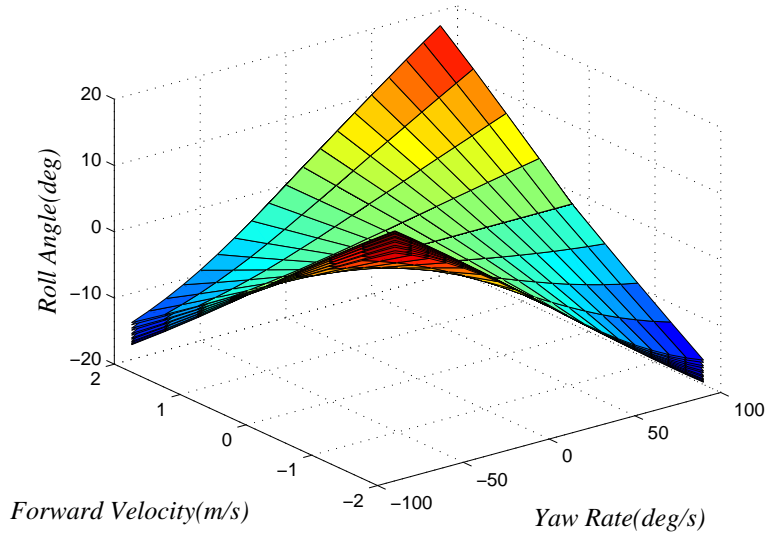


Figure 2.8: Roll Angle for Different Trimmed States

*Yaw rate & Forward Velocity vs Thrust for Different Climb Velocity*

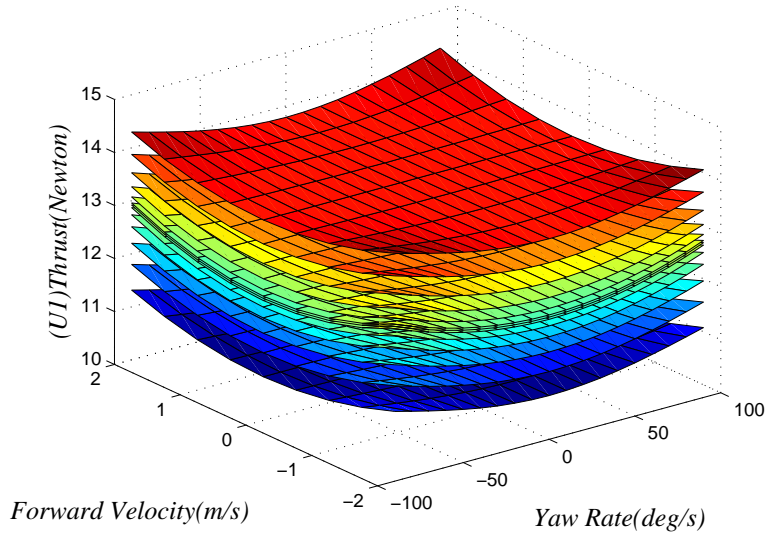


Figure 2.9: Thrust for Different Trimmed States

*Yaw rate & Forward Velocity vs Roll Moment for Different Climb Veloc*

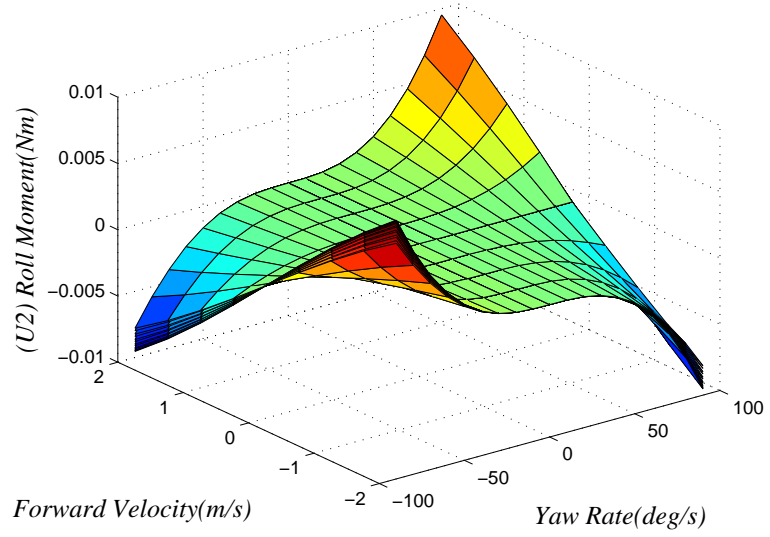


Figure 2.10: Roll Moment for Different Trimmed States

*Yaw rate & Forward Velocity vs Pitch Moment for Different Climb Veloc*

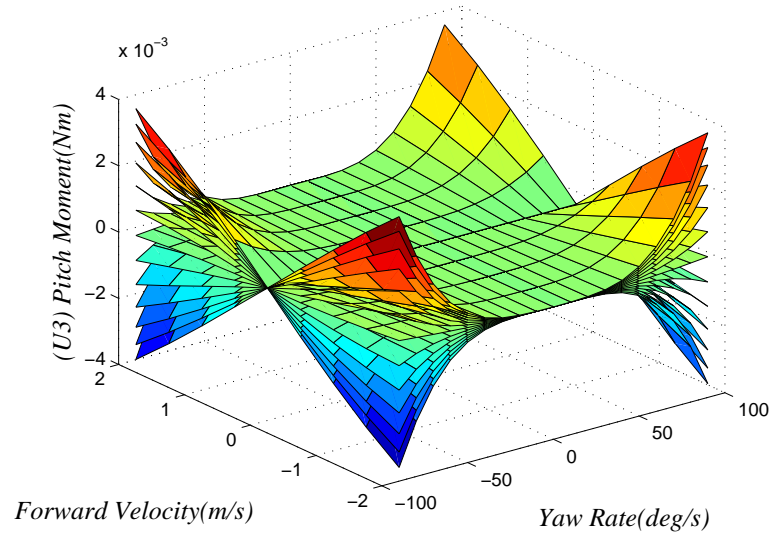


Figure 2.11: Pitch Moment for Different Trimmed States

## 2.6.2 Obtaining the State Matrices

Taylor expansion is used to linearize a nonlinear equation around an equilibrium point. The states obtained by using a linear model are quite accurately model the exact states when they are not far from the equilibrium points. Assume that  $\delta x$  and  $\delta u$  are the deviations from the trim conditions ( $\delta x \triangleq x - \bar{x}$ ,  $\delta u \triangleq u - \bar{u}$ ). Taylor expansion gives ;

$$\dot{x} = f(x, u) = f(\bar{x}, \bar{u}) + \frac{\delta f}{\delta x} \Big|_{\bar{x}, \bar{u}} \delta x + \frac{\delta f}{\delta u} \Big|_{\bar{x}, \bar{u}} \delta u + \frac{1}{2} \frac{\delta^2 f}{\delta x^2} \Big|_{\bar{x}} \delta x^2 + \frac{1}{2} \frac{\delta^2 f}{\delta u^2} \Big|_{\bar{u}} \delta u^2 \dots \quad (2.43)$$

At the trim conditions,  $\ddot{x} = 0$   $f(\bar{x}, \bar{u}) = 0$ . In order to simplify the Taylor expansion, higher order terms are truncated, so the equation consists of only the first order derivation terms (Jacobian Linearization)[13]. Taylor expansion can be rewritten in a simple form;

$$\dot{\delta x} \approx A \delta x + B \delta u \quad (2.44)$$

where  $A = \frac{\delta f}{\delta x} \Big|_{\bar{x}, \bar{u}}$  and  $B = \frac{\delta f}{\delta u} \Big|_{\bar{x}, \bar{u}}$  The linear controllers are designed for only the dynamics of the hover condition. It is certain that controllers give supreme performance at the hover condition when they are compared to the other trim conditions. The designed controllers are tested under several conditions and their responses are examined whether they are sufficient or not. Unless the system behavior is good, new linear controllers should be designed for several trim conditions (Gain scheduling).

Procedure of the linearization about hover condition is obtained by using MATLAB, for the state vector  $[u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r]^T$  and the input vector  $[U_1 \ U_2 \ U_3 \ U_4]^T$ , The state matrices are found as;



$$A = \begin{bmatrix} 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.45)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/J_{xx} & 0 & 0 \\ 0 & 0 & 1/J_{yy} & 0 \\ 0 & 0 & 0 & 1/J_{zz} \end{bmatrix} \quad (2.46)$$

The output matrix  $C$  is selected according to desired output of the system and these are  $u, v, w, \psi$ . Furthermore, there is no input-output coupling ( $D = 0$ ).

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.47)$$



## CHAPTER 3

### CONTROLLER DESIGN

#### 3.1 Introduction

The design steps of the controllers of the autopilot system are introduced in this chapter. Three different control strategies are proposed and they are PI controller, linear quadratic regulator (LQR) and sliding mode controller. First two methods are applied to the linearized system whereas the third one is implemented on the nonlinear system.

#### 3.2 PI Controller

PI controllers generate outputs based on the error between the commanded input and the actual output. The proportional (P) term of a PI controller generates a control signal proportional to the error and increasing it improves the rise time of a system. The integral term (I) generates a control signal proportional to the integral of the error. It is slow but increasing the I term decreases the steady state error. For the quadrotor systems, the controller produces the thrust ( $U_1$ ) or torques ( $U_2, U_3, U_4$ ).

If  $r(t)$  is a reference input,  $y(t)$  is a system output, and  $e(t)$  is the error, the output ( $U_c$ ) of a PI controller is given as;

$$\begin{aligned} e(t) &= r(t) - y(t) \\ u_c(t) &= K_P e(t) + K_I \int_0^t e(\tau) d\tau \end{aligned} \tag{3.1}$$

In Laplace domain, it is represented as;

$$U_c(s) = (K_P + \frac{K_I}{s})E(s) \quad (3.2)$$

### 3.2.1 PI Controller of the Quadrotor

The guidance system sets the the desired  $u, v, w, \psi$  references for the control system. The controllers are designed to track these reference commands. The structure of the control system is shown in Figure 3.1. For the quadrotor model, PI controller will be shown to ensure stability and meet the desired specifications. However, it results in a large overshoot. Therefore, a pre-filter is designed to decrease overshoot. The pre-filter is a first order low pass filter with unity gain (Equation: 3.3). In Laplace domain, pre-filter transfer function ( $F(s)$ ) is;

$$F(s) = \frac{K}{s + K} \quad (3.3)$$

In principle, the pole of the pre-filter is used to cancel out the zero of the closed loop system. As a result, high frequency components of the input (reference) signal is filtered out. As a drawback, the pre-filter makes the system response slow down. Moreover, in real applications, the place of the zero of the system is not precisely known, so the pole of the pre-filter does not completely cancel the zero. In such a case, the overshoot may still be large or the system may extremely slow down.

The error is computed from filtered reference and sensor measurement, then controller generates the control signal. The control signals are transformed into rotational speed of propellers with the motor inversion block, which consists of Equation 2.30.

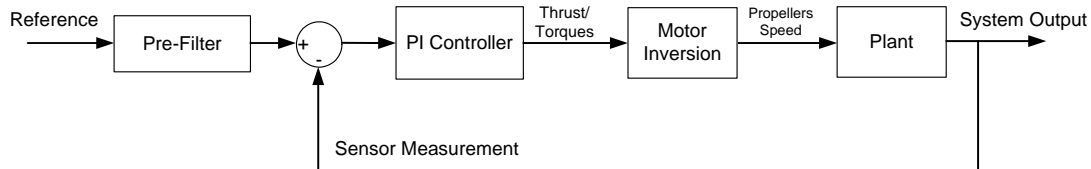


Figure 3.1: PI+PreFilter Control Structure

In order to control the quadrotor, cascade controller loops are used. The cascade control has superior disturbance rejection properties. In the quadrotor, three cascaded loops are formed for the control of horizontal speeds ( $u, v$ ). The yaw ( $\psi$ ) angle is controlled by two cascaded loops. The vertical speed ( $w$ ) controller is not cascaded and has only one PI loop.

When the controllers are designed, some system requirements should be determined; the damping ratio ( $\zeta$ ) is selected as 0.7, which is the value for second order system to make its response underdamped. The controller bandwidths are chosen according to the system dynamics.

Due to diagonality of the inertia matrix, the roll and pitch dynamics are identical. In this thesis, only the design steps of the roll rate and the roll controller are mentioned. The same procedures are followed for the pitch rate and the pitch angle controls.

### 3.2.1.1 Roll & Pitch Rate Control

The roll and pitch rates ( $\dot{\phi}, \dot{\theta}$ ) are equal to the  $p$  and  $q$  values. This assumption is valid only when the roll and pitch angles are small.

The roll rate dynamic and its Laplace transform are given as follows;

$$\dot{p} = \frac{U_2}{J_{xx}} \Rightarrow p(s) = \frac{U_2(s)}{J_{xx}s} \quad (3.4)$$

The transfer function ( $G_p(s)$ ) between  $U_2(s)$  and  $p(s)$ ;

$$G_p(s) = \frac{p(s)}{U_2(s)} = \frac{1}{J_{xx}s} \quad (3.5)$$

The PI controller is formed in Laplace domain as  $K_p(s) = K_P + \frac{K_I}{s}$ . In that case, the closed loop transfer function will be ( $T_p(s) = \frac{K_p G_p}{1 + K_p G_p}$ );

$$T_p(s) = \frac{K_P s + K_I}{J_{xx}s^2 + K_P s + K_I} \quad (3.6)$$

A second order closed loop system in general form is written as;

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.7)$$

The system natural frequency is equal to the bandwidth of the system ( $\omega_n = B_\omega$ ). When the characteristic polynomial of  $T_p(s)$  and  $T(s)$  are equalized,  $K_P$  and  $K_I$  values can be computed for the damping ratio  $\zeta = 0.7$  and the bandwidth  $B_\omega = 15\text{Hz}$ .

$$\omega_n^2 = \frac{K_I}{J_{xx}} \Rightarrow K_I = J_{xx}\omega_n^2 \quad (3.8)$$

$$2\zeta\omega_n = \frac{K_P}{J_{xx}} \Rightarrow K_P = 2J_{xx}\zeta\omega_n \quad (3.9)$$

The closed loop system has a zero at  $-K_I/K_P$  which causes an overshoot. To decrease the overshoot, a pre-filter is added to cancel the zero of the system.

$$F_p(s) = \frac{K_I/K_P}{s + K_I/K_P} \quad (3.10)$$

The effect of the pre-filter on the step response of the system are shown in Figure 3.2. Furthermore, the closed loop Bode plot shows that desired bandwidth is achieved (Figure 3.3).

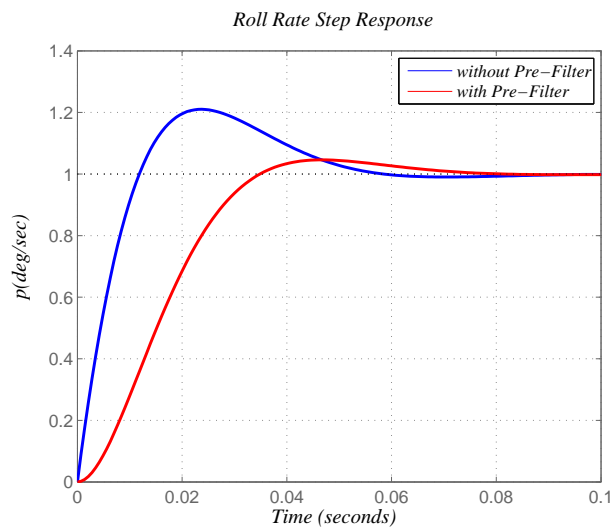


Figure 3.2: Roll Rate Step Response

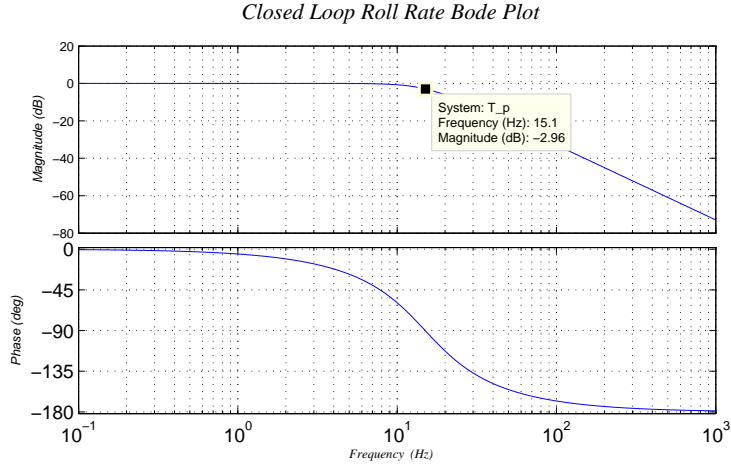


Figure 3.3: Closed Loop Roll Rate Bode Plot

### 3.2.1.2 Roll & Pitch Control

The roll and pitch controllers generate the reference signals for the previously designed  $p$  and  $q$  control loops respectively. For a cascaded loop controller, one can assume that if the inner loop is almost ten times faster than the outer loop, the inner loop transfer function can be thought as  $T_{in}(s) = 1$ . Thus, the roll and pitch angle dynamics are taken into consideration in a very simple form. For example, the relationship between  $p$  and  $\phi$  is just an integrator.

$$\dot{\phi} = p \Rightarrow \frac{\phi(s)}{p(s)} = \frac{1}{s} \quad (3.11)$$

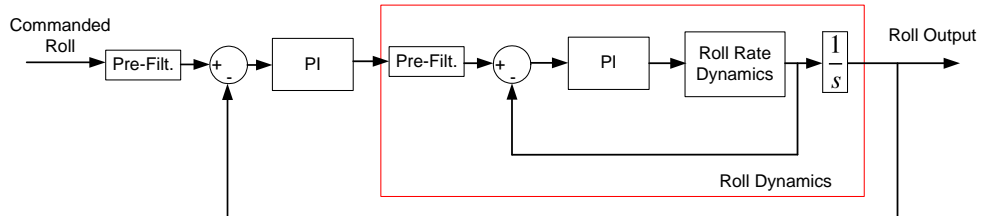


Figure 3.4: Roll Dynamics of the System

The PI controller is designed as the damping ratio  $\zeta = 0.7$  and the bandwidth  $B_\omega =$

1.5Hz. Then, PI parameters of the roll angle will be;

$$\begin{aligned} K_I &= \omega_n^2 \\ K_P &= 2\zeta\omega_n \end{aligned} \quad (3.12)$$

The pre-filter model is chosen as it is in the previous section. Therefore, it is not mentioned again. The step response of the closed loop system is shown in Figure 3.5.

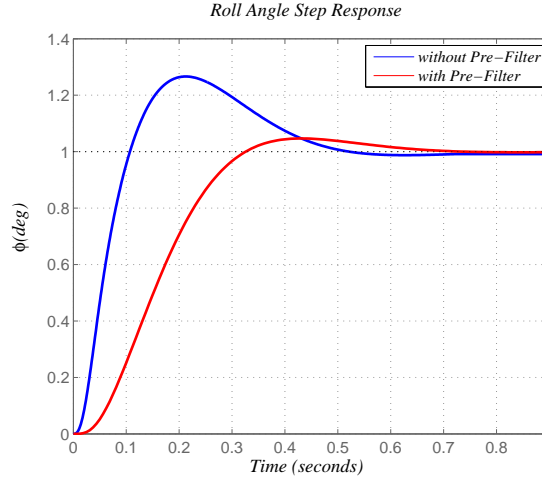


Figure 3.5: Roll Angle Step Response

### 3.2.1.3 Horizontal Velocities Control

The quadrotor changes its roll/ pitch angle to move horizontally. After the linearization in Chapter 1, the  $u$  and  $v$  dynamics are found as below;

$$\begin{aligned} \dot{u} &= -g\theta \Rightarrow \frac{u(s)}{\theta(s)} = \frac{-g}{s} \\ \dot{v} &= g\phi \Rightarrow \frac{v(s)}{\phi(s)} = \frac{g}{s} \end{aligned} \quad (3.13)$$

The damping ratio is chosen as  $\zeta = 0.7$  and the bandwidth decreases seven times than roll/ pitch controllers and it is set to be 0.2 Hz. For  $u$  controller, the parameters are given as;

$$\begin{aligned} K_I &= \omega_n^2 / -g \\ K_P &= 2\zeta\omega_n / -g \end{aligned} \quad (3.14)$$



$v$  controller has the same parameters with opposite sign and the step response of the closed  $v$  loop is shown in Figure 3.6. In conclusion, by designing three cascaded loops, the quadrotor can track the horizontal speed commands of the guidance algorithm.

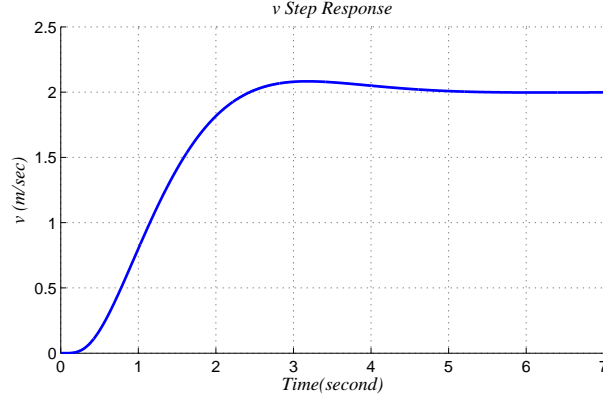


Figure 3.6:  $v$  Step Response

### 3.2.1.4 Vertical Velocity Control

The  $w$  dynamic of the quadrotor is given as follows;

$$\dot{w} = -\frac{U_1}{m} + g \quad (3.15)$$

The vertical velocity is controlled by the thrust. At the hover condition,  $U_1$  value is equal to the whole system's weight ( $mg$ ) and in that case, the thrust is called as the nominal thrust. The gravity can be considered as a constant disturbance and to suppress this disturbance, a feed forward controller is used in the system (Figure 3.7). Total output of the feed forward and PI control are equal to  $U_1$ . Equation 3.15 can be written as follows;

$$\dot{w} = -\frac{U_{1(nom)} + \delta U_1}{m} + g \quad (3.16)$$

where  $U_{1(nom)}$  is the output of the feed forward controller and it is equal to the weight of the system. So, the  $w$  controller ( $\delta U_1$ ) can be considered as a perturbation onto the nominal thrust. Equation 3.16 will be;

$$\dot{w} = -\frac{\delta U_1}{m} \quad (3.17)$$

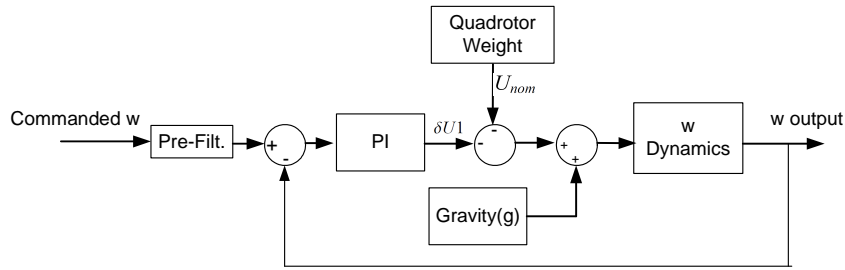


Figure 3.7: Vertical Speed Closed Loop Structure

The damping ratio and the bandwidth are chosen as  $\zeta = 0.7$  and  $0.5 \text{ Hz}$  respectively. After the PI controller is formed, the step response of  $w$  is shown in Figure 3.8.

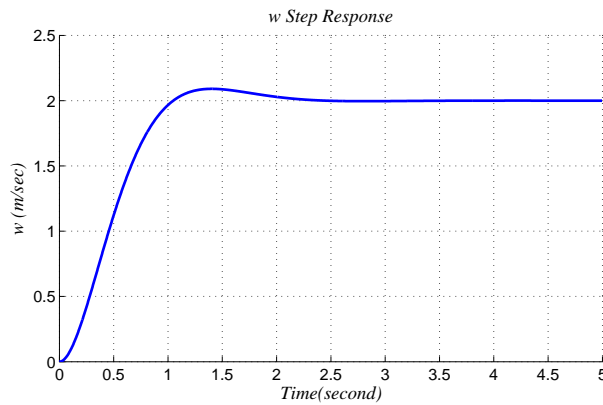


Figure 3.8:  $w$  Step Response

### 3.2.1.5 Yaw Rate and Yaw Angle Control

The yaw rate dynamic is similar to the roll rate dynamics. However, the yaw moment ( $U_4$ ) and inertia ( $J_{zz}$ ) values are substituted with the roll moment ( $U_2$ ) and  $J_{xx}$ . A PI controller and pre-filter are designed with the same manner. The damping ratio and bandwidth are selected  $\zeta = 0.7$ ,  $B_\omega = 8 \text{ Hz}$  respectively.

The yaw rate response of the system is given in Figure 3.9

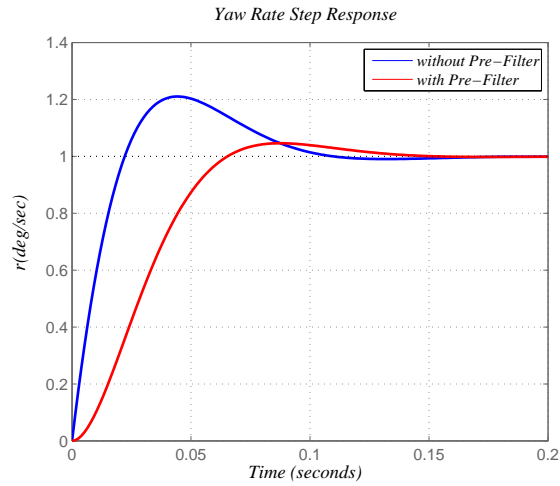


Figure 3.9: Yaw Rate( $r$ ) Step Response

The yaw angle model is similar to the roll/ pitch angle models. Once again the damping ratio is chosen as  $\zeta = 0.7$  and the bandwidth is arranged to be  $0.2 \text{ Hz}$  as well. The step response of the yaw angle is shown below in Figure 3.10.

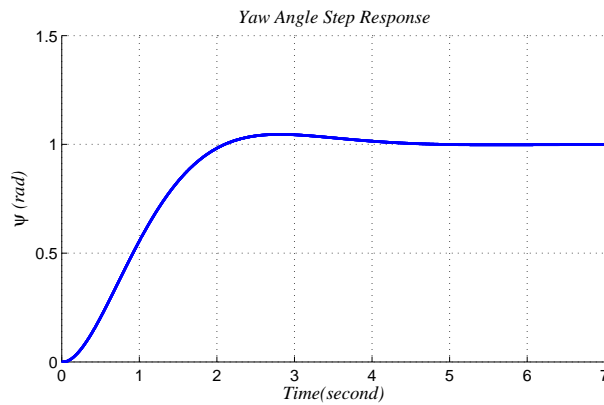


Figure 3.10: Yaw Angle Step Response

The PI+ Pre-filter controllers are tested on the nonlinear model of the quadrotor in Simulink. For low speed profile, the linear and nonlinear system response are so similar. Therefore, only one set of controller design gives sufficient responses. Constructed Simulink blocks are given in Figure 3.11.

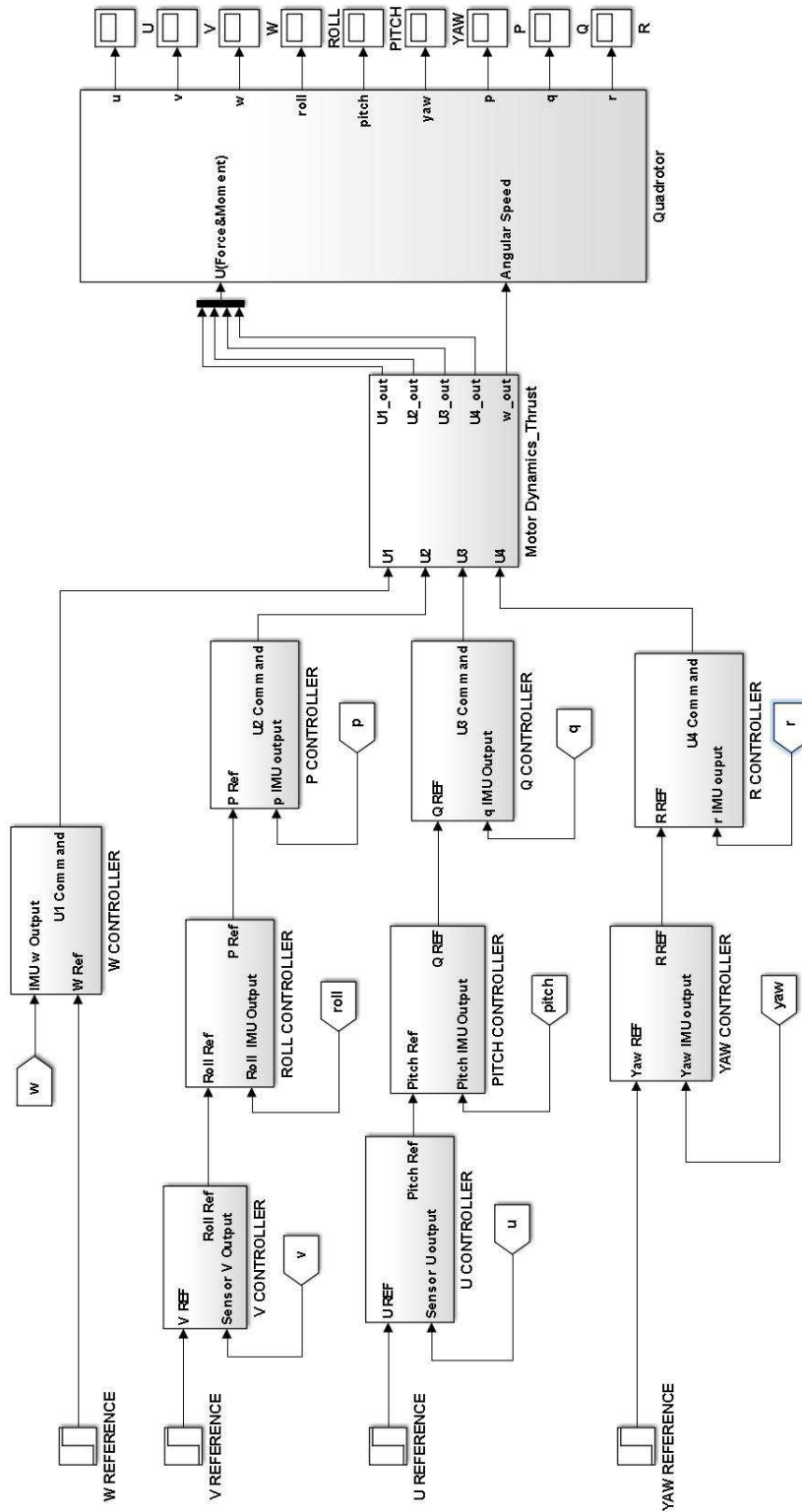


Figure 3.11: PI Cascade Loops for Quadrotor Autopilot

### 3.3 Linear Quadratic Regulator (LQR)

In this section, first of all, a brief summary is presented about the LQR. Then, LQR design steps for the quadrotor are explained and designed controller is tested in MATLAB.

A time invariant linear system is given as:

$$\begin{aligned} \dot{x} &= Ax + Bu, & x(t_0) &= x_0 \\ y &= Cx \end{aligned} \quad (3.18)$$

It is aimed to design a state feedback controller to stabilize the system ( $u = Kx$ ). Determination of the state feedback vector  $K$  is a tradeoff between the transient response and the control action. According to performance index (cost function), the optimal control method handles this tradeoff by finding  $u$  which minimizes the cost function.

$$J = \frac{1}{2} \int_0^{\infty} [x^T Qx + u^T Ru] dt \quad (3.19)$$

The controller design is a constrained optimization whose constraints are managed by the system dynamics. The cost function is formed as control inputs and states in which time  $t$  is unbounded and final value of  $x(t_f)$  have no constraints. In Equation 3.19,  $Q$  is a positive semi definite and  $R$  is a symmetric positive definite matrix.  $x^T Qx$  limits transient state deviation,  $u^T Ru$  limits control effort [9].

Hamiltonian can be written after a Lagrange multiplier is added to the cost function;

$$H = \frac{1}{2} x^T Qx + u^T Ru + p^T (Ax + Bu) \quad (3.20)$$

where  $p$  is the adjoint variable or co-state.  $x$  is known initially,  $p$  is known at the final time, since it is  $p(t_f) = P x(t_f)$ . The necessary condition for optimality is computed by differentiating  $H$  with respect to the states and co-states.

1.  $\dot{x} = \frac{\partial H^T}{\partial p} = Ax + Bu$  with  $x(t_0) = x_0$  (State Equations )
2.  $\dot{p} = -\frac{\partial H^T}{\partial x} = -(Qx + A^T p)$  with boundary condition  $p(t_f) = P_{t_f} x(t_f)$  (Co-state Equation )
3.  $\frac{\partial H}{\partial u} = 0 \rightarrow Ru + B^T p = 0$ , so  $u = -R^{-1} B^T p$  (Optimal Control Equation)

(For being minimum  $\frac{\partial^2 H}{\partial u^2} \leq 0$  so it should be  $R \geq 0$ )

$u$  is substituted with  $-R^{-1}B^T p$  and the above state and co-state expressions are combined. As a result, The Hamiltonian matrix is constructed.

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \quad (3.21)$$

The matrix shows closed loop dynamics of  $x$  and  $p$ . By introducing  $p = Px$  ( $P$  is positive semi definite symmetric), the optimization problem with two boundary values can be solved by Riccati Equation [5]. The derivation of Riccati Equation for a time invariant system is given below;

$$\begin{aligned} p &= Px \\ \dot{p} &= \dot{P}x + P\dot{x} \\ &= \dot{P}x + P(Ax + Bu) \\ &= \dot{P}x + P(Ax - BR^{-1}B^T p) \\ &= \dot{P}x + P(Ax - BR^{-1}B^T Px) \\ -(Qx + A^T Px) &= (\dot{P} + PA - PBR^{-1}B^T P)x \\ -\dot{P} &= PA + A^T P - PBR^{-1}B^T P + Q \end{aligned} \quad (3.22)$$

For infinite time horizon case ( $t \rightarrow \infty$ ),  $\dot{P} = 0$ , the solution is obtained from the algebraic Riccati Equation (ARE) [9]. The solution of the AREs can give different  $P$  matrices but only one of them should be the positive semi definite one.

LQR design procedures can be summarized as;

1. Determine proper  $Q$ ,  $R$  matrices such as positive semi definite and positive definite respectively
2. Solve Riccati equation and calculate  $K = -R^{-1}B^T P$
3. Simulate the system
4. If system response does not meet the system requirements or constraints, change the  $Q$  and  $R$ , follow the same design procedure.

### 3.3.1 LQR Design for Quadrotor

The LQR design is a regulator problem which means that designed controller/ regulator makes the system stable and all the states of the system will go to zero. In this thesis, it is aimed that the quadrotor will track desired velocities and heading angle commands. So, the LQR should solve a tracking/ servo problem.

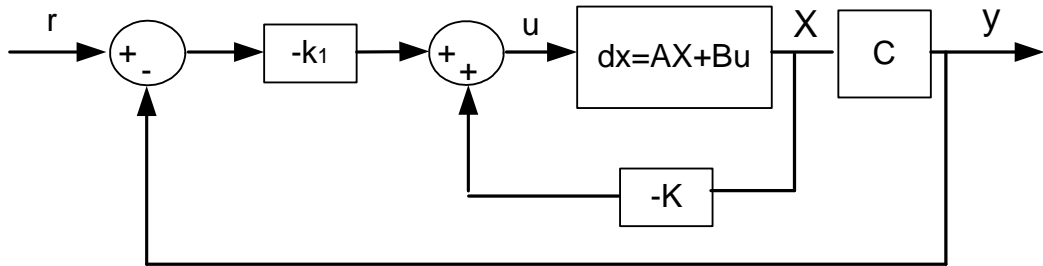


Figure 3.12: LQR Design for Tracking(Type 1)

In order to design a state feedback controller, the linear model of the system is firstly examined; if the linear model contains an integrator, which means that the system is type 1/ servo system, an extra integral term does not needed to be added to achieve tracking action. The structure of full state feedback of type 1 system is shown in Figure 3.12. So, control input for tracking will be;

$$\begin{aligned}
 u &= - [0 \quad k_2 \quad \dots \quad k_n] \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + k_1(r - x_1) \\
 &= - Kx + k_1r
 \end{aligned} \tag{3.23}$$

where  $K = [k_1 \quad k_2 \quad \dots \quad k_n]$  is the full state feedback vector and it is computed by solving the algebraic Riccati equation. After the state feedback ( $K$ ) is found, the state

equation can be written as follows;

$$\dot{x} = Ax + Bu = (A - BK)x + Bk_1r \quad (3.24)$$

At the steady state ( $t \rightarrow \infty$ );

$$\dot{x}(\infty) = (A - BK)x(\infty) + Bk_1r(\infty) \quad (3.25)$$

$r(\infty)$  is equal to any  $r(t)$  for  $t \rightarrow \infty$ , subtracting Equation 3.25 from Equation 3.24;

$$\dot{x}(t) - \dot{x}(\infty) = (A - BK)(x(t) - x(\infty)) \quad (3.26)$$

$e = x(t) - x(\infty)$ , so the equation becomes;

$$\dot{e}(t) = (A - BK)e(t) \quad (3.27)$$

Consequently, tracking problem is transformed into a regulator problem, if eigenvalues of  $(A - BK)$  have negative real part, the system is asymptotically stable and  $e(t)$  goes to zero for any initial value.

After the linearization, the quadrotor dynamics are written again as follows:

$$\begin{aligned} \dot{q} &= \frac{U_3}{J_{yy}}, & \dot{\theta} &= q, & \dot{u} &= -g\theta \\ \dot{p} &= \frac{U_2}{J_{xx}}, & \dot{\phi} &= p, & \dot{v} &= g\phi \\ \dot{r} &= \frac{U_4}{J_{zz}}, & \dot{\psi} &= r \\ \dot{w} &= -\frac{U_1}{m} + g \end{aligned} \quad (3.28)$$

The quadrotor's inputs are not coupled with each others. Therefore, the states and associated inputs are divided into four groups (Equation: 3.29).



$$\begin{aligned}
\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} A_1 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ v \end{bmatrix} + \begin{bmatrix} B_1 \end{bmatrix} [U_3] \\
\begin{bmatrix} \dot{p} \\ \dot{\phi} \\ \dot{u} \end{bmatrix} &= \begin{bmatrix} A_2 \end{bmatrix} \begin{bmatrix} p \\ \phi \\ u \end{bmatrix} + \begin{bmatrix} B_2 \end{bmatrix} [U_2] \\
\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} A_3 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} B_3 \end{bmatrix} [U_4] \\
\begin{bmatrix} \dot{w} \end{bmatrix} &= \begin{bmatrix} A_4 \end{bmatrix} \begin{bmatrix} w \end{bmatrix} + \begin{bmatrix} B_4 \end{bmatrix} [U_1]
\end{aligned} \tag{3.29}$$

In Equation 3.29, state matrices are written as follows;

$$\begin{aligned}
A_1 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -g & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, A_4 = \begin{bmatrix} 0 \end{bmatrix} \\
B_1 &= \begin{bmatrix} 1/J_{yy} \\ 0 \\ 0 \end{bmatrix}, B_2 = \begin{bmatrix} 1/J_{xx} \\ 0 \\ 0 \end{bmatrix}, B_3 = \begin{bmatrix} 1/J_{zz} \\ 0 \end{bmatrix}, B_4 = \begin{bmatrix} 1/m \end{bmatrix} \\
C_1 = C_2 &= [0 \quad 0 \quad 1], C_3 = [0 \quad 1], C_4 = [1]
\end{aligned}$$

Designed state feedback controllers are simulated in the MATLAB/ Simulink environment and resultant system responses are given in Figures 3.13, 3.14 and 3.15.

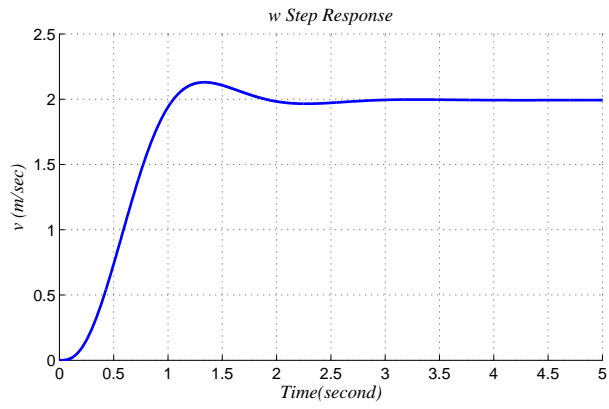


Figure 3.13: v Step Response

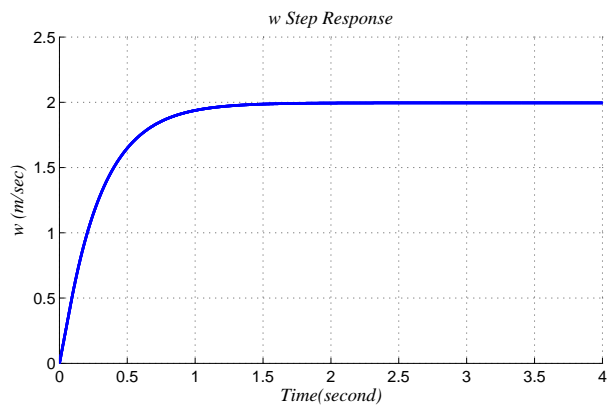


Figure 3.14: w Step Response

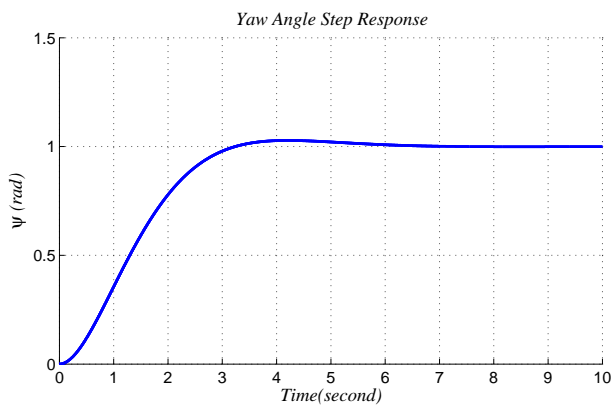


Figure 3.15: Yaw Angle Step Response

### 3.4 Sliding Mode Control

A control action should be designed considering the contribution of uncontrolled inputs, which are noises, disturbances or modifications of the system. The sliding mode control is a robust control method against bounded disturbance and system uncertainty.

In the sliding mode control, feedback path switch between two values according to the rule that depends on the value of state at each time. The aim of switching control law is to drive trajectories of state onto a surface for all subsequent times. The surface is determined by the control engineer and it is called the sliding surface (the sliding manifold, switching function /surface) and the resultant trajectory motion is a sliding mode. Furthermore, the sliding surface does not depend on the plant dynamics or control law. If the state is above the surface, a feedback path has one gain, if the state is below the surface, the feedback has a different gain. Therefore, the sliding mode control is a variable structure control (VSC) method [29].

The sliding mode control (SMC) contains two design steps:

1. Selection of a stable sliding surface, in which system states or errors are restricted, has desired dynamics.
2. Determination of a control law which makes the states/ errors move towards the sliding surface.

In the sliding mode control, the trajectory of the states generally follow two phases (modes). First, they reach the sliding surface called the reaching phase. Then, the states stay in the sliding surface which is called as the sliding phase. If the states are defined as error states, they reach the origin after a while.

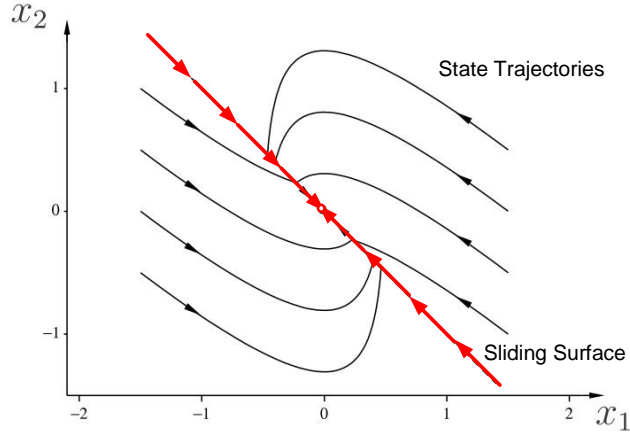


Figure 3.16: State Trajectories and Sliding Surface

Suppose  $x$  represent the state vector,  $d$  represents the uncontrolled input. Assume that a system which is given as;

$$\dot{x} = f(x, t) + B(x, t)u(x, t) + d(x) \quad (3.30)$$

where  $x \in R^n$ ,  $u \in R^m$ ,  $B \in R^{n \times m}$ .

The first design step is the determination of the sliding surface. Nonlinear switching surfaces are not appropriate to design and the order of the switching function should be less than order of the plant. The sliding surface can be chosen as [48];

$$S(x) = \left(\frac{d}{dt} + \lambda\right)^{n-1} x = 0 \quad (3.31)$$

where  $S(x) = [S_1(x) \ S_2(x) \ S_3(x) \dots \ S_m(x)]^T = 0$

In Equation 3.31, the upper limit of  $\lambda$  depends on the physical property of the system and the lower limit depends on the allowable tracking time [35].

After a sliding surface is selected, the next step is the determination of a control law. The control law makes the states converge to the sliding surface. When the states reach the surface, they remain on the surface for all time. This step is satisfied by defining a Lyapunov function  $V(x)$ .

$V(x)$  is the energy like positive definite function and a system is globally asymptotically stable for Lyapunov stability (Second/ Direct method), if

$$\begin{aligned} V(x) > 0 \quad \text{for } x \neq 0 \quad \& \quad V(x) = 0 \quad \text{for } x = 0 \\ \dot{V}(x) < 0 \quad \text{for } x \neq 0 \quad \& \quad \dot{V}(x) = 0 \quad \text{for } x = 0 \end{aligned} \quad (3.32)$$

For  $S(x)$ , which is written as  $S$  in short, the positive definite quadratic Lyapunov function is given as;

$$V(x) = S^T W S, \quad W : \text{ a symmetric positive definite matrix} \quad (3.33)$$

$u(x)$  must be selected that makes  $\dot{V}(x)$  negative. The derivation of control input is given as follows;

$$\dot{V}(x) = \dot{S}^T W S + S^T W S^T \dot{V}(x) = 2S^T W \dot{S} \quad (3.34)$$

Suppose that  $\dot{S}$  has no  $t$  but only  $x$  dependency and it will be;

$$\dot{S} = \frac{\partial S}{\partial x} \dot{x} = \frac{\partial S}{\partial x} [f(x, t) + B(x, t)u(x, t)] \quad (3.35)$$

$$\dot{V}(x) = 2S^T W \left( \frac{\partial S}{\partial x} f(x, t) + \frac{\partial S}{\partial x} B(x, t)u(x, t) \right) < 0 \quad (3.36)$$

Here the control input is decomposed by equivalent ( $u_{eq}$ ) and corrective ( $u_c$ ).

$$u(x, t) = u_c(x, t) + u_{eq}(x, t) \quad (3.37)$$

The equivalent control ( $u_{eq}(x, t)$ ) is the control action which operates the system on the sliding surface, on which the state trajectories converge to zero.

$$\dot{S} = 0 \Rightarrow u_{eq}(x, t) = - \left[ \frac{\partial S}{\partial x} B(x, t) \right]^{-1} \frac{\partial S}{\partial x} f(x, u) \quad (3.38)$$

The corrective control ( $u_c$ ) makes  $\dot{V}(x)$  negative for any  $x \neq 0$ . It becomes active when the trajectories are outside of sliding surfaces.

$u_c(x, t)$  should be given as follows;

$$\dot{V}(x) = 2S^T W \left[ \frac{\partial S}{\partial x} B(x, t) \right] u_c(x, t) < 0 \quad (3.39)$$

Assume that  $[\frac{\partial S}{\partial x}B(x, t)]$  is invertible and  $u_c(x, t)$  is written as;

$$u_c(x, t) = [\frac{\partial S}{\partial x}B(x, t)]^{-1}\hat{u}_c(x, t) \quad (3.40)$$

To make the calculation easy, assume that  $W = I$ . Hence,  $\dot{V}(x)$  is simplified to find  $\hat{u}_c(x, t)$ ;

$$\dot{V}(x) = 2S^T W [\frac{\partial S}{\partial x}B(x, t)] [\frac{\partial S}{\partial x}B(x, t)]^{-1}\hat{u}_c(x, t) = 2S^T \hat{u}_c(x, t) < 0 \quad (3.41)$$

$\hat{u}_c(x, t)$  is found with Equation 3.41. There are some different structures for the corrective control. But generally relay with constant gain is chosen in the literature [29].

$$\hat{u}_c(x, t) = U \text{sgn}(S) \quad (3.42)$$

where "sgn" is the signum function

$$\text{sgn}(S) = \begin{cases} 1, & S > 0 \\ 0, & S = 0 \\ -1, & S < 0 \end{cases} \quad (3.43)$$

Consequently, to ensure stability of the system ( $\dot{V}(x) < 0$ ),  $U$  should be negative. Furthermore, magnitude of  $U$  may be increased to make system more robust against the modelling errors and disturbances.

### **Chattering**

After a trajectory reaches a sliding surface, it slides along the surface. This may require infinitely fast switching. However, switching components have imperfection such as delay, hysteresis etc. in reality. Therefore, oscillations may occur on the neighborhood of the sliding surface. This may cause high controller efforts and stimulate frequency response. This phenomena is called as chattering.

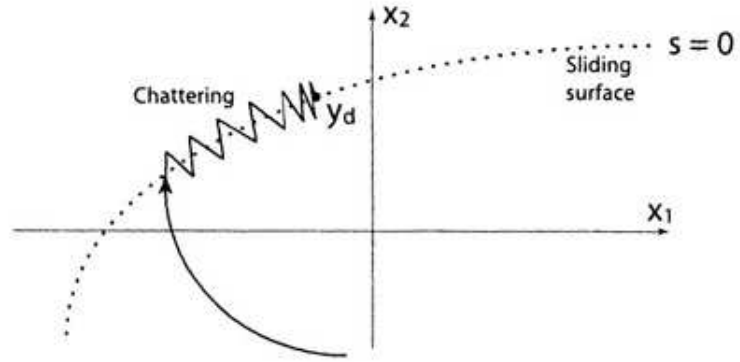


Figure 3.17: Chattering

To avoid the chattering, some functions are used instead of signum which smooth the discontinuity between positive and negative numbers. In this thesis, the saturation (*sat*) function is used [45].

$$sat(S/\epsilon) = \begin{cases} S/\epsilon, & |S/\epsilon| \leq 1 \\ sgn(S/\epsilon), & otherwise \end{cases} \quad (3.44)$$

where  $\epsilon$  is a very small number and system robustness increases and if  $\epsilon$  is chosen as small as possible.

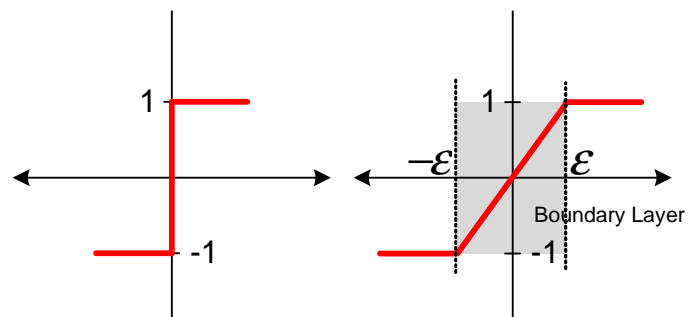


Figure 3.18: Signum & Saturation Function

### 3.4.1 Sliding Mode Control of Quadrotor

Assume that  $p, q, r$  are very small perturbations and in that case,  $\dot{\phi} = p, \quad \dot{\theta} = q$ . The rotation dynamic equations are constructed again according to this assumption. Consequently, six state equations are obtained to design the sliding mode controller.

$$\begin{aligned}
 \ddot{\phi} &= (U_2 + (J_{yy} - J_{zz})\dot{\theta}\dot{\psi} + J_p\dot{\theta}\omega_R)/J_{xx} \\
 \ddot{\theta} &= (U_3 + (J_{zz} - J_{xx})\dot{\psi}\dot{\phi} - J_p\dot{\phi}\omega_R)/J_{yy} \\
 \ddot{\psi} &= (U_4 + (J_{xx} - J_{yy})\dot{\phi}\dot{\theta})/J_{zz} \\
 \dot{u} &= -g \sin(\theta) + rv - qw - \frac{1}{2m}\rho A_{wx} C_{Dx} \dot{u} |\dot{u}| \\
 \dot{v} &= g \sin(\phi) \cos(\theta) + pw - ru - \frac{1}{2m}\rho A_{wy} C_{Dy} \dot{v} |\dot{v}| \\
 \dot{w} &= -U_1/m + g \cos(\phi) \cos(\theta) + qu - pv - \frac{1}{2m}\rho A_{wz} C_{Dz} \dot{w} |\dot{w}|
 \end{aligned} \tag{3.45}$$

The attitude dynamics are second order, the orientation dynamics are first order differential equations. The proper sliding surface will be chosen with respect to Equation 3.31.

If the states are converted to error states, sliding mode controller guarantees that the states converge to zero that means the system follows the reference signals.

By defining the error states as;  $e_\phi = \phi - \phi_d, e_\theta = \theta - \theta_d, e_\psi = \psi - \psi_d$  and for  $\lambda_i > 0$ , the sliding surfaces are selected for the attitude controllers as;

$$\begin{aligned}
 S_\phi &= \dot{e}_\phi + \lambda_\phi e_\phi \\
 S_\theta &= \dot{e}_\theta + \lambda_\theta e_\theta \\
 S_\psi &= \dot{e}_\psi + \lambda_\psi e_\psi
 \end{aligned} \tag{3.46}$$

Lyapunov function is selected as  $V(x) = \frac{1}{2} S^T S$  and the time derivative of  $V(x)$  satisfy the inequality  $\dot{V}(S) = S \dot{S} < 0$ . Then  $u(x, t)$  should be configured to ensure  $\dot{S} = -k \text{sgn}(S)$  where  $k > 0$ . It carries out Lyapunov stability condition. The differentiation



of the sliding surfaces;

$$\begin{aligned}\dot{S}_\phi &= \ddot{e}_\phi + \lambda_\phi \dot{e}_\phi \\ \dot{S}_\theta &= \ddot{e}_\theta + \lambda_\theta \dot{e}_\theta \\ \dot{S}_\psi &= \ddot{e}_\psi + \lambda_\psi \dot{e}_\psi\end{aligned}\tag{3.47}$$

When  $e$  is substituted with its definition, the equations are given as;

$$\begin{aligned}\dot{S}_\phi &= \ddot{\phi} - \ddot{\phi}_d + \lambda_\phi(\dot{\phi} - \dot{\phi}_d) \\ \dot{S}_\theta &= \ddot{\theta} - \ddot{\theta}_d + \lambda_\theta(\dot{\theta} - \dot{\theta}_d) \\ \dot{S}_\psi &= \ddot{\psi} - \ddot{\psi}_d + \lambda_\psi(\dot{\psi} - \dot{\psi}_d)\end{aligned}\tag{3.48}$$

Instead of the second order derivative of the Euler angles ( $\ddot{\phi}$ ,  $\ddot{\theta}$ ,  $\ddot{\psi}$ ), associated state equations are written (Equation 3.45). As a result, the control inputs will be as follows;

$$\begin{aligned}U_2 &= J_{xx}[\ddot{\phi}_d - \lambda_\phi(\dot{\phi} - \dot{\phi}_d) - (J_{yy} - J_{zz})\dot{\theta}\dot{\psi} - J_r\dot{\theta}w_R - k_\phi \text{sgn}(S_\phi)] \\ U_3 &= J_{yy}[\ddot{\theta}_d - \lambda_\theta(\dot{\theta} - \dot{\theta}_d) - (J_{zz} - J_{xx})\dot{\psi}\dot{\phi} + J_r\dot{\phi}w_R - k_\theta \text{sgn}(S_\theta)] \\ U_4 &= J_{zz}[\ddot{\psi}_d - \lambda_\psi(\dot{\psi} - \dot{\psi}_d) - k_\psi \text{sgn}(S_\psi)]\end{aligned}\tag{3.49}$$

To design the sliding mode controller for the orientation, first of all, the air friction is considered as a disturbance and neglected for the linear velocity controls. The error states are defined as  $e_u = u - u_d$ ,  $e_v = v - v_d$ ,  $e_w = w - w_d$  and the sliding surfaces are obtained according to Equation 3.31. The surfaces are very simple in form, because the state equation contains first order terms only.

$$\begin{aligned}S_u &= e_u \\ S_v &= e_v \\ S_w &= e_w\end{aligned}\tag{3.50}$$

In case the sliding surfaces are not two dimensional, they can be as phase lines instead of sliding surfaces [4]. Figure 3.19 illustrates a phase line.



Figure 3.19: One Dimension Phase Portrait (Phase Line)

The state equations, which are associated with  $\dot{u}$  and  $\dot{v}$ , do not include the controller efforts, but  $U_1$  directly affects the horizontal motion ( $w$ ). The quadrotor is underactuated and  $X$  and  $Y$  axes motions depend on the pitch and the roll angles. So, the tilt angle can be considered as virtual control inputs to design the sliding mode controller. The virtual control inputs produce the reference signal for the roll and pitch angles [50], [15].

The virtual inputs and the modified state equations are given as;

$$\begin{aligned} U_u &\triangleq \sin(\theta_d) \Rightarrow \dot{u} = -gU_u \\ U_v &\triangleq \cos(\theta) \sin(\phi_d) \Rightarrow \dot{v} = gU_v \end{aligned} \quad (3.51)$$

Lyapunov function is chosen as in the previous case, the derivative of the sliding surfaces are computed as follows;

$$\begin{aligned} \dot{S}_u &= \dot{u} - \dot{u}_d \\ \dot{S}_v &= \dot{v} - \dot{v}_d \\ \dot{S}_w &= \dot{w} - \dot{w}_d \end{aligned} \quad (3.52)$$

For  $\dot{S} = -k \operatorname{sgn}(S)$ , control inputs will be;

$$\begin{aligned} U_u &= -\frac{1}{g}[\dot{u}_d - rv + qw - k_u \operatorname{sgn}(S_u)] \\ U_v &= \frac{1}{g}[\dot{v}_d - pw + ru - k_v \operatorname{sgn}(S_v)] \\ U_1 &= -[w_d - g \cos(\theta) \cos(\phi) - qu + pv - \operatorname{sgn}(S_w)] \end{aligned} \quad (3.53)$$

The desired tilt angle can be revealed from trigonometric calculation.

$$\begin{aligned} \theta_d &= \arcsin(U_u) \\ \phi_d &= \arcsin\left(\frac{U_v}{\cos \theta}\right) \end{aligned} \quad (3.54)$$

Due to the chattering, The "sgn" function is replaced with the "sat" function. The resultant system responses are presented in Figures 3.20, 3.21 and 3.22.

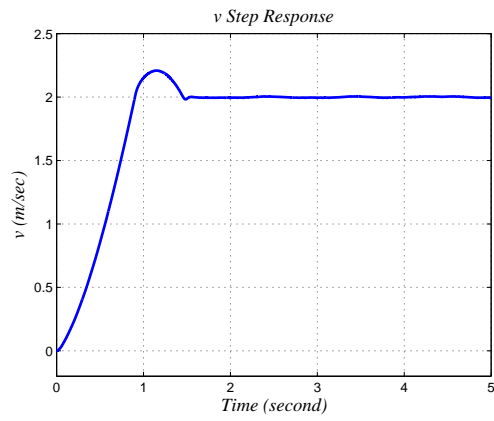


Figure 3.20:  $v$  Step Response

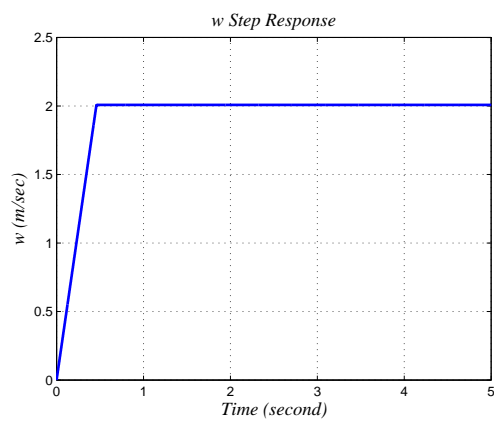


Figure 3.21:  $w$  Step Response

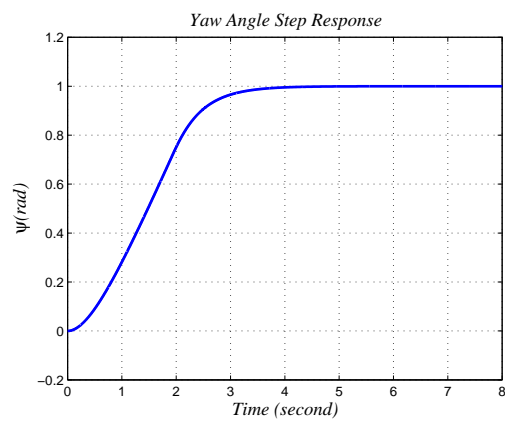


Figure 3.22:  $\psi$  Step Response

### 3.5 Comparison of the Designed Controllers

This section compares the designed PI+Pre filter, LQR and sliding mode controllers. All three controllers enable to secure system stability and their time responses are sufficiently good.

The settling time for  $v$  is nearly 4 *sec* for the PI, 2.5 *sec* for the LQR, and 1.5 *sec* for the sliding mode controller.  $U_2$  (the roll moment) which is the control signal is far from the saturation limits of the system for all designed controllers (Figure 3.23). Although the PI has the slowest response, it has largest control signal. The performance of the LQR is better than the PI, because it has shorter settling time with smaller controller effort. The sliding mode controller has the fastest response, but the controller effort makes periodical peaks that wastes energy during the steady states.

The control signals ( $U_1$ , the total thrust) are shown in Figure 3.24. The controller affords does not exceed the saturation limits. For  $w = 2m/sec$ , the sliding mode controller has the fastest response. The system response does not oscillate and its shape is like a ramp input before it reaches the reference. The controller generates a constant output during half a second, then it suddenly decreases to the system's weight. The LQR's response is faster than the PI's and as expected the controller signal is larger than it. Furthermore, it is seen in Figure that  $U_1$  is equal to weight of the system at the steady state.

The settling times of yaw are 4 *sec*, 5.5 *sec* and 4 *sec* for the PI, LQR and sliding mode controller respectively. The sliding mode controller generates an impulsive output and the signal is saturated. The PI controller has faster response than the LQR and the its control afford is approximately two times larger than that of LQR's.

There are some performance indices to compare the designed controllers. In this thesis, the integral of squared error (ISE) and the integral of time multiply absolute error (ITAE) methods are used to compare the performances of the controllers. The IAE and ITAE performance indices are given in Equation 3.55.

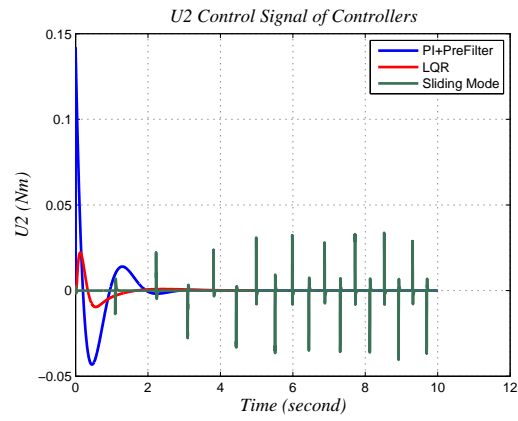


Figure 3.23:  $U_2$  Control Signal

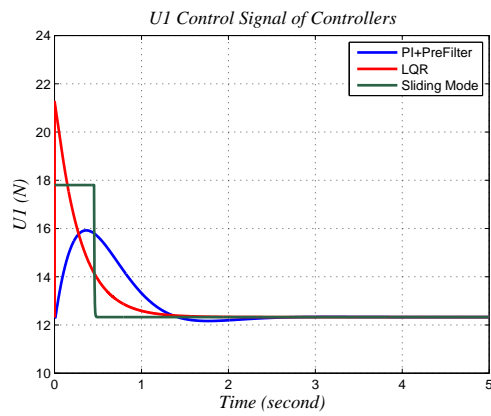


Figure 3.24:  $U_1$  Control Signal

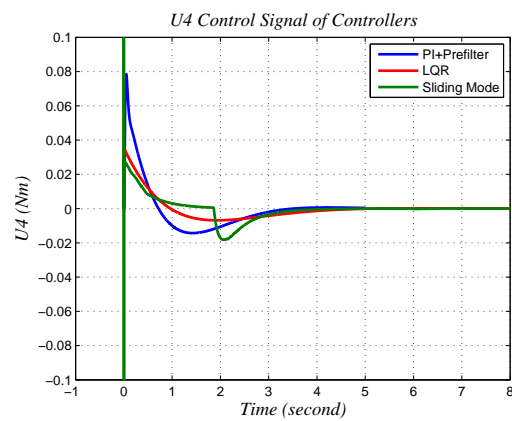


Figure 3.25:  $U_4$  Control Signal

$$IAE = \int_0^{\infty} |e^2(t)|dt \tag{3.55}$$

$$ITAE = \int_0^{\infty} t|e^2(t)|dt$$

The performance indices of the controllers are shown in Table 3.1. With respect to the IAE,  $v$  and  $w$  loop controllers are nearly the same. The PI controller is superior when  $\psi$  response are compared with others. According to the ITAE, the sliding mode shows the best performance for  $v$  tracking.  $w$  performance of the controllers can be said to be identical. However, if  $\psi$  performances are compared, the PI controller is extremely good. Its results are three times better than that of LQR's and five times better than that of the sliding mode controller's results.

Table3.1: Comparison of the Controllers with Different Performance Indices

Designed Methods	Reference	IAE	ITAE
PI+Pre Filter	$v$	1287	1092.5
	$w$	523.7	202.1
	$\psi$	516.4	393.3
LQR	$v$	1327.7	774.1
	$w$	601.8	216.83
	$\psi$	1424.2	1403.3
Sliding Mode	$v$	1318.6	571.2
	$w$	493.3	170
	$\psi$	1755.4	2214.1

## CHAPTER 4

### GUIDANCE

#### 4.1 Introduction

Guidance is the action of determining the desired course, attitude and speed of the vehicle with respect to some reference frames [51]. A guidance system determines the vehicle's trajectories and enables the system to work autonomously.

Guidance is a part of navigation, guidance and control (NGC) system. For a UAV system, navigation computes current location and orientation of the system by using sensors such as IMU, compass, altimeter, GPS, etc. Outputs of the navigation become inputs for the guidance. Then, reference signals which are inputs to the control system are calculated by the guidance. The control system drives actuators in consideration of system saturation and aerodynamic effects, so the system follows reference signals and preserve own stability.

In the literature, several guidance techniques are available, such as waypoint guidance by line of sight (LOS), proportional navigation, vision based guidance, Lyapunov based guidance, etc [30].

The LOS is a widely used guidance method due to the implementation simplicity [51]. Firstly, the LOS guidance algorithm finds the line of sight angle that is the angle between the vehicle speed vector and the target. Then, the angle will be set to the zero by vehicle steering (changing  $\psi$  angle). Consequently, the velocity vector points the target and the vehicle approaches the target. The LOS guidance is used for stationary targets although it is not suitable for manoeuvring or moving targets.

## 4.2 Line of Sight (LOS) Guidance

A way point is set as a coordinate in the state space. If path following is desired, the path should be split into a certain number of way points by the flight management. The flight management stores all way point coordinates and reference speed magnitudes in a sequence.

The way point that is in front of the vehicle is called the desired/ present way point. The desired way point is a sphere in three dimensional space whose origin is exactly at desired way point coordinates. Sphere's radius is called as the region of acceptance ( $R_{acc}$ ) and usually it is determined by the velocity of the vehicle. When the quadrotor enters inside of the spherical region, the flight management considers that the vehicle has reached the desired way point. Afterwards, new desired way point and speed magnitude are given to the guidance algorithm according to stored way point sets. Later on, the guidance computes necessary reference body fixed velocities ( $u_{ref}$ ,  $v_{ref}$ ,  $w_{ref}$ ) and heading angle ( $\psi_{ref}$ ). This action repeats itself until no desired way point is left to reach. When the final desired way point is reached, depending on the flight management, all velocity commands can be set to the zero, so the system preserves its own location, or it can land on a secured region or return to launch/take off point.

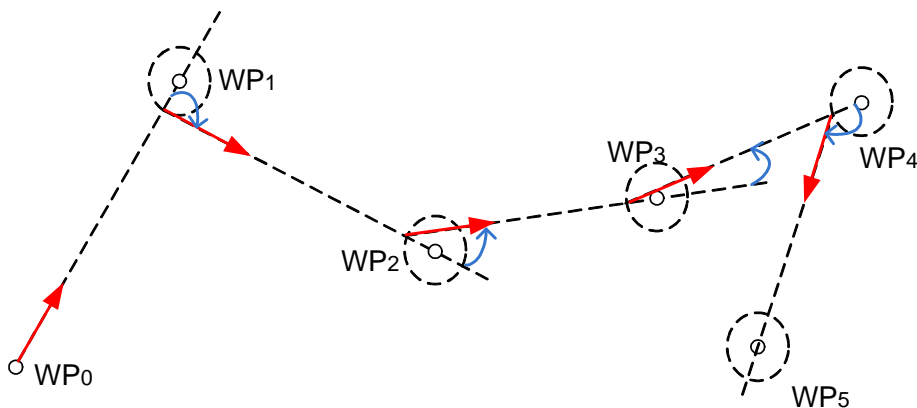


Figure 4.1: Way Point Guidance



### 4.3 LOS Guidance of a Quadrotor

In this thesis, the aim is to move the quadrotor between constant way points with the LOS guidance. Assume that initial point or current location of the quadrotor ( $WP_0$ ) is given as  $[x_0 \ y_0 \ z_0]^T$  and desired way point ( $WP_1$ ) is given as  $[x_1 \ y_1 \ z_1]^T$ . It is planned that the quadrotor will fly from  $WP_0$  to  $WP_1$  on a straight line with a given constant speed ( $v$ ).

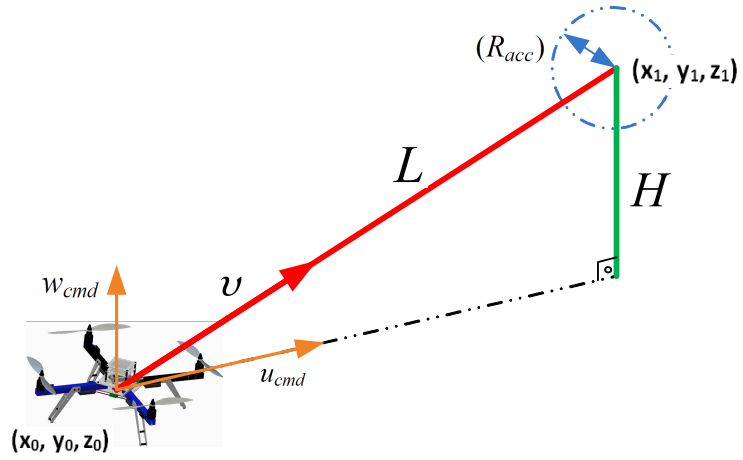


Figure 4.2: LOS Guidance of a Quadrotor

The path of the quadrotor is a straight line between the vehicle and the desired way point. LOS angle ( $\alpha_{LOS}$ ) is the angle between north and this line. The first step of the guidance is the calculation of  $\alpha_{LOS}$  by using Equation 4.1. Then  $\alpha_{LOS}$  is set to be the desired yaw angle ( $\psi_{ref}$ ). When the actual yaw angle is equal to  $\alpha_{LOS}$ , the speed vector of the vehicle is coincident with the line of sight and its direction points towards the desired yaw point. The difference of these two angles is called as the delta yaw ( $\Delta\psi$ ) and it should be zero during the flight. So, the desired way point always lies in front of the vehicle. The guidance algorithm calculates the desired heading angle of the quadrotor and keeps delta yaw at the zero to avoid being out of desired line.

$$\alpha_{LOS} = \text{atan2}\left(\frac{y_1 - y_0}{x_1 - x_0}\right) \quad (4.1)$$

$$\psi_{cmd} = \alpha_{LOS} \quad (4.2)$$

The desired way point in three dimensional space is projected into two different spaces which represent vertical position (*onedimensional*) and horizontal position (*twodimensional*). In order to reach the desired vertical position and horizontal position, the guidance algorithm adjusts  $w_{ref}$ ,  $u_{ref}$  and  $\psi_{ref}$ .  $u_{ref}$  is always kept positive, and the sign/ direction of horizontal speed vector with respect to earth is arranged by changing  $\psi$  angle. This means that the quadrotor always moves forward, toward to own its  $X$  axis with the speed  $u_{ref}$ . If it is necessary to fly back, it changes the  $\psi_{ref}$  angle and rotates itself. Meanwhile  $v_{ref}$  is always zero, which means that there is no lateral velocity of the vehicle.

$w_{ref}$  and  $u_{ref}$  speeds are calculated with respect to the distance between two successive way points and the total speed magnitude. At the beginning of the guidance procedure, the total distance between two way points ( $L$ ) and horizontal difference ( $H$ ) are calculated.

$$L = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (4.3)$$

$$H = z_1 - z_0 \quad (4.4)$$

It is aimed that the vehicle reaches both desired horizontal and vertical positions at the same time. Therefore, the forward speed ( $u_{ref}$ ) and the vertical speed ( $w_{ref}$ ) are adjusted to achieve this aim by taking into consideration of allowable total speed ( $u = \sqrt{u_{ref}^2 + w_{ref}^2}$ ). So the reference speeds are computed as follows;

$$\begin{aligned} w_{ref} &= |u| \frac{L}{H} \\ u_{ref} &= \sqrt{u^2 - w_{cmd}^2} \\ v_{ref} &= 0 \end{aligned} \quad (4.5)$$

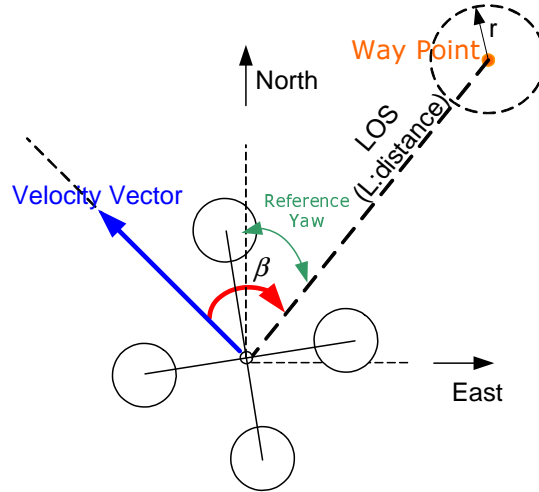


Figure 4.3: Velocity Vector and Reference Heading ( $\psi$ ) Angle

$w_{ref}$  can be either positive or negative according to the sign of  $H$ .  $u_{ref}$  is always positive and it is previously mentioned that the sign of the speed vector is arranged by the actual  $\psi$  angle.

It is shown at Figure 4.4, the yaw angle becomes zero at north, and it is defined in the region of the first or fourth quadrant positive, the second and third quadrant negative. There exists a discontinuity in the yaw angle between  $-\pi$  and  $\pi$ . If the actual and desired yaw angle signs are different, the system always passes over zero radians. However, the vehicle can reach the desired yaw angle through the opposite direction (clockwise and counterclockwise rotation). So it is desired that the system always prefers the shorter rotation. Therefore, an additional algorithm, which is called as yaw angle correction is inserted to the guidance algorithm. In Figure 4.4, the orange arrow shows the rotation without yaw angle correction. In that case, quadrotor would rotate more than  $\pi$  radians. Nevertheless, with the yaw angle correction (green arrows in the figure), the quadrotor always chooses a rotation and it is less than  $\pi$  (Figure 4.4).

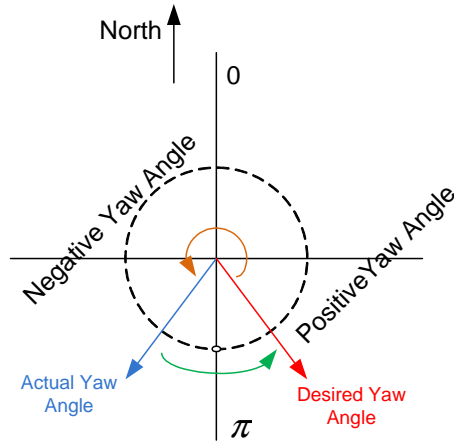


Figure 4.4: Yaw Angle Correction

Changing the yaw angle with a large amount can cause a stability problem, when the quadrotor has relatively high speed [39]. So, previously defined delta yaw can be used in an algorithm, which limits the forward velocity of the system while rotating (Figure 4.5). For example, if there is a sharp rotation between successive way points of the quadrotor velocity vector, the system firstly decreases its velocity and starts to change the heading angle. While reducing the delta yaw, the speed is increased again. Relationship between the forward speed and the delta yaw is expressed in Equation 4.6 ( $\kappa$  is a user defined constant).

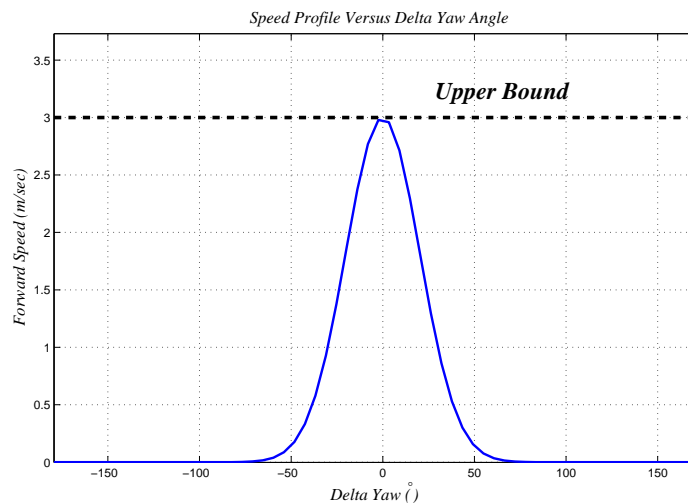


Figure 4.5: Delta Yaw vs. Forward Speed ( $\kappa = 0.5, u = 3m/s$ )

$$\begin{aligned}
 w_{ref} &= w_{ref} e^{(\Delta\psi/\kappa^2)} \\
 u_{ref} &= u_{ref} e^{(\Delta\psi/\kappa^2)}
 \end{aligned}
 \tag{4.6}$$

The model is designed in Simulink to simulate the guidance algorithm. Flight management and guidance blocks are inserted into the model (Figure: 4.7).  $R_{acc}$  depends on the quadrotor velocity and it is chosen as two times larger than the vehicle velocity ( $v$ ). In simulation, five way points and speed profiles are stored in the flight management. When the vehicle enters into a  $R_{acc}$ , trigger signal will be high, and the flight management sends new way point and speed information to the guidance block. Figure 4.6 shows the result of the simulated system.

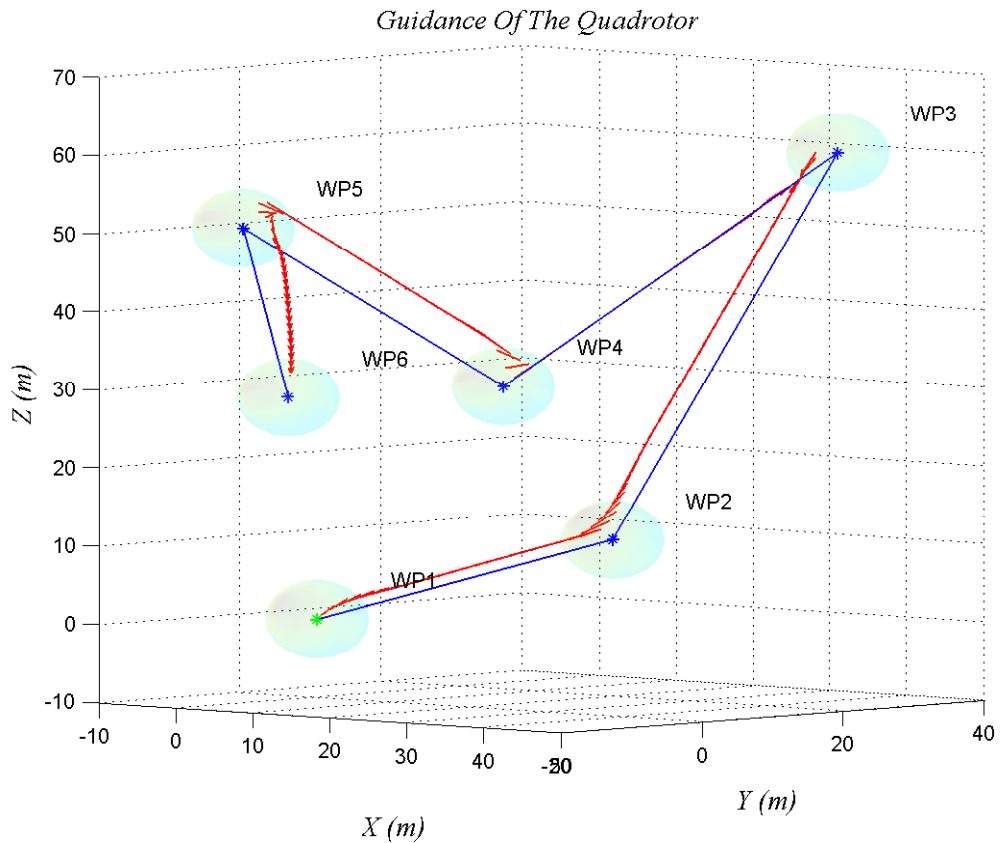


Figure 4.6: Guidance of Quadrotor

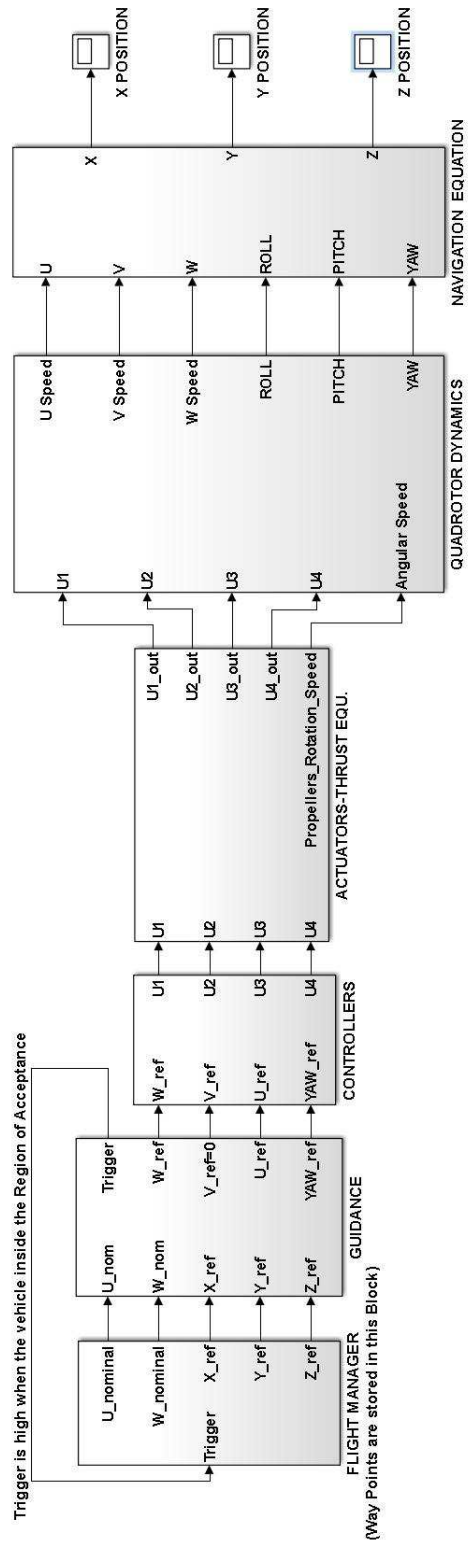


Figure 4.7: Simulink Model of Autonomous System

## **CHAPTER 5**

### **SYSTEM HARDWARE & SOFTWARE**

#### **5.1 Introduction**

RTF Quad Kit, which is designed by 3DRobotics Inc., is used to implement hardware applications. It is a ready-to-fly kit and the user needs to assemble the system components properly. The kit includes quadrotor frame, autopilot board, motors, ESCs, propellers, power boards and all necessary wirings. RC receiver/ transmitter, wireless telemetry, and battery are bought individually. In this chapter, all the system components are introduced.

#### **5.2 Hardware Components**

##### **5.2.1 Auto Pilot Board**

Ardu Pilot Mega (APM) is a complete open source autopilot system, which is based on Arduino. This compact system gives the users chance to develop own autopilot software. One can design an autopilot for any fixed wing planes, multi rotor vehicles, cars even boats with APM.

APM was introduced at 2011 and since then APM board has been used in many different areas; agriculture, mining, entertainment and scientific research [12].

APM has mainly three pin groups which are labeled as outputs, inputs, analogs. The outputs give PWM signals to drive motors via ESCs. The inputs are connected with

RC receiver. The analogs can be used as input or output. An additional sensor such as pitot, sonar, infrared range finder can be integrated by using these channels. They are also configured as digital outputs depending on the application like camera shutter [10]. Except these pin groups, there exists connector slots for GPS, wireless telemetry, power and micro USB connections.



Figure 5.1: APM 2.6

APM can be programmable with Arduino IDE. Main processor is Atmel's ATMEGA 2560 which has 16 channel, 10 bit A/D converter. [11]. The system has a 4 megabyte data flash chip for data logging.

The APM includes an InvenSense's six degree of freedom MEMS IMU (MPU-6000) which contains a 3 axis gyroscope, a 3 axis accelerometer, and a temperature sensor. Moreover, APM 2.6 also has a MEMS pressure sensor/ barometer (MS5611-01BA) that is used to measure altitude. [13]. The barometric sensor is covered by foam because it is sensitive to light. Furthermore, turbulence and propeller air stream can affect the sensor outputs.

### 5.2.2 Quadrotor Frame

Quadrotor frame contains four aluminum arms with carbon fiber legs and hub mounting plates. The frame is constructed according to the X configuration. The system components are placed carefully to make fixed center of gravity into quadrotor body frame. (Figure 5.6).



### 5.2.3 Magnetometer (Compass)

A magnetometer (Digital compass) gives heading information to the autopilot according to earth magnetic field. It is a highly sensitive sensor to magnetic interference, so it is mounted on the top of the quadrotor, away from the electrical circuit. Honeywell's HMC5883L 3 axis digital compass is used in the setup which has from  $1^\circ$  to  $2^\circ$  heading accuracy [14].



Figure 5.2: Three Axis Magnetometer

### 5.2.4 GPS

One can find his/her own position at any point on Earth by using the Global Positioning System (GPS). GPS was developed by U.S. Department of Defence at 1970s and since 1996 it has been used at both civil and military applications. Its accuracy depends on many parameters and an exact location can be determined within a range of 20 m. Speed and direction of travel (course) can be derived from GPS time and position data about  $0.1m/s$  precision. Depending on the communication protocol and message type, GPS gives position and speed information in geodetic, or ECEF coordinate frame.

For fully autonomous mission, GPS must be utilized. There is no sensor that can give information directly about vehicle's speed and position with respect to ground. However, GPS update rate is about 1 Hz and it is slow for the control system. Therefore, the orientation should be estimated by combining GPS, barometer, and accelerometer. GPS and barometer signals corrects and calibrates the accelerometer outputs with the use of a mathematical algorithm such as the Kalman Filter. The accelerometer has an unknown bias that causes a drift in velocity and position estimation for long term.

Thus, it provides an accurate estimation only for a short period.

3D robotics Ublox LEA-6 GPS module is easily connected to APM 2.6 through SPI bus. The GPS module should be mounted away from noise sources like the magnetometer.

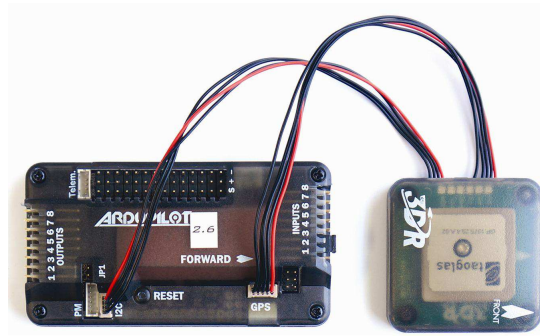


Figure 5.3: Connection of GPS to APM 2.6

### 5.2.5 Motors & Propellers

RTF Quad Kit includes outrunner brushless DC motors (BLDC) as actuators. A BLDC motor does not have brushes for commutation; instead it is electronically commutated. This brings some advantages to BLDC motor over brushed DC motor. The BLDC is highly efficient and has a longer lifetime, lower noise, better top speed [54]. In the quadrotor, 2 pole-pair 850 Kv AC2830 – 358 motor is placed on the tip of the each arm. The abbreviations Kv describe the motor RPM value for per volt. For example, if this system BLDC motor is 850 Kv and is supplied by 7 volts, it rotates 5950 RPM ( $7 \times 850$ ) without loading. But in the real case, this number will be less due to friction.

The outrunner term refers to outer permanent magnet shell spins around stationary coil on the motor shaft. They are suitable for small aircrafts, because of their high power to weight ratios.

APC  $10 \times 47$  is used in the system. A propeller is named by two numbers; the first number represents the diameter of the propeller in inches; the second one is the pitch value. Theoretically the propeller diameter is proportional to the thrust value and

the pitch value is the forward movement of the propeller during one rotation which shows how much rotor blade is tilt from the rotor shaft. If the pitch rate increases, the quadrotor will be more aggressive but have poor stability.

In the quadrotor, there exists two types of propeller. Neighboring propellers should rotate to opposite directions to balance yaw torque while the thrust vectors always point upwards. Therefore, propellers are designed with a special geometry and they are called as pusher and normal.

### 5.2.6 RC Transmitter & Receiver

Radio Control (RC) system uses the radio signal to control a device at a distant place. Thus, they are used for the manual flight of the quadrotor. The reference signal is converted to radio signals by the transmitter. Then, the receiver at the vehicle collects these signals and generates appropriate PWM values to drive the vehicle. Before the first flight, the transmitter and the receiver should be bound together, so that the receiver recognizes own transmitter signals.

2.4 GHz Spectrum DX7s 7 channel transmitter and AR8000 receiver are used in this thesis.



Figure 5.4: RC Transmitter & Receiver

### 5.2.7 Electronic Speed Controller

Electronic speed control (ESCs) is a device which controls the rotational speed of the BLDC motor. It is frequently used in RC models. 20 A continuous current SimonK

ESC is used in this thesis. Control pins of the ESC are connected to APM's output. The autopilot sends the desired PWM values for each motors and the ESC drives the motors according to these values. Furthermore, SimonK ESC includes battery eliminator circuit (BEC). The BEC can supply the power for other components such as APM 2.6, GPS, RC receiver etc.

### 5.2.8 Wireless Telemetry

A wireless telemetry provides communication between the quadrotor and the ground station. It is used for the system's data logging. The IMU, GPS, compass, and controller outputs can be transmitted to the computer for analyzing of the system. Furthermore; during the autonomous flight, new missions can be uploaded via wireless telemetry (i.e., new way points, speed profile, landing command etc.).



Figure 5.5: Connection of Wireless Telemetry Kit to APM 2.6

### 5.2.9 Battery

The battery gives energy to the system. For the UAV system, the weight and capacity of the battery bound the flight time and agility. The lithium polymer (LiPo) batteries are the most common power supply for the UAV because they have very good power to weight ratio and they are lighter than NiCad or NiMH batteries.

The LiPo battery is mainly classified according to cell numbers, cell connection, and discharge rate. Each LiPo cell provides nominally 3.7 V and the cells can be con-

nected with both parallel or serial which determine the voltage and current specifications of battery. Capacity of the battery is expressed by milliampere rate (mAh) which show how much current is supplied to the circuit for one hour after the battery is fully charged. Discharge rate simply indicates how fast the battery is discharged safely.

In quadrotor setup, the motors are the main load and the other components consume very low power. In the motor's datasheet maximum power is given 2000 W and nominal voltage of the LiPo battery is 11.1 V. Then, the maximum current can be estimated around 17 A for the motor. This equals to 68 A for four motors. A battery with the capacity 3000 mAh and discharge rate 25 C can safely supply 75 A and the value is sufficient for the system.

#### 5.2.10 Other Components

The power distribution board, which is also designed by 3D Robotics, transfers the power to the ESCs. The power module contains a regulator circuit and supplies unrippled power for APM 2.6, GPS, magnetometer, wireless telemetry and RC transmitter [12].



Figure 5.6: Fully Assembled Quadrotor System

### 5.3 Software

Arduino IDE is the development platform for the autopilot. The C++ code is written and uploaded to the APM 2.6 with the use of Arduino IDE. APM has an open source autopilot code and more than 15000 developers donated the autopilot. Therefore, its libraries are pretty reliable. While developing the custom autopilot, some libraries are used in autopilot such as AHRS.h and InertialNav.h which are associated with state estimation

Arduino IDE does not provide powerful debugging tools. So that, a C# program is prepared to read serial port for debugging and data logging. During the tests, data is stored within PC with the use of this program.

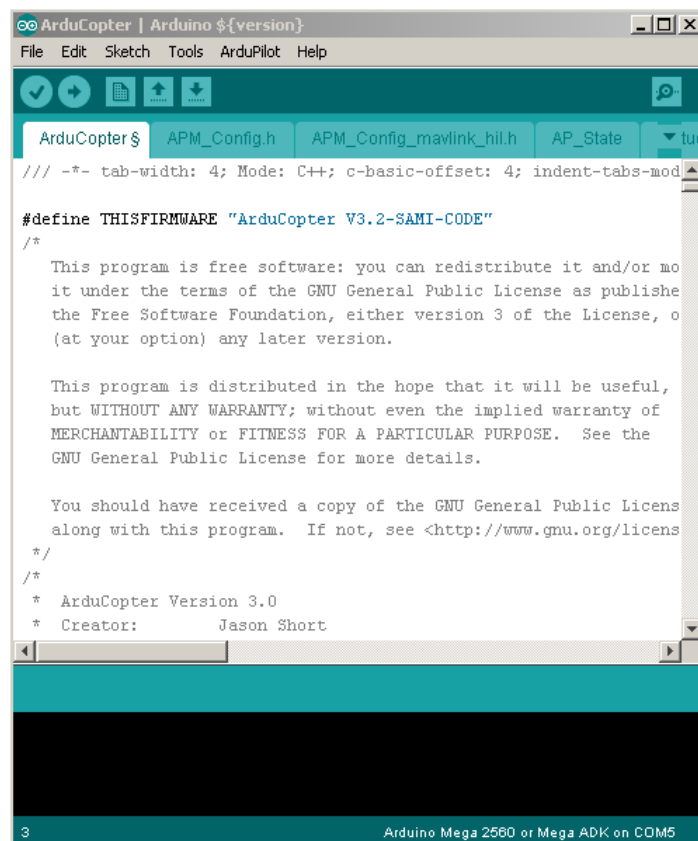


Figure 5.7: Arduino IDE

## CHAPTER 6

### HARDWARE IMPLEMENTATIONS

This chapter includes the hardware implementation of the autopilot which provides the operator the means to control the system with a remote controller. Not all of the autopilot algorithms and simulated scenarios were implemented on real hardware due to both the lack of time and some sensor problems encountered.

As a first step, a code is written with C# to log the vehicle's data. Using wireless telemetry, desired data can be transmitted to the computer at 115200 *bit/sec*. A buffer stores the data packets and the program reads all buffer contents each 100 milliseconds. System messages should be sent from the vehicle in a proper structure. Each message carries special marker, so that the data can be parsed. Furthermore, controller parameters, saturation limits, filter coefficients and reference signals can be sent to the vehicle with telemetry. It is very useful that the controller parameters can be adjusted while the system is being tested without rebuilding the autopilot code.

Before the controller design, a test setup was built to secure both the environment and the vehicle (Figure 6.1). The test setup has a rotary joint with three degrees of freedom. The quadrotor is mounted on a plate at the top of the rotary joint, on which quadrotor moves freely. Attitude controllers are tested with the use of this setup.



Figure 6.1: Test Setup for Attitude Control

Autopilot code is written with C++ in Arduino IDE. Before the controller design, system infrastructures should be build. Initially, RC signals are gathered as the inputs. Five channels of RC are necessary to control the vehicle. Mission of the channels are listed as follows;

- **First Channel** controls the roll angle of the vehicle. PWM output of the channel is converted to desired roll angle, then cascaded roll control provides desired angle and system stability. The roll angle output is bounded between  $-30^\circ$  and  $30^\circ$ .
- **Second Channel** controls the pitch angle of the vehicle. Operating principles are completely the same as the first channel.
- **Third Channel** controls the thrust value and it is responsible for  $U_1$ . Due to the barometer problem, altitude and climb/ descent rate data are not collected. Therefore, the controller loops for the horizontal motion are not constructed. However, the autopilot code arranges the PWM values, when the third channel stick of RC is in the middle position, the system holds its own at hover.



- **Fourth Channel** controls the yaw rate of the vehicle. During manual use controlling the yaw angle directly is not desirable. The RC stick position determines the rotational speed of the yaw angle ( $\dot{\psi}$ ). The channel output is mapped between  $-15^\circ/sec$  and  $15^\circ/sec$ . Desired yaw rate is a reference value for yaw rate controller.
- **Fifth Channel** is not obligatory for the system. These three positioned channel allows users to change the flight modes.

After the RC signals are collected and mapped into appropriate values, sensor data should be obtained properly. It is the most significant step of the autopilot; outputs of the gyroscope are used as feedback signals for the rate loop controllers ( $p$ ,  $q$ ,  $r$ ). Furthermore, gyroscope outputs are combined with the accelerometer outputs to obtain the Euler angles. The roll and pitch angles are obtained by using a pre-built library called AHRS.lib. It consists of an attitude estimation algorithm based on Kalman filter approach.

The next step is the implementation of the motor inversion block. This block takes the control signals and distributes the necessary PWM values to each motor with respect to X configuration. PWM values become the outputs of the autopilot and they are sent to ESCs to drive the motors. Associated code is given in Figure 6.2 ( $U1$  is the value of channel 3 of RC,  $U2$ ,  $U3$  and  $U4$  is the control signals,  $PWM_i$  is the  $i^{th}$  motor PWM value and  $K\_CT2PWM$  is the constant gain and it transforms the control signals to the PWM values).

```

////////////////////////////////////
////////////////////////////////////Controller to Motor Conversion////////////////////////////////////
////////////////////////////////////
PWM_1= U1 -K_CT2PWM*U2 +K_CT2PWM*U3 +K_CT2PWM*U4;
PWM_2= U1 +K_CT2PWM*U2 -K_CT2PWM*U3 +K_CT2PWM*U4;
PWM_3= U1 +K_CT2PWM*U2 +K_CT2PWM*U3 -K_CT2PWM*U4;
PWM_4= U1 -K_CT2PWM*U2 -K_CT2PWM*U3 -K_CT2PWM*U4;

```

Figure 6.2: Motor Inversion Block of the Autopilot

The final step is the design of the controllers. In Chapter 3, it is explained in detail that cascade PI loops are built on the system. For manual flight, the roll/ pitch angle controllers include two cascaded loops and the yaw rate controller includes only one loop. Sampling rates of the inner and outer loops are adjusted at 100 Hz and 50 Hz, respectively. The controller parameters are adjusted by hand tuning and they are not coherent with MATLAB simulations. This situation is expected due to several reasons; the main reason is that relationships between motor input PWMs and propeller rotational speeds are uncertain. Moreover, the motor-propeller model is not available and their dynamics are neglected in MATLAB/ Simulink simulations. Despite all these uncertainties, cascaded PI loop structure is able to stabilize the quadrotor on the test setup and track the reference signals. During the tuning procedure, tilt angles were observed to oscillate. Therefore a D term is added. Consequently the oscillation is substantially eliminated. When D term is used, instantaneous change of error can cause an infinite (in practice, a very large) output. Hence, a low pass filter with a cut-off frequency of 10 Hz is added in order to limit high frequency gain and noise. In the last case, the structure of designed controllers are PID for the inner loops (the roll and pitch rate control) and PIs for the outer loops (the roll and pitch control) and the yaw rate loop. In Figure 6.3, discrete implementation of the PID controller is given.

After the autopilot implementation is completed, reference signals are commanded to observe the system performance. The following graphs are given when all reference inputs are zero as if the system is at the hover condition. Furthermore, input disturbance effect is shown under these conditions. RC is not appropriate to produce exactly zero degree for angle reference. Therefore, the input signals are commanded from a computer via wireless telemetry.

```

////////////////////////////////////
////////////////////////////////////PID CONTROLLER////////////////////////////////////
////////////////////////////////////

float pid_compute ( float Kp, float Ki, float Kd, float Sat, ...
                  float Ts, float error, float& Intg_Sum, ...
                  float error_previous, float& D_term_previous)
{
    float P_term, I_term, D_term, alpha_d;
    float RC_d=0.016; // Filter Bandwidth for D term (cut-off: 10 Hz)
    alpha_d=Ts/(RC_d+Ts);

    P_term = Kp*error; // P term is calculated

    Intg_Sum = Intg_Sum + error*Ts; // Error is accumulated

    if(Intg_Sum > Sat) // integral max. saturation
        Intg_Sum = Sat;

    if(Intg_Sum < -Sat) // integral min. saturation
        Intg_Sum = -Sat;

    I_term=Ki*Intg_Sum; // I term is calculated

    D_term=Kd*((error-error_previous)/Ts); // D term is calculated

    D_term=alpha_d*D_term+(1-alpha_d)*D_term_previous
    // D Term is filtered

    D_term_previous=D_term; // Previous D term is
    //kept for the next filtering

    return P_term+I_term+D_term; // PID output is calculated
}

```

Figure 6.3: PID Implementation of the Autopilot

Figure 6.4 shows that the roll angle is kept at zero degree with the error and it is less than 1°. The pitch angle controller is more successful than the roll angle; the error is nearly 0.5° (Figure 6.5). the Yaw rate error is perturbed into  $\pm 1.5$ °/sec which is shown in Figure 6.6.

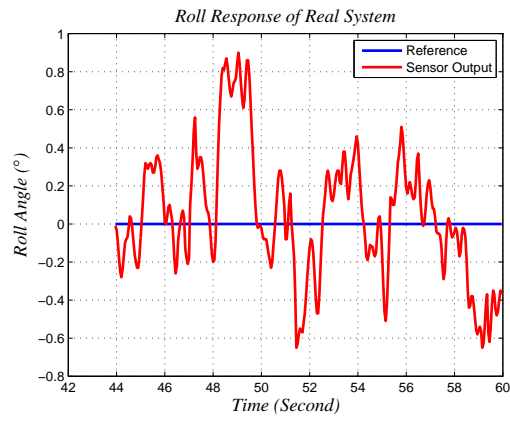


Figure 6.4:  $\phi = 0$  Response of the Real System

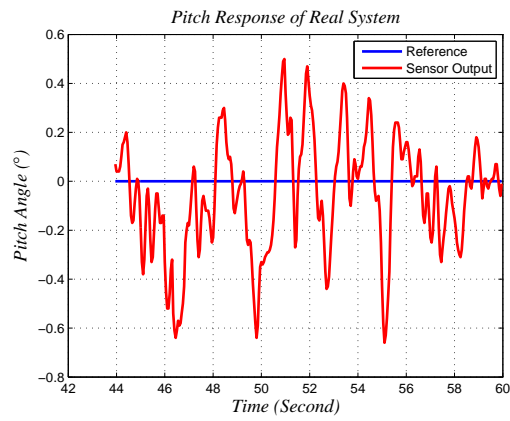


Figure 6.5:  $\theta = 0$  Response of the Real System

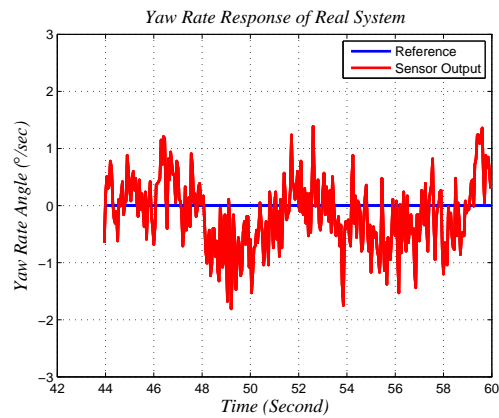


Figure 6.6:  $\dot{\psi} = 0$  Response of the Real System

Disturbances are given to the system by hand. The experiments show that the system is quite robust with respect to disturbances. The roll angle behaviour of the system is shown in Figure 6.7. Peaks in figure indicates instants of disturbance. As a result, it can be said that the system secures its own stability against the disturbance.

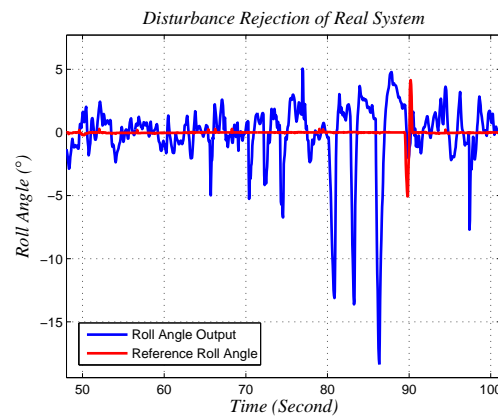


Figure 6.7: Disturbance Rejection of the Real System

In the next steps, various reference signals are command to examine the system's reference tracking. The desired roll/ pitch angle and the yaw rate commands are given individually via the RC.

It is seen that the roll/ pitch angle controller has good reference tracking ability (Figure 6.8 and Figure 6.9). The same result is observed for the yaw rate response of the system (Figure 6.10).

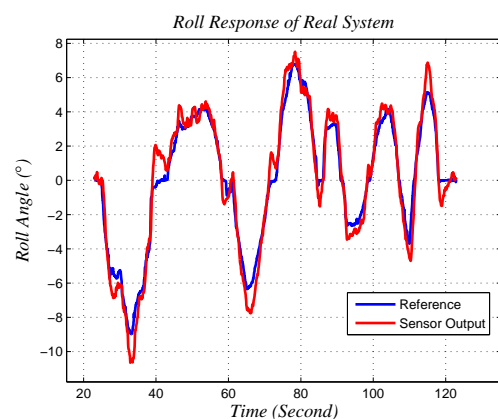


Figure 6.8: Roll Angle Tracking of the Real System

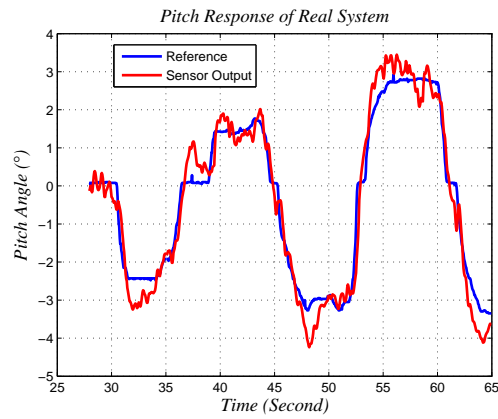


Figure 6.9: Pitch Angle Tracking of the Real System

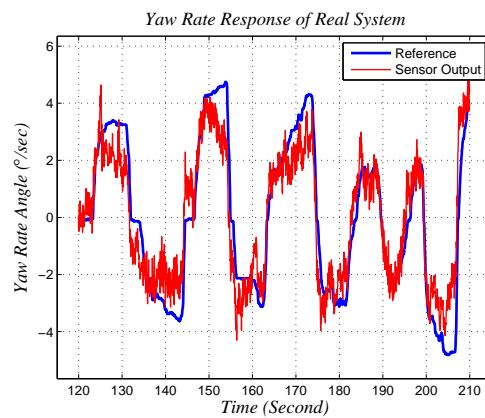


Figure 6.10: Yaw Rate Tracking of the Real System

As a result, the quadrotor is stable and follows the RC reference signals. They are sufficient for the manual flight. Autonomous mission tasks will be conducted after the barometer and compass work properly. There is an incompatibility between sensor outputs and expected real values. Furthermore, the wireless telemetry connection may be lost, when the distance becomes more than 30 m between ground station (PC) and the quadrotor. After these problems are solved, the linear velocity controller and the guidance algorithm will be designed, so that the vehicle can manage to carry out autonomous missions.

## **CHAPTER 7**

### **CONCLUSION**

This chapter is reserved for main results and contributions of this thesis. Furthermore, the challenges, which were confronted during the study, and the future work will be mentioned.

The area of UAV research has been gaining more importance recently. This project was created within this context. It is aimed to understand quadrotor dynamics through modelling, simulation and analysis. An essential view about the autopilot, vehicle dynamics, aerodynamics, control designs, embedded software application and hardware knowledge has been acquired.

The system kinematics and dynamics were obtained. Various forces and torque effects were investigated. Especially, the propeller gyro effect for an X configuration quadrotor was explained in detail. After the modelling, twelve-state equations of the system were obtained. Three of them were associated with the guidance and navigation and they were redundant when the controllers were designed. The others were used during linearization and controller design procedures.

General information about trim conditions and their calculations were analysed. In this thesis, despite the fact that linear controller was based on only one trimmed state, the states and inputs of the system were investigated for different operation points. It was observed that a single trim state is sufficient to design linear controllers for the quadrotor when the speed profile was not so harsh.

An autopilot system which includes a guidance-navigation-control algorithm was built in MATLAB/ Simulink environment. Relationships between the blocks were

explained. The guidance basics were studied and a simple LOS algorithm was implemented on MATLAB/ Simulink environments.

Different control structures were implemented on the system: PI in cascaded form, LQR control for system type 1 and sliding mode control. PI was the simple one to be implemented on hardware, however its performance highly depends on the model accuracy. LQR mainly intends to minimize the system energy and sliding mode is a robust technique against disturbance and the model uncertainty.

At the end of this study, a powerful basis about controller design was obtained. In hardware implementation of PI controller; pre filter and D term filter effects were seen very clearly. Furthermore, the importance of integral saturation was observed. During the hardware study, controller coefficients were mostly found by tuning operations. After the constructed PI-PID cascaded loops, several tunings were made for a while and later the parameters, which had sufficient responses, were obtained.

3DRobotics' ready-to-fly kit made it possible to study on a trustworthy hardware. So, the designer was able to focus on investigation of the system dynamics and the control implementation. Thereby, no time was consumed for the construction of the vehicle. All the components of the system were easily combined with each other. APM 2.6 is a very versatile autopilot board which has many features. However, some other autopilot boards are also available today. They have better micro controllers and various sensors and features. Also they have more empty places for coding and data logging. It is expected that for a complicated autopilot code, APM 2.6 will be inadequate.

One of the main benefits of this thesis was that a wide ability was acquired about the embedded systems. During the study, the basics about real time applications and programming micro controllers were found out. Furthermore, C++ and C# programming languages are studied to design autopilot software. The autopilot code was written with C++ and data logging program was prepared with C#. This was a tough process and the educational attainments were quite high. RC inputs, IMU and GPS's outputs were collected properly, loop configuration was arranged and motors were driven by the autopilot code. Another important step was to clarify the relationship between the controller outputs and the motors.



During the experiments, it was seen that test setup increased the system inertia. The parameters, which were configured at the test setup, were not proper for flight testing. After getting sufficiently good results and making sure that the controller structure worked properly, real flight tests were carried out. At the end of the thesis; attitude control was completed with cascaded loops on the real system and verified control algorithm on both test setup and real flight.

Motor thrust/ drag behaviours were not derived experimentally due to the lack of thrust measurement setup. If the behaviours had been revealed, consistency between MATLAB simulations and hardware implementations might have been observed. Obtaining the altitude output of the barometer, heading information of the magnetic compass and GPS position outputs should be completed. For now, there is an incompatibility between sensors outputs and real values. Furthermore, there is a problem, which should be solved in wireless telemetry. When the distance was more than 30 *m* between the vehicle and PC, the wireless telemetry connection was lost (expected distance: 700 *m* – 1300 *m*). This is another problem which should be overcome.

For future works, first of all, the attitude control of the system will be completed by using LQR and sliding mode control. The compass/ barometer outputs will be secured, and also it may be possible to add a SONAR to measure the altitude correctly. After sensor improvement, controller for linear motion will be constructed. Finally, a simple guidance algorithm will be implemented. So, the vehicle will be able to fly between way points in a straight line. During autonomous mission study, it will be also necessary to analyse vehicle take off and landing.



## REFERENCES

- [1] <http://www.vuse.vanderbilt.edu/adamsja/Courses/Robotics/Lectures/UAVs.pdf>, last accessed at 02.11.2014.
- [2] <http://terpconnect.umd.edu/leishman/Aero/Breguet.pdf>, last accessed at 05.11.2014.
- [3] [http://www.aviastar.org/helicopters\\_eng/oemichen.php](http://www.aviastar.org/helicopters_eng/oemichen.php), last accessed at 05.11.2014.
- [4] [www.scholarpedia.org/article/State\\_Space](http://www.scholarpedia.org/article/State_Space), last accessed at 05.11.2014.
- [5] Topic #18 deterministic lqr, lecture notes. [http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16\\_30F10\\_lec18.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec18.pdf), last accessed at 05.12.2014.
- [6] [http://www.aviastar.org/helicopters\\_eng/bothezat.php](http://www.aviastar.org/helicopters_eng/bothezat.php), last accessed at 06.11.2014.
- [7] [http://www.aviastar.org/helicopters\\_eng/convertawings.php](http://www.aviastar.org/helicopters_eng/convertawings.php), last accessed at 06.11.2014.
- [8] <http://www.boeing.com/boeing/rotorcraft/military/v22/>, last accessed at 06.11.2014.
- [9] Linear quadratic optimal control. <http://www.me.umn.edu/courses/me8281/notes/Old%20S06/Chapter%206%20LQ%20Control.pdf>, last accessed at 10.09.2014.
- [10] <http://copter.ardupilot.com/wiki/common-apm25-and-26-overview/>, last accessed at 11.09.2014.
- [11] <http://atmel.com/device/atmega2560.aspx>, last accessed at 15.08.2014.
- [12] The apm universal autopilot. <http://3drobotics.com>, last accessed at 15.08.2014.
- [13] Measurement Specialities, MS5611-01BA03 Barometric Pressure Sensor. last accessed at 20.08.2014.
- [14] Honeywell HMC5883L 3-Axis Digital Compass IC. last accessed at 21.08.2014.

- [15] K. S. A. Das, F. L. Lewis. Sliding mode approach to control quadrotor using dynamic inversion. <http://cdn.intechopen.com/pdfs-wm/23869.pdf>, last accessed at 18.10.2014.
- [16] P. Adigbli. Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques. *3<sup>rd</sup> US-Europeana Competition and Workshop on Micro Air Vehicle Systems*, 2007.
- [17] E. Altug and R. Mahony. Control of a quadrotor helicopter using visual feedback. *IEEE International Conference on Robotics and Automation*, 2002.
- [18] A. C. Andrew. Octocopter aerodynamics and control. *National University of Singapore*, 2012.
- [19] C. Balas. Modelling and linear control of a quadrotor. *Cranfield University*, 2007.
- [20] M. Bangura and R. Mahony. Nonlinear dynamic modelling for high performance control of a quadrotor. *Australasion Conference of Robotics and Automation*, 2012.
- [21] R. W. Beard. Quadrotor dynamics and control. *Brigham Young University*, 2008.
- [22] J. L. Boiffier. The dynamics of flight the equations. *John Wiley & Sons*, 1998.
- [23] S. Bouabdallah. Design and control of quadrotors with application to autonomous flying. *Ecole Polytechnique Federale De Lausanne*, 2007.
- [24] S. Bouabdallah and R. Siegwart. Sliding mode control based on backstepping approach for an uav type-quadrotor. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 2005.
- [25] H. Bouadi and M. Bouchoucha. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 2007.
- [26] T. Bresciani. Modelling, identification and control of a quadrotor helicopter. *Lund University*, 2008.
- [27] T. Buchholz and D. Gretarsson. Construction of a four rotor helicopter control system. *Technical University of Denmark*, 2009.
- [28] G. Cai. Unmanned rotorcraft systems. *Springer*, 2011.
- [29] R. D. Carlo and S. Zak. Variable structure, sliding-mode controller desing. *Control System Advanced Methods*, 2011.
- [30] F. Y. Cevher. Autopilot and guidance design for a mini rov (remotely operated underwater vehicle). *Middle East Technical University*, 2012.

- [31] I. Cheeseman and W. Bennett. The effect of the helicopter rotor by ground on a forward flight. *Ministry of Supply Aeronautical Research Council and Memoranda*, 1957.
- [32] M. J. Cutler. Design and control of an autonomous variable pitch-control quadrotor helicopter. *MIT*, 2012.
- [33] R. Czyba and G. Szafranski. Control structure impact on the flying performance of the multi-rotor vtol platform-design, analysis and experimental validation. *Internation Journal of Advanced Robotic Systems*, 2012.
- [34] B. Erginer and E. Altug. Modeling and pd control of a quadrotor vtol vehicle. *IEEE Intelligent Vehicles Symposium*, 2007.
- [35] M. S. Fadali. Sliding mode control, lecture notes. <http://wolfweb.unr.edu/fadali/EE776/Sliding-Mode%20Control.pdf>, last accessed at 21.10.2014.
- [36] G. Fay. Derivation of the aerodynamic force for the mesicopter simulation. <http://adg.stanford.edu/mesicopter/ProgressReports/mesicopteraeromodel.pdf>, last accessed at 15.03.2014, 2001.
- [37] T. I. Fossen. Guidance and control of ocean vehicles. *John Wiley & Sons*, 1999.
- [38] G. Hoffmann and H. Huang. Quadrotor helicopter flight dynamics and control theory and experiment. *AIAA Guidance Navigarion and Control Conference and Exhibit*, 2007.
- [39] T. M. Jensen. Waypoint-following guidance based on feasibility algorithms. *Norwegian University of Science and Technology*, 2011.
- [40] T. Madani and A. Benallegue. Control of a quadrotor mini-helicopter via full state back-stepping technique. *IEEE International Conference on Decision and Control*, 2006.
- [41] A. D. Marco. General solution to the aircraft trim problem. 2007.
- [42] A. Neff. Linear and non-linear control of a quadrotor uav. *The Graduate School of Clemson University*, 2007.
- [43] M. Oliviera. Modelling, identification and control of a quadrotor aircraft. *Czech Technical Univesity in Prague*, 2011.
- [44] M. Orsag. Influence of forward and descent flight on quadrotor dynamics. *Recent Advances in Aircraft Technology*, 2012.
- [45] H. A. Oyman. Kayan kipli denetim yönteminin 2 serbestlik dereceli helikopter sistemine uygulanması. *Boğaziçi Üniversitesi*, 2012.

- [46] A. H. Pedersen. Test and modelling of four rotor helicopter rotors. *Technical University of Denmark*, 2006.
- [47] J. Seddon and S. Newman. Basic helicopter aerodynamics. *Blackwell Science*, 2002.
- [48] J. Slotine. Applied nonlinear control. *Prentice Hall*, 1991.
- [49] I. Sonnevend. Analysis and model based control of a quadrotor helicopter. *Budapest Pazmany Peter Catholic University*, 2010.
- [50] A. Swarup and Sudhir. Comparison of quadrotor performance using backstepping and sliding mode control. *International Conference on Circuits, Systems and Controls*, 2014.
- [51] S. M. A. W. Naeem, R. Sutton and R. S. Burns. A review of guidance laws applicable to unmanned underwater vehicles. *The Journal of Navigation*, 56(1):15–29, 2003.
- [52] M. Wierema. Design, implementation and flight test of indoor navigation and control system for a quadrotor uav. *TU Delft*, (2008).
- [53] R. Xu and U. Özgüner. Sliding mode control of a class of underactuated system. *Automatica*, 44(1):233–241, 2008.
- [54] P. Yedamale. Microchips AN885 BLDC Motor Fundamentals. last accessed at 17.08.2014.
- [55] S. Ying. Advanced dynamics. *AIAA Education Series*, 1997.