TRAJECTORY TRACKING OF A QUADROTOR UNMANNED AERIAL VEHICLE (UAV) VIA ATTITUDE AND POSITION CONTROL

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

BY

EMRE CAN SUİÇMEZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

JULY 2014

Approval of the thesis:

TRAJECTORY TRACKING OF A QUADROTOR UNMANNED AERIAL VEHICLE (UAV) VIA ATTITUDE AND POSITION CONTROL

submitted by **EMRE CAN SUİÇMEZ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp _____
Head of Department, **Aerospace Engineering**

Assist. Prof. Dr. Ali Türker Kutay _____
Supervisor, **Aerospace Enginnering Department, METU**

Examining Committee Members:

Assoc. Prof. Dr. Melin Şahin
Aerospace Engineering Department, METU _____

Assist. Prof. Dr. Ali Türker Kutay
Aerospace Engineering Department, METU _____

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Mustafa Kaya
Flight Training Department, UTAA _____

Dr. Sebahattin Topal
Space Technologies Research Institude, TÜBİTAK _____

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: EMRE CAN SUİÇMEZ

Signature :

ABSTRACT

TRAJECTORY TRACKING OF A QUADROTOR UNMANNED AERIAL VEHICLE (UAV) VIA ATTITUDE AND POSITION CONTROL

SUİÇMEZ, EMRE CAN
M.S., Department of Aerospace Engineering
Supervisor : Assist. Prof. Dr. Ali Türker Kutay

July 2014, 109 pages

In this thesis, trajectory tracking of a quadrotor UAV is obtained by controlling attitude and position of the quadrotor simultaneously. Two independent control methods are used to track desired trajectories accurately. One of these methods is a nonlinear control approach called as "backstepping control". The other method is a more unique optimal control approach called as "Linear Quadratic Tracking(LQT)". In addition, fixed-gain LQR controller which is widely used in literature is also used for comparison analysis. First, nonlinear dynamic model of quadrotor is obtained by using Newton's equations of motion. Then, backstepping controller is obtained in three steps and simulation model of the backstepping controller is formed. On the other hand, time-varying optimal control gains of LQT controller are found offline by solving matrix difference Riccati equation(DRE) backwards in time. Then, LQT controller is modeled by using time-varying optimal control gains as a state feedback controller. Several trajectories to be followed are generated in MATLAB and sent into the simulation models as inputs. Finally, backstepping, LQT and LQR controllers are simulated in MATLAB/Simulink environment, for inital validation. Several trajectories are tried to be followed by each controller and simulation results of controllers are compared to each other. It is observed that, LQT controller could track relatively complex trajectories more accurately and efficiently compared to backstepping and LQR controllers. Other advantageous and disadvantageous characteristics of each

control method are also analyzed in details. In this thesis, "AscTech Hummingbird" quadrotor manufactured by Ascending Technologies is used. "AscTech Hummingbird" quadrotor gives opportunity to test high level control algorithms generated in MATLAB/Simulink environment. Therefore, complete validation of controllers obtained in this thesis could be performed by real time experiments in future.

Keywords: UAV, Quadrotor, trajectory/path tracking, nonlinear control, optimal control, disturbance rejection, backstepping, LQT, LQR

ÖZ

DÖRT ROTORLU BİR İNSANSIZ HAVA ARACININ (İHA) YÖNELİM VE POZİSYON KONTROLÜ ARACILIĞIYLA YÖRÜNGE TAKİBİ

SUİÇMEZ, EMRE CAN
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Ali Türker Kutay

Temmuz 2014 , 109 sayfa

Bu tez çalışmasında, dört rotorlu bir İHA'nın yörünge/yol takibi, dört rotorlu'nun yönelimi ve pozisyonu aynı anda kontrol edilerek elde edilmiştir. İstenilen yörüngelerin yüksek doğrulukla takip edilmesi için, birbirinden bağımsız iki farklı kontrol yöntemi kullanılmıştır. Bu yöntemlerden biri "geri adımlamalı kontrol" olarak adlandırılan doğrusal olmayan bir kontrol yaklaşımıdır. Diğer yöntem ise "Linear Quadratic Tracking(LQT)" olarak adlandırılan daha özgün bir optimum kontrol yaklaşımıdır. Ayrıca, literatürde yaygın olarak kullanılan sabit-kazançlı LQR(Linear Quadratic Regulator) kontrolcü karşılaştırma analizlerinde kullanılmıştır. İlk olarak, dört-rotorlu'nun doğrusal olmayan dinamik modeli Newton hareket denklemleri kullanılarak elde edilmiştir. Sonrasında, geri adımlamalı kontrolcü üç aşamada elde edilmiştir ve geri-adımlamalı kontrolcünün simülasyon modeli oluşturulmuştur. Diğer taraftan, LQT kontrolcünün zamana göre değişen optimum kontrol kazançları matris DRE zamanda tersine çözülerek çevrimdışı olarak elde edilmiştir. Sonrasında, LQT kontrolcü, zamana göre değişen optimum kazançları durum geribeslemeli kontrolcü olarak modellenmiştir. Takip edilecek olan çeşitli yörüngeler MATLAB kullanılarak oluşturulmuş ve simülasyon modellerine girdi olarak gönderilmiştir. Son olarak, geri adımlamalı, LQT ve LQR kontrolcülerin ilk doğrulanması amacıyla, kontrolcüler MATLAB/Simulink ortamında benzetilmiştir. Çeşitli yörüngeler her bir kontrolcü tarafından takip edilmeye çalışılmıştır ve kontrolcülerin simülasyon sonuçları birbiriyle karşılaş-

tırılmıştır. LQT kontrolcünün, görece kompleks yörüngeleri geri adımlamalı ve LQR kontrolcülere göre daha doğru ve verimli olarak takip edebildiği gözlenmiştir. Ayrıca, her bir kontrolcünün avantajlı ve dezavantajlı özellikleri detaylı olarak analiz edilmiştir. Bu tez çalışmasında, "Ascending Technologies" firması tarafından üretilen "AscTech Hummingbird" dört rotorlu insansız hava aracı kullanılmıştır. "AscTech Hummingbird" dört rotorlu hava aracı, MATLAB/Simulink ortamında oluşturulan yüksek seviye kontrol algoritmalarının test edilebilmesine olanak sağlamaktadır. Bu nedenle gelecek çalışmalarda, bu tez çalışmasında elde edilen kontrolcülerin tam anlamıyla doğrulanması gerçek zamanlı deneyler aracılığıyla sağlanabilir.

Anahtar Kelimeler: İHA, Dört rotorlu, yörünge/yol takibi, doğrusal olmayan kontrol, optimum kontrol, bozucu bastırma, geri-adımlama, LQT, LQR

To my family and people who are reading this page

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor Assist. Prof. Dr. Ali Türker Kutay for its valuable guidance, support and advice throughout my research and education.

I would also like to express my gratitude to my thesis committee members, Assoc. Prof. Dr. Melin Şahin, Prof. Dr. Kemal Leblebicioğlu, Assoc. Prof. Dr. Mustafa Kaya and Dr. Sebahatin Topal for their significant reviews and comments.

I would like to express my special appreciation to Prof. Dr. Kemal Leblebicioğlu, Prof. Dr. Erol Kocaoğlan, Prof. Dr. Mustafa Kuzuoğlu and all other academic staff of METU for their instructive and rewarding lectures.

I would like to thank entire academic staff of Department of Aerospace Engineering. I'm also grateful to administrative staff of Aerospace Engineering Department and special thanks to Derya Kaya and Nilgün Kaplan for their understanding and supportive attitudes.

Special thanks to my dear friend and colleague Emre Yılmaz for his support and sincere friendship. I would also specially thank to Muharrem Özgün, Özcan Yırtıcı, Özgür Harputlu, Özgür Yalçın, Pınar Eneren, Levent Cevher and all of my other colleagues and friends at METU for their help and fellowship. I also thank to my dear friend Erdinç Mermer for his supports. I would also thank to my special friends Gökhan, Burak, Mert, Anıl, Yasin, Fırat, Berk, Canbert, Çınar, Mehmet that I really spent great time at some stages of my life.

I am also grateful to my lovely family, Halit, Nurcan, Şencan, Tekin and our little sweet guest Ezgi that we hope to join our family soon. I would like to specially thank my father and mother for their invaluable support and love especially at hard times of my life.

I also thank God to give me the opportunity to meet my sweetheart Tuğba who helped me at each stage of my life with her endless love and support. I am also grateful to Büşra, Murat, Gülay, Fatih, Ceyda and all other friends that I meet via my beloved Tuğba.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for its scholarship support.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xix

CHAPTERS

1 INTRODUCTION 1

1.1 Unmanned Air Vehicles(UAV) 2

1.2 UAV Types 2

1.2.1 Fixed-wing UAVs 3

1.2.2 Rotary-wing UAVs 4

1.2.3 Hybrid(convertible) configurations UAVs 5

1.3 Quadrotor and comparison with other rotary-wing UAVs . . . 5

1.3.1 Control of the quadrotor 9

1.4 Motivation and Aim of This Work 12

1.5 Previous works 13

1.6 Contributions of This Work 15

2 DYNAMIC MODEL 19

2.1 Definition of Control Inputs 20

2.2 Reference Frames and Transformation Matrices 24

2.2.1 Earth Fixed Reference Frame, $\boldsymbol{F_E}$ 24

2.2.2 Body Fixed Reference Frame, $\boldsymbol{F_B}$ 25

2.2.3 Transformation Matrices 25

2.2.3.1 Transformation of Translational Velocities 26

2.2.3.2 Transformation of Angular Velocities . 28

2.3 Defining External Forces and Moments 29

2.3.1 Gravitational Force 29

2.3.2 Force Generated by Propeller System 30

2.3.3 Aerodynamic Drag Force 30

2.3.4 Moment Generated by Propeller System 31

2.4 Obtaining The Nonlinear Dynamic Model 32

2.4.1 The Nonlinear Dynamic Model for Translational Motion 33

2.4.2 The Nonlinear Dynamic Model for Rotational Motion 35

2.4.3 The Simplified Nonlinear Dynamic Model for Rotational Motion 36

3 CONTROL METHODS 39

3.1 Backstepping Controller 39

3.1.1 Step 1: Attitude Control 41

3.1.2 Step 2: Position Control 44

3.1.3 Step 3: Obtaining Desired Angles 48

3.2 Linear Quadratic Tracking(LQT) Control 51

3.2.1 Obtaining Discrete-Time State Space Equations . . 52

3.2.2 Defining Performance Index 53

3.2.3 Optimal Control Algorithm 55

3.3 Linear Quadratic Regulator(LQR) Control 57

4 RESULTS AND DISCUSSION 61

4.1 The Nonlinear Dynamic Model of the Quadrotor 61

4.2 Dynamic Model of the Motor 63

4.3 Angular Speed Converter 64

4.4 Simulation Results of Backstepping Controller 66

4.4.1 Desired Trajectory A 66

4.4.2 Results of Backstepping Controller 67

4.5 Simulation Results of LQT Controller and Comparison with LQR Controller 73

4.5.1 Desired Trajectory B 73

4.5.2 Results of LQT Controller 73

4.5.3 Comparison Between LQT and LQR Controllers . 78

4.5.3.1 Comparison of Energy Consumption . 79

4.5.3.2 Comparison of Error Terms 82

4.5.3.3 Saturation of Motors 83

4.5.3.4 Summary of the comparison between LQT and LQR controllers 85

4.6 Disturbance Rejection Properties of Backstepping, LQT and LQR Controllers 86

4.6.1 Disturbance Rejection Comparison of Backstepping, LQT and LQR controller 89

4.7 Comparison of Backstepping, LQT and LQR Controllers . . 92

4.7.1 Desired Trajectory C 92

4.7.2 Comparison Results 94

4.7.3 Summary of Comparison Results 97

5 CONCLUSION 101

REFERENCES 103

LIST OF TABLES

TABLES

Table 1.1 General comparison between UAVs with VTOL and CTOL abilites. 3

Table 1.2 Comparison between fixed pitch and variable pitch propellers. . . . 9

Table 1.3 Comparison between several VTOL configurations [9]. (1=Worst, 4=Best) 10

Table 2.1 Symbols used in the derivation of dynamic model and abbreviations 21

Table 3.1 Symbols used in backstepping controller. 41

Table 3.2 Optimized control parameters of backstepping controler. 51

Table 4.1 Summary of the comparison between LQT and LQR controllers. . . 85

Table 4.2 Steady state tracking errors of backstepping, LQT and LQR controllers while tracking desired trajectory C. 95

Table 4.3 Summary of the comparison among backstepping, LQT and LQR controllers, 1=Best, 3=Worst. 99

LIST OF FIGURES

FIGURES

Figure 1.1 "AscTech Hummigbird" quadrotor used in this thesis [13]. 6

Figure 1.2 Some of the popular quadrotors and research projects 7

Figure 1.3 Forces and torques generated by each propeller. 11

Figure 2.1 Rotor configuration, force and torque generated by each propeller of the quadrotor, reference frames F_B and F_E. 22

Figure 3.1 Top view of the quadrotor, motion in x-y plane. 48

Figure 3.2 Each part of the bacsktepping controller with its inputs and outputs. Green blocks represent the backstepping controller and red block represents the nonlinear quadrotor dynamics. 50

Figure 3.3 The LQT control system as a state feedback control with time-varying and offline calculated optimal control gains L, L_g and g [74]. . . . 56

Figure 3.4 The LQR control system as a state feedback control with fixed(time-invariant) optimal control gain K. 58

Figure 4.1 Simulink block of the nonlinear dynamic model of quadrotor with its inputs and outputs. 62

Figure 4.2 Simulink model of the backstepping controller. 65

Figure 4.3 3D plot of desired trajectory A. 68

Figure 4.4 x, y, z coordinates of desired trajectory A and x, y, z coordinates obtained by backstepping controller. 68

Figure 4.5 Error between x, y, z coordinates of desired trajectory A and x, y, z coordinates obtained by backstepping controller. 69

Figure 4.6 Euler angles obtained by backstepping controller while tracking desired trajectory A. 70

Figure 4.7 Control input U_1 obtained by backstepping controller while tracking desired trajectory A. 71

Figure 4.8 Control inputs U_2, U_3, U_4 obtained by backstepping controller while tracking desired trajectory A. 72

Figure 4.9 Translational velocities obtained by backstepping controller while tracking desired trajectory A. 72

Figure 4.10 3D plot of desired trajectory B. 74

Figure 4.11 x, y, z coordinates of desired trajectory B and x, y, z coordinates obtained by LQT controller. 74

Figure 4.12 Error between x, y, z coordinates of desired trajectory B and x, y, z coordinates obtained by LQT controller. 75

Figure 4.13 Euler angles obtained by LQT controller while tracking desired trajectory B. 76

Figure 4.14 Translational velocities obtained by LQT controller while tracking desired trajectory B. 76

Figure 4.15 Optimal control input U_1 obtained by LQT controller while tracking desired trajectory B. 77

Figure 4.16 Angular velocities $(\omega_1, \omega_2, \omega_3, \omega_4)$ obtained by LQT controller while tracking desired trajectory B. 78

Figure 4.17 Total power consumed by four motors of quadrotor for LQT and LQR controllers at each time instant while tracking desired trajectory B. . 80

Figure 4.18 Desired y coordinate and y coordinates obtained by LQT and LQR controllers while tracking desired trajectory B. 81

Figure 4.19 Error between x, y, z coordinates of desired trajectory B and x, y, z coordinates obtained by LQT and LQR controllers while tracking desired trajectory B. 82

Figure 4.20 Angular velocities $(\omega_1, \omega_2, \omega_3, \omega_4)$ obtained by LQT and LQR controllers while tracking desired trajectory B. 83

Figure 4.21 Optimal control input U_1 obtained by LQT and LQR controllers while tracking desired trajectory B. 84

Figure 4.22 Pure disturbance d_x added between 5-5.02 seconds and net unit force f_{distx} obtained by simulations with additional disturbances. 87

Figure 4.23 Pure disturbance d_y added between 10-10.02 seconds and net unit force f_{disty} obtained by simulations with additional disturbances. 87

Figure 4.24 Pure disturbance d_z added between 20-25 seconds and net unit force f_{distz} obtained by simulations with additional disturbances. 88

Figure 4.25 Tracking errors of each controller when disturbances are added to the system. 89

Figure 4.26 Euler angles obtained by each controller when disturbances are added to the system. 90

Figure 4.27 Control inputs obtained by each controller when disturbances are added to the system. 91

Figure 4.28 x, y, z coordinates of desired trajectory C. 93

Figure 4.29 Tracking errors of backstepping, LQT and LQR controllers while tracking desired trajectory C. 94

Figure 4.30 Total power consumed by four motors of quadrotor for backstepping, LQT and LQR controllers while tracking desired trajectory C. 95

Figure 4.31 Euler angles obtained by each controller while tracking desired trajectory C. 97

Figure 4.32 Translational velocities obtained by each controller while tracking desired trajectory C. 98

Figure 4.33 Angular velocities of rotors obtained by each controller while tracking desired trajectory C. 98

LIST OF ABBREVIATIONS

AIBC	Adaptive Integral Backstepping Controller
COG	Center of gravity
CTOL	Conventional Take-off and Landing
DRE	Difference Riccati Equation
ESC	Electronic Speed Control
GPS	Global Positioning System
IB	Integral Backstepping
IMU	Inertial Measurement Unit
LHS	Left hand side
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LQT	Linear Quadratic Tracking
MEMS	Micro Electro Mechanical Systems
PID	Proportional Integral Derivative
RHS	Right hand side
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-off and Landing

LIST OF UNITS

deg	Degree
Kg	Kilogram
m/s	Meter per seconds
N	Newton
rad	Radians
rpm	Revolutions per minute
sec	Seconds

LIST OF SYMBOLS

A	Continuous time state matrix
A_d	Discrete time state matrix
B	Continuous time input matrix
B_d	Discrete time input matrix
C	Continuous time output matrix
C_d	Discrete time output matrix
d	Level arm
d_x	Unit disturbance force added to the system in x direction.
d_y	Unit disturbance force added to the system in y direction.
d_z	Unit disturbance force added to the system in z direction.
$E, back$	Total energy consumed by four motors of the quadrotor for backstepping controller.
E, lqr	Total energy consumed by four motors of the quadrotor for LQR controller.
E, lqt	Total energy consumed by four motors of the quadrotor for LQT controller.
F_B	Body fixed reference fram
F_E	Earth fixed reference frame
F_i	Force generated by i_{th} rotor
g	Gravitational acceleration
$I_{n\times n}$	n_{th} order identity matrix
I_x	Moment of inertia along x axis
I_y	Moment of inertia along y axis
I_z	Moment of inertia along z axis
J	Body inertia matrix
J_d	Discrete-time performance index(cost function) of LQT controller
$J(u)$	Continuous time cost function of LQR controller
K	Fixed (time-invariant) optimal control gain of LQR controller
k	Discrete time step
K_t	Aerodynamic drag matrix for translational motion
k_f	Final time
k_g	Motor gain

k_m	Torque factor(Constant that relates F_i and T_i)
k_n	Thrust factor(Constant that relates F_i and ω_i)
L, L_g, g	Time-varying optimal control gains of LQT controller
L_{BE}	Transformation matrix from F_E to F_B
L_{EB}	Transformation matrix from F_B to F_E
L_R	Transformation matrix between $[p, q, r]$ and $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$
m	Mass of the quadrotor
p, q, r	Body angular velocities
P_i	Power consumption of i_{th} motor.
$P_{back,i}$	Power consumption of i_{th} motor for backstepping controller.
$P_{lqr,i}$	Power consumption of i_{th} motor for LQR controller.
$P_{lqt,i}$	Power consumption of i_{th} motor for LQT controller.
Q	State weight matrix
R	Control weight matrix
T_i	Torque generated by i_{th} rotor
U	Input vector
U_1, U_2, U_3, U_4	Control inputs
U^*	Optimal control input of LQT controller
V_B	Translational velocity vector of quadrotor relative to F_E expressed in F_B
X	State vector
X_d	Desired state vector
x_{1d}, x_{3d}, x_{5d}	Desired roll, pitch, yaw angles
x_{7d}, x_{9d}, x_{11d}	Desired x, y, z coordinates
$x_{7d,i}, x_{9d,i}, x_{11d,i}$	Desired x, y, z coordinates at time step i
Y	Output vector

LIST OF GREEK LETTERS

ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
ϕ, θ, ψ	Euler angles
ω_i	Angular velocity of i_{th} rotor

$\omega_{i,des}$	Desired angular velocity of i_{th} rotor
ω	Rotational velocity vector of quadrotor relative to F_E expressed in F_B
ξ	Position vector of quadrotor relative to F_E expressed in F_E
η	Orientation vector of quadrotor relative to F_E expressed in F_E
ζ	Damping ratio of command filter used in backstepping controller
ω_n	Natural frequency of command filter used in backstepping controller

LIST OF MATHEMATICAL SYMBOLS

c	Cosine function
s	Sine function
$\mathbf{s}$	Laplace variable
$\dot{f}(\cdot)$	First order time derivative
$\ddot{f}(\cdot)$	Second order time derivative
$\times$	Cross product

CHAPTER 1

INTRODUCTION

The aim and the motivation of this work is trajectory tracking of a quadrotor Unmanned Aerial Vehicle(UAV) by controlling attitute and position of the quadrotor simultaneously. Two different and independent control approaches will be used to obtain trajectory tracking of the quadrotor. The first one is a continuous-time nonlinear control method called as "backstepping" and the second one is a discrete-time optimal control method called as "Linear Quadratic Tracking(LQT)". In addition, fixed gain LQR controller which is widely used in literature is also used for comparison analysis. LQT control algorithm used in this thesis is a unique optimal control approach that is specifically developed to track desired trajectories [1]. On the other hand, backstepping control techniques have been used by many researchers in literature. However, most of these works deal with attitude control/stabilization or tracking relatively simple trajectories. Therefore, in this thesis, backstepping controller is designed to track relatively complex trajectories. Backstepping controller obtained in this thesis is based on the method used by [2].

In this introductory chapter, first, general definitions and information about UAVs will be given in Section 1.1 and 1.2. In Section 1.3, more detailed information about quadrotors and the one used in this paper, "AscTech Hummingbird", will be explained. In addition, in Section 1.3, a comparison between quadrotors and other rotary wing UAVs will also be given since quadrotor is a rotary-wing UAV. In Section 1.4, the motivation and aim of the study will be mentioned briefly. In Section 1.5 previous works about the control of quadrotor will be summarized. Finally, in Section 1.6 the contribution and original results of this thesis will be explained.

1.1 Unmanned Air Vehicles(UAV)

In the last twenty years, unmanned air vehicles(UAV) made a great impact on aviation. As the name implies, "An UAV is a powered aerial vehicle that does not carry an onboard crew, can operate with varying degrees of autonomy, and can be expendable or reusable" as defined in [3]. The control of the vehicle can be obtained by a remote pilot or onboard autonomous control systems [4]. Therefore, unlike manned air vehicles, UAVs could avoid risking human lives especially for dangerous situations such as, search and rescue at damaged disaster areas, military operations, firefighting etc.. Performing these hazardous tasks without direct human interaction is highly valuable [5].

In addition, UAVs can be very small in size compared to manned air vehicles since additional space and weight considerations for human crew are not included to the air vehicle. With the latest development of Micro Electro Mechanical Systems(MEMS) technology, human crew with large weight and space can be replaced with very small electronic devices in UAVs [6]. Therefore, for specific operations/missions that require small size aerial vehicles, UAVs can be very effective.

UAVs can be categorized into several groups according to their configurations and abilities. In section 1.2, the basic types of UAVs will be discussed in more details.

1.2 UAV Types

Unmmanned Air Vehicles can be basically categorized into three main groups which are [3, 7]:

-Fixed-wing UAVs

-Rotary-wing UAVs

-Hybrid(convertible) configuration UAVs

These are the most general categories of UAVs and each of them will be explained in the following subsections. However, there are also some other types of UAVs such as

blimps, flapping wing UAVs, etc. which are briefly mentioned in [3].

1.2.1 Fixed-wing UAVs

As the name implies, fixed-wing UAVs have fixed-wings at a certain place which produce lift by using the forward speed of the vehicle. Lift generated is basically related to the shape of the wings. Therefore, fixed-wing UAVs need a runway to take-off and land or use some systems for launching and safe landing such as catapult launch mechanisms or parachute landing. In literature, to express the usage of runways for take-off and landing, acronym CTOL(Conventional Take-off and Landing) is used.

The main advantages of fixed-wing UAVs are long endurance and high cruise speeds since they are aerodynamically more efficient [3, 8]. Moreover, they are relatively more simple to design compared to rotary-wing or hybrid designs. On the other hand, since fixed-wing UAVs require runways or launch systems to take-off and land, they are disadvantageous compared to rotary wing UAVs which have Vertical Take-off and Landing(VTOL) ability [3, 7, 8]. Fixed-wing UAVs are also more prone to be damaged during landing compared to VTOL UAVs. Summary of the comparison between fixed-wing UAVs with CTOL ability and rotary-wing UAVs with VTOL ability can be seen in Table 1.1.

Table 1.1: General comparison between UAVs with VTOL and CTOL abilites.

	Advantages	Disadvantages
VTOL UAVs (rotary-wing)	-No need for runways -Hovering ability -High maneuverability	-Short range and endurance -Limited speed and altitude flights -More complex mechanism -High energy consumption
CTOL UAVs (fixed-wing)	-Long range and endurance -High speed and altitude flights -More simple mechanism -Energy efficient	-Requires runways -No hovering -Less maneuverability -More prone to be damaged during landing

Fixed-wing UAVs are mainly classified as: "tailplane aft", "tailplane forward" and "tailless". For detailed information about these specific types of fixed-wing UAVs, [7] can be examined.

1.2.2 Rotary-wing UAVs

Rotary-wing UAVs use rotors instead of fixed-wings to provide lift and thrust force. Rotary-wing UAVs don't require long and smooth runways for take-off and landing since they can vertically take-off and land. Therefore, their main advantage is VTOL ability which enables to operate in complicated and limited environments which are not appropriate for fixed-wing UAVs [9]. Another important advantage of rotary-wing UAVs compared to fixed-wing ones is hovering ability. Hovering is important if operation requires to stay in the air at a specific location, such as mapping, aerial photography, surveillance, etc.. Moreover, rotary-wing UAVs are highly maneuverable compared to fixed-wing UAVs [3]. Therefore, for specific type of missions rotary-wing UAVs can be very advantageous. On the other hand, rotary-wing UAVs are not as aerodynamically efficient as fixed-wing UAVs; therefore, they don't have long flight time and they couldn't reach high speeds and altitudes compared to fixed-wing UAVs [7, 8]. Therefore, rotary-wing UAVs are generally advantageous for short range and low speed flights when no runway is available [7, 8].

Configurations of VTOL UAVs could change according to their rotor number or positions basically. The most common example is helicopters with one main rotor and a tail rotor. Rotary wing UAVs generally can be classified as: single rotor, coaxial rotor, tandem rotor and multi-rotors [3, 7].

"AscTech Hummingbird" [10] quadrotor used in this thesis is also a rotary-wing VTOL UAV. In section 1.3, a more detailed information about quadrotors and "AscTech Hummingbird" will be given and a comparison between other rotary-wing VTOL UAVs will be discussed.

1.2.3 Hybrid(convertible) configurations UAVs

As discussed in Subsections 1.2.1 and 1.2.2, both fixed-wing UAVs with conventional take-off and landing(CTOL) and rotary-wing UAVs with vertical take-off and landing(VTOL) have its own advantageous and disadvantageous properties. VTOL UAVs are not efficient for long range and high speed flights compared to fixed-wing ones. On the other hand, fixed-wing UAVs with CTOL requires runways and couldn't perform hovering ability which is invaluable for specific type of missions. Therefore, for several years, engineers and researchers have been searching for an UAV design which could both have long range and endurance flights with high aerodynamic efficiency and VTOL ability [7].

At this point, hybrid(convertible) designs have appeared. Hybrid design UAVs generally have both wings and rotors at the same time. They could perform VTOL and hovering by tilting their rotors, wings or body and with the help of their wings they could have long range and endurance with high speed flights [7]. Although, convertible designs have advantageous properties of both fixed-wing and rotary-wing UAVs, they are mechanically and aerodynamically very complex due to tilting mechanisms [14]. Therefore, design and maintenance costs are very high compared to other UAV types.

Hybrid designs can be generally classified as: convertible rotor, tilt-wing, tilt body, ducted fan and jet life aircrafts. More detailed information about each configuration can be obtained in [7].

1.3 Quadrotor and comparison with other rotary-wing UAVs

Quadrotor is a small rotary-wing UAV with vertical take-off and landing(VTOL) ability. Figure 1.1, shows the "AscTech Hummingbird" quadrotor used in this thesis [10]. As can be seen in Figure 1.1, "AscTech Hummingbird" quadrotor has four brushless motors attached to the body in cross configuration. Each motor is connected to fixed-pitch propellers which provide thrust force independently if coupling effects between propellers are assumed to be negligible. There are also four ESC(Electronic Speed Control) units to control motors. Moreover, "AscTech Hummingbird" has sev-

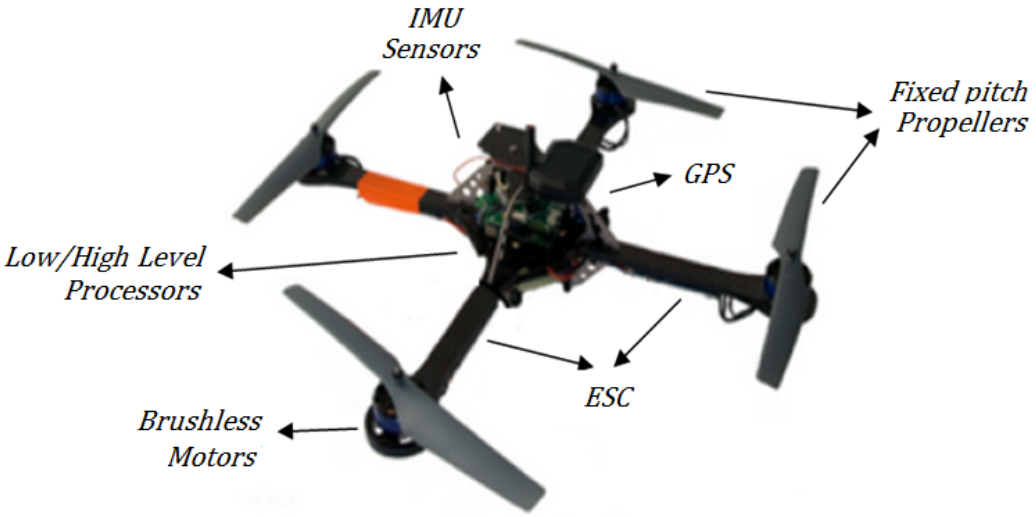


Figure 1.1: "AscTech Hummigbird" quadrotor used in this thesis [13].

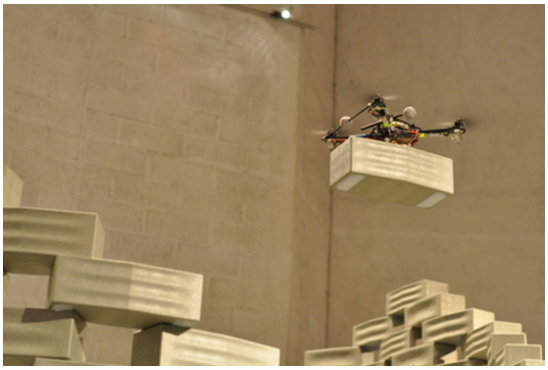
eral sensor units such as IMU, GPS, barometer, etc. to obtain feedback information. The mainboard of "AscTech Hummingbird" has two ARM7 microprocessors called as LowLevel(LL) and HighLevel(HL) processors which could be used for different tasks [12]. One could test own control algorithms by flashing the algorithm into HL processor which could run at 1KHz [12]. First, control algorithms(models) generated in MATLAB/Simulink is build as a C++ code. Then, C++ code is flashed(embedded) into the HL processor by using "Eclipse" program. There is also a LL processor which has a well proven attitude control algorithm that is already embedded by producer. It is possible to switch between LL and HL processors during flight as a safety backup [12]. The communication between "AscTech Hummingbird" and a ground PC is also obtained by a wireless serial link called as "XBee" [12]. It can be concluded that, one of the main advantages of "AscTech Hummingbird" quadrotor is opportunity to test own high level control algorithms generated in MATLAB/Simulink environment with a very fast rate of 1KHz.

There are also several quadrotor producers for research/experiment and civil use. Figure 1.2 shows some of the most popular quadrotors in the field produced by aerospace companies or universities [15, 16, 17, 18, 19, 20, 21, 22].

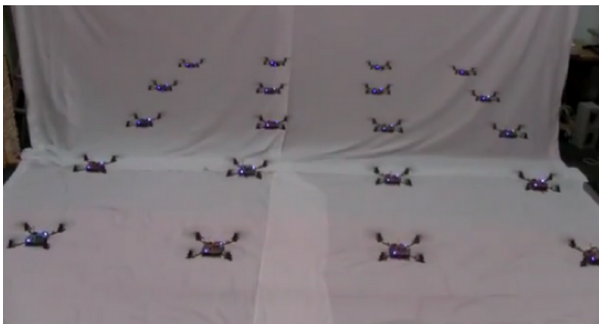
The main difference of quadrotors compared to other conventional VTOL UAVs

(a) Parrot AR. Drone

(b) DraganFlyer X4



(c) ETH Zurich project: Quadrotors build a wall



(d) UPenn GRASP Lab.: Swarm of nano quadrotors

(e) MIT: Variable-pitch quadrotor

Figure 1.2: Some of the popular quadrotors and research projects

such as helicopters is using fixed-pitch propellers to control the quadrotor instead of variable-pitch propellers which are generally used in helicopters. As the name implies, fixed pitch-propeller systems could not change the pitch angle of the propeller, instead, control problem can be handled by changing the angular velocities of each propeller properly. [23]. On the other hand, in variable-pitch propellers, control of the vehicle could be achieved by changing the pitch angle of the propeller [24]. Variable-pitch propeller systems used in conventional helicopters require very complex swash plate or mechanical transmission mechanisms to change the pitch angle of the propellers properly [14]. However, fixed-pitch propeller systems don't require swash plates or any other mechanical transmission systems [7, 14]. Therefore, fixed-pitch propeller systems are more robust and reliable in terms of design and maintenance due to mechanical simplicity. In addition, cost of production and maintenance of fixed-pitch propellers are very low compared to the variable pitch propellers. Aerodynamic complexity of fixed-pitch propellers are also relatively low compared to variable-pitch ones. However, although variable pitch propellers are more complex and fragile, they can handle aggressive maneuvers more efficiently with the help of increased control bandwidth and efficient reverse thrust [23]. The control limitations of fixed-pitch propellers could limit very aggressive maneuvers for large size quadrotors(above 2 Kg) [23]. However, in literature, it is seen that for small size quadrotors, like "AscTech Hummingbird"(0.48 Kg) used in this thesis, control bandwidth doesn't restrict aggressive maneuvers significantly [23, 25, 26]. Table 1.2 gives a detailed comparison of fixed-pitch and variable-pitch propellers and it can be concluded that, for small size quadrotors like "AscTech Hummingbird", fixed-pitch propellers could provide sufficient control inputs with simple dynamics and low cost design and maintenance, even for relatively aggressive maneuvers [23].

According to these information, one can conclude that, the basic difference and advantage of quadrotors compared to other VTOL UAVs with variable-pitch propellers is the removal of complex mechanical transmission mechanisms that complicates both structural and aerodynamic design [7]. In literature, there are also some pioneer quadrotor designs that use variable-pitch propellers [23, 27]. However, most of the quadrotors in use have fixed-pitch propeller mechanisms due to mechanical and aerodynamic simplicity [7, 14, 23].

Table 1.2: Comparison between fixed pitch and variable pitch propellers.

	Advantages	Disadvantages
Fixed Pitch Propellers	-Mechanical simplicity -Simple dynamics -Robust and reliable -Easy and low cost design and maintenance	-Less maneuverability due to control limitations
Variable Pitch Propellers	-Efficient reverse thrust -Increased control bandwidth -Aggressive and aerobatic maneuverability	-Complicated mechanical systems -Complex dynamics -Fragile -Difficult and high cost design and maintenance

On the other hand, the basic disadvantages of quadrotors are high energy consumption, low speed flight and short range and endurance. However, other VTOL UAVs (except hybrid designs) are also inefficient in terms of energy consumption and have short range and endurance with limited flight speeds according to Table 1.3 prepared by [9]. Therefore, it is more reasonable to claim that, quadrotors consume high energy and they have short range and endurance with low flight speeds compared to fixed-wing UAVs, as can be seen in Table 1.1.

A complete and very detailed comparison of small quadrotors and other types of small VTOL configurations can be seen in Table 1.3, prepared by [9].

According to Table 1.3, quadrotor is one of the most advantageous small VTOL UAV concept. Therefore, it is not extraordinary that quadrotor UAVs are very popular and have been worked by many researchers and organizations in the last decade [10, 15, 16, 17, 18, 19, 20, 21].

1.3.1 Control of the quadrotor

As stated earlier, since quadrotors have generally fixed-pitch propellers, control of the vehicle is obtained by adjusting angular velocity of each rotor. Then, as can be seen in Figure 1.3, four rotors provide four thrust force(F_1, F_2, F_3, F_4) to the system.

Table 1.3: Comparison between several VTOL configurations [9]. (1=Worst, 4=Best)

	A	B	C	D	E	F	G	H
Power cost	2	2	2	2	1	4	3	3
Control cost	1	1	4	2	3	3	2	1
Payload/volume	2	2	4	3	3	1	2	1
Maneuverability	4	2	2	3	3	1	3	3
Mechanical simplicity	1	3	3	1	4	4	1	1
Aerodynamics complexity	1	1	1	1	4	3	1	1
Low speed flight	4	3	4	3	4	4	2	2
High speed flight	2	4	1	2	3	1	3	3
Miniaturization	2	3	4	2	3	1	2	4
Survivability	1	3	3	1	1	3	2	3
Stationary flight	4	4	4	4	4	3	1	2
TOTAL	**24**	**28**	**32**	**24**	**33**	**28**	**22**	**24**

A:Single rotor, B:Axial rotor, C:Coaxial rotors, D:Tandem rotors, E:Quadrotor, F:Blimp, G:Bird-like, H:Insect-like.

Front and rear rotors rotate rotate clockwise while right and left rotors rotate counter-clockwise as can be seen in Figure 1.3. Therefore, torques generated by front and rear rotors are in the opposite direction with the torques generated by left and right rotors. By this way, the net torque generated by four propellers are balanced and yaw motion is automatically stabilized.

The total thrust is the sum of four thrust forces F_1, F_2, F_3, F_4. Therefore, the motion in z direction is directly related to the total force generated. On the other hand, the motion in x and y directions occur by changing the pitch(θ) and roll(ϕ) angles properly. Pitch motion(θ) is obtained by increasing/reducing the angular velocity of the rear rotor while reducing/increasing the angular velocity of the front rotor [14]. Similarly, roll motion(ϕ) is obtained by increasing/reducing the angular velocity of the right rotor while reducing/increasing the angular velocity of the left rotor [14]. Yaw motion(ψ) is obtained by increasing /decreasing the angular velocity of the front and rear motors while decreasing/increasing the angular velocity of the right and left rotors. Therefore, heading of the quadrotor can be adjusted by changing yaw angle(ψ) of the quadrotor. Then, four control inputs U_1, U_2, U_3, U_4 used to control quadrotor are defined as in Equation 1.1.

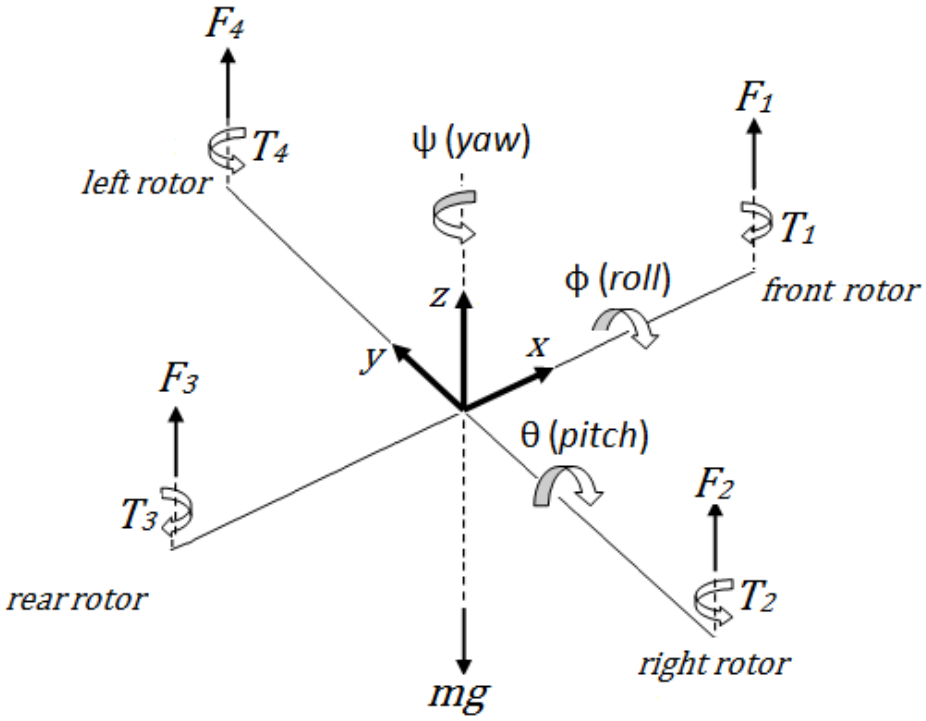


Figure 1.3: Forces and torques generated by each propeller.

$$\left.\begin{aligned} U_1 &= F_1 + F_2 + F_3 + F_4 \\ U_2 &= F_4 - F_2 \\ U_3 &= F_3 - F_1 \\ U_4 &= T_2 + T_4 - T_1 - T_3 \end{aligned}\right\} \tag{1.1}$$

According to Equation 1.1, U_1 is the total force generated by four propellers and directly related to the motion in z direction, whereas U_2, U_3 and U_4 are related to roll, pitch and yaw motion respectively. As explained earlier, motion in x and y directions occur by changing pitch(θ) and roll(ϕ) angles which tilt U_1 in corresponding direction. A more detailed information about the control of the quadrotor is given in Section 2.1 of Chapter 2.

As defined in Equation 1.1, there are four independent control inputs(U_1, U_2, U_3, U_4) and six degrees of freedom to be controlled($x, y, z, \phi, \theta, \psi$) [2, 14, 28, 29]. There-

fore, quadrotor is an under-actuated system and it is also dynamically unstable which makes it difficult to control [2, 28, 29, 30]. However, under-actuated control problem can be solved efficiently by using θ and ϕ as additional virtual control inputs to control the motion in x and y directions [30].

To conclude, quadrotor is an under-actuated system with highly nonlinear, unstable and coupled dynamics. [1, 2, 28, 29, 35] Therefore, the aim of this study is obtaining trajectory tracking of a quadrotor via attitude and position control since it is a challenging and motivating task.

1.4 Motivation and Aim of This Work

As can be seen in Table 1.3, quadrotor UAV is a very advantageous VTOL concept especially in terms of aerodynamic and mechanical simplicity [2, 28, 29, 30]. However, control of the vehicle is not straightforward and requires solving some problems due to under-actuated, highly nonlinear and unstable dynamics of the vehicle [28, 29, 30, 34, 36]. Therefore, it is really motivating, challenging and satisfying to design and test autonomous control systems for quadrotor UAVs.

In this thesis, the aim is obtaining trajectory tracking of a quadrotor UAV by controlling attitude and position simultaneously. Two different control methods are used independently. One of the control method is a nonlinear control technique called as "backstepping" and it has been used by various researchers and experimentally proven control methodology for quadrotors [2, 34, 36, 38]. The other method is an optimal control algorithm called as "Linear Quadratic Tracking(LQT)", and it is a more unique approach and hasn't been used and experimentally verified on quadrotors by other researchers yet. As stated earlier, the backstepping method have been used by many researchers in literature, however, in this thesis some modifications are added to the control system for more accurate and efficient tracking of relatively complex trajectories. These contributions will be explained particularly in Section 1.6. On the other hand, LQT control algorithm proposed by [1] could be considered as a more unique approach and original results obtained by using optimal LQT control technique will be also explained in Section 1.6.

1.5 Previous works

Since autonomous control of quadrotor UAV is a challenging and motivating task, there are a lot of works in literature related to this area. Most of the works in literature focus on the attitude control/stabilization of the quadrotor since it is a basis for the complete control of the quadrotor. On the other hand, trajectory/path tracking of the quadrotor requires simultaneous control of both attitude and position, therefore, it is more complicated and the number of works related to trajectory tracking of the quadrotor is not very much compared to the works related to attitude control/ stabilization.

Since the topic of this thesis is trajectory tracking of the quadrotor, the literature review will be focused on this content. However, works related to attitude control/ stabilization will be also briefly mentioned. In literature, autonomous control of quadrotor UAV is tried to be achieved by using various control approaches. These approaches could be basically divided into three groups which are linear control techniques, nonlinear control techniques and intelligent/adaptive/robust control techniques.

The most common linear control techniques are PID and LQR control. It is seen that, since quadrotor is a highly nonlinear, coupled and under-actuated system, classical PID control approaches couldn't give satisfactory results for quadrotors [37, 38]. Therefore, some researchers combined classical PID control with intelligent control methods and some nonlinear techniques to handle the insufficiency of classical PID control [38, 39, 40, 41]. In [42], complex trajectory tracking of a quadrotor is obtained successfully by using feedback linearization for attitude control as inner loop and classical PID control for position control as outer loop.

The other most common linear control approach used by researchers is LQR which is an optimal control method. In LQR method, a quadratic cost function(performance index) is tried to be minimized to obtain optimum control gains [37]. To solve LQR algorithm, linear dynamic model of the system is required. Therefore, nonlinear dynamic model of the system have to be linearized around a trim condition such as hovering. LQR control could be advantageous to satisfy some design requirements since these requirements could be defined and constrained in cost function. In literature,

LQR control have been used by some researchers as a regulator to stabilize the attitude of the quadrotor [29, 37, 38, 43]. On the other hand, for trajectory tracking of the quadrotor, LQR methods have been also used by some researchers [29, 37, 44, 45, 46]. Linear Quadratic Gaussian(LQG) control approach is also used in literature to obtain trajectory tracking and collision avoidance [44, 45]. In some of the works, integral action is also added to the system to decrease steady-state errors [37, 45]. These works generally use fixed(time-invariant) optimal control gain "K" as a state feedback control [37, 38, 43, 44, 45]. Only some of the works use different optimal control gains for different trim conditions which can be considered as gain scheduling. [46]. The LQT control algorithm used in this thesis [1] uses time-varying optimal control gains that are calculated offline instead of fixed gain LQR which is widely used in literature [37, 38, 43, 44, 45]. The advantageous and disadvantageous properties of using time-varying and offline calculated optimal control gains will be explained in details in Section 1.6.

As opposed to linear control methods, there have been so many works related to nonlinear control of quadrotor UAV. The most common nonlinear control techniques used to control quadrotor are backstepping, integral backstepping, sliding-mode control, feedback linearization and combination of these methods [2, 11, 28, 29, 30, 34, 36, 42, 47, 48, 49, 50, 51, 52, 53, 54]. There are also some works that use adaptive and intelligent control methods such as nonlinear H_∞ control, L_1 adaptive control, Adaptive Integral Backstepping Controller(AIBC), intelligent fuzzy control [35, 37, 39, 55, 56, 57, 58, 59]. More unique approaches are also tried by some researchers such as geometric tracking control [59], minimum snap trajectory generation [60], visual feedback control systems [61, 63] and balancing an inverted pendulum on top of a quadrotor [62]. Most of these works deal with the attitude control/stabilization of the quadrotor or trajectory tracking of relatively simple trajectories [2, 11, 29, 34, 39, 47, 48, 50, 51, 53, 55, 56]. On the other hand, more complex trajectory tracking have been used by relatively fewer researchers due to increased complexity [42, 54, 58, 60, 64].

As stated earlier, in this thesis, backstepping and LQT approaches are used to obtain trajectory tracking of quadrotor. In addition a fixed-gain LQR controller is obtained for detailed comparison analysis. The contributions of both LQT and backstepping

approaches are mentioned particularly in Section 1.6.

1.6 Contributions of This Work

In this section, original results obtained by using LQT approach and modifications added to backstepping method will be explained in details.

The main difference or uniqueness of the LQT algorithm [1] used in this paper can be explained as follows: in LQT algorithm, offline calculated optimal control gains are time-varying unlike the time-invariant(fixed) control gain obtained in classical LQR control that is widely used in literature. In other words, time-varying control gains of LQT are optimized to track desired trajectory specifically. It is observed that, using time-varying control gains optimized to track desired trajectory results in decreased energy consumption compared to the fixed gain LQR control and backstepping. In addition, trajectory tracking errors decreased significantly in LQT control compared to the fixed-gain LQR control. Also, for sharp maneuvers like step commands, LQT control could react before the input is commanded. By this way, control inputs changes more slowly and motors do not saturate. The decrease in energy consumption is also related to this behavior of the LQT algorithm.

Then, three main advantages of variable gain LQT control algorithm used in this thesis compared to fixed-gain LQR control could be summarized as following:

1) Decreased energy consumption especially for trajectories that involve sharp maneuvers.

2) Decreased tracking errors.

3) Saturation of motors is automatically avoided.

More detailed analysis of the advantageous results of LQT controller are explained in Subsection 4.5.3.4.

It is also important to mention the main drawback of LQT controller. As explained in Section 3.2, LQT algorithm calculates time varying control gains offline and these gains are the inputs of the LQT control system as can be seen in Figure 3.3. In other

words, time varying control gains of LQT are specific for commanded trajectory and have to be calculated separately for each trajectory. This behavior of LQT controller could be interpreted as a disadvantage compared to fixed-gain LQR and backstepping controllers. However, it should be also mentioned that, backstepping and fixed-gain LQR controllers require desired trajectory information directly as inputs, as can be seen in the yellow blocks of Figures 4.2 and 3.4. That means if the quadrotor is desired to be flown to a specific location, an algorithm is required to produce a suitable trajectory to take the quadrotor from its current position to the final destination. This step requires an online guidance algorithm, or an offline trajectory generator. In this thesis, desired trajectory information are given to the backstepping and fixed-gain LQR controllers offline. On the other hand, as can be seen in Figure 3.3, LQT controller use time varying optimal control gains as inputs of the controller and doesn't require desired trajectory information since offline calculated control gains are already optimized to track desired trajectory and possess desired trajectory information. Therefore, time-varying control gains of LQT controller could be considered as the offline trajectory input of fixed-gain LQR and backstepping controllers.

To conclude, the main drawback of LQT controller is the offline calculation of control gains for each trajectory specifically. In other words, an online guidance algorithm couldn't be used in LQT controller. Therefore, the drawback of LQT controller comes out if desired trajectory is given to the controllers by using an online guidance algorithm instead of an offline trajectory generator.

As mentioned previously, the other method used in this thesis which is called as "backstepping" has been used by various researchers [2, 29, 34, 47, 48, 50, 53, 55, 56] and most of these works focus on tracking relatively simple trajectories(ex:step inputs) [2, 29, 34, 50, 55]. However, the backstepping controller used in this thesis designed in such a way that more complex trajectories could be tracked accurately. It is important to remind that, backstepping controller designed in this thesis is based on the work proposed by [2]. In this thesis, the following modifications are added to the backstepping controller proposed by [2]:

1) A basic drag model is added to the controller to obtain more realistic results.

2) Second order derivatives of desired trajectory are added to the controller to increase

accuracy.

3) A command filter is used to track trajectories that involve sharp maneuvers.

4) A simple and reasonable flight approach is obtained by adjusting the heading of the quadrotor and direction of motion in $x - y$ plane.

More detailed information about the design, formulation and contribution of the backstepping controller obtained in this thesis are explained in Section 3.1 of Chapter 3.

The contributions and results of both backstepping and LQT controllers could be seen in Chapter 3 and Chapter 4 with more details. Moreover, a detailed comparison analysis among backstepping, LQT and LQR controllers is also made in Subsection 4.7.3.

After this intoductory part, first, dynamic model of the quadrotor is obtained in Chapter 2. Then, backstepping, LQT and fixed-gain LQR methods are explained and obtained in Chapter 3. In Chapter 4, simulation results of each control system are illustrated and detailed comparison analysis are made. It is important to remind that all of the simulations, algorithms, codes and optimizations are performed and solved by using MATLAB environment. Finally, a conclusion of this thesis work and possible future works are presented in Chapter 5.

CHAPTER 2

DYNAMIC MODEL

In this chapter, dynamic model of the quadrotor will be derived by using Newton's equations of motion for translational and rotational dynamics. First, the control inputs U_1, U_2, U_3, U_4 will be defined, and their relation with the motor dynamics will be explained in Section 2.1. Then, reference frames used in this thesis and transformation matrices between them will be defined in Section 2.2. After that, external forces and moments act on quadrotor body will be explained in Section 2.3. Finally, a fully nonlinear dynamic model of the quadrotor will be obtained by using Newton's equations of motion in Section 2.4. This model will be used to test the controllers designed in Chapter 3 via simulation. In addition, the nonlinear dynamic model obtained will be simplified to use in the derivation of control laws that are explained in Chapter 3.

In this chapter, following assumptions are made before deriving the dynamic model of the quadrotor [29, 55, 65].

Assumptions:

1) Quadrotor is a rigid body and mass distribution is symmetrical.

2) Propellers are rigid.

3) Center of gravity and body fixed frame origin coincides.

4) Earth's gravitational field (g), mass of the quadrotor (m) and body inertia matrix of the quadrotor(J) are constants.

5) Thrust factor(k_n) and torque factor(k_m) of motors are constants.

6) Inertia of motors and rotors are neglected.

7) Aerodynamic drag force is assumed to be proportional with translational velocity.

8) Rotation of the Earth relative to distant stars is negligible.

It is also noted that, mathematical symbols used in the derivation of dynamic model are defined in Table 2.1.

2.1 Definition of Control Inputs

As can be seen in Figure 1.1, "AscTech Hummingbird" quadrotor used in this thesis, has four rotors connected to propellers. The quadrotor used in this thesis has "fixed-pitch propellers" instead of "variable-pitch propellers" which are used mostly in commercial or military helicopters. For fixed-pitch propellers, angle of attack of propellers are constant and thrust force produced by each propeller changes by adjusting the angular velocity of corresponding rotor. On the other hand, for variable-pitch propellers, angle of attack of propellers could be changed properly. [24] The basic advantages of using fixed-pitch propellers are mechanical and aerodynamic simplicity. Production and maintenance costs of fixed pitch propellers are also very low compared to variable-pitch ones. On the other hand, variable-pitch propellers are mechanically more complex but they can handle aggressive maneuvers more effectively with the help of increased control bandwidth and reverse thrust abilities. [23, 24]. However, for small quadrotors like the one used in this thesis, it is seen that aggressive maneuvers could also be achieved with fixed-pitch propellers [23]. This is because rotors of small quadrotors have low inertias and hence they could be accelerated at fast rates, yielding a sufficiently high control bandwidth. To conclude, control of the quadrotor used in this thesis is achieved by adjusting the angular velocity of each propeller.

Before obtaining control inputs mathematically, the relation between the angular velocity of propellers and thrust and torque generated by propellers need to be defined.

Let F_i represents the thrust generated by i_{th} propeller and ω_i represents the angular velocity of i_{th} propeller. Then, for "AscTech Hummingbird" quadrotor used in this

Table 2.1: Symbols used in the derivation of dynamic model and abbreviations

Symbol	Definition
F_i	Force generated by i_{th} rotor
T_i	Torque generated by i_{th} rotor
ω_i	Angular velocity of i_{th} rotor
k_n	Constant that relates F_i and ω_i
k_m	Constant that relates F_i and T_i
U_1, U_2, U_3, U_4	Control inputs
F_E	Earth fixed reference frame
F_B	Body fixed reference frame
m	Mass of the quadrotor
I_x,I_y,I_z	Body inertia in x, y, z directions
J	Body inertia matrix
g	Gravitational acceleration
d	Level arm
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
ϕ, θ, ψ	Euler angles
p, q, r	Body angular velocities
ξ	Position vector of quadrotor expressed in F_E
η	Orientation vector of quadrotor expressed in F_E
V_B	Translational velocity vector of quadrotor expressed in F_B
ω	Rotational velocity vector of quadrotor expressed in F_B
L_{BE}	Transformation matrix from F_E to F_B
L_{EB}	Transformation matrix from F_B to F_E
L_R	Transformation matrix between $[p, q, r]$ and $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$
K_t	Aerodynamic drag matrix for translational motion
$I_{n\times n}$	n_{th} order identity matrix
RHS	Right hand side
LHS	Left hand side
$c(\cdot)$	Cosine function
$s(\cdot)$	Sine function
$\dot{f}(\cdot)$	First order time derivative
$\ddot{f}(\cdot)$	Second order time derivative
$\times$	Cross product

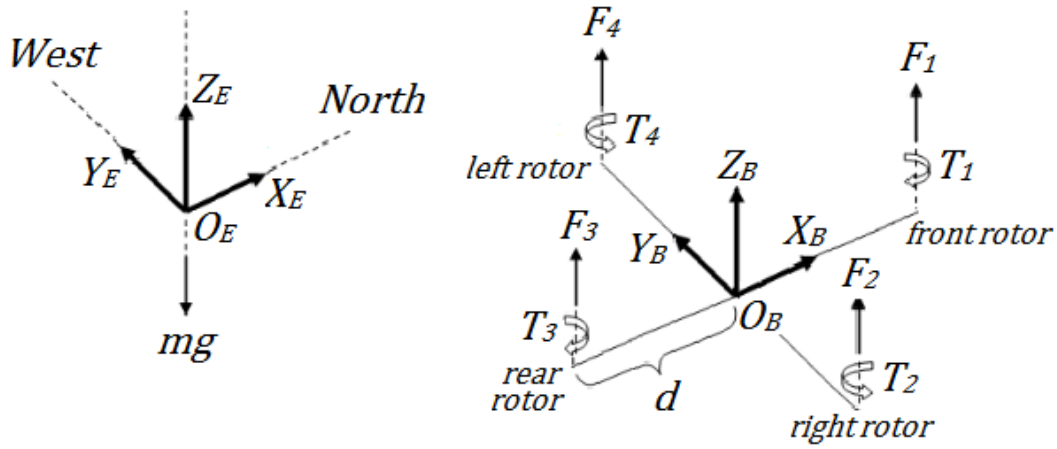


Figure 2.1: Rotor configuration, force and torque generated by each propeller of the quadrotor, reference frames F_B and F_E.

thesis, relation between thrust and angular velocity of the propeller is obtained as follows [66]:

$$F_i = k_n \omega_i^2, \quad [N] \tag{2.1}$$

In Equation (2.1), k_n is a constant that relates angular velocity and thrust generated by propellers. Value of k_n is calculated at [66] as $k_n = 5.7 \cdot 10^{-8}\ N/rpm^2$.

Relation between thrust and torque generated by propellers also have to be defined. Let T_i represent the torque generated by i_{th} propeller. According to [66], torque generated by i_{th} propeller is defined as follows:

$$T_i = k_m F_i, \quad [N \cdot m] \tag{2.2}$$

In Equation (2.2), k_m is a constant that relates thrust and torque generated by propellers. Value of k_m is found at [66] as $k_m = 0.016\ m$. Since k_m and k_n are constants, it can be concluded that, relation between thrust and torque generated by propellers and angular velocity of propellers is quadratic. Force(F_i) and torque(T_i) generated by each propeller are illustrated Figure 2.1.

By using Equations (2.1) and (2.2), control inputs can be defined. There are four control inputs which are expressed symbolically as U_1, U_2, U_3, U_4. The mathematical

definitions of U_1, U_2, U_3, U_4 are as follows:

$$U_1 = F_1 + F_2 + F_3 + F_4 \tag{2.3}$$

$$U_2 = F_4 - F_2 \tag{2.4}$$

$$U_3 = F_3 - F_1 \tag{2.5}$$

$$U_4 = T_2 + T_4 - T_1 - T_3 \tag{2.6}$$

As can be seen in Equation (2.3) and Figure 2.1, U_1 is the total force generated by four propellers, therefore, it is directly related to the motion in z direction(altitude). In other words, U_1 is the force that overpowers gravity and make the quadrotor to stay in the air. Moreover, to move the quadrotor in x and y directions, orientation of U_1 is adjusted according to desired direction by changing roll(ϕ) and pitch(θ) angles.

As can be seen in Equation (2.4), U_2 is the differential force between the left and right propellers, therefore, it is directly related to the "roll motion" and angle ϕ. Similarly according to Equation (2.5), U_3 is the differential force between the front and rear propellers, therefore, it is directly related to the "pitch motion" and angle θ. In addition, the motion in x and y directions mainly occurs by changing angles ϕ and θ, respectively. By changing angles ϕ and θ, contribution of U_1 in x and y directions change and motion in x and y directions occur.

By analyzing Equation (2.6), one can see that, unlike U_1, U_2 and U_3 which are forces and have the units of $[N]$, U_4 is torque so that has the units of $[N \cdot m]$. U_4 is the net torque generated by four propellers of the quadrotor. Therefore, it is related to the "yaw motion" and angle ψ. As can be seen in Figure 2.1, front and rear propellers rotate clockwise, while right and left propellers rotate counterclockwise. Therefore, contributions of front and rear propellers(T_1, T_3) are subtracted from the contributions of right and left propellers(T_2, T_4) to obtain the net torque generated.

The control inputs defined in Equations (2.3), (2.4), (2.5) and (2.6) will be used in Section (2.3) while defining forces and moments generated by the propeller system.

2.2 Reference Frames and Transformation Matrices

Before deriving the dynamic model of the quadrotor, reference frames have to be identified. In this section reference frames and transformation matrices between them will be defined.

Two reference frames will be used in this thesis, these are Earth fixed reference frame, F_E and body fixed reference frame, F_B.

2.2.1 Earth Fixed Reference Frame, F_E

Earth fixed reference frame is fixed to the Earth and it can be defined such that the position of the quadrotor on the earth surface (x, y, z coordinates) can be expressed in Earth fixed reference frame [65]. As can be seen in Figure 2.1, $O_E x_E$ points North, $O_E y_E$ points West and $O_E z_E$ points opposite direction to the center of the Earth.

To derive dynamic model of any physical system, an inertial reference frame have to be defined at first. Newton's second law, $\vec{f} = m\vec{a}$, is valid for inertial reference frames. Here, $\vec{f}$ represents net external forces acting on the particle and $\vec{a}$ represents acceleration of the particle relative to inertial reference frame [65].

Earth fixed reference frame can be considered as an inertial reference frame since the rotation of the Earth relative to distant stars (inertial reference frame) can be neglected [65]. This is a very reasonable assumption since quadrotors typically have ranges on the order of few kilometers and flight times on the order of minutes. Therefore, in the derivation of dynamic model of the quadrotor, Earth fixed reference frame is assumed to be inertial.

The position and orientation vector (ξ, η) of the quadrotor relative to the Earth surface are defined as follows, respectively:

$$\xi \triangleq [x, y, z]^T \tag{2.7}$$

$$\eta \triangleq [\phi, \theta, \psi]^T \tag{2.8}$$

In Equation (2.7), $[x, y, z]^T$ represents the position of the quadrotor relative to Earth

fixed reference frame F_E and also expressed in F_E.

Likewise, Equation (2.8) represents the orientation of the quadrotor about its center of mass, expressed in Earth fixed reference frame. To clarify, $[\phi, \theta, \psi]$ are called as "Euler angles" or roll, pitch and yaw angles, respectively [67, 68].

To conclude, by using the vectors defined in Equations (2.7) and (2.8), one can determine the position and orientation of the quadrotor relative to the Earth surface.

2.2.2 Body Fixed Reference Frame, F_B

Body fixed reference frame is fixed to the quadrotor body and it translates and rotates with the body [67]. The origin of the body fixed reference frame coincides with the COG of the quadrotor. As can be seen in Figure 2.1, $O_B x_B$ points front rotor, $O_B y_B$ points left rotor and $O_B z_B$ points the direction such that the orthogonal triad and right hand rule are satisfied [69].

As mentioned previously, body fixed reference frame moves and rotates with the quadrotor. By using traditional notation, rotational and translational velocities of F_B relative to our inertial frame of reference F_E are defined as follows, respectively [65]:

$$\omega \triangleq [p, q, r]^T \tag{2.9}$$

$$V_B \triangleq [u, v, w]^T \tag{2.10}$$

In Equations (2.9) and (2.10), ω and V_B represent the rotational and translational velocities of quadrotor relative to Earth fixed reference frame F_E and expressed in body fixed reference frame F_B.

2.2.3 Transformation Matrices

As defined previously, two reference frames(F_B and F_E) will be used in the derivation of dynamic model. To find the components of a vector in both of the frames, transformation matrices have to be formulated at first. By making successive rotations around corresponding axis, one can move from one frame to another.

2.2.3.1 Transformation of Translational Velocities

To relate the translational velocity of quadrotor in the components of body fixed frame and Earth fixed frame, we have to make three sequence of rotations [69]. The order of rotations are important and in this thesis, the common order of yaw, pitch, roll(ψ, θ, ϕ) will be used [70].

By making three successive transformations, we can find the transformation matrix L_{BE} that transforms F_E to F_B as follows [65]:

$$L_{BE} = L(\phi)L(\theta)L(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

For notational simplicity, trigonometric functions cosine() and sine() are shortened as c() and s(). By extending Equation (2.11), final form of L_{BE} is obtained as follows [65]:

$$L_{BE} = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & s(\phi)c(\theta) \\ c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) & c(\phi)c(\theta) \end{bmatrix} \tag{2.12}$$

Let's V_B and V_E be the components of the translational velocity of quadrotor in frames F_B and F_E, respectively. Then by using transformation matrix obtained in Equation (2.12), one can obtain V_B as follows:

$$V_B = L_{BE}V_E \tag{2.13}$$

It is important to note that, transformation matrix L_{BE} is nonsingular, therefore inverse of L_{BE} exists. By multiplying both sides of Equation (2.13) with L_{BE}^{-1}, one can also obtain V_E as follows:

$$V_E = L_{BE}^{-1}V_B \tag{2.14}$$

By using the same terminology as in Equation (2.13), we can define L_{EB} (transformation matrix that transforms F_B to F_E) as following:

$$V_E = L_{EB} V_B \tag{2.15}$$

Then, from Equations (2.14) and (2.15), one can find that:

$$L_{EB} = L_{BE}^{-1} \tag{2.16}$$

To obtain L_{EB} more easily , instead of finding inverse of L_{BE}, one can use the orthogonality property of transformations. In Equation (2.11), all of the transformation matrices are orthogonal, then the product L_{BE} is also orthogonal. Therefore, following property of orthogonal matrices can be used to find the inverse of L_{BE} more easily [71]:

$$L_{BE}^{-1} = L_{BE}^{T} \tag{2.17}$$

Then, by using Equations (2.16), (2.17) and (2.12), L_{EB} which transforms F_B to F_E is obtained as following:

$$L_{EB} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \tag{2.18}$$

By differentiating Equation (2.7) with respect to time, V_E can be expressed as in equation 2.19. V_E is translational velocity of quadrotor relative to reference frame F_E and expressed in reference frame F_E.

$$V_E = \dot{\xi} = [\dot{x}, \dot{y}, \dot{z}]^T \tag{2.19}$$

Then, by using Equations (2.15), (2.18), (2.19) and (2.10), kinematic equation for translational motion can be obtained as follows:

$$\dot{\xi} = L_{EB} V_B \tag{2.20}$$

2.2.3.2 Transformation of Angular Velocities

First, by taking derivative of Euler angles defined in equation (2.8) with respect to time, rate of change of Euler angles are defined as following:

$$\dot{\eta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T \tag{2.21}$$

To find the transformation between body angular velocities $[p, q, r]$ and rate of change of Euler angles $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$, a similar approach used in previous Subsection (2.2.3.1) can be used. However, transformation is not straightforward since the rate of change of Euler angles expressed in Equation (2.21) are not directly defined in reference frame F_E.

By using transformations defined in Equation 2.22, one can obtain body angular velocities $[p, q, r]$ as follows [72]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = L(\phi)L(\theta)L(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + L(\phi)L(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + L(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{2.22}$$

Transformation matrices $L(\phi)$, $L(\theta)$ and $L(\psi)$ are already defined in Equation (2.11). By putting $L(\phi)$, $L(\theta)$ and $L(\psi)$ into Equation (2.22), final form of Equation (2.22) is obtained as follows:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.23}$$

Transformation matrix found in Equation (2.23) is defined as L_R:

$$L_R \triangleq \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \tag{2.24}$$

By multiplying both sides of Equation (2.23) by L_R^{-1}, rate of change of Euler angles can also be found as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)/cos(\theta) & cos(\phi)/cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.25}$$

By integrating Equation (2.25), Euler angles can be obtained. However, as can be seen in Equation (2.25), at $\theta = \mp\ 90°$, there exists a singularity. For some inertial measurement systems, this singularity could cause the phenomenon called as "gimbal lock" [72]. To avoid this problem, pitch angle θ is constrained between ($-89°$,$+89°$) which is a reasonable limitation.

To conclude, by using Equations (2.23), (2.24), (2.21) and (2.9), kinematic equation for rotational motion is obtained as follows:

$$\omega = L_R \dot{\eta} \tag{2.26}$$

In Equation (2.26), ω represents body angular velocities $[p, q, r]^T$ and $\dot{\eta}$ represents rate of change of Euler angles $[\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$, as explained previously.

2.3 Defining External Forces and Moments

In this section external forces and moments act on quadrotor body will be defined. Some of the forces and moments will be neglected for simplicity. The most dominant forces and moments defined in this section will be used to derive the nonlinear dynamic model in Section (2.4). While deriving the dynamic model in Section (2.4), the forces and moments are used such that they are expressed in body fixed reference frame(F_B). Therefore, the final forms of the forces and moments defined in this section are expressed in F_B.

2.3.1 Gravitational Force

As explained previously, gravitational field is assumed to be constant. Then, gravitational acceleration is taken as $g = 9.81\ m/s^2$. Gravity acts at the center of gravity(COG) of the quadrotor. Then, gravitational force expressed in F_E, $F_{grav,E}$ can be

written as follows:

$$F_{grav,E} = -m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2.27}$$

By using transformation matrix L_{BE} obtained in Equation (2.12), gravitational force expressed in F_B, $F_{grav,B}$ can be written as follows:

$$F_{grav,B} = -L_{BE}\, m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2.28}$$

As mentioned previously, in the derivation of dynamic model, Equation (2.28) which is expressed in F_B will be used.

2.3.2 Force Generated by Propeller System

As explained previously, quadrotor has four propellers. The total force generated by four propellers defined in Equation (2.3) as control input, U_1. In Equation (2.3), U_1 is expressed in F_B. U_1 is in the direction of $O_B z_B$ that is illustrated in Figure 2.1 and has no contributions in $O_B x_B$ and $O_B y_B$ axes. Then, force generated by propeller system expressed in F_B, $F_{prop,B}$ can be written as follows:

$$F_{prop,B} = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} \tag{2.29}$$

2.3.3 Aerodynamic Drag Force

The effect of aerodynamic drag is considered for translational dynamics only. Actually, as a basic knowledge in aerodynamics, aerodynamic drag force is proportional to the square of velocity. However, most of the works related to small size quadrotors like "AscTech Hummingbird" used in this thesis, assume that the relation between aerodynamic drag force and translational speed could be taken as proportional [30, 31, 32, 34]. Therefore, in this thesis, it is also assumed that aerodynamic

drag force for translational motion is proportional to translational velocities $[u, v, w]^T$. This assumption is also tested by simulations in Chapter 4 and it is seen that the aerodynamic drag force could be approximated as in Equation (2.30), instead of a quadratic relation. The reason for this behavior might depend on small scale of "AscTech Hummingbird" quadrotor and low translational speeds (5-10 m/s) [32, 33].

Then, aerodynamic drag force for translational motion expressed in F_B, $F_{aero,B}$ is defined as following:

$$F_{aero,B} = -K_t V_B = -K_t \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.30}$$

K_t is a matrix with constant diagonal entries which are taken as in Equation (2.31) according to previous works on small size quadrotors [30, 31, 32, 34]:

$$K_t = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{2.31}$$

Diagonal entries of K_t matrix could be found and updated by using a more accurate drag model in future works.

2.3.4 Moment Generated by Propeller System

By using Equations (2.4), (2.5) and (2.6), moment generated by propellers system expressed in F_B, $M_{prop,B}$ can be written as follows:

$$M_{prop,B} = \begin{bmatrix} U_2 d \\ U_3 d \\ U_4 \end{bmatrix} \tag{2.32}$$

In Equation (2.32), d represents the moment arm (distance between propeller and COG of the quadrotor) which can be seen in Figure 2.1. U_2, U_3 and U_4 are control inputs and they are defined in Equations (2.4), (2.5) and (2.6), respectively.

In Equation (2.32), the term "$U_2 d$" is differential moment generated between left and right propellers and it is in the direction of $O_B x_B$. Similarly, the term "$U_3 d$" is differential moment generated between front and rear propellers and it is in the direction of $O_B y_B$, whereas U_4 is the net torque generated by all of the propellers in the direction of $O_B z_B$. The directions of $O_B x_B$, $O_B y_B$ and $O_B z_B$ are illustrated in Figure 2.1.

2.4 Obtaining The Nonlinear Dynamic Model

In this section, the nonlinear dynamic model of the quadrotor will be derived by using Newton's equations of motion. As stated earlier, Newton's equations of motion, $\vec{f} = m\vec{a}$, is valid for inertial reference frames. However, in this thesis, Newton's equations of motion will be written in body fixed reference frame F_B which is not an inertial frame of reference. Therefore, additional terms are included into $\vec{f} = m\vec{a}$.

Then, Newton's law for translational and rotational motion written in the reference frame F_B can be obtained as follows, respectively [34]:

$$\sum F_{ext} = m\dot{V}_B + \omega \times (mV_B) \tag{2.33}$$

$$\sum M_{ext} = J\dot{\omega} + \omega \times (J\omega) \tag{2.34}$$

In Equations (2.33) and (2.34), $\sum F_{ext}$ and $\sum M_{ext}$ represent the net force and moment acting on quadrotor body expressed in body fixed reference frame F_B, respectively. Then, $\sum F_{ext}$ and $\sum M_{ext}$ are expressed as follows:

$$\sum F_{ext} = F_{grav,B} + F_{prop,B} + F_{aero,B} \tag{2.35}$$

$$\sum M_{ext} = M_{prop,B} \tag{2.36}$$

By substituting Equations (2.28), (2.29), (2.30) and (2.32) into Equations (2.35) and (2.36), extended form of $\sum F_{ext}$ and $\sum M_{ext}$ which are written in body fixed frame

F_B can be obtained as follows:

$$\sum F_{ext} = -L_{BE}\, m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - K_t V_B \tag{2.37}$$

$$\sum M_{ext} = \begin{bmatrix} U_2\, d \\ U_3\, d \\ U_4 \end{bmatrix} \tag{2.38}$$

2.4.1 The Nonlinear Dynamic Model for Translational Motion

In the derivation of the nonlinear dynamic model for translational motion, Equation (2.33) will be used as a basis. LHS of Equation (2.33) is obtained in Equation (2.37). On the other hand, RHS of Equation (2.33) includes the term $\dot{V}_B$ which can be obtained by differentiating kinematic Equation (2.20) with respect to time as following:

$$\ddot{\xi} = \dot{L}_{EB} V_B + L_{EB} \dot{V}_B \tag{2.39}$$

In Equation (2.39) , time derivative of transformation matrix L_{EB} can be obtained as in Equation (2.40) by using the orthogonality property of transformation matrix L_{EB} [67].

$$\dot{L}_{EB} = L_{EB}\, \tilde{\omega} \tag{2.40}$$

where:

$$\tilde{\omega} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}, \quad \tilde{\omega} S = \omega \times S, \quad S \in \mathbb{R}^3 \tag{2.41}$$

In Equation (2.41), the term "$\times$" represents "cross product", ω defined in Equation (2.9) represents the angular velocity of quadrotor defined in body fixed frame F_B and S is any vector in $\mathbb{R}^3$.

By substituting Equations (2.40) and (2.41) into Equation (2.39), one can obtain:

$$\ddot{\xi} = L_{EB}(\omega \times V_B) + L_{EB} \dot{V}_B \tag{2.42}$$

Then, rearranging Equation (2.42), $\dot{V}_B$ can be obtained as follows:

$$\dot{V}_B = L_{EB}^{-1}\, \ddot{\xi} - (\omega \times V_B) \tag{2.43}$$

Then, substituting Equations (2.43) and (2.37) into Equation (2.33), following form can be obtained.

$$-L_{BE}\, m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - K_t V_B = m(L_{EB}^{-1}\, \ddot{\xi} - (\omega \times V_B)) + \omega \times (mV_B) \quad (2.44)$$

In Equation (2.44), the last term $\omega \times (mV_B)$ can be rearranged as $m(\omega \times V_B)$ since m is a constant. Also the term $(K_t V_B)$ can be written as $(K_t L_{EB}^{-1}\dot{\xi})$ by using Equation (2.20). Then, Equation (2.44) can be rewritten as follows:

$$-L_{BE}\, m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - K_t L_{EB}^{-1}\dot{\xi} = m(L_{EB}^{-1}\, \ddot{\xi} - (\omega \times V_B)) + m(\omega \times V_B) \quad (2.45)$$

Multiplying both sides of Equation (2.45) by L_{EB} from left and by making simplifications Equation (2.45) is rewritten as follows:

$$-L_{EB}L_{BE}\, m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + L_{EB} \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - L_{EB}K_t L_{EB}^{-1}\dot{\xi} = L_{EB} m L_{EB}^{-1}\, \ddot{\xi} \quad (2.46)$$

In Equation (2.46), the matrix product $(L_{EB}L_{BE})$ equals to $I_{3\times3}$ since $L_{EB} = L_{BE}^{-1}$ according to Equation (2.16). The term $(L_{EB}K_t L_{EB}^{-1}\dot{\xi})$ can be rewritten as $(L_{EB}L_{EB}^{-1}$ $K_t\dot{\xi})$ since K_t is an identity matrix multiplied with a constant. In addition, the term $(L_{EB}mL_{EB}^{-1})$ can be rewritten as $(mL_{EB}L_{EB}^{-1})$ since m is a constant. After simplifications, Equation (2.46) can be rearranged as following:

$$\ddot{\xi} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + L_{EB} \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} - (K_t/m)\dot{\xi} \quad (2.47)$$

Replacing $\ddot{\xi} = [\ddot{x}, \ddot{y}, \ddot{z}]^T$ and $\dot{\xi} = [\dot{x}, \dot{y}, \dot{z}]^T$ into Equation (2.47), final form of the translational equations of motion can be obtained as follows:

$$\boxed{\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + L_{EB} \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} - (K_t/m) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}} \quad (2.48)$$

where L_{EB} is found in Equation (2.18) as following:

$$L_{EB} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix}$$

In Equation (2.48), the position(x, y, z) of quadrotor is defined in Earth fixed inertial reference frame F_E, as stated earlier. Therefore, Equation (2.48) can be directly used in simulations to find the position of quadrotor expressed in Earth fixed reference frame F_E.

2.4.2 The Nonlinear Dynamic Model for Rotational Motion

Equation (2.34) will be used as a basis to derive the nonlinear dynamic model for rotational motion. LHS of Equation (2.34) is expressed in Equation (2.38). The terms used in RHS of Equation (2.34) are body inertia matrix J and body angular velocity vector ω and its time derivative $\dot{\omega}$. ω is defined in Equation (2.9) and its time derivative is defined as follows:

$$\dot{\omega} = [\dot{p}, \dot{q}, \dot{r}]^T \tag{2.49}$$

In addition, body inertia matrix J is defined as follows:

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.50}$$

By substituting Equations (2.38), (2.9), (2.49) and (2.50) into Equation (2.34), following equation can be obtained:

$$\begin{bmatrix} U_2 d \\ U_3 d \\ U_4 \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \right) \tag{2.51}$$

Rearranging Equation (2.51), the following form is obtained for rotational dynamic

model of quadrotor:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_y - I_z)qr/I_x \\ (I_z - I_x)pr/I_y \\ (I_x - I_y)pq/I_z \end{bmatrix} + \begin{bmatrix} U_2 d/I_x \\ U_3 d/I_y \\ U_4/I_z \end{bmatrix} \tag{2.52}$$

Equation (2.52) is written in terms of body angular velocities $[p, q, r]$. Therefore, to obtain the orientation of the quadrotor relative to the Earth surface, Euler angles $[\phi, \theta, \psi]$ have to be calculated.

To calculate Euler angles, first, $[\dot{p}, \dot{q}, \dot{r}]$ obtained in Equation (2.52) are integrated and body angular velocities $[p, q, r]$ are obtained. Then, by using transformation between body angular velocities $[p, q, r]$ and rate of change of Euler angles $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$ which is defined in Equation (2.25), one can obtain $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$ as following;

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)/cos(\theta) & cos(\phi)/cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.53}$$

Finally, $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$ obtained in Equation (2.53) are integrated to obtain Euler angles $[\phi, \theta, \psi]$. As explained earlier, in Equation (2.53), at $\theta = \mp\ 90^\circ$ a singularity occurs. Therefore, to avoid this problem, pitch angle θ is constrained between (-89°,$+89^\circ$) which is a reasonable limitation.

2.4.3 The Simplified Nonlinear Dynamic Model for Rotational Motion

As explained earlier, dynamic model of the quadrotor is used to obtain control methods in Chapter 3. To simplify the control laws and algorithms, we can simplify the dynamic model used in the derivation of control laws. Since, translational dynamics obtained in Equation (2.48) is not very complex, no simplification is needed. However, for rotational dynamics obtained in Equation (2.52) simplification is required since transformation between $[\dot{p}, \dot{q}, \dot{r}]$ and $[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}]$ is very complex and involves so many trigonometric terms and derivatives, as can be seen in Equation (2.53).

Therefore, rotational dynamic model obtained in Equation (2.52) can be simplified by making the assumption that body angular velocities $[p, q, r]$ and rate of change of Euler angles $[\dot{\phi}, \dot{\theta}, \dot{\psi}]$ are equal if perturbations from hover condition are small [29].

To clarify this assumption, we can examine tranformation matrix L_R defined in Equations (2.23) and (2.24). For small ϕ and θ, L_R can be taken as $I_{3\times3}$. Then, relation between body angular velocities and rate of change of Euler angles become as following:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{2.54}$$

Substituting the relations defined in Equation (2.54) into Equation (2.52), the simplified nonlinear dynamic model for rotational motion is obtained as follows:

$$\boxed{\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} (I_y - I_z)\dot{\theta}\dot{\psi}/I_x \\ (I_z - I_x)\dot{\phi}\dot{\psi}/I_y \\ (I_x - I_y)\dot{\phi}\dot{\theta}/I_z \end{bmatrix} + \begin{bmatrix} U_2 d/I_x \\ U_3 d/I_y \\ U_4/I_z \end{bmatrix}} \tag{2.55}$$

While deriving control laws in Chapter 3, for rotational dynamics, simplified form obtained in Equation (2.55) will be used. The simplification is based on the assumption defined in Equation (2.54). The simulation results obtained in Chapter 4 show that this assumption is valid even if Euler angles are not small. According to the simulation results obtained in Chapter 4, for Euler angles up to $\mp 60^{\circ}$, control systems work properly. It is important to note that, to validate the assumption made in Equation (2.54), the non simplified equations of motion which are obtained in Equations (2.52) and (2.53) are used as rotational dynamic model, while testing control systems by simulations.

To conclude, the nonlinear equations of motion which are not simplified and obtained in Equations (2.48), (2.52) and (2.53) are used in simulations as dynamic model of the quadrotor. However, while obtaining controllers in Chapter 3, simplified rotational equations of motion obtained in Equations (2.55) are used for simplicity, whereas translational equations motion obtained in Equation (2.48) are used without simplification.

CHAPTER 3

CONTROL METHODS

In this chapter, theory of backstepping, LQT and fixed-gain LQR controllers are explained and each control system are formulated and modeled in MATLAB/Simulink.

3.1 Backstepping Controller

In this section, a nonlinear backstepping controller will be designed based on the method described in [2]. The aim of the controller is tracking of a desired trajectory asymptotically.

To get more accurate results, the effects of aerodynamic drag for translational motion and the second order derivatives of desired trajectory are added into the control law. In addition, a simpler flight approach is obtained by controlling the yaw angle such that the heading of the quadrotor and the direction of motion are along the same line. To achieve good tracking performance for relatively complex trajectories, a command filter is also added to the controller.

The backstepping control method can be explained in three steps. First step is the attitude control of the quadrotor. In this step, control inputs U_2, U_3, U_4 which are directly related to the attitude of the quadrotor, are formulated. Second step is about the position control of the quadrotor. In this step, control input U_1 and functions u_x and u_y related to the orientation of U_1, are formulated. In the third step, desired "Euler angles" (desired orientation) which are used to find control inputs U_2, U_3 and U_4 are found by using the functions u_x, u_y formulated in the position control.

It is also important to note that, one of the most difficult and important part of backstepping controller design is the selection of control parameters. The backstepping controller obtained in this thesis has 12 control parameters($\alpha_1, \alpha_2, ...\alpha_{11}, \alpha_{12}$) to be determined. Therefore, "MATLAB Optimization Toolbox" is used to obtain control parameters that give satisfactory results.

The simplified nonlinear dynamic model of the quadrotor obtained in Equations (2.48) and (2.55) of Chapter 2, rewritten in a complete form in Equation (3.1). Complete nonlinear dynamic model of the quadrotor expressed in Equation (3.1) is used in the formulation of backstepping controller [2].

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{pmatrix} = \begin{pmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} x_2 \\ x_4 x_6 a_1 + b_1 U_2 \\ x_4 \\ x_2 x_6 a_3 + b_2 U_3 \\ x_6 \\ x_4 x_2 a_5 + b_3 U_4 \\ x_8 \\ u_x U_1/m - K_x x_8/m \\ x_{10} \\ u_y U_1/m - K_y x_{10}/m \\ x_{12} \\ -g + (cos(x_1)cos(x_3))U_1/m - K_z x_{12}/m \end{pmatrix} \tag{3.1}$$

where:

$$a_1 = (I_y - I_z)/I_x$$

$$a_3 = (I_z - I_x)/I_y$$

$$a_5 = (I_x - I_y)/I_z$$

$$b_1 = d/I_x$$

$$b_2 = d/I_y$$

$$b_3 = 1/I_z$$

$$u_x = (cos(x_1)sin(x_3)cos(x_5) + sin(x_1)sin(x_5))$$

$$u_y = (cos(x_1)sin(x_3)sin(x_5) - sin(x_1)cos(x_5))$$

The terms a_1, a_3, a_5 and b_1, b_2, b_3 are defined according to Equation (2.55) which is the final form of the simplified nonlinear dynamic model for rotational motion. Similarly, the terms u_x and u_y are defined according to Equations (2.48) and (2.18) which is the final form of the nonlinear dynamic model for translational motion. In addition, U_1, U_2, U_3,U_4 and K_x, K_y, K_z are control inputs and aerodynamic drag coefficients for translational motion, respectively. All of the other parameters of Equation (3.1) are defined in Table 3.1 and in Chapter 2.

The symbols and parameters used for the backstepping controller design are defined in Table 3.1.

Table 3.1: Symbols used in backstepping controller.

Symbol	Definition
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
Euler angles	ϕ,θ,ψ
x_{1d},x_{3d},x_{5d}	Desired roll,pitch,yaw angles
x_{7d},x_{9d},x_{11d}	Desired x,y,z coordinates
U_1,U_2,U_3,U_4	Control inputs
$\alpha_1, \alpha_2, ..., \alpha_{11}, \alpha_{12}$	Control parameters
K_x,K_y,K_z	Aerodynamic drag coefficients for translational motion
m	Mass of the quadrotor
I_x,I_y,I_z	Body inertia terms
g	Gravitational acceleration
d	Level arm
$\dot{f}(\cdot)$	First order time derivative
$\ddot{f}(\cdot)$	Second order time derivative

3.1.1 Step 1: Attitude Control

In this part, control inputs U_2, U_3, U_4 are formulated to obtain desired orientation or "Euler Angles" of the quadrotor.

First, error term for "Euler angle" $\phi(x_1)$ is defined as:

$$z_1 = x_{1d} - x_1 \tag{3.2}$$

Here x_{1d} represents the desired value of angle ϕ. By choosing a Lyapunov function of z_1 which is positive definite itself and has negative semi-definite first order derivative with respect to time, error term z_1 converges to zero [73]. Then, Lyapunov function is chosen as:

$$V(z_1) = (1/2)z_1^2 \tag{3.3}$$

First order derivative of V(z_1) with respect to time is found as:

$$\dot{V}(z_1) = z_1\dot{z}_1 = z_1(\dot{x}_{1d} - \dot{x}_1) \tag{3.4}$$

By using Equation (3.1), we can replace $\dot{x}_1$ by x_2. Then Equation (3.4) becomes:

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - x_2) \tag{3.5}$$

Here, if x_2 is taken as, $x_2 = \dot{x}_{1d} + \alpha_1 z_1$, then $\dot{V}(z_1)$ is negative semi-definite for α_1>0. In other words, $x_{2d}(desired\ x_2)$ is chosen as following:

$$x_{2d} = \dot{x}_{1d} + \alpha_1 z_1 \quad for \quad \alpha_1 > 0 \tag{3.6}$$

To clarify, by controlling x_2 $(\dot{\phi})$, desired ϕ (x_{1d}) can be obtained.

Second, it is obvious that, x_2 in Equation (3.5) is coming from quadrotor dynamics defined in Equation (3.1). Therefore, to obtain desired $x_2(x_{2d})$ which is defined in Equation (3.6), we need another error term which converges to zero.

Then, error term z_2 related to $x_2(\dot{\phi})$ is defined as following:

$$z_2 = x_{2d} - x_2 \tag{3.7}$$

To obtain x_{1d} and x_{2d}, we want both of the error terms z_1 and z_2 converge to zero.

Therefore, another Lyapunov function which is positive definite itself and has negative semi-definite first order derivative with respect to time, have to be defined.

This new Lyapunov function is chosen as follows:

$$V(z_1, z_2) = (1/2)(z_1^2 + z_2^2) \tag{3.8}$$

By differentiating Equation (3.8) with respect to time, $\dot{V}(z_1, z_2)$ is obtained as:

$$\dot{V}(z_1, z_2) = z_1\dot{z}_1 + z_2\dot{z}_2 \tag{3.9}$$

We need $\dot{z}_1$ and $\dot{z}_2$ to replace into Equation (3.9). By using Equations (3.6) and (3.7), z_2 can be written as:

$$z_2 = \dot{x}_{1d} + \alpha_1 z_1 - x_2 \tag{3.10}$$

In Equation (3.10) we can replace $\dot{z}_1$ with the term $(\dot{x}_{1d} - x_2)$ by using Equation (3.5). Then z_2 becomes:

$$z_2 = \dot{z}_1 + \alpha_1 z_1 \tag{3.11}$$

From Equation (3.11), $\dot{z}_1$ is found as:

$$\dot{z}_1 = z_2 - \alpha_1 z_1 \tag{3.12}$$

In addition by differentiating Equation (3.10) with respect to time, $\dot{z}_2$ is obtained as:

$$\dot{z}_2 = \ddot{x}_{1d} + \alpha_1 \dot{z}_1 - \dot{x}_2 \tag{3.13}$$

By replacing Equations (3.12) and (3.13) into Equation (3.9), $\dot{V}(z_1, z_2)$ is obtained as:

$$\dot{V}(z_1, z_2) = z_1(z_2 - \alpha_1 z_1) + z_2(\ddot{x}_{1d} + \alpha_1 \dot{z}_1 - \dot{x}_2) \tag{3.14}$$

In Equation (3.14), $\dot{x}_2$ comes from quadrotor dynamics and can be replaced by the term $(x_4 x_6 a_1 + b_1 U_2)$ by using Equation (3.1).

By replacing $\dot{x}_2$ with $(x_4 x_6 a_1 + b_1 U_2)$ and rearranging the terms, final form of $\dot{V}(z_1, z_2)$ is obtained as:

$$\dot{V}(z_1, z_2) = z_1 z_2 - \alpha_1 z_1^2 + z_2 \ddot{x}_{1d} + z_2^2 \alpha_1 - \alpha_1^2 z_1 z_2 - z_2 x_4 x_6 a_1 - z_2 b_1 U_2 \tag{3.15}$$

In Equation (3.15), U_2 represents the control input. One can choose U_2 such that $\dot{V}(z_1, z_2)$ is negative semi-definite, therefore, error terms z_1 and z_2 converge to zero according to the Lyapunov theory [73].

The effect of the second order derivatives of desired angles($\ddot{x}_{1d}$, $\ddot{x}_{3d}$, $\ddot{x}_{5d}$) can be neglected for attitude control, then, the term $\ddot{x}_{1d}$ in Equation (3.15) is taken as zero. However, for position control explained in Subsection 3.1.2, second order derivatives of desired positions are included into the control law for better tracking performance.

Then, control input U_2 is chosen as:

$$U_2 = (1/b_1)(z_1 + \alpha_1 z_2 - \alpha_1^2 z_1 - x_4 x_6 a_1 + \alpha_2 z_2), \quad for \quad \alpha_1 > 0, \quad \alpha_2 > 0 \quad (3.16)$$

By substituting Equation (3.16) into Equation (3.15), $\dot{V}(z_1, z_2)$ becomes:

$$\dot{V}(z_1, z_2) = -\alpha_1 z_1^2 - \alpha_2 z_2^2 < 0, \quad for \quad \alpha_1 > 0, \alpha_2 > 0 \quad and \quad z_1 \neq 0, z_2 \neq 0 \quad (3.17)$$

In other words, if control input U_2 is chosen as Equation (3.16), $\dot{V}(z_1, z_2)$ is negative semi-definite and convergence of error terms z_1 and z_2 are guaranteed [73].

By using the same method, control inputs U_3 and U_4 related to θ and ψ respectively, are found as following:

$$U_3 = (1/b_2)(z_3 + \alpha_3 z_4 - \alpha_3^2 z_3 - x_2 x_6 a_3 + \alpha_4 z_4), \quad for \quad \alpha_3 > 0, \quad \alpha_4 > 0 \quad (3.18)$$

$$U_4 = (1/b_3)(z_5 + \alpha_5 z_6 - \alpha_5^2 z_5 - x_4 x_2 a_5 + \alpha_6 z_6), \quad for \quad \alpha_5 > 0, \quad \alpha_6 > 0 \quad (3.19)$$

The terms $\ddot{x}_{3d}$ and $\ddot{x}_{5d}$ are also taken as zero while finding U_3 and U_4.

Error terms used in Equations (3.18) and (3.19) are defined as follows:

$$z_3 = x_{3d} - x_3 \quad (3.20)$$

$$z_4 = x_{4d} - x_4 \quad (3.21)$$

$$z_5 = x_{5d} - x_5 \quad (3.22)$$

$$z_6 = x_{6d} - x_6 \quad (3.23)$$

where, x_{3d}, x_{4d}, x_{5d}, x_{6d} are desired θ, $\dot{\theta}$, ψ, $\dot{\psi}$, respectively. In Equations (3.21) and (3.23), x_{4d} and x_{6d} are also chosen such that the first order derivative of Lyapunov functions with respect to time are negative semi-definite.

$$x_{4d} = \dot{x}_{3d} + \alpha_3 z_3 \quad for \quad \alpha_3 > 0 \quad (3.24)$$

$$x_{6d} = \dot{x}_{5d} + \alpha_5 z_5 \quad for \quad \alpha_5 > 0 \quad (3.25)$$

In Equations (3.2), (3.20) and (3.22), desired Euler angles x_{1d}(desired ϕ), x_{3d}(desired θ), x_{5d}(desired ψ) are obtained in Subsection 3.1.3.

3.1.2 Step 2: Position Control

In this part, first, control input U_1 will be obtained by using the same method explained in Subsection 3.1.1. Control input U_1 is the total force generated by four

propellers of quadrotor. Therefore, it is directly related to the motion in translational directions(x, y, z), which can be clearly seen in the dynamic model of quadrotor defined in Equation (3.1). Moreover, by considering the dynamic model that is defined in Equation (3.1), one can see that motion in x and y directions are also effected by u_x and u_y, respectively. Therefore, in the first part of position control, altitude control will be obtained by choosing U_1 properly and in the second part, motion in x and y directions will be obtained by choosing u_x and u_y properly.

As mentioned before, in the first part of position control, U_1 will be obtained such that the error between desired and actual altitude converges to zero according to the Lyapunov theory [73].

The error term for altitude is defined as:

$$z_{11} = x_{11d} - x_{11} \tag{3.26}$$

A Lyapunov function of z_{11} is chosen as $V(z_{11}) = (1/2)z_{11}^2$, such that it is positive definite.

Then, by using Equations (3.26) and (3.1), first order derivative of $V(z_{11})$ with respect to time is obtained as:

$$\dot{V}(z_{11}) = z_{11}\dot{z}_{11} = z_{11}(\dot{x}_{11d} - \dot{x}_{11}) = z_{11}(\dot{x}_{11d} - x_{12}) \tag{3.27}$$

If, $\dot{V}(z_{11})$ is negative semi-definite, error term z_{11} converges to zero according to the Lyapunov theory [73]. Then, if x_{12} is chosen as, $x_{12} = \dot{x}_{11d} + \alpha_{11}z_{11}$ for α_{11}>0, then $\dot{V}(z_{11})$ defined in Equation (3.27) is negative semi-definite and error term z_{11} defined in Equation (3.26) converges to zero. Therefore, desired x_{12}(x_{12d}) is defined as follows:

$$x_{12d} = \dot{x}_{11d} + \alpha_{11}z_{11}, \quad for \;\; \alpha_{11} > 0 \tag{3.28}$$

To obtain x_{12d} defined in Equation (3.28), one can control $x_{12}(\dot{z})$ which comes from quadrotor dynamics defined in Equation (3.1)

Therefore, to obtain x_{12d}, another error term z_{12} is defined as following:

$$z_{12} = x_{12d} - x_{12} = \dot{x}_{11d} + \alpha_{11}z_{11} - x_{12} \tag{3.29}$$

We want both of the error terms z_{11}, z_{12} defined in Equations (3.26) and (3.29) respectively, converge to zero. Then, a Lyapunov function of z_{11} and z_{12} is chosen

as:

$$V(z_{11}, z_{12}) = (1/2)(z_{11}^2 + z_{12}^2) \tag{3.30}$$

If, first order derivative of $V(z_{11}, z_{12})$ with respect to time is negative semi-definite, then convergence of error terms z_{11} and z_{12} are guaranteed [73]. By taking first order derivative of Equation (3.30) with respect to time, $\dot{V}(z_{11}, z_{12})$ is obtained as:

$$\dot{V}(z_{11}, z_{12}) = z_{11}\dot{z}_{11} + z_{12}\dot{z}_{12} \tag{3.31}$$

$\dot{z}_{11}$ and $\dot{z}_{12}$ are required to replace into Equation (3.31). By taking first order derivative of Equation (3.26) with respect to time and using Equation (3.1) , one can obtain:

$$\dot{z}_{11} = \dot{x}_{11d} - \dot{x}_{11} = \dot{x}_{11d} - x_{12} \tag{3.32}$$

By replacing $\dot{z}_{11}$ found in Equation (3.32) into Equation (3.29), final form of $\dot{z}_{11}$ is obtained as:

$$\dot{z}_{11} = z_{12} - \alpha_{11} z_{11} \tag{3.33}$$

In addition, by differentiating Equation (3.29) with respect to time, $\dot{z}_{12}$ is obtained as:

$$\dot{z}_{12} = \ddot{x}_{11d} + \alpha_{11}\dot{z}_{11} - \dot{x}_{12} \tag{3.34}$$

By replacing Equations (3.33) and (3.34) into Equation (3.31), one can obtain:

$$\dot{V}(z_{11}, z_{12}) = z_{11}(z_{12} - \alpha_{11} z_{11}) + z_{12}(\ddot{x}_{11d} + \alpha_{11}\dot{z}_{11} - \dot{x}_{12}) \tag{3.35}$$

In Equation (3.35), $\dot{x}_{12}$ comes from quadrotor dynamics and can be replaced by the term $(-g + (cos(x_1)cos(x_3))U_1/m - K_z x_{12}/m)$ by using Equation (3.1). After replacing $\dot{x}_{12}$ and making rearrangements, final form of $\dot{V}(z_{11}, z_{12})$ is obtained as:

$$\begin{aligned} \dot{V}(z_{11}, z_{12}) = z_{11}z_{12} - \alpha_{11}z_{11}^2 + z_{12}\ddot{x}_{11d} + z_{12}^2\alpha_{11} - \alpha_{11}^2 z_{11}z_{12} + z_{12}g \\ -z_{12}(cos(x_1)cos(x_3))U_1/m + z_{12}K_z x_{12}/m \end{aligned} \tag{3.36}$$

In Equation (3.36), one can choose control input U_1 such that $\dot{V}(z_{11}, z_{12})$ is negative semi-definite. Then, proper choose of U_1 is defined as:

$$\begin{aligned} U_1 = (m/(cos(x_1)cos(x_3)))(z_{11} + \ddot{x}_{11d} + \alpha_{11}z_{12} - \alpha_{11}^2 z_{11} + g \\ +K_z x_{12}/m + \alpha_{12}z_{12}), \quad for \quad \alpha_{11} > 0, \;\; \alpha_{12} > 0 \end{aligned} \tag{3.37}$$

As can be seen in Equation (3.37), the effect of second order derivative of desired altitude($\ddot{x}_{11d}$) is added to the controller for better tracking performance. This effect was neglected while driving attitude controller in Subsection 3.1.1.

By choosing U_1 as in Equation (3.37), $\dot{V}(z_{11}, z_{12})$ which is defined in Equation (3.36) becomes:

$$\dot{V}(z_{11}, z_{12}) = -\alpha_{11} z_{11}^2 - \alpha_{12} z_{12}^2 < 0, \quad for \;\; \alpha_{11} > 0, \alpha_{12} > 0 \;\; and \;\; z_{11} \neq 0, z_{12} \neq 0 \tag{3.38}$$

Then, it can be clearly seen that in Equation (3.38), $\dot{V}(z_{11}, z_{12})$ is negative semi-definite. Besides, Lyapunov function $V(z_{11}, z_{12})$ was defined in Equation (3.30) as $V(z_{11}, z_{12}) = (1/2)(z_{11}^2 + z_{12}^2)$ which is positive definite. Then, according to Lyapunov theory, error terms z_{11} and z_{12} which are defined in Equations (3.26) and (3.29) converge to zero [73].

By using the same method, functions u_x and u_y which are directly related to the motion in x and y directions are found as following:

$$u_x = (m/U_1)(z_7 + \ddot{x}_{7d} + \alpha_7 z_8 - \alpha_7^2 z_7 + K_x x_8/m + \alpha_8 z_8) \quad for \;\; \alpha_7 > 0, \alpha_8 > 0 \tag{3.39}$$

$$u_y = (m/U_1)(z_9 + \ddot{x}_{9d} + \alpha_9 z_{10} - \alpha_9^2 z_9 + K_y x_{10}/m + \alpha_{10} z_{10}) \quad for \;\; \alpha_9 > 0, \alpha_{10} > 0 \tag{3.40}$$

In Equations (3.39) and (3.40) error terms are defined as follows:

$$z_7 = x_{7d} - x_7 \tag{3.41}$$

$$z_8 = x_{8d} - x_8 \tag{3.42}$$

$$z_9 = x_{9d} - x_9 \tag{3.43}$$

$$z_{10} = x_{10d} - x_{10} \tag{3.44}$$

In Equations (3.26), (3.41) and (3.43), the terms x_{11d}(desired z),x_{7d}(desired x) and x_{9d}(desired y) are given to the system as inputs that define desired trajectory.

In addition, terms x_{8d} and x_{10d} defined in Equations (3.42) and(3.44) are also chosen such that the first order derivative of Lyapunov functions with respect to time are negative semi-definite.

$$x_{8d} = \dot{x}_{7d} + \alpha_7 z_7 \quad for \quad \alpha_7 > 0 \tag{3.45}$$

$$x_{10d} = \dot{x}_{9d} + \alpha_9 z_9 \quad for \quad \alpha_9 > 0 \tag{3.46}$$

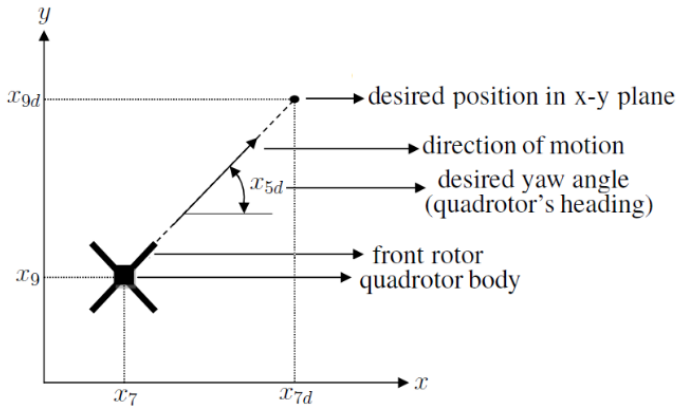


Figure 3.1: Top view of the quadrotor, motion in x-y plane.

The functions u_x and u_y which are obtained in Equations (3.39) and (3.40) respectively, are not directly control inputs, instead they will be used to find desired Euler angles x_{1d} and x_{3d} which will be the inputs of attitude controller explained in Subsection 3.1.1.

3.1.3 Step 3: Obtaining Desired Angles

In this part, desired Euler angles x_{1d}, x_{3d} and x_{5d} which are the inputs of attitude controller, will be obtained.

First, desired yaw angle(x_{5d}) will be obtained such that quadrotor's heading and direction of motion in $x-y$ plane are on the same line.

By using Figure 3.1 and basic trigonometry, desired yaw angle(x_{5d}) can be obtained as follows:

$$x_{5d} = \arctan[(x_{9d} - x_9)/(x_{7d} - x_7)] \tag{3.47}$$

In Equation (3.47), x_7 and x_9 represent the actual position in x-y plane whereas, x_{7d} and x_{9d} represent the desired position in x-y plane. Therefore, x_{7d} and x_{9d} are given to the system as inputs that define desired trajectory to be followed.

Once the desired yaw angle(x_{5d}) is obtained from Equation (3.47), desired roll(x_{1d}) and desired pitch(x_{3d}) angles can be obtained by using following equations which are

defined in quadrotor dynamic model, equation (3.1):

$$u_x = (cos(x_1)sin(x_3)cos(x_5) + sin(x_1)sin(x_5)) \tag{3.48}$$

$$u_y = (cos(x_1)sin(x_3)sin(x_5) - sin(x_1)cos(x_5)) \tag{3.49}$$

Multiply both sides of Equation (3.48) by $sin(x_5)$ and multiply both sides of Equation (3.49) by $cos(x_5)$.

$$u_x sin(x_5) = (cos(x_1)sin(x_3)cos(x_5)sin(x_5) + sin(x_1)sin^2(x_5)) \tag{3.50}$$

$$u_y cos(x_5) = (cos(x_1)sin(x_3)sin(x_5)cos(x_5)) - sin(x_1)cos^2(x_5)) \tag{3.51}$$

Subtract Equation (3.51) from Equation (3.50):

$$(u_x sin(x_5) - u_y cos(x_5)) = sin(x_1)(sin^2(x_5) + cos^2(x_5)) \tag{3.52}$$

By using the fact $[sin^2(x_5) + cos^2(x_5)] = 1$, desired roll angle ($x_{1d}$) is obtained from Equation (3.52) as:

$$x_{1d} = \arcsin[u_x sin(x_{5d}) - u_y cos(x_{5d})] \tag{3.53}$$

In Equation (3.53), value of x_{5d} is obtained from Equation (3.47) and functions u_x and u_y are obtained from Equations (3.39) and (3.40) respectively.

By using a similar approach desired pitch angle(x_{3d}) is obtained as;

$$x_{3d} = \arcsin[(u_x cos(x_{5d}) + u_y sin(x_{5d}))/cos(x_{1d})] \tag{3.54}$$

In Equation (3.54), value of x_{1d}, x_{5d}, u_x and u_y are obtained from Equations (3.53), (3.47), (3.39), (3.40), respectively.

By analyzing Equation (3.53) and Equation (3.54), it can be seen that no singularity exists for Equation (3.53). However, in Equation (3.54), for $x_{1d} = k\pi$ where $k = \mp 1, 3, 5, ..n$, singularity exist. To overcome this problem, value of x_{1d} obtained from Equation (3.53) is limited between ($-89°$,$+89°$) by using a saturation function. Limiting desired roll angle (x_{1d}) between ($-89°$,$+89°$) is a reasonable approach.

To sum up, in this step, desired roll, pitch and yaw angles(x_{1d}, x_{3d}, x_{5d}) are obtained in Equations (3.53),(3.54) and (3.47) respectively. Desired angles obtained in this step are used as inputs for the attitude controller explained in Subsection 3.1.1.

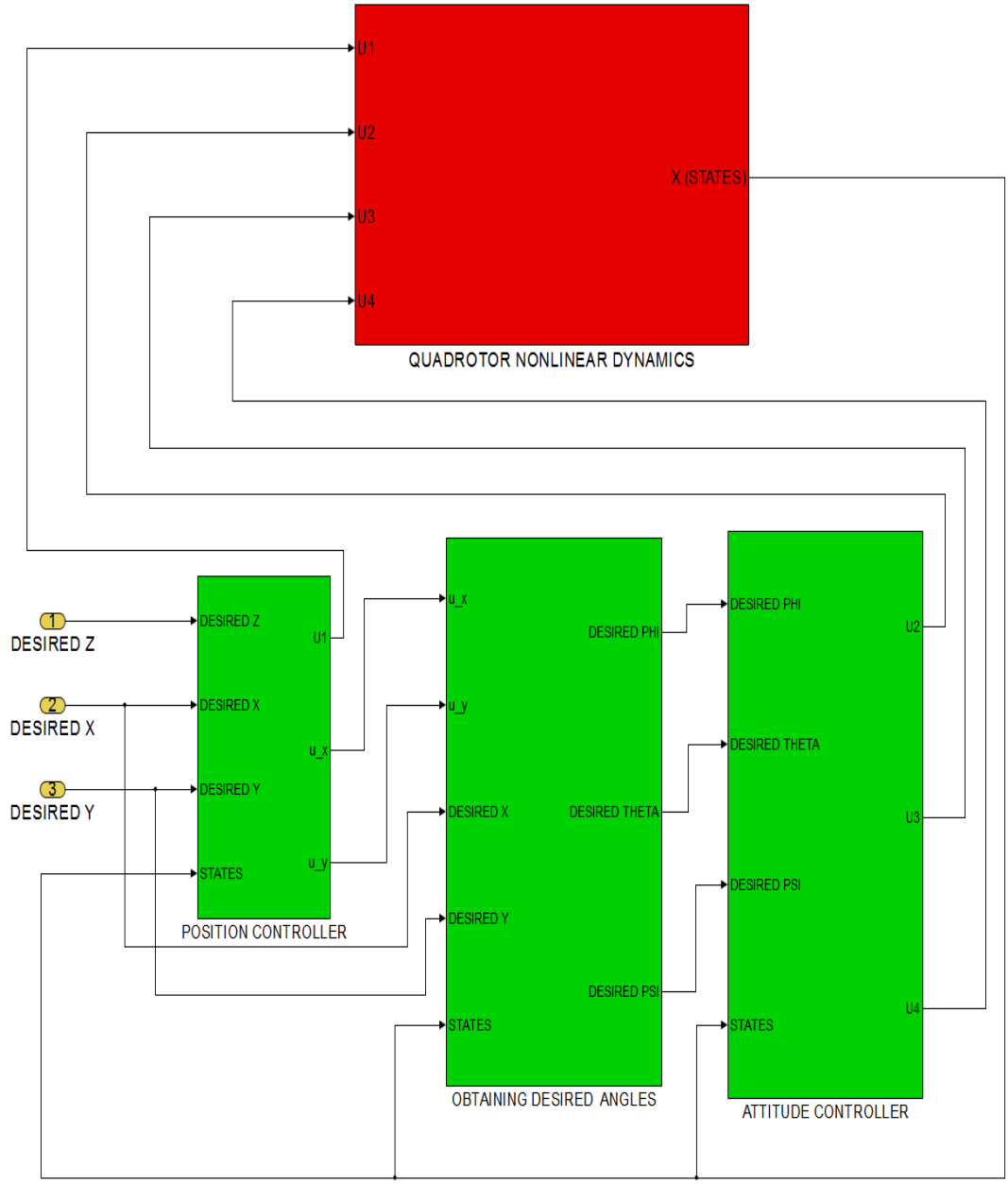


Figure 3.2: Each part of the bacsktepping controller with its inputs and outputs. Green blocks represent the backstepping controller and red block represents the nonlinear quadrotor dynamics.

Finally, backstepping controller to track desired trajectories is obtained in three steps. Figure 3.2 shows each part of the backstepping controller with its inputs and outputs to clarify the design process.

As mentioned previously, optimized control parameters of backstepping controller are found by using "MATLAB Optimization Toolbox" and illustrated in Table 3.2.

Table 3.2: Optimized control parameters of backstepping controler.

Parameter	Optimized value	Parameter	Optimized value
α_1	13.6185	α_7	4.3529
α_2	0.5994	α_8	0.0533
α_3	13.6185	α_9	4.3529
α_4	0.5994	α_{10}	0.0533
α_5	9.412	α_{11}	3.9018
α_6	12.213	α_{12}	0.2734

Moreover, results obtained in Chapter 4 shows that performance of the backstepping controller is directly related to the selection of control parameters. Therefore, optimization of control parameters is one of the most important and time consuming part of backstepping controller design.

3.2 Linear Quadratic Tracking(LQT) Control

In this section, our second and more unique control approach which is called as Linear Quadratic Tracking(LQT) will be explained. In this thesis, LQT controller is obtained by solving discrete-time optimal control algorithm defined in [1]. As explained previously, LQT is a discrete-time optimal control approach and hasn't been used widely in literature unlike fixed-gain LQR control which has been used by many researchers. LQT is similar to the classical fixed gain LQR control, however, the main difference and advantage of the LQT control is time varying control gains which are optimized to track desired trajectory in an offline way. Therefore, LQT control could be advantageous in some ways compared to fixed gain LQR control which uses time-invariant(fixed) control gains.

This section focus on obtaining time-varying optimal control gains of LQT controller by solving a discrete-time optimal control algorithm defined in [1]. The advantageous and disadvantageous properties of LQT control could be seen in details in Chapter 4.

First, the nonlinear quadrotor dynamics obtained in Chapter 2 will be linearized and expressed as a discrete time state space form in Subsection 3.2.1. Then, discrete time performance index(cost function) and state and control weight matrices will be defined in Subsection 3.2.2. Finally, discrete-time LQT algorithm which is solved to

obtain time-varying optimal control gains will be explained in Subsection 3.2.3.

3.2.1 Obtaining Discrete-Time State Space Equations

The LQT algorithm requires linear dynamic model of the system. Therefore, first the nonlinear dynamic model of the quadrotor obtained in Equations 2.48 and 2.55 for translation and rotational motion, have to be linearized around a trim (equilibrium) condition. The simplified nonlinear dynamic model of the quadrotor is expressed as a complete form in Equation 3.1.

Equation 3.1 is linearized around trim condition which is hovering at an altitude of 1 meter. Then, at trim condition, $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_7 = x_8 = x_9 = x_{10} = x_{12} = 0$, only z coordinate, x_{11} = 1 meter. Linearization is performed in a MATLAB code by using "MATLAB Symbolic Toolbox" which calculates Jacobian of the nonlinear system obtained in Equation 3.1 at hover condition.

Then, dynamic model of the quadrotor is linearized around hover condition and obtained as a continuous time state space form as following:

$$\left.\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX \end{aligned}\right\} \tag{3.55}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{pmatrix} = \begin{pmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{pmatrix}, \quad U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix}, \quad Y = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{pmatrix}$$

In Equation (3.55), A, B and C are state, input and output matrices of the linearized state space system and X, U and Y are state, input and output vectors, respectively. As mentioned previously, linearized A, B and C matrices are found by using "MATLAB Symbolic Toolbox" and are not explicitly expressed due to the large size of matrices.

The state space form of the quadrotor dynamics obtained in Equation 3.55 is continuous time, however, LQT algorithm is solved in discrete-time as can be seen in Equation set (3.60). Then, continuous time state space equations obtained in Equation (3.55) is discretized by choosing 0.01 seconds time interval. Discrete time state space equations obtained by using MATLAB are defined as following:

$$\left.\begin{aligned} X(k+1) &= A_d X(k) + B_d U(k) \\ Y(k) &= C_d X(k) \end{aligned}\right\} \tag{3.56}$$

In Equation (3.56), A_d, B_d and C_d are discrete time state, input and output matrices respectively. "k" represents discrete time step i.e. $k = 1, 2, ..., k_f$, such that $k \in \mathbb{Z}^+$.

3.2.2 Defining Performance Index

To obtain time-varying optimal control gains, a performance index(cost function) have to be defined and tried to be minimized. Performance index is defined in discrete time and it possess desired trajectory information at each time step since the aim is tracking a desired trajectory. Then, discrete time performance index J_d is defined as following [1]:

$$\begin{aligned} J = \frac{1}{2}[C_d X(k_f) - r(k_f)]^T F [C_d X(k_f) - r(k_f)] \\ + \frac{1}{2} \sum_{k=k_0}^{k_f - 1} \left([C_d X(k) - r(k)]^T Q [C_d X(k) - r(k)] + U^T(k) R U(k) \right) \end{aligned} \tag{3.57}$$

In Equation (3.57), $r(k)$ is a 12×1 vector that represents the desired state vector at time step k such that the first six rows are related to desired orientation and the last six rows are related to desired position. In other words, $r(k)$ contains information

about desired trajectory to be followed, at each time step k. C_d is discrete time output matrix and equals to 12×12 identity matrix. $X(k)$ and $U(k)$ are state and input vectors at time step k and defined in Equation (3.55).

Boundary condition at time step zero, $X(k_0) = X_0$ is initial hover condition defined in Subsection 3.2.1. At final time k_f, $X(k_f)$ is free and k_f is fixed so that F is taken as 12×12 zero matrix.

In equation (3.57), Q and R matrices are state weight and control weight matrices respectively. Diagonal elements of Q and R matrices are chosen as in Equation (3.58) by trial and error such that good tracking performance is achieved while satisfying the constraints due to mechanical limits of the quadrotor.

$$\left.\begin{aligned} Q = diag[100, 50, 10, 5, 0, 0, 100, 1, 100, 1, 1000, 0.1] \\ R = diag[10, 0, 0, 0] \end{aligned}\right\} \tag{3.58}$$

It is also important to remind that, LQT controller is obtained by using the simplified and linearized dynamics which is approximation of the nonlinear dynamic model in the vicinity of the hover condition. Therefore, it is also preferable to satisfy constraints on Euler angles defined in equation (3.59) to get more accurate results in simulations which are performed by using the nonlinear dynamic model as plant [74].

$$-20^\circ < \phi < 20^\circ, \quad -20^\circ < \theta < 20^\circ, \quad -20^\circ < \psi < 20^\circ \tag{3.59}$$

Therefore, in the selection of Q and R matrices, satisfying constraints on Euler angles defined in Equation (3.59) is also considered. However, it should be also noted that, according to the results obtained in Chapter 4, LQT controller works properly for Euler angles up to $\mp 60^\circ$ for short periods of time (4-5 seconds). Therefore, it can be concluded that, robustness level of LQT controller could compensate the errors between linearized and true nonlinear dynamics [74].

3.2.3 Optimal Control Algorithm

To solve the optimal control problem, Equation set (3.60) defined in [1] is used.

$$\left.\begin{aligned} P(k) &= A_d^T P(k+1)[I + EP(k+1)]^{-1} A_d + V \\ V &= C_d^T Q C_d \\ E &= B_d R^{-1} B_d^T \\ g(k) &= [A_d^T - A_d^T P(k+1)[I + EP(k+1)]^{-1} E] \\ & \quad g(k+1) + C^T Q r(k) \\ X^*(k+1) &= [A_d - B_d L(k)] X^*(k) + \\ & \quad B_d L_g(k) g(k+1) \\ L(k) &= [R + B_d^T P(k+1) B_d]^{-1} B_d P(k+1) A_d \\ L_g(k) &= [R + B_d^T P(k+1) B_d]^{-1} B_d^T \\ U^*(k) &= -L(k) X^*(k) + L_g(k) g(k+1) \end{aligned}\right\} \tag{3.60}$$

As stated earlier, Equation set (3.60) is in discrete-time and our aim is finding time-varying optimal control gains L, L_g and g offline. By using the boundary conditions defined in Equation (3.61), Equation set (3.60) can be solved by integrating backwards in time.

$$\left.\begin{aligned} P(k_f) &= C_d^T F C_d \\ g(k_f) &= C_d^T F \; r(k_f) \end{aligned}\right\} \tag{3.61}$$

Then, by generating and running a MATLAB code, time-varying optimal control gains L, L_g, g and optimal control input U^* can be obtained at each time step k. The size of L and L_g matrices are 4×12 and size of g matrix is 12×1 at each time step. On the other hand, as defined in Equation set (3.55), optimal control input U^* is 4×1 vector at each time step since there are four control inputs.

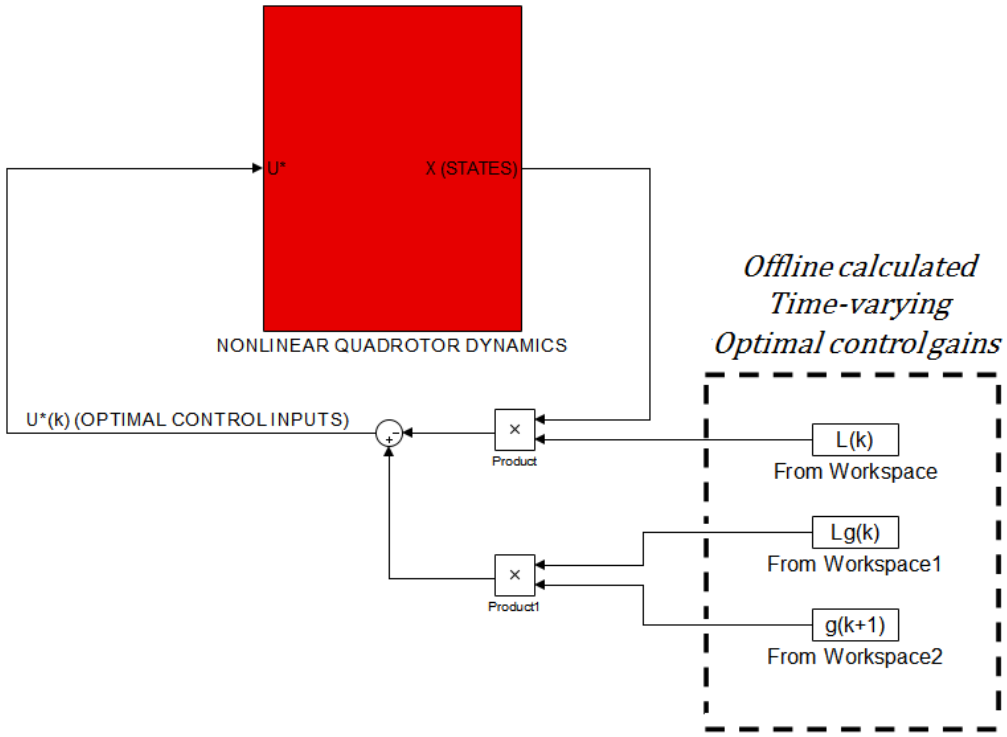


Figure 3.3: The LQT control system as a state feedback control with time-varying and offline calculated optimal control gains L, L_g and g [74].

It is important to note that $r(k)$ represents the information about the desired trajectory to be followed. Desired trajectories are generated in MATLAB at each time step k before solving Equation set (3.60). In Equation set (3.60), it can be seen that only optimal control gain g includes the term $r(k)$. The results obtained in Section 4.5 also showed that, optimal control gain g varies in proportion to the desired trajectory to be followed and therefore it is time-varying. On the other hand, according to the results of Section 4.5 , optimal control gains L and L_g are not directly related to the desired trajectory and they can be approximately considered as time-invariant(fixed) optimal control gains. Then, one can conclude that tracking desired trajectory is actually achieved by using time-varying optimal control gain g while calculating optimal control input U^* in the last equation of Equation set (3.60).

It should be also reminded that, desired trajectory information is given to the LQT algorithm offline. Therefore, it is stated that, optimal control gains of LQT algorithm are calculated offline.

Figure 3.3 shows the basic simulink model of LQT controller [74]. As can be seen

in Figure 3.3, time-varying control gains L, L_g and g that are optimized to track desired trajectory are found by using the MATLAB code and sent into the Simulink model by using "From Workspace" blocks. However, simulation step size is variable although optimal control gains are found by using a fixed step size of 0.01 seconds. This problem is solved by using interpolation and extrapolation to find the values of L, L_g and g at time steps which are not exact multiple of 0.01, for example at time steps 0.0116 or 0.0282 [74].

It is also important to remind that, as can be seen in Figure 3.3, desired trajectory information is not included into the LQT controller since offline calculated control gains of LQT are already optimized to track desired trajectory and possess desired trajectory information.

Finally, LQT control system is obtained the results of the LQT controller and comparison with fixed-gain LQR controller are presented in Chapter 4.

3.3 Linear Quadratic Regulator(LQR) Control

As stated earlier, LQT controller with time-varying control gains used in this thesis will be compared to fixed-gain LQR controller in Chapter 4. The detailed theory of fixed gain LQR control is not given in this thesis since it is a widely known optimal control technique. On the other hand, it is noted that, fixed gain matrix of LQR controller which is defined as K is obtained by using the "lqr" function of MATLAB. "lqr" function tries to find the value of optimal control gain K such that the state feedback law $u = -KX$ minimizes the cost function defined in Equation (3.62) [75].

$$J_u = \int_0^\infty (X^T Q X + u^T R u) dt \tag{3.62}$$

which subject to the state space model of $\dot{X} = AX + Bu, \quad Y = CX.$

In Equation (3.62), Q and R matrices are equal to the ones used in LQT controller to make reasonable comparisons between LQT and LQR controllers. The values of Q and R matrices can be seen in Equation (3.58). Moreover, the state space model used in LQR controller, in other words A, B and C matrices, are also the same as the one

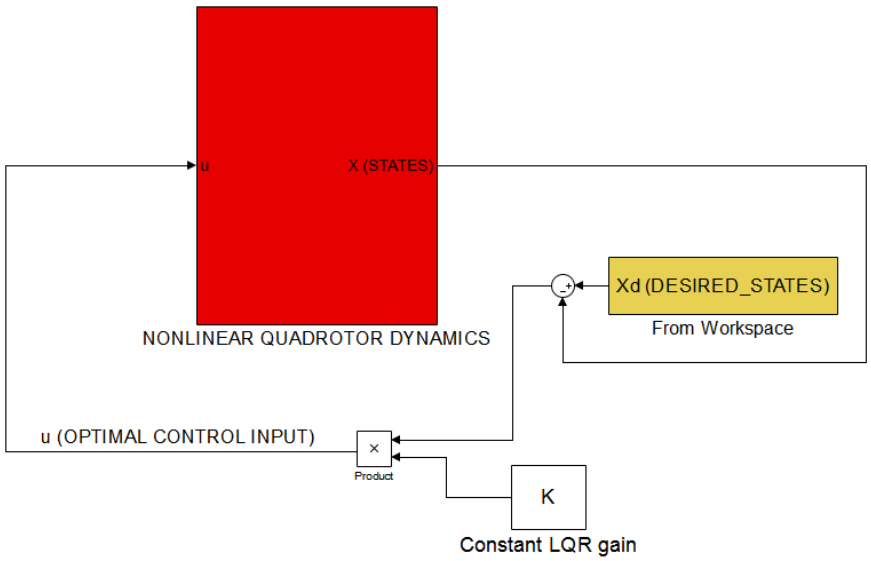


Figure 3.4: The LQR control system as a state feedback control with fixed(time-invariant) optimal control gain K.

used in LQT controller which are defined in Equation (3.55).

To conclude, to obtain reasonable comparison results between LQT and fixed-gain LQR controllers in Chapter 4, same Q and R matrices and state space models are used in both controllers.

Figure 3.4 shows the basic Simulink model of fixed-gain LQR controller. As can be seen in Figure 3.4, unlike LQT controller illustrated in Figure 3.3, optimal control gain K is a 4×12 constant matrix and desired trajectory to be followed is given into the LQR controller offline. Yellow block in Figure 3.4 represents the desired states(X_d), in other words the desired trajectory to be followed. Similar to the optimal control gains of LQT(L, L_g, g) controller which are calculated offline, desired trajectory information have to be given to the fixed-gain LQR controller offline since optimal control gain of LQR(K) is not optimized to follow desired trajectory specifically.

LQR controller is also designed to follow a desired trajectory instead of regulating the system, as can be seen in Figure 3.4. Therefore, the control law u is expressed as following:

$$u = -K(X - X_d) \tag{3.63}$$

In Equation (3.63), X represents the state vector and X_d represents the desired state vector which contains information about the desired trajectory to be followed.

As stated earlier, fixed-gain LQR controller is used to compare the results of LQT control system obtained in this thesis. The advantageous and disadvantageous properties of LQT controller are explained in details in Chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, controllers obtained in Chapter 3 will be tested by simulations and results will be analyzed. In other words, initial validation of the controllers by simulations will be performed.

First, the equations of the nonlinear dynamic model of quadrotor obtained in Chapter 2 will be repeated since this model is used in simulations as the dynamic model of quadrotor. Then, motor model of "AscTech Hummingbird" quadrotor will be defined in Section 4.2. A converter will be defined in Section 4.3 to switch between angular velocity of propellers and control inputs. Then, in Section 4.4, simulation results of backstepping controller will be given and analyzed. In Section 4.5, results of LQT controller will be presented and comparison analysis between LQT and LQR controllers will be performed. In Section 4.6, disturbance rejection characteristics of each control system will be observed. Finally, a detailed comparison analysis among backstepping, LQT and LQR controllers will be made in Section 4.7.

4.1 The Nonlinear Dynamic Model of the Quadrotor

It is important to note that, to obtain more realistic results, simulations are performed by using the nonlinear dynamic model which is not simplified by assuming that perturbations from hover condition are small. The nonlinear dynamic model for translational and rotational motion are obtained in Equations (2.48), (2.52), (2.53) and repeated as following:

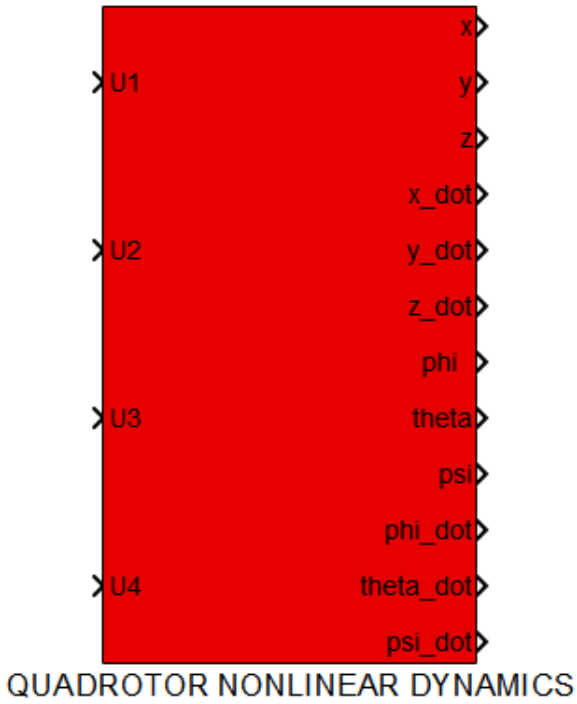


Figure 4.1: Simulink block of the nonlinear dynamic model of quadrotor with its inputs and outputs.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + L_{EB} \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} - (K_t/m) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{4.1}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_y - I_z)qr/I_x \\ (I_z - I_x)pr/I_y \\ (I_x - I_y)pq/I_z \end{bmatrix} + \begin{bmatrix} U_2 d/I_x \\ U_3 d/I_y \\ U_4/I_z \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)/cos(\theta) & cos(\phi)/cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.3}$$

Equation (4.1) represents the nonlinear dynamic model for translational motion and Equations 4.2 and (4.3) represent the nonlinear dynamic model for rotational motion.

Since, trajectory tracking of quadrotor is tried to be achieved, position and orientation

of the quadrotor relative to and expressed in Earth surface are needed. Then, as can be seen in Equation (4.1), translation motion is expressed in Earth fixed reference frame F_E. On the other hand, for rotational motion, Equation (4.2) gives the body angular velocities p, q, r and they are expressed in body fixed reference frame F_B. However, to obtain orientation of the quadrotor, Euler angles ϕ, θ, ψ are needed. To obtain ϕ, θ, ψ transformation Equation (4.3) is used. To obtain more realistic results in simulations, as can be seen in Equation (4.3), transformation matrix between $\dot{\phi}, \dot{\theta}, \dot{\psi}$ and p, q, r is not taken as identity matrix to simplify the equations.

As can be seen in Equations (4.1) and (4.2), the inputs of the nonlinear dynamic model are U_1, U_2, U_3 and U_4 which are defined in Equations (2.3), (2.4), (2.5), (2.6) of Chapter 2.

To sum up, as can be seen in Figure 4.1, the inputs and outputs of the nonlinear dynamic model are (U_1, U_2, U_3, U_4) and $(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}, x, y, z, \dot{x}, \dot{y}, \dot{z})$, respectively. According to Figure 4.2, the outputs of the nonlinear dynamic model goes to the controllers and the inputs of the nonlinear dynamic model comes from the motor model which will be defined in Section 4.2.

4.2 Dynamic Model of the Motor

As stated earlier, AscTech Hummingbird quadrotor is powered by four electric motors [10]. Since control of the quadrotor is obtained by changing the angular velocities of each propellers, the effect of the motor, motor controller and propeller dynamics should be included into the simulations to obtain more realistic results.

For AscTech Hummingbird quadrotor used in this thesis, a simple dynamic model of the motor as a first-order differential equation is found in [5] as following:

$$\frac{dw_i}{dt} = k_g(w_{i,des} - w_i) \tag{4.4}$$

Then, as a motor model in simulations, relationship expressed in Equation (4.4) will be used. In Equation (4.4), k_g represents the time delay of the motors and it is defined as motor gain. The numerical value of k_g is found as $20\ s^{-1}$ for AscTech Hummingbird quadrotor [5]. In Equation (4.4), $w_{i,des}$ represents the desired angular velocity of

the propellers and determined by the control system, on the other hand, w_i represents the angular velocity of the propellers generated by motors.

The relation between the force,torque generated by each propeller and the angular velocty of each propeller is also required to complete motor model and obtain control inputs U_1, U_2, U_3, U_4. For AscTech Hummingbird quadrotor used in this thesis, these relations are expressed in Equations (2.1) and (2.2) of Chapter 2 and repeated as following [66]:

$$F_i = k_n \omega_i^2, \quad [N] \tag{4.5}$$

$$T_i = k_m F_i, \quad [N \cdot m] \tag{4.6}$$

Moreover the relation between the control inputs U_1, U_2, U_3, U_4 and F_i, T_i are defined in Equations (2.3), (2.4), (2.5), (2.6) of Chapter 2. By using Equations (4.5), (4.6), (2.3), (2.4), (2.5), (2.6), control inputs U_1, U_2, U_3, U_4 can be obtained.

To sum up, as can be seen in the blue block of Figure 4.2, the inputs and outputs of the motor model are desired angular velocities($w_{i,des}$) and control inputs (U_1, U_2, U_3, U_4), respectively. According to Figure 4.2, the inputs of the motor model($w_{i,des}$) are determined by the controllers and their values are found by using the angular speed converter which will be expressed in Section 4.3. On the other hand, the outputs of the motor model (U_1, U_2, U_3, U_4) goes to the nonlinear dynamic model of quadrotor as control inputs.

4.3 Angular Speed Converter

As can be seen in Figure 4.2, output of the control systems designed in Chapter 3 are the control inputs U_1, U_2, U_3, U_4. In other words, control systems don't provide desired angular velocities($w_{i,des}$) directly. Therefore, control inputs have to be converted into desired angular velocities to input into the motor model expressed by Equation (4.4). Conversion can be done by using Equations (4.5), (4.6), (2.3), (2.4), (2.5), (2.6).

As can be seen in the white block of Figure 4.2, inputs and outputs of the angular speed converter are U_1, U_2, U_3, U_4 and desired angular velocities $w_{1,des}$, $w_{2,des}$,

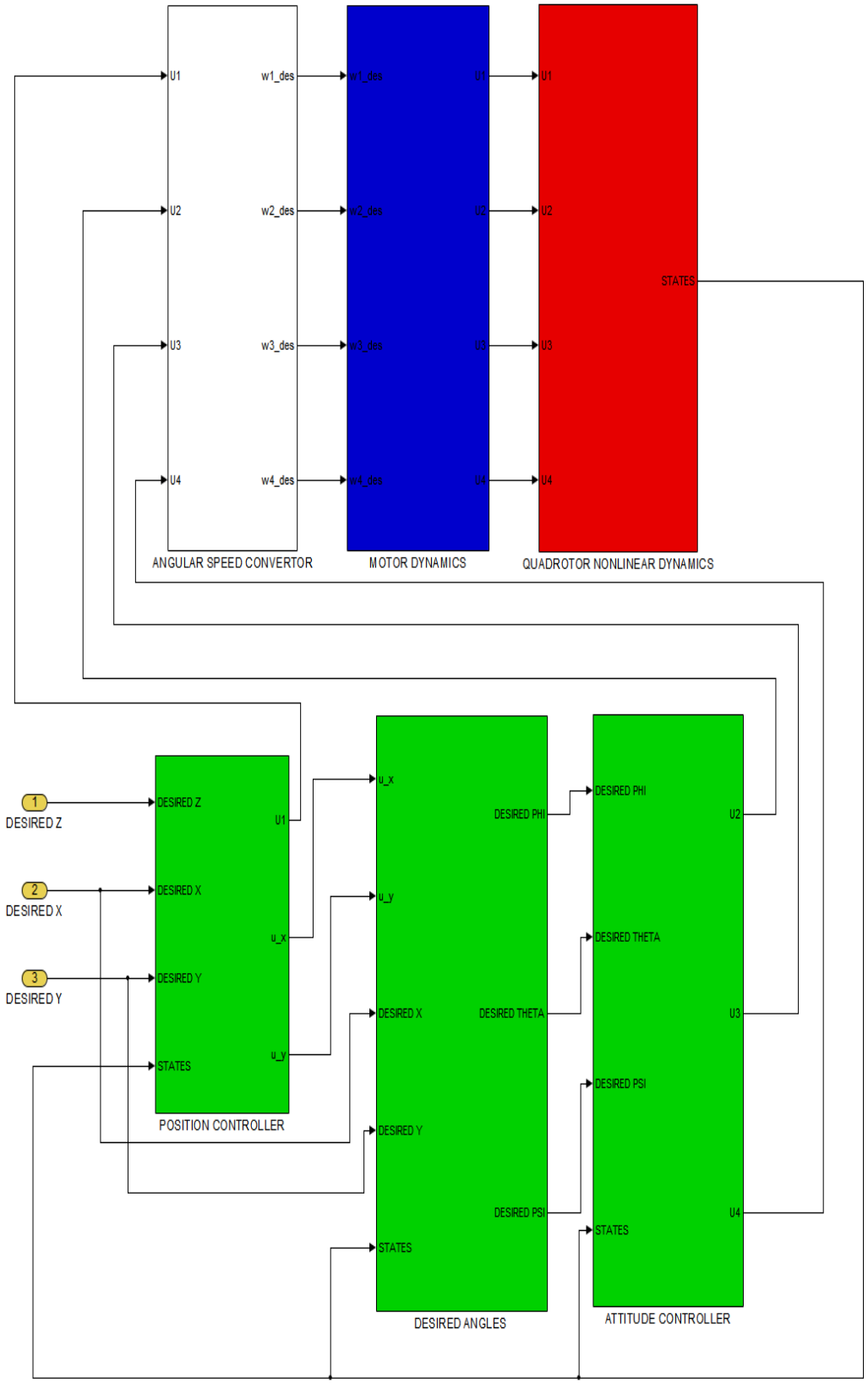


Figure 4.2: Simulink model of the backstepping controller.

$w_{3,des}$, $w_{4,des}$, respectively. The outputs of the angular speed converter go to the motor model(dynamics) and inputs of the angular speed converter come from controllers, as can be seen in Figure 4.2.

4.4 Simulation Results of Backstepping Controller

In this section, simulation results of the backstepping controller obtained in Section 3.1 will be expressed and analyzed. As defined in Section 3.1, backstepping controller consists of three parts which are attitude control, position control and obtaining desired angles. Figure 4.2 shows the simulink model of the backstepping control system. Green blocks represent the each part of the backstepping controller and yellow blocks represent the desired trajectory generated in Equation set 4.7 which is given to the system as inputs. Red, blue and white blocks of Figure 4.2 represent the nonlinear dynamic model of the quadrotor, motor dynamics and angular speed convertor blocks, respectively.

4.4.1 Desired Trajectory A

As stated in Section 1.4, the aim is tracking desired trajectories accurately. Therefore, a trajectory have to be generated at first. In this subsection, desired trajectory A will be defined.

Desired trajectory A is generated and used as inputs for the backstepping controller as can be seen in Figure 4.2. In Figure 4.2, yellow blocks represent the desired trajectory, in other words, desired x, y, z coordinates(x_{7d}, x_{9d}, x_{11d}).

Desired trajectory A is defined for 60 seconds and formulated as following:

$$\left.\begin{aligned} T_s &= 0.01 \\ T_{final} &= 60 \\ t_i = 0 : T_s : T_{final} = [0, 0.01, 0.02, ..., T_{final}], \quad for \quad i &= 1, 2, ...6001 \\ x_{7d,i} &= t_i \\ x_{9d,i} &= 2t_i \\ x_{11d,i} &= 1 + sin(t_i/4) \end{aligned}\right\} \tag{4.7}$$

In Equation set 4.7, T_s represents the time interval and it is taken as 0.01 seconds. T_{final} is the final time which is 60 seconds and i represents the time step. $x_{7d,i}, x_{9d,i}$, $x_{11d,i}$ that defines desired trajectory A at corresponding time step i. As can be seen in Equation set 4.7, our desired trajectory A involves sinusoidal motion in z direction (altitude), and linear motions in x and y directions with different speeds.

It is important to note that desired trajectory A obtained in Equation set 4.7 is filtered for each coordinate(x_{7d}, x_{9d}, x_{11d}) by using a second order filter defined as follows:

$$x_{d,filtered} = \frac{\omega_n^2}{\mathbf{s}^2 + 2\zeta\omega_n\mathbf{s} + \omega_n^2} \; x_d \tag{4.8}$$

The second order filter is used to track relatively complex trajectories without blowing up the simulations. Therefore, this problem could be solved by smoothing desired trajectory via filtering. By trial and error, damping factor (ζ) and natural frequency (ω_n) of command filter defined in Equation 4.8 are chosen as 0.7 and 10, respectively.

Therefore we can conclude that, the actual desired trajectory A is filtered by a second order filter and all of the error analysis in Subsection 4.4.2 will be done according to the filtered desired trajectory A obtained in Equation 4.8.

4.4.2 Results of Backstepping Controller

As can be seen in Equation set 4.7, desired trajectory is relatively complex since it involves sinusoidal motion in z direction and linear motion in x and y directions with different translational velocities.

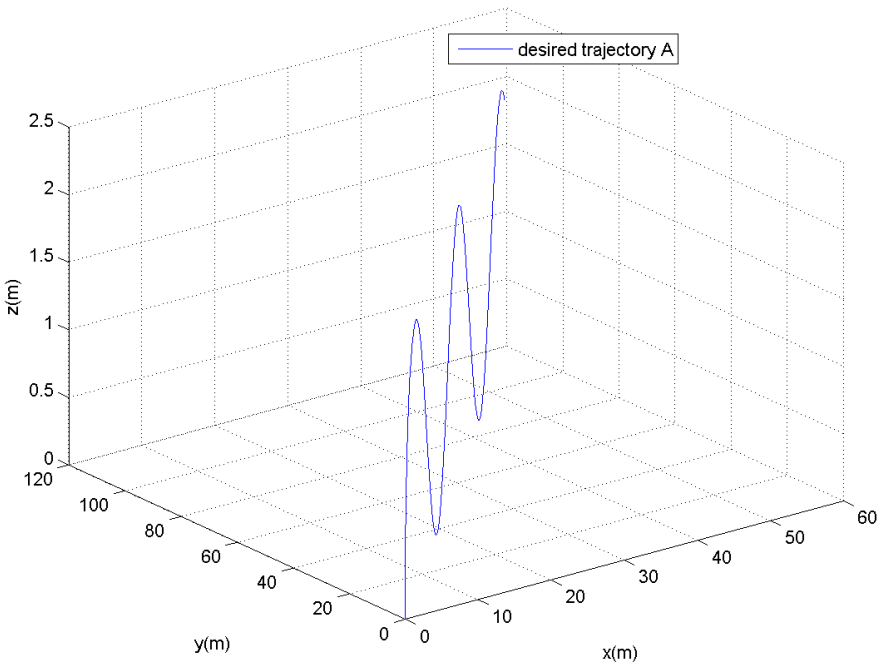


Figure 4.3: 3D plot of desired trajectory A.

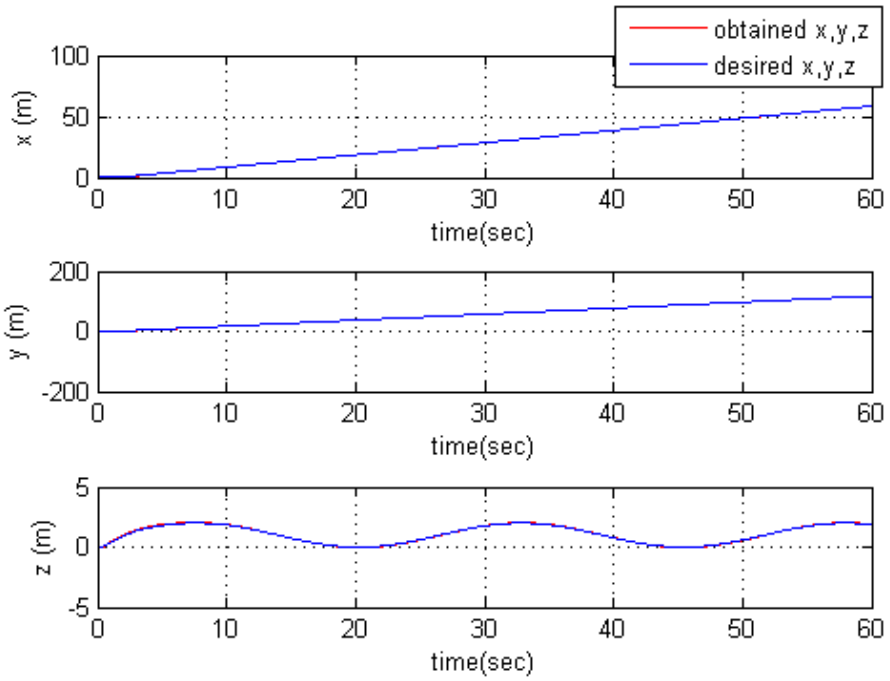


Figure 4.4: x, y, z coordinates of desired trajectory A and x, y, z coordinates obtained by backstepping controller.

It is important to remind that the actual desired trajectory is filtered by using equation 4.8. Therefore, the actual desired trajectory used in simulations as inputs to the position controller(yellow blocks in Figure 4.2 is filtered by using equation 4.8 for each coordinate(x_{7d}, x_{9d}, x_{11d}) and illustrated in Figure 4.3.

Figure 4.4 shows the tracking results of backstepping controller obtained in MATLAB/Simulink. According to Figure 4.4, desired trajectory is accurately tracked. To see the accuracy of the controller more clearly, Figure 4.5 can be examined which shows the results of the error between desired and obtained trajectory. The error functions obtained in Figure 4.5 for x, y, z coordinates are defined as follows:

$$error(t) = x_{d,filtered}(t) - x_{obtained}(t) \tag{4.9}$$

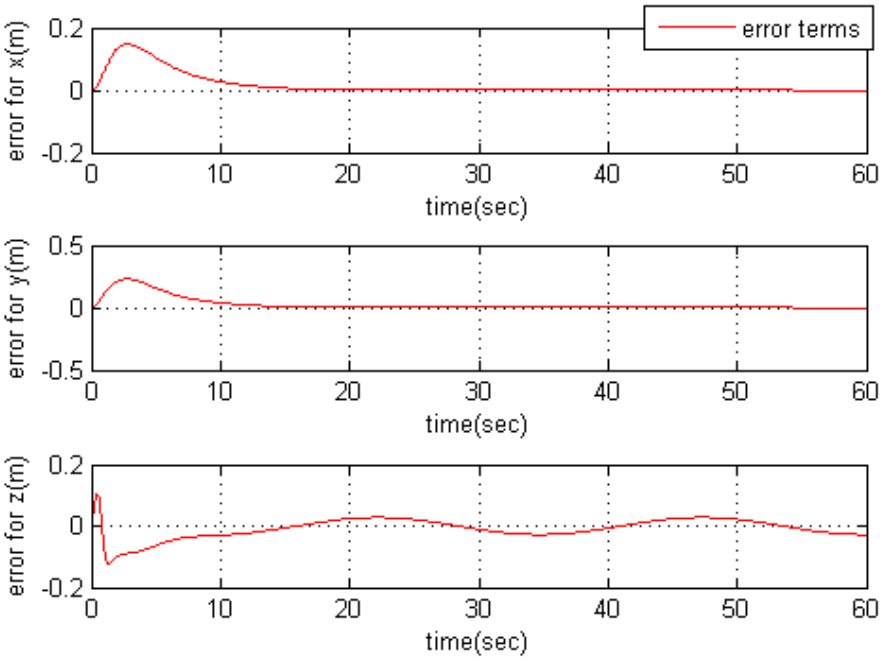


Figure 4.5: Error between x, y, z coordinates of desired trajectory A and x, y, z coordinates obtained by backstepping controller.

According to Figure 4.5, for the motion in x and y directions the value of the error terms at steady state are $7.38 \cdot 10^{-5}$ and $1.22 \cdot 10^{-4}$ meters which correspond to 0.0738 and 0.122 millimeters, respectively. Moreover, the maximum value of the error for the motion in x and y directions are 0.147 and 0.229 meters. On the other hand, as can be seen in Figure 4.5, the error terms for the motion in z direction has more higher

values and can be approximated as a sinusoidal function since desired trajectory is a more complex sinusoidal function. However , maximum values of the error term for z coordinates is 0.1225 meters which is still reasonable. To sum up, the error analysis show that desired trajectory is followed with high accuracy.

As mentioned previously, Euler angles ϕ and θ are responsible for the motion in x and y directions, respectively. The Euler angles obtained are plotted in Figure 4.6. As can be seen in Figure 4.6, quadrotor moves in both x and y directions simultaneously by only using pitch angle(θ) of 2.74° although roll angle(ϕ) is zero. The reason for this kind of motion is explained previously in Figure 3.1. As can be seen in Figure 3.1, quadrotor adjusts its yaw(ψ) angle such that quadrotor's heading and direction of motion in $x-y$ plane are on the same line. Then, yaw angle(ψ) equals to 63.44° during flight as can be seen in the last plot of Figure 4.6. By this way, by only changing pitch angle(θ), quadrotor can translates both in x and y directions and a more simpler flight approach can be obtained.

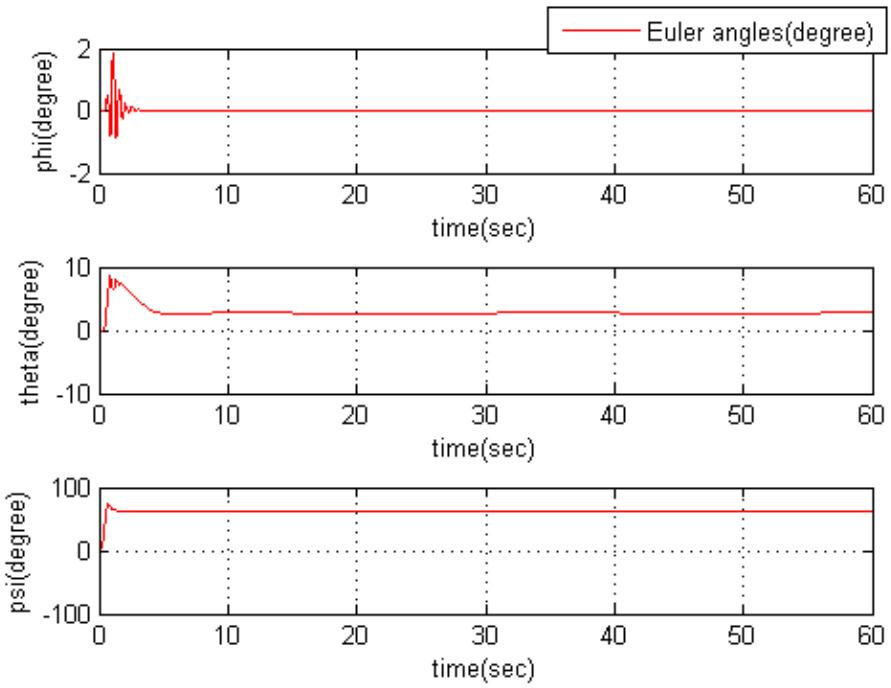


Figure 4.6: Euler angles obtained by backstepping controller while tracking desired trajectory A.

It is also important to analyze control inputs U_1, U_2, U_3, U_4 since these control inputs

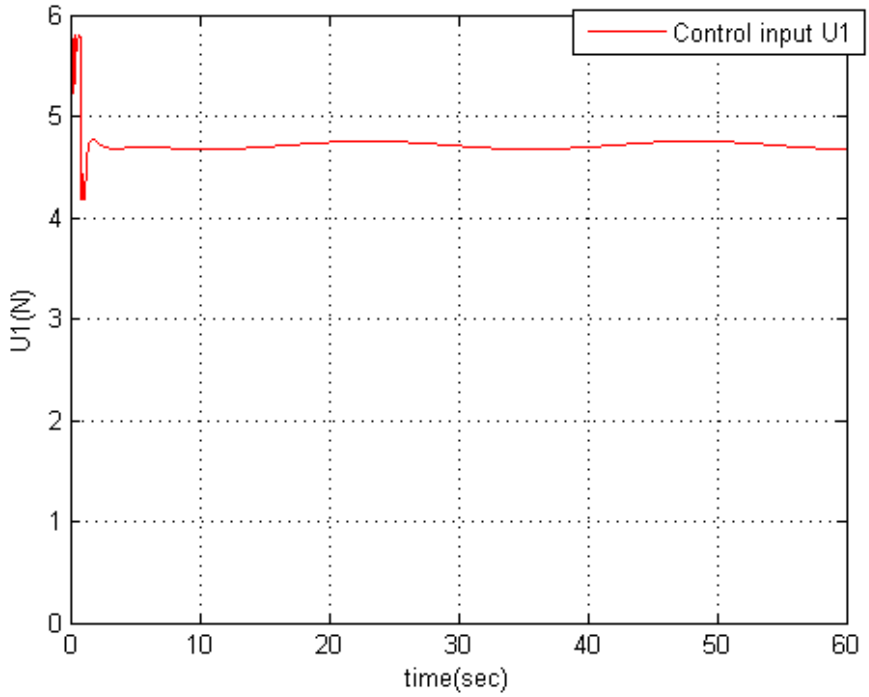


Figure 4.7: Control input U_1 obtained by backstepping controller while tracking desired trajectory A.

are related to force and torque generated by propellers which are constrained by motor dynamics defined in Equations (4.4), (4.5) and (4.6). Figure 4.7, shows the values of U_1 and it can be seen that U_1 changes sinusoidally since it is directly related to motion in z direction which is defined as a sine function in Equation (4.7). Another result is that maximum U_1 value is 5.7986 N which is reasonable according to the limits of motors defined in Equation (4.5). In Figure 4.8, values of other control inputs U_2, U_3, U_4 are also plotted. To sum up, control inputs are found reasonable for desired trajectory A and constraints due to motor dynamics are satisfied.

Figure 4.9 shows the translational velocities of the quadrotor along x, y, z directions. It is observed that translational velocities in x and y directions are 1 m/s and 2 m/s respectively. On the other hand, velocity in z direction changes sinusoidally as expected.

To conclude, backstepping controller could track desired trajectory A accurately while satisfying the limits of the quadrotor.

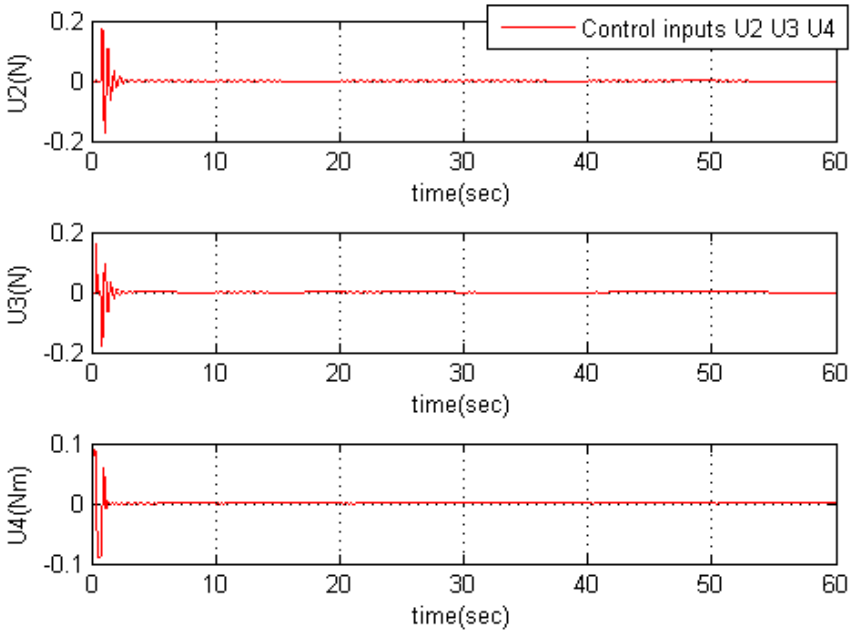


Figure 4.8: Control inputs U_2, U_3, U_4 obtained by backstepping controller while tracking desired trajectory A.

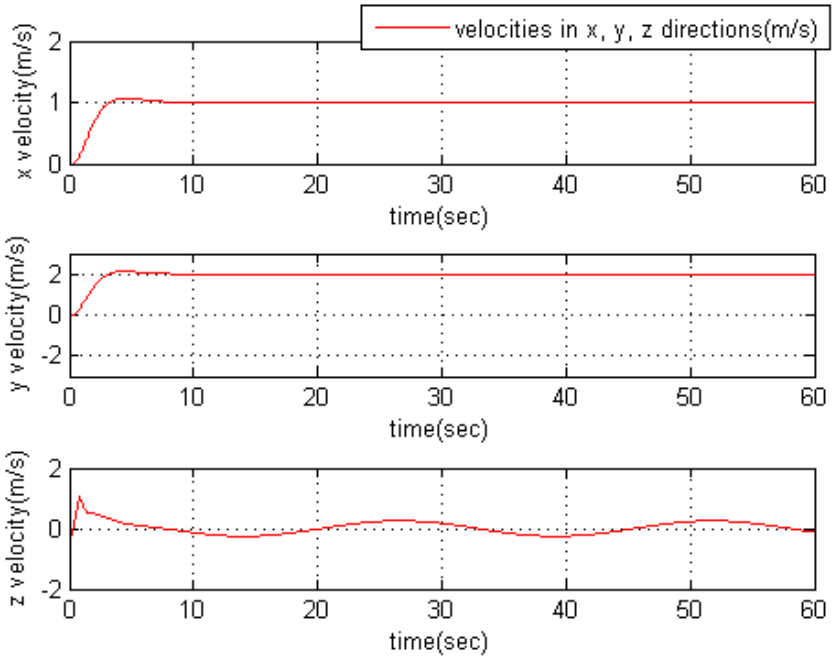


Figure 4.9: Translational velocities obtained by backstepping controller while tracking desired trajectory A.

4.5 Simulation Results of LQT Controller and Comparison with LQR Controller

In this section, simulation results of LQT controller designed in this thesis will be presented and compared with classical fixed gain LQR control. Theory of both LQT and fixed-gain LQR control methods are explained in Sections 3.2 and 3.3, respectively.

As mentioned previously, LQT control approach uses time varying and offline calculated control gains that are optimized to track desired trajectory, unlike fixed gain LQR control. Therefore, it is expected to see some advantageous properties of LQT controller compared to fixed gain LQR controller.

4.5.1 Desired Trajectory B

Before presenting the results of LQT controller, a trajectory to be followed will be generated in a MATLAB code as similar to the desired trajectory A generated in Section 4.4.1. As can be seen in Figure 4.10, desired trajectory B consists of cruise flight between 0-15 seconds, climb between 15-30 seconds, sharp turn at 30 seconds (step input of 5 meters in y coordinate), descent between 30-45 seconds and ends with cruise flight between 45-60 seconds. Desired trajectory is generated in discrete time by using a sample time of 0.01 seconds and final time is 60 seconds.

Desired trajectory B is chosen in this way to observe different behaviors of LQT and LQR controllers for several types of flight/motion.

4.5.2 Results of LQT Controller

In this subsection, simulation results of LQT control system will be presented. As stated earlier, the aim is tracking desired trajectory B defined in Subsection 4.5.1. 3-dimensional plot of desired trajectory B can also be seen Figure 4.10.

As can be seen in Figures 4.11 and 4.12 , tracking performance of LQT controller is very accurate. According to Figure 4.12, steady state error for x coordinate is 0.01 meters or 1 cm. Besides, error for the motion in z direction are 0.0001 for cruise

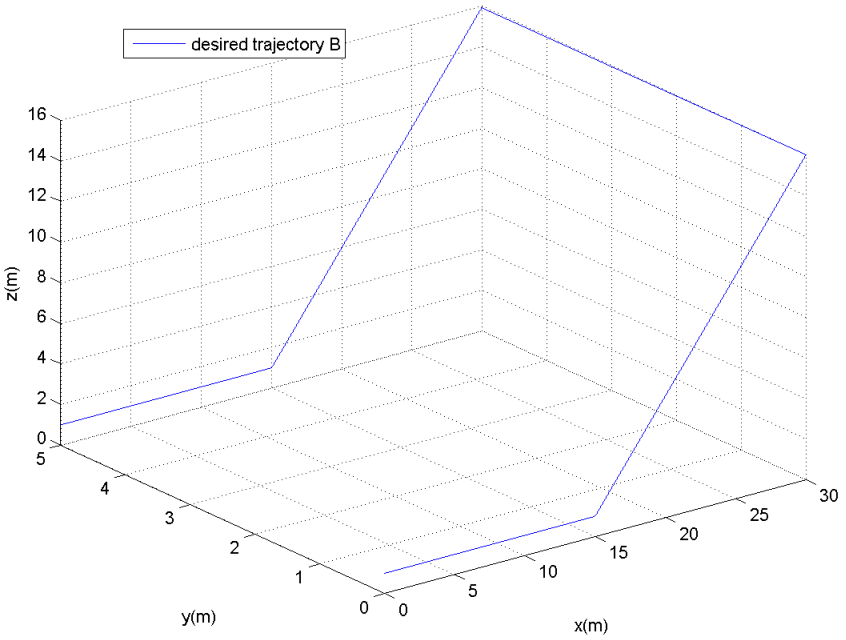


Figure 4.10: 3D plot of desired trajectory B.

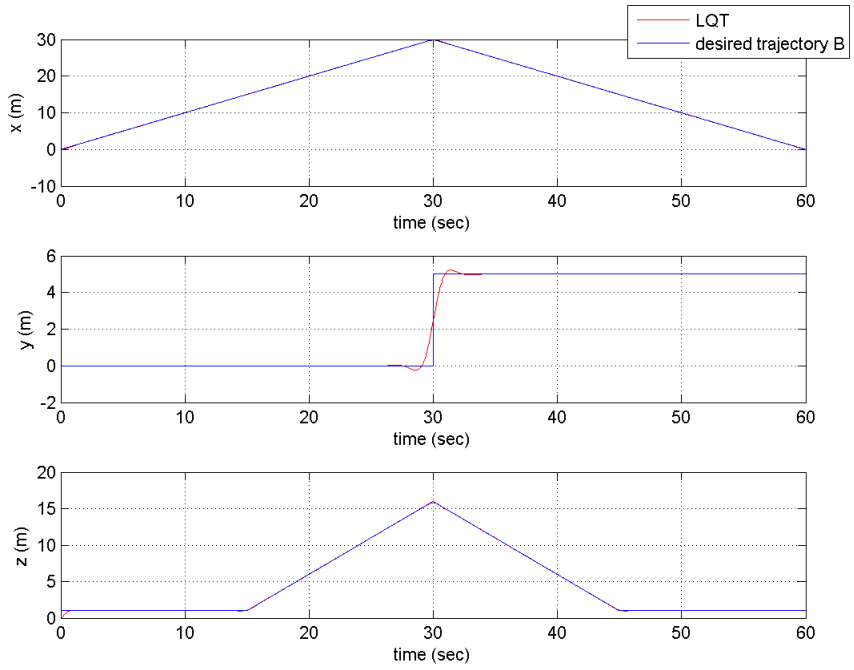


Figure 4.11: x, y, z coordinates of desired trajectory B and x, y, z coordinates obtained by LQT controller.

flights (0-15 and 45-60 seconds) and 0.0101 meters for climb (15-30 seconds) and descent (30-45 seconds).

Figure 4.13 shows Euler angles obtained while tracking desired trajectory B. Since quadrotor translates in x direction and there exist an aerodynamic drag force, pitch angle (θ) remains a constant value during cruise flight,climb and descent. However, roll angle (ϕ) which is related to the motion in y direction equals to zero during the whole flight regime except for 28 -32 seconds when a step input of 5 meters for y coordinate is commanded. Yaw angle(ψ) is also equals to zero except for 28-32 seconds. This result is also expected since a specific yaw motion is not desired at first while defining desired trajectory B. To conclude, according to Figure 4.13, LQT controller gives expected results in terms of Euler angles.

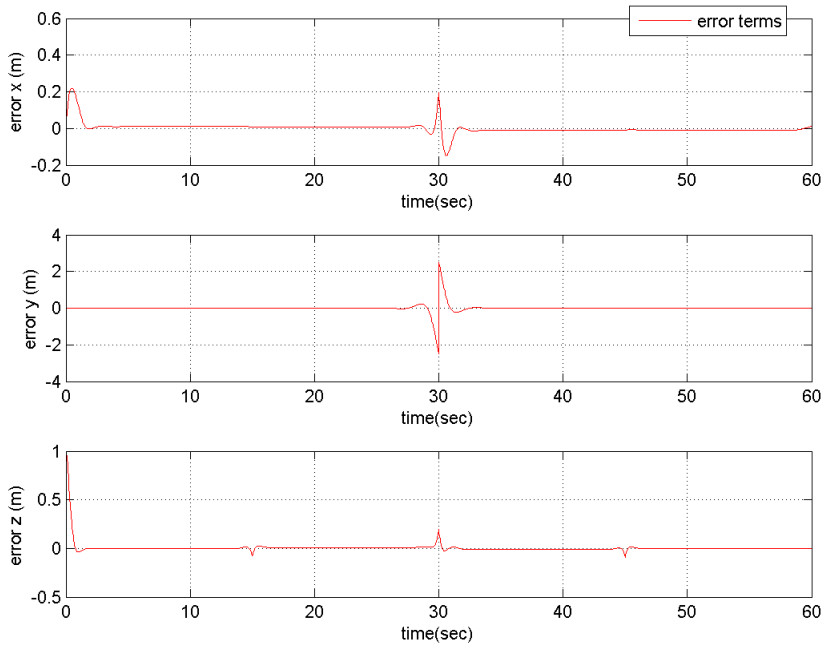


Figure 4.12: Error between x, y, z coordinates of desired trajectory B and x, y, z coordinates obtained by LQT controller.

As can be seen Figure 4.14, translational velocities are also consistent with desired trajectory B. An interesting result is that, although a step input is commanded at 30 seconds for y coordinate which can be seen in Figure 4.11 , LQT controller reacts to this step command before 30 seconds according to the second plot of Figure 4.11. The reason for this behavior is that optimal control gains of LQT controller are optimized

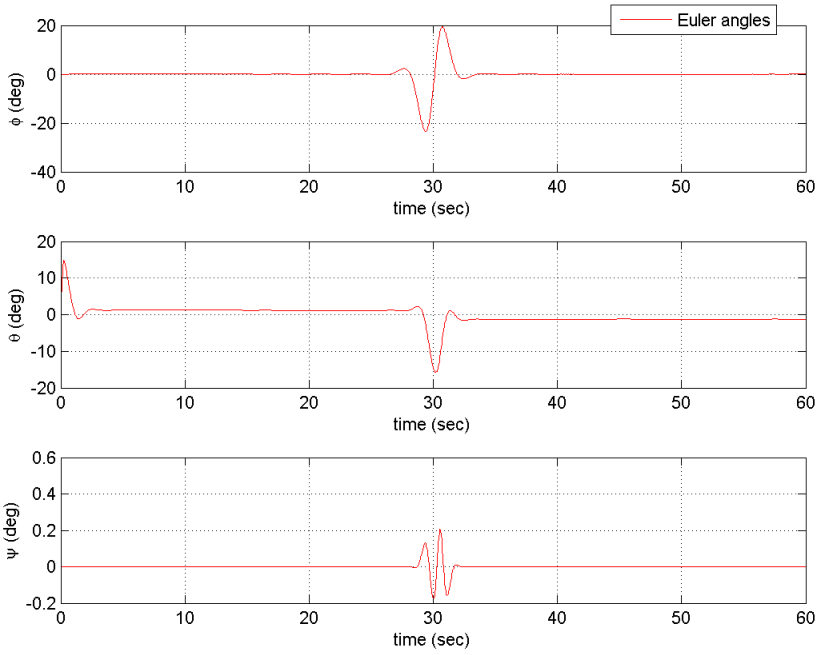


Figure 4.13: Euler angles obtained by LQT controller while tracking desired trajectory B.

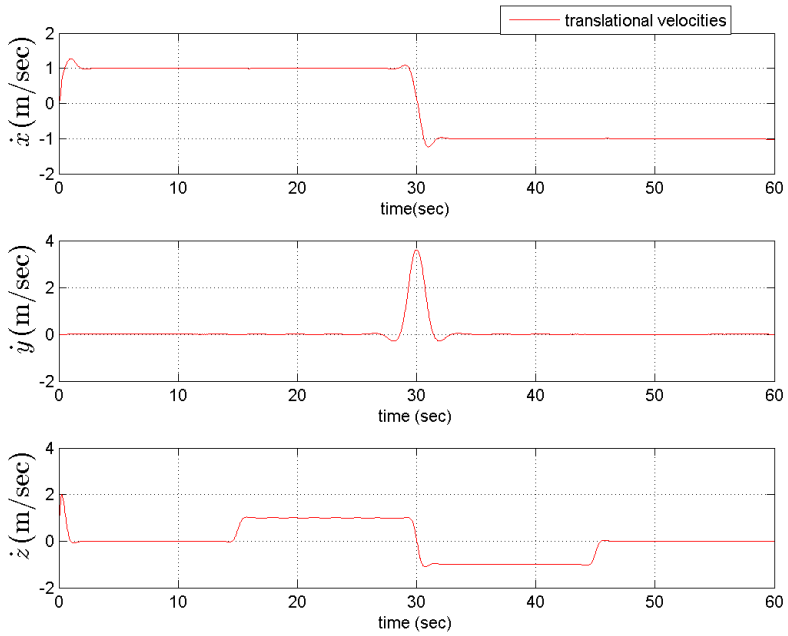


Figure 4.14: Translational velocities obtained by LQT controller while tracking desired trajectory B.

for desired trajectory to be followed and calculated off-line. This kind of behavior can not be obtained in fixed-gain LQR controller as explained in Subsection 4.5.3.

As can be seen Figure 4.15 optimal control input U_1 is also as expected. To remind, U_1 is defined as the total force generated by four rotors. Therefore it is directly related to the motion in z direction. According to Figure 4.15, for climb (15-30 seconds), U_1 is slightly higher than weight of the quadrotor and for descent (30-45 seconds), U_1 is slightly less than weight of the quadrotor. For cruise flights (0-15 and 45-60 seconds) U_1 is equal to the weight of the quadrotor and translational motion in x and y directions occur by changing Euler angles accordingly as can be seen in Figure 4.13.

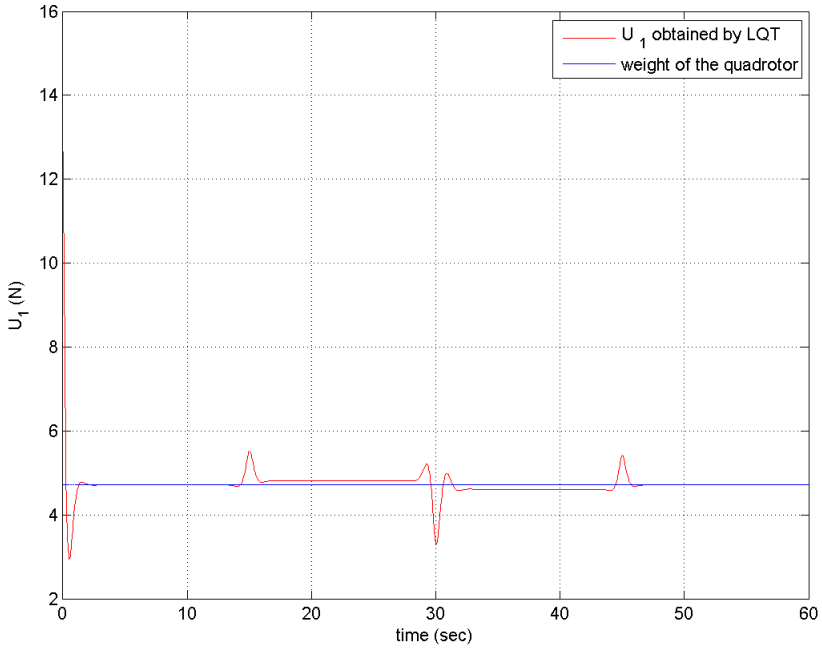


Figure 4.15: Optimal control input U_1 obtained by LQT controller while tracking desired trajectory B.

Finally, angular velocities of each rotor is presented in Figure 4.16 and results show that maximum values of angular velocity reach to 7163 rpm which is in the limits of "AscTech Hummingbird" quadrotor used in this thesis.

To conclude, according to the simulation results, LQT controller gives accurate tracking performance for desired trajectory B which can be considered as a relatively complex trajectory.

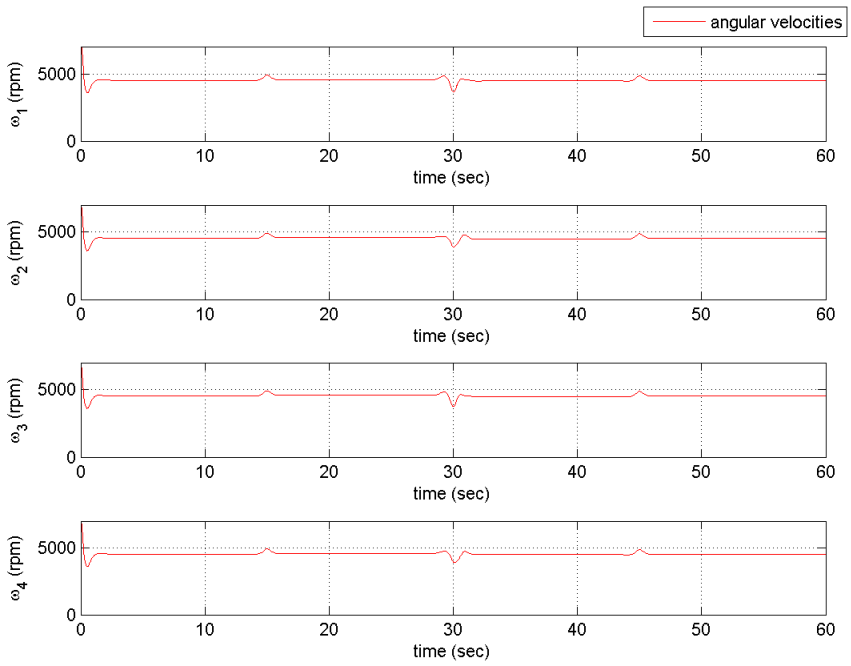


Figure 4.16: Angular velocities ($\omega_1, \omega_2, \omega_3, \omega_4$) obtained by LQT controller while tracking desired trajectory B.

4.5.3 Comparison Between LQT and LQR Controllers

In this subsection, simulation results of LQT controller will be compared to fixed gain LQR controller. As mentioned earlier, LQT controller used in this thesis is a more unique approach since it uses time varying and offline calculated control gains which are optimized to track desired trajectory. Therefore, a comparison between fixed gain LQR control that is widely used in literature would be meaningful to see the advantageous and disadvantageous characteristics of LQT control method used in thesis.

Before presenting comparison results, reminding theoretical differences of LQT and LQR controllers could be beneficial. As stated in Section 3.2, desired trajectory is given to the LQT algorithm defined in Equation set 3.60 and optimal control gains of LQT controller are found offline. Then, these time-varying optimal control gains are used as a state feedback controller as can be seen in Figure 3.3. Therefore, for LQT controller, desired trajectory is not given to the system as inputs since control gains are already optimized to follow desired trajectory and possess information about

desired trajectory. However, as can be seen in Figure 3.4, for fixed gain LQR control, desired trajectory have to be given to the system as inputs since constant LQR gain K is not optimized to follow desired trajectory specifically. Therefore, in fixed gain LQR controller, desired trajectory is given to the controller offline.

To conclude, the main difference between LQT and fixed gain LQR controllers is that LQT controller uses time-varying and offline calculated control gains that are optimized to follow a specific trajectory, whereas, fixed gain LQR controller uses a fixed gain of K which is not optimized to follow a specific trajectory.

It is also noted that, to make reasonable comparisons, same dynamic model and cost functions are used to find the optimal control gains of LQT and LQR controllers. Moreover, LQT and LQR controllers are simulated by using the same nonlinear dynamic model as a plant. Basic simulation models of LQT and LQR control systems can be seen in Figures 3.3 and 3.4 respectively.

4.5.3.1 Comparison of Energy Consumption

First, energy consumption rates of each controller are analyzed. To find energy consumption rates the following equation can be used:

$$P_i = T_i \omega_i \quad for \quad i = 1, 2, 3, 4 \tag{4.10}$$

In Equation (3.40), P_i represents power consumption of i_{th} motor and T_i and ω_i represent torque and angular velocity of i_{th} rotor. It is noted that, ω_i in Equation (4.10) is in the units of rad/sec such that P_i is in the units of $joule/sec$.

By using Equation (4.10), power consumption values at each time instant are calculated for both LQT and LQR controllers and illustrated in Figure 4.17. According to Figure 4.17, especially between 28-34 seconds, LQT controller consumes less energy than LQR controller. To calculate total energy consumed by four motors between 28-34 seconds, power consumption of each motor found by using Equation (4.10) are integrated and summed for both controllers as following:

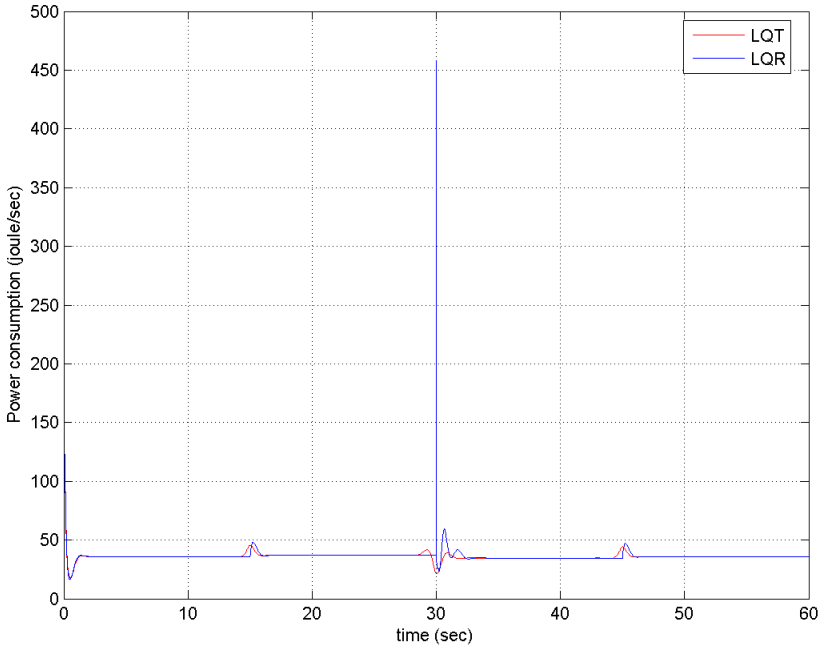


Figure 4.17: Total power consumed by four motors of quadrotor for LQT and LQR controllers at each time instant while tracking desired trajectory B.

$$E_{lqt} = \sum_{i=1}^{4} \int_{28}^{34} P_{lqt,i}\, dt = 210.2405 \quad [joule] \tag{4.11}$$

$$E_{lqr} = \sum_{i=1}^{4} \int_{28}^{34} P_{lqr,i}\, dt = 227.7995 \quad [joule] \tag{4.12}$$

According to the values obtained in Equations 4.11 and 4.12, LQT controller consumes $\mathbf{8.3519}$ $\%$ less energy between 28-34 seconds when a sharp maneuver occurs. If a trajectory involves more than one sharp maneuvers, the energy efficiency of LQT controller compared to LQR controller would be more dominant and significant.

The reason for this behavior can be seen more clearly in Figure 4.18. As can be seen in Figure 4.18, a step input of 5 meters is commanded to the system at 30 seconds for y coordinate and LQT controller can respond to this command before 30 seconds unlike LQR controller. Then, LQT controller could track step command by consuming less energy compared to LQR controller. Figure 4.20 also shows angular velocities of each rotors and it is observed that angular velocities obtained by LQR

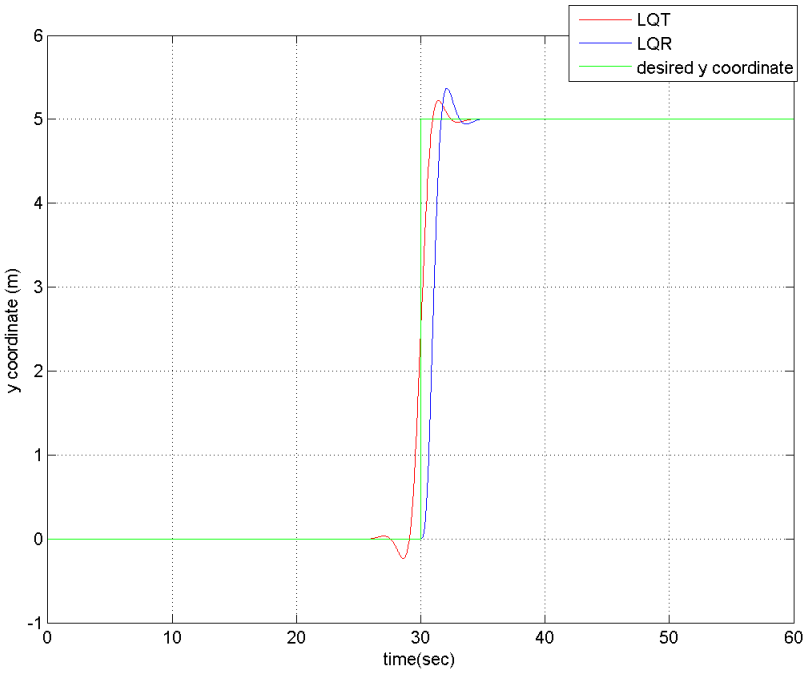


Figure 4.18: Desired y coordinate and y coordinates obtained by LQT and LQR controllers while tracking desired trajectory B.

controller peaks to 13283 rpm at 30 seconds unlike LQT controller.

Then, it can be concluded that, LQT controller is advantageous since it consumes less energy than LQR controller while tracking trajectories that involve sharp maneuvers like a step input. The main reason of this behavior can be explained as follows: time-variant control gains of LQT controller are optimized and calculated off-line to follow a specific trajectory, while LQR controller uses a fixed optimal control gain to follow any kind of trajectory. At this point, it can be also concluded that, LQT controller is disadvantageous since optimal control gains should be calculated off-line for every specific trajectory. However, it should be also stated that, desired trajectory have to be given to the LQR controller as inputs for each time step as can be seen in Figure 3.4 since the nature of LQR control is regulating the system instead of tracking a desired trajectory.

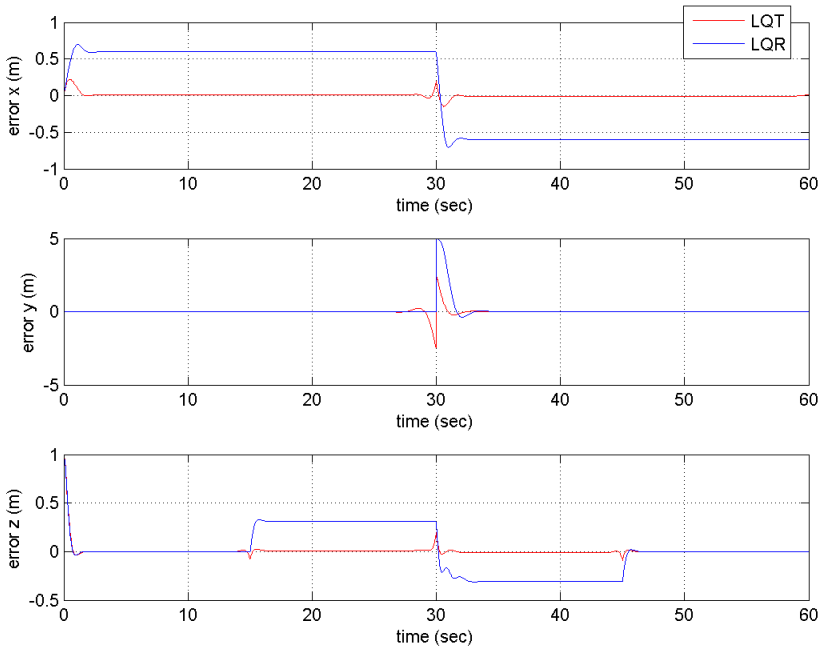


Figure 4.19: Error between x, y, z coordinates of desired trajectory B and x, y, z coordinates obtained by LQT and LQR controllers while tracking desired trajectory B.

4.5.3.2 Comparison of Error Terms

Second comparison analysis is made for the error between desired trajectory and trajectory obtained by LQT and LQR controllers. Error terms z_7, z_9, z_{11} for x, y, z coordinates are previously defined in Equations (3.41), (3.43) and (3.26), respectively. Figure 4.19 shows the values of error terms z_7, z_9, z_{11} for both LQT and LQR controllers.

According to Figure 4.19, error terms of LQT and LQR controller at steady state are 0.01 and 0.6 meters for x coordinates, respectively. Then, for the motion in x direction, LQT controller has approximately 60 times less steady state error.

As can be seen in Figure 4.19, for the motion in y direction, error terms of LQT controller are approximately two times smaller than LQR controller for the sharp maneuver (step input) commanded at 30 seconds. As can be seen in Figure 4.18, since time variant gains of LQT are optimized for desired trajectory, it could respond to step command before it is applied at 30 seconds. On the other hand, LQR controller

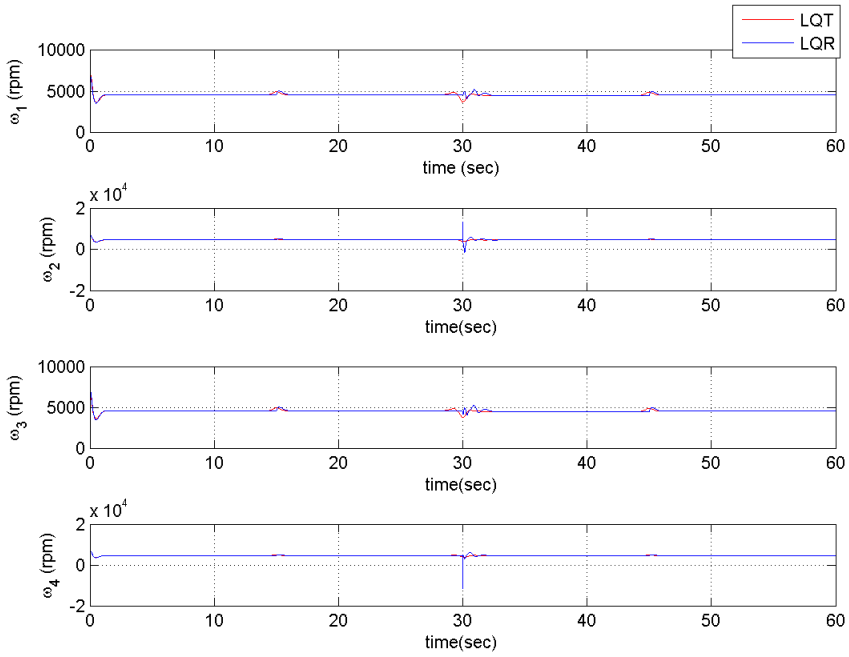


Figure 4.20: Angular velocities ($\omega_1, \omega_2, \omega_3, \omega_4$) obtained by LQT and LQR controllers while tracking desired trajectory B.

uses a fixed optimal control gain K and couldn't respond to the step command before it is applied. For this reason LQT controller has smaller tracking errors compared to LQR controller for the sharp maneuver commanded at 30 seconds.

For the motion in z direction, LQT controller has 0.0101 meters error during climb (15-30 seconds) and descent (45-60 seconds), while error values of LQR controller are 0.3103 meters as can be seen in Figure 4.19. ,

To conclude, compared to fixed gain LQR controller, LQT controller track desired trajectory more accurately since it uses time-variant control gains optimized to follow desired trajectory specifically.

4.5.3.3 Saturation of Motors

In this part, the saturation of motors would be considered for both LQT and LQR controllers. As explained in Sections 3.2 and 3.3, both LQT and LQR methods are linear control techniques and optimal control gains of both controllers are obtained by

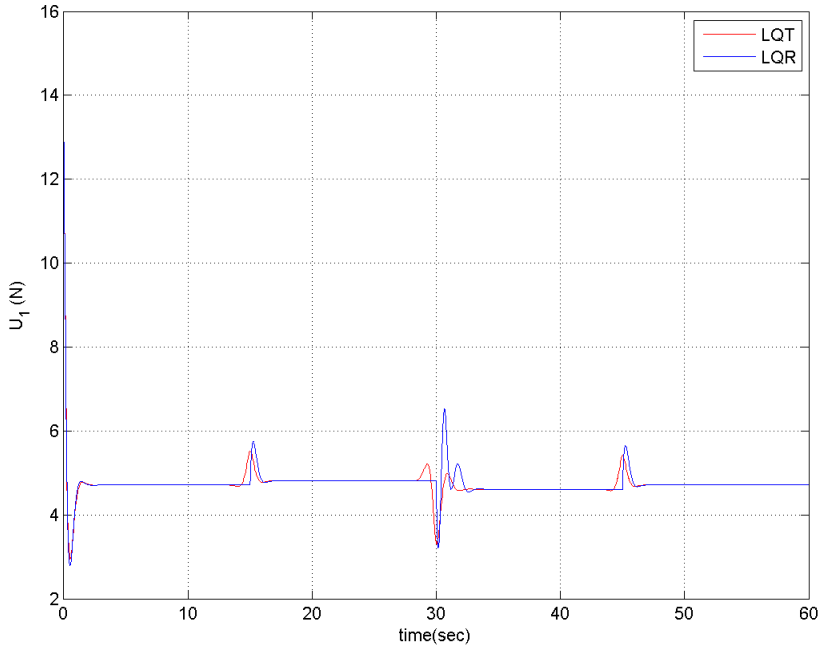


Figure 4.21: Optimal control input U_1 obtained by LQT and LQR controllers while tracking desired trajectory B.

using linearized and simplified dynamic model defined in Equation (3.55). Therefore, adding nonlinearities (ex: saturation) to LQT and LQR control systems are undesirable and could deteriorate the performance of both controllers. For this reason both LQT and LQR controllers are simulated without using any kind of saturations in motor dynamics.

As mentioned previously, Figure 4.20 shows the angular velocities of each rotor while tracking desired trajectory B by using LQT and LQR controllers. According to Figure 4.20, angular velocity of rotors could reach to 13283 rpm for LQR controller, while the maximum angular velocity for LQT controller equals to 7163 rpm. This result also shows that LQR controller is not applicable in reality without saturating motor dynamics since the safe limits of AscTech Hummingbird quadrotor is around 8000 rpm for each rotor [10].

To conclude, LQT controller could be used without using any saturations which is advantageous since both LQT and LQR are linear control methods and addition of any kind of nonlinearity could worsen the performance of both controllers. It is important

Table 4.1: Summary of the comparison between LQT and LQR controllers.

	Advantages	Disadvantages
LQT (time-variant control gains)	-Decreased energy consumption. -More accurate tracking. -Saturation of motors are automatically avoided. -Optimized control gains already contain information about desired trajectory.	-Each trajectory requires off-line calculation of time-variant control gains.
LQR (fixed control gain)	-Fixed control gain K can be used for each trajectory.	-Increased energy consumption. -Less accurate tracking. -Motor dynamics have to be saturated. -Desired trajectory information have to be given as inputs.

to note that, by changing the control weight matrix (R) of LQR controller, limitation on control inputs and angular velocities could be adjusted as desired. However, this would also cause a slower response time and worsen the performance of the controller. Therefore, to make reasonable comparisons, both LQT and LQR controllers are compared by using the same Q and R matrices defined in Equation (3.58).

4.5.3.4 Summary of the comparison between LQT and LQR controllers

According to the comparison results, the advantageous and disadvantageous properties of LQT controller compared to LQR can be summarized as in Table 4.1. It is important to note that energy efficiency of LQT controller becomes more significant for trajectories that involve sharp maneuvers like a step input.

Since advantageous properties of LQT controller are already discussed in Subsection 4.5.3, in this paragraph, drawback of LQT controller would be explained in details. As can be seen in Table 4.1, LQT controller requires to calculate time-variant control gains off-line for each trajectory. Therefore, it could be interpreted as a disadvantage of LQT controller. However, it is also important to note that, LQR controller

requires desired trajectory information as input as can be seen in Figure 3.4, whereas LQT controller doesn't require desired trajectory as input since optimized control gains already possess this information, as can be seen in Figure 3.3. Therefore, off-line calculated control gains of LQT controller could be considered as desired trajectory input of LQR controller. It is also noted that, run time of off-line calculation of LQT control gains and generation of desired trajectory are almost equal.

To conclude, compared to fixed gain LQR control used in literature widely, LQT controller could be advantageous to track a specific trajectory since it uses time-variant control gains optimized to track desired trajectory.

4.6 Disturbance Rejection Properties of Backstepping, LQT and LQR Controllers

In this section, disturbance rejection characteristics of backstepping, LQT and LQR controllers are analyzed and compared to each other. Disturbances are added to the translational quadrotor dynamics obtained in Equation (2.48).

$$\begin{bmatrix} f_{distx} \\ f_{disty} \\ f_{distz} \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + L_{EB}\begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} - (K_t/m)\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \tag{4.13}$$

As can be seen in Equation (4.13), disturbances are symbolized as d_x, d_y, d_z and added to the system as unit forces in the units of $[N/Kg]$. In Equation (4.13), f_{distx}, f_{disty}, f_{distz} represent net unit forces act on x, y, z directions when disturbances are added to the system. To clarify, f_{distx}, f_{disty}, f_{distz} are net unit forces when simulations are run with additional disturbance terms, d_x, d_y, d_z.

Figures 4.22, 4.23 and 4.24 shows the numerical values of pure disturbances d_x, d_y, d_z and net unit force terms f_{distx}, f_{disty}, f_{distz} defined in equation 4.13. f_{distx}, f_{disty}, f_{distz} terms are also plotted to see the strength of disturbances. It is noted that, values of f_{distx}, f_{disty}, f_{distz} are obtained from the nonlinear dynamic model block of LQT controller simulated in MATLAB/Simulink. Since all of the controllers are tested by using the same nonlinear dynamic model block, similar f_{distx}, f_{disty}, f_{distz} values are

obtained in simulations for backstepping, LQT and LQR controllers.

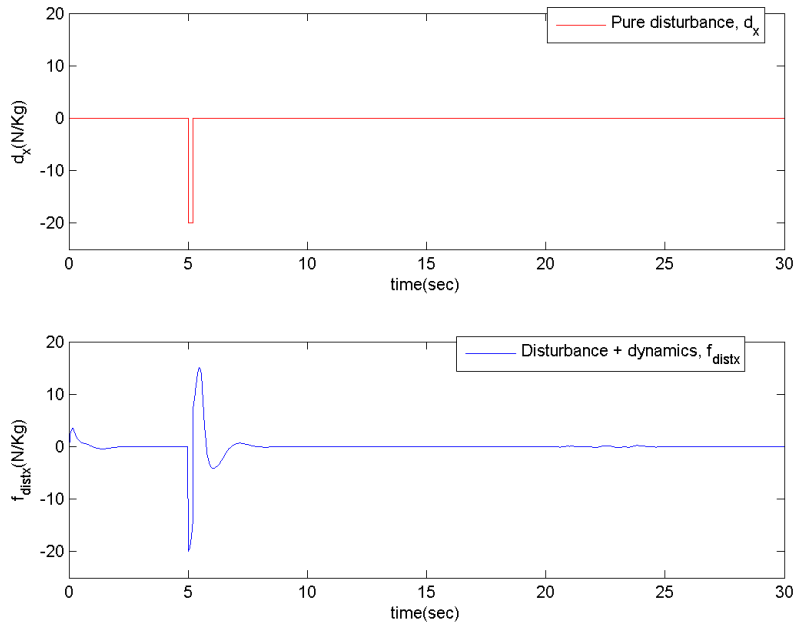


Figure 4.22: Pure disturbance d_x added between 5-5.02 seconds and net unit force f_{distx} obtained by simulations with additional disturbances.

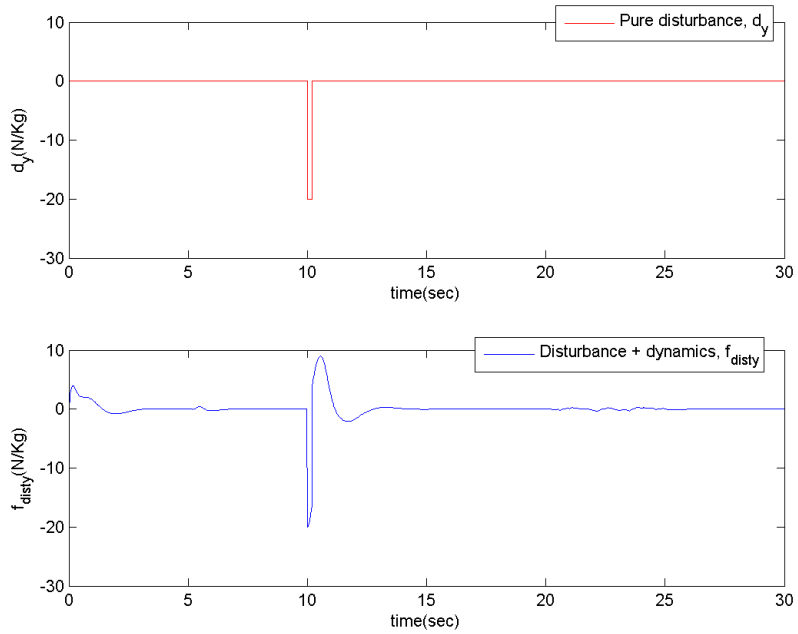


Figure 4.23: Pure disturbance d_y added between 10-10.02 seconds and net unit force f_{disty} obtained by simulations with additional disturbances.

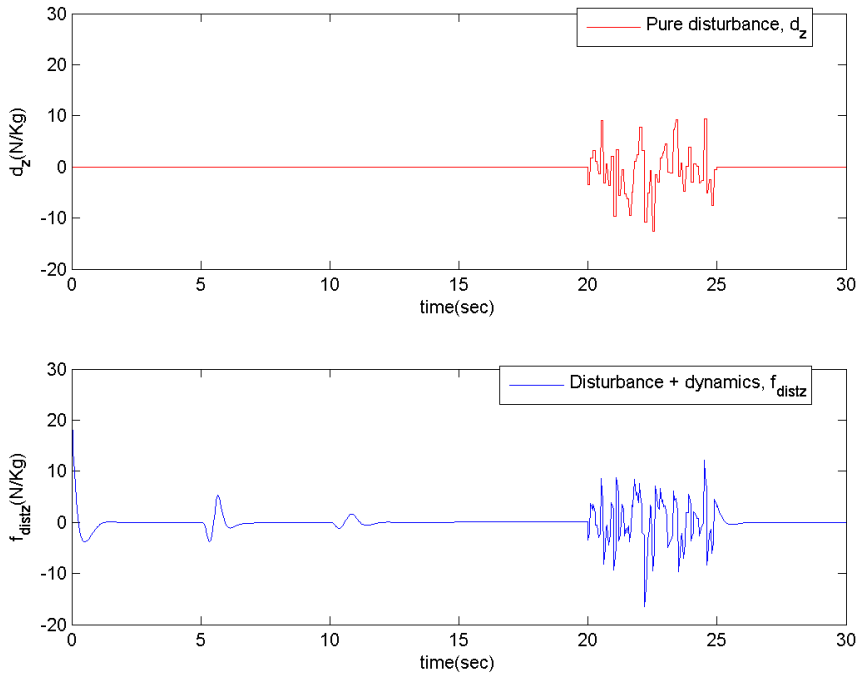


Figure 4.24: Pure disturbance d_z added between 20-25 seconds and net unit force f_{distz} obtained by simulations with additional disturbances.

As can be seen in Figures 4.22 and 4.23, strong impulse disturbances are added to the system in x and y directions at 5 and 10 seconds, respectively. According to Figures 4.22 and 4.23, disturbances d_x and d_y could be considered as strong impulse disturbances since they change the total unit net forces f_{distx},f_{disty} considerably and acts for 0.2 seconds only. On the other hand, d_z which acts on z direction is a Gaussian distributed random disturbance with mean and variance equal to 0 and 1, respectively. d_z acts between 20-25 seconds as can be seen in Figure 4.24.

In the following sections disturbance rejection properties of backstepping, LQT and LQR controllers are analyzed. To make reasonable comparisons, same disturbance models presented in Figures 4.22, 4.23 and 4.24 are added to each controller. In addition, each controller tries to track "desired trajectory A" defined in Subsection 4.4.1. However, simulations are performed for 30 seconds to see the effects of disturbances more clearly.

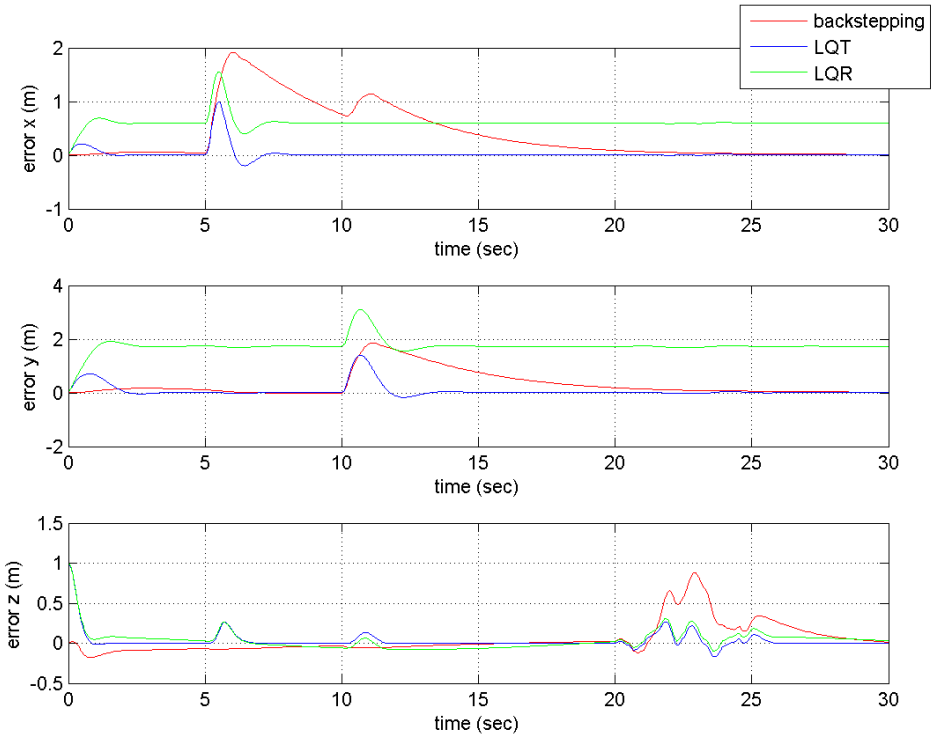


Figure 4.25: Tracking errors of each controller when disturbances are added to the system.

4.6.1 Disturbance Rejection Comparison of Backstepping, LQT and LQR controller

As stated earlier, each control system are simulated in MATLAB/Simulink with same disturbance models added to the nonlinear dynamic model of quadrotor.

Figure 4.25 shows tracking errors for x, y, z coordinates, in other words, the error terms z_7, z_9, z_{11} defined in Equations (3.41), (3.43), (3.26), respectively. To compare the disturbance rejection properties according to Figure 4.25, the change in error terms should be analyzed when disturbances are added to the system at 5,10 and 25 seconds. According to the results plotted in Figure 4.25, backstepping controller could not respond to disturbances as fast as LQT and LQR controllers. On the other hand, LQT and LQR controllers show similar disturbance rejection properties. In Figure 4.25, it is also possible to obverse that, LQT controller has less steady state error compared to LQR as explained in Subsection 4.5.3.2. However, both LQR and

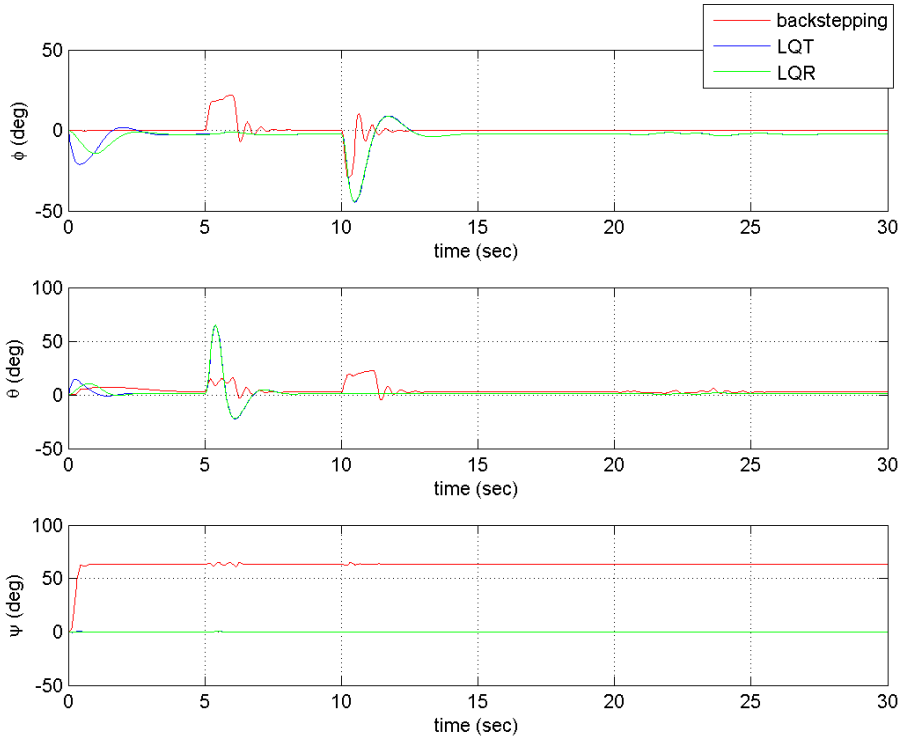


Figure 4.26: Euler angles obtained by each controller when disturbances are added to the system.

LQR controllers show very similar disturbance rejection properties, since the change in error terms is important while analyzing disturbance rejection characteristics.

As mentioned in Section 2.1, translational motion of quadrotor basically occurs by changing Euler angles. Therefore, when an external disturbance is present, quadrotor tries to cancel it out by changing Euler angles properly. Euler angles are also adjusted by changing the control inputs U_1, U_2, U_3, U_4 which are responsible for the desired angular speed (rpm) of each rotors. Therefore, to understand the mechanism of disturbance rejection, Figures 4.26 and 4.27 which show Euler angles and control inputs obtained by controllers when disturbances are present could be analyzed.

As can be seen in Figures 4.26 and 4.27, at 5 and 10 seconds when strong impulse disturbances are added to the system, each controller adjust its control inputs and change Euler angles properly to reject disturbances. By this way, quadrotor try to return to its desired trajectory and error terms shown in Figure 4.25 decreases. On the other hand, for the Gaussian distributed disturbance added between 20-25 seconds in

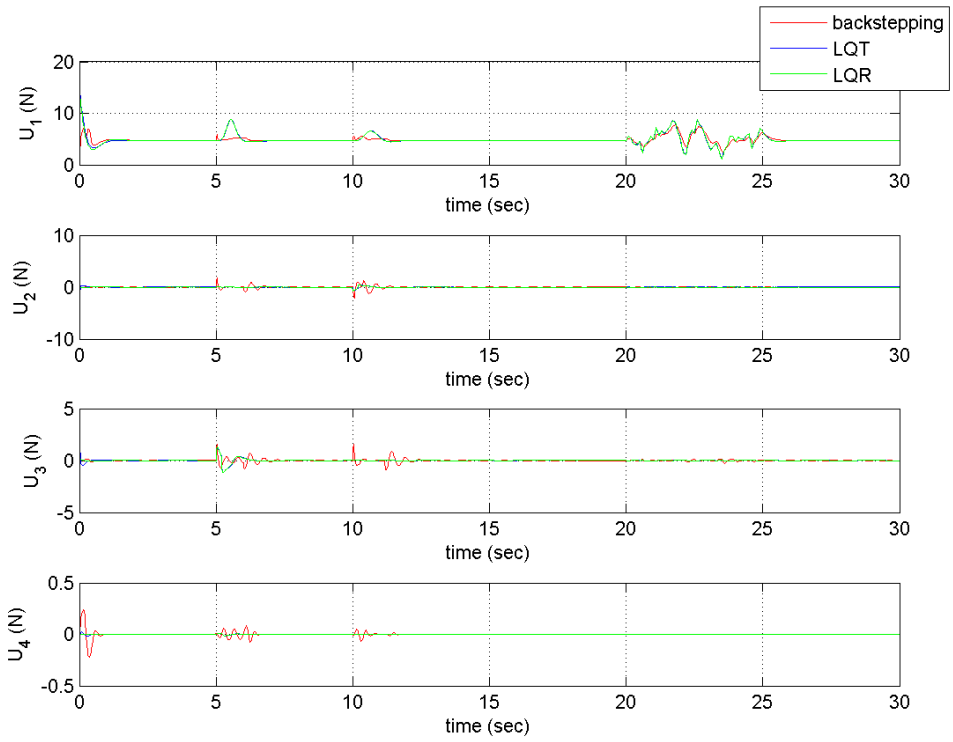


Figure 4.27: Control inputs obtained by each controller when disturbances are added to the system.

z direction, disturbances are rejected by adjusting control input U_1 which is directly responsible for the motion in z direction. First plot of Figure 4.27 shows that between 20-25 seconds each controller adjust U_1 properly to reject the disturbance added in z direction.

In the last plot Figure 4.26 yaw angle(ψ) or heading of the quadrotor is given. According to Figure 4.26, yaw angle is not effected by disturbances since it is not directly related to the motion in x, y, z directions. At this point, a basic difference between the backstepping and LQT,LQR controllers could be observed such that the backstepping controller adjusts its heading according to the direction of motion in $x-y$ plane, unlike LQT,LQR controllers. Therefore, as can be seen in Figure 4.26, yaw angle (ψ) for backstepping controller is 63.5 degree while the values for LQT and LQR controllers are 0 degree, at steady state.

The reason of the slow response time of backstepping controller could also be seen in Figure 4.26. According to Figure 4.26, when disturbances are added to the sys-

tem at 5 and 10 seconds, backstepping controller could not adjust the Euler angles as fast as LQT and LQR controllers. Therefore, LQT and LQR controllers reject disturbances faster than backstepping controller as can be seen in Figure 4.25.

To conclude, it is observed that LQT and LQR controllers have better disturbance rejection characteristics compared to the backstepping controller. It is also seen that both LQT and LQR controllers have almost identical disturbance rejection properties. At this point, relatively poor disturbance rejection characteristics of backstepping controller could be related to the parameter selection of backstepping controller. As explained in Section 3.1, performance of backstepping controller is directly related to some control parameters. Although these parameters are optimized by using "MATLAB Optimization Toolbox" and illustrated in Table 3.2, another parameter set could give better results in terms of disturbance rejection. Therefore, it could be concluded that, finding optimized parameters for different cases is one of the most important drawback of backstepping controller.

4.7 Comparison of Backstepping, LQT and LQR Controllers

In Subsection 4.5.3 a comparison analysis between LQT and LQR controllers is performed to observe the differences between two optimal control approach. On the other hand, in this section, backstepping controller which is a nonlinear control technique is also compared with optimal control techniques LQT and LQR. Therefore, results obtained in this section could give a more complete comparison analysis that include each of the controllers used in this thesis. Before presenting comparison results, desired trajectory to be followed is defined in the following subsection.

4.7.1 Desired Trajectory C

Desired trajectory C is similar to the desired trajectory A defined in Subsection 4.4.1 and defined as in Equation set 4.14.

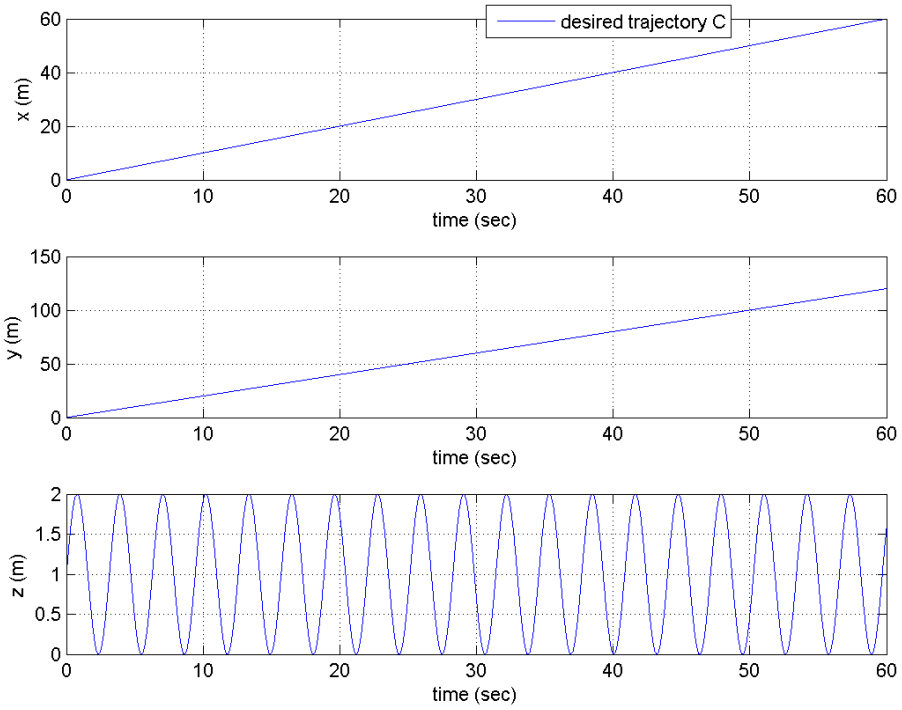


Figure 4.28: x, y, z coordinates of desired trajectory C.

$$\left.\begin{aligned} t_i = 0 : T_s : T_{final} = [0, 0.01, 0.02, ..., T_{final}], \quad for \quad i = 1, 2, ...6001 \\ T_s = 0.01 \\ T_{final} = 60 \\ x_{7d,i} = t_i \\ x_{9d,i} = 2t_i \\ x_{11d,i} = 1 + sin(2t_i) \end{aligned}\right\} \quad (4.14)$$

In Equation (4.14), $x_{7d,i}, x_{9d,i}, x_{11d,i}$ represent the x, y, z coordinates of desired trajectory C at time step i, respectively. Then, as can be seen in Figure 4.28, quadrotor is desired to make linear motion in x and y directions with different velocities and sinusoidal motion in z direction with increased frequency compared to desired trajectory A.

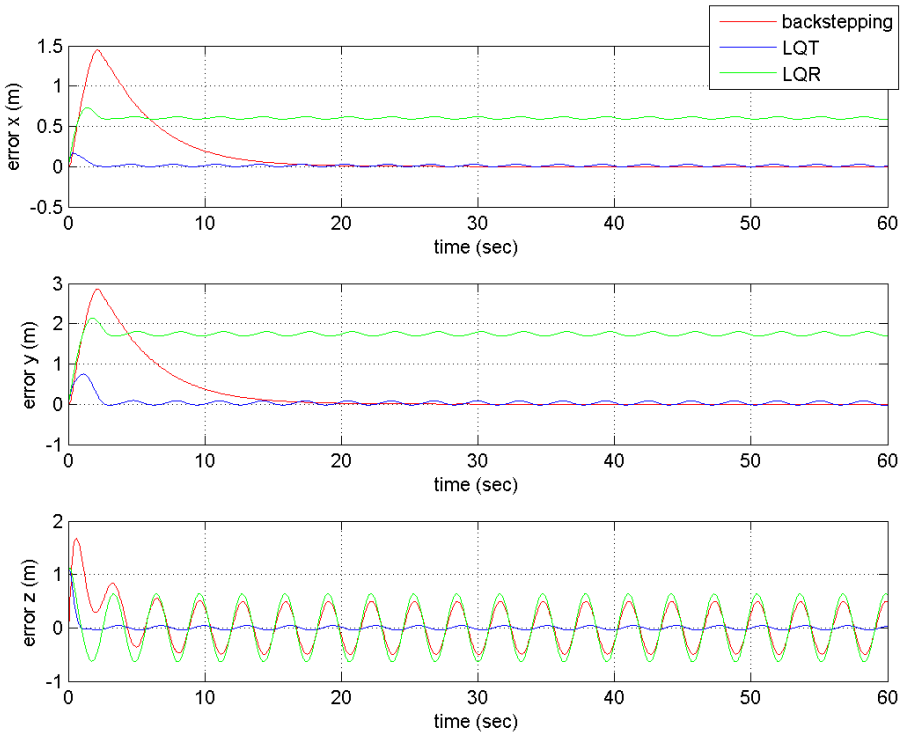


Figure 4.29: Tracking errors of backstepping, LQT and LQR controllers while tracking desired trajectory C.

4.7.2 Comparison Results

First, tracking error terms which represent the error between desired and obtained x, y, z coordinates are compared. As can be seen in Figure 4.29, for the motion in x and y directions, backstepping controller could not respond as fast as LQT controller at first, however, tracking errors of backstepping controller converges to zero at steady state. Steady state errors of LQT controller are also very close to zero (0.01 meters), unlike LQR controller which has considerably high steady state errors (0.6 and 1.8 meters) compared to LQT and backstepping controllers. For the motion in z direction, LQT controller has approximately 10 times smaller error compared to backstepping and LQR controllers.

Table 4.2 shows the exact values of tracking errors for each controller. Then, it can be concluded that LQT controller is more accurate than backstepping and LQR controllers. The reason for this behavior could depend on the usage of time-variant

control gains optimized to track desired trajectory in LQT controller. According to Table 4.2, it could be also concluded that, backstepping controller tracks relatively simple trajectories more accurately such as the linear motion in x and y directions. However, LQT controller is the most accurate one while tracking relatively complex trajectories such as high frequency sinusoidal motion in z direction.

Table 4.2: Steady state tracking errors of backstepping, LQT and LQR controllers while tracking desired trajectory C.

	Error x (m)	Error y (m)	Error z (m)
Bacstepping	0.002	0.005	0.494
LQT	0.032	0.087	0.044
LQR	0.622	1.805	0.638

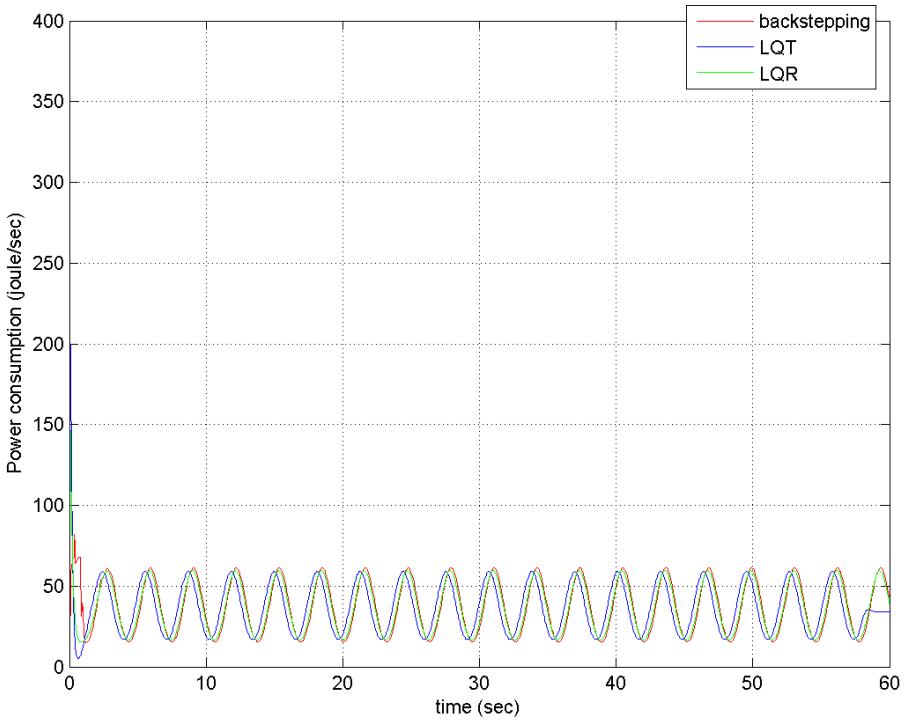


Figure 4.30: Total power consumed by four motors of quadrotor for backstepping, LQT and LQR controllers while tracking desired trajectory C.

Second comparison analysis is related to the energy consumption values. As explained in Subsection 4.5.3.2, LQT controller consume approximately 7.7 % less energy compared to LQR controller while tracking a sharp maneuver such as a step input. However, while tracking desired trajectory C, total energy consumption values

does not vary significantly. Figure 4.30 shows the power consumption values at each time step for each controllers. Total energy consumed while tracking desired trajectory C are calculated by integrating power consumption values over the whole flight as in Equations (4.15), (4.16), (4.17). According to the values obtained in Equations (4.15), (4.16), (4.17), LQT seems to be most efficient controller although the difference is very small. Therefore, it can be concluded that, the energy efficiency of LQT controller is more significant when trajectory involves sharp maneuvers (ex:step input) as explained in Subsection 4.5.3.1

$$E_{lqt} = \sum_{i=1}^{4} \int_{0}^{60} P_{lqt,i}\, dt = 2219 \quad [joule] \tag{4.15}$$

$$E_{lqr} = \sum_{i=1}^{4} \int_{0}^{60} P_{lqr,i}\, dt = 2240 \quad [joule] \tag{4.16}$$

$$E_{back} = \sum_{i=1}^{4} \int_{0}^{60} P_{back,i}\, dt = 2259 \quad [joule] \tag{4.17}$$

Third, Euler angles obtained by each controller, in other words, orientation of the quadrotor during flight is compared. According to Figure 4.31, LQT and LQR controllers have very similar results, however, backstepping controller has its own characteristics. As can be seen in the third plot of Figure 4.31, yaw angle(ψ) equals to zero for LQT and LQR controllers, however, backstepping controller has yaw angle of 63.43 degree during flight. As explained in Subsection 3.1.3, backstepping controller is designed in such a way that quadrotor's heading(ψ) and direction of motion in $x-y$ plane are on the same line. By this way, quadrotor could translate in both x and y directions by changing only pith angle(θ) as can be seen in Figure 4.31. Therefore, complexity of flight and coupling effects between roll and pitch motion are avoided. Another advantage is that if visual feedback is used in controllers instead of sensors, camera mounted on quadrotor is automatically adjusted in the direction of motion in $x-y$ plane. It is important to note that, LQT and LQR controllers could not perform at high Euler angles for long time since they are derived by using linearized quadrotor dynamics which is valid near hover condition. Therefore, adjusting heading and direction of motion is not possible for LQT and LQR controllers for yaw angles higher than 15 degrees. To conclude, compared to linear control techniques LQT and LQR,

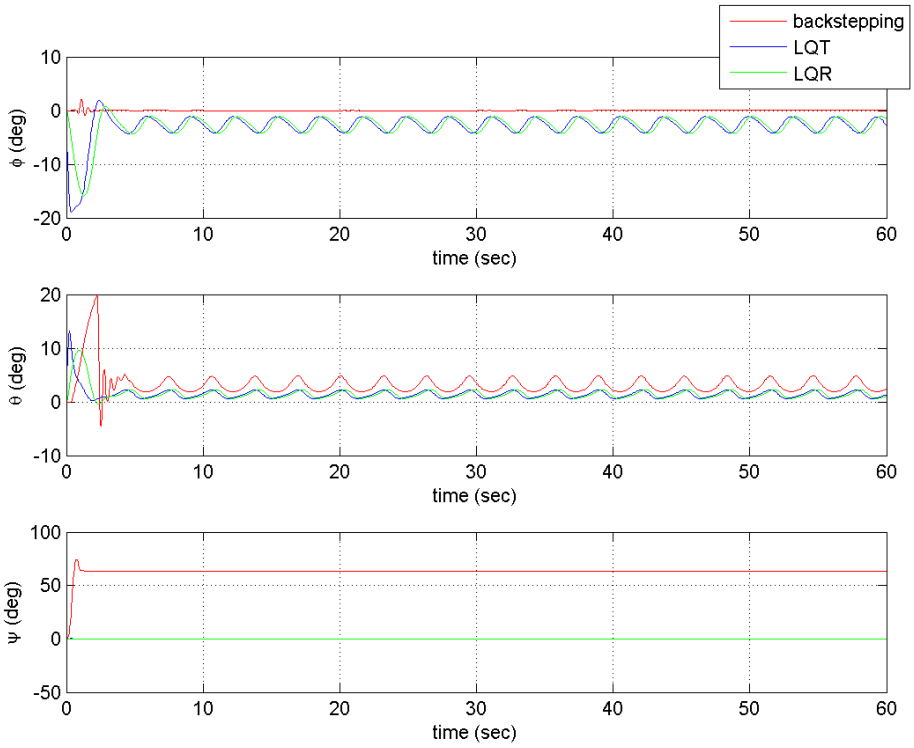


Figure 4.31: Euler angles obtained by each controller while tracking desired trajectory C.

backstepping controller has a more simple and advantageous flight approach since it is a nonlinear control method.

Figure 4.32 shows translation velocities during flight for each controller. As can be seen in the first and second plots of Figure 4.32, $\dot{x}$ and $\dot{y}$ are more steady for backstepping controller unlike LQT and LQR controllers which have more oscillatory values although amplitude of oscillations are very low (0.08 meters). Finally, Figure 4.33 shows angular velocity values in the units of rpm and it can be concluded that, rpm values for all controllers, stay in the constraints of quadrotor.

4.7.3 Summary of Comparison Results

Table 4.3 sums the comparison results of backstepping, LQT and LQR controllers. As can be seen in Table 4.3, LQT controller is advantageous especially for tracking relatively complex trajectories. In addition, as explained in Subsection 4.5.3.1, LQT

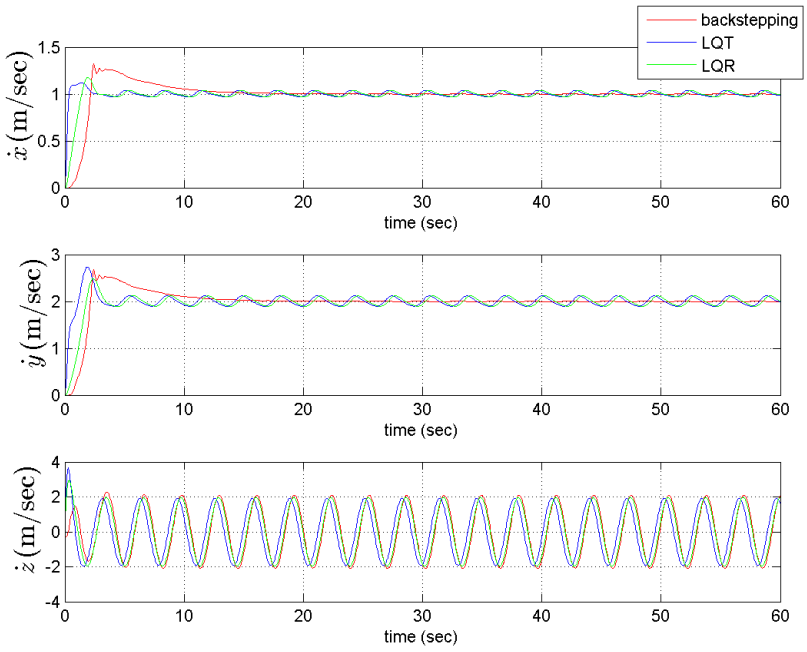


Figure 4.32: Translational velocities obtained by each controller while tracking desired trajectory C.

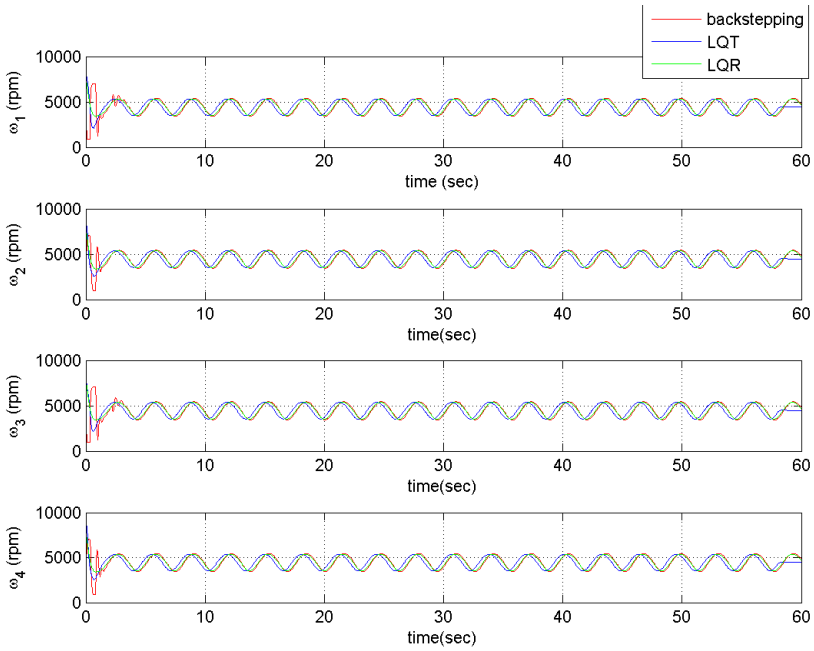


Figure 4.33: Angular velocities of rotors obtained by each controller while tracking desired trajectory C.

controller is energy efficient especially for trajectories that involve sharp maneuvers such as step inputs.

As can be seen in Table 4.3, backstepping controller has a highly complex structure compared to LQT and LQR. This could result in several problems to be solved while experimentally validating backstepping controller. It is also observed that, performance of backstepping controller is directly related to control parameters. One set of optimized parameters could not give satisfactory results for all cases. Therefore, obtaining a global optimized set of control parameters is really hard and requires so much computational time. Based on these observations, simplicity and applicability of backstepping controller is ranked as lower than LQT and LQR controllers.

According to the results obtained in Subsection 4.6.1, disturbance rejection characteristics of LQT and LQR controllers are satisfactory and very similar. However, backstepping controller could not reject disturbances as effective as LQT and LQR controllers.

Table 4.3: Summary of the comparison among backstepping, LQT and LQR controllers, 1=Best, 3=Worst.

	Backstepping	LQT	LQR
Tracking relatively complex trajectories accurately	2	1	3
Tracking simple trajectories accurately	1	2	3
Energy consumption	3	1	2
Simplicity/Applicability	3	2	1
Disturbance Rejection	3	1	1

Although backstepping controller has some drawbacks as can be seen in Table 4.3, it could perform well for high Euler angles since it is a nonlinear control method. For example, backstepping controller could adjust its heading(ψ) according to the direction of motion in $x - y$ plane. By this way, a more practical flight approach could be obtained as explained in Subsection 4.7.2. On the other hand, for large periods of time, LQT and LQR controllers could not perform well at high Euler angles since LQT and LQR are linear control methods which are valid near hover condition.

However, according to the results of Subsection 4.6.1, for short periods of time, LQT and LQR controllers are valid even for high Euler angles up to 60 degrees.

It is also important to note that, for backstepping and LQR controllers, desired trajectory is given to the controller as direct inputs as can be seen in the yellow blocks of Figures 4.2 and 3.4. However, in LQT controller desired trajectory is indirectly given to the controller by using time-variant gains g, L_g, L which are calculated off-line by solving LQT algorithm defined in Equation set 3.60. Therefore, LQT controller could be disadvantageous since time-variant control gains of LQT have to be calculated off-line for each trajectory. However, it is also important to remind that, the code written in this thesis solves LQT algorithm very fast. Therefore, calculating time-variant control gains of LQT does not take significant computational time.

To conclude, according to the comparison results, LQT control technique which is a more original approach compared to backstepping and LQR could be advantageous for tracking relatively complex trajectories accurately and efficiently.

CHAPTER 5

CONCLUSION

Quadrotor UAV is very popular and advantageous VTOL concept especially for specific type of missions/operations and autonomous control of the vehicle has been worked by many researchers and organizations for the last decade. Although quadrotor have many advantageous properties, it has a highly nonlinear, coupled and underactuated dynamics. Therefore, control of the vehicle is not straightforward and many researchers interested in designing and verifying control methods for quadrotors.

In this thesis, trajectory tracking of a quadrotor is obtained by using two independent control methods called as "nonlinear backstepping control" and "optimal LQT (Linear Quadratic Tracking) control". In addition, a fixed-gain LQR controller is also used to compare backstepping, LQT and LQR control methods. Compared to backstepping and LQR, LQT control method is a more original approach and it is an optimal control technique specialized to track desired trajectories. LQT controller uses time-varying control gains optimized to track desired trajectories, unlike fixed-gain LQR control. On the other hand, backstepping controller has been used by many researchers in literature, however, a complete study of tracking relatively complex trajectories by using backstepping controller hasn't been tried widely.

Throughout the thesis, first, dynamic model of the quadrotor is derived by using Newton's equations of motion. Then, each control methods are formulated and designed by using MATLAB/Simulink environment. Finally, initial validation of control systems are obtained by simulations in MATLAB/Simulink and control systems are compared to each other. First, LQT controller is compared to fixed-gain LQR controller since they are both optimal control techniques with different approaches. Then, a

more detailed comparison analysis is made among backstepping, LQT and LQR controllers.

According to the comparison results, it could be stated that, LQT controller could track relatively complex trajectories with high accuracy compared to backstepping and LQR controllers. In addition, energy consumption of LQT controller is lesser than LQR and backstepping controllers especially for trajectories that involve sharp maneuvers. A summary of the positive and negative characteristics of each controllers are also summarized in Subsection 4.7.3.

The quadrotor used in this thesis, "AscTech Hummingbird", is designed and manufactured by "Ascending Technologies" company. "AscTech Hummingbird" quadrotor gives opportunity to develop and test high level control algorithms by embedding algorithms into the board of the quadrotor via MATLAB/Simulink and C++ conversion. Therefore, "AscTech Hummingbird" is a good choice for the purpose of this thesis which is designing and validating high level control algorithms for a quadrotor.

Although simulations give good and motivating results, to achieve a complete validation, controllers have to be verified by real time experiments. Therefore, experimental validation of obtained control algorithms could be performed as a future work. In addition, dynamic model of the quadrotor can be detailed to obtain more realistic and accurate results and performance of controllers could be increased by making some modifications.

REFERENCES

[1] D. S. Naidu, *Optimal Control Systems*, CRC Press LLC, Florida, USA, 2003.

[2] S. Bouabdallah and R. Siegwart, ”Backstepping and sliding-mode techniques applied to an indoor micro quadrotor", *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2247-2252.

[3] K. Nonami, M. Kartidjo, K. J. Yoon and A. Budiyono, *Autonomous Control Systems and Vehicles Intelligent Unmanned Systems*, Springer, Japan, 2013.

[4] G. Coelho, "Ota-Quadrotor: An Object-Tracking Autonomous Quadrotor For Real-Time Detection And Recognition", M.Sc. thesis, University of North Texas, Texas, USA, 2012.

[5] D. Mellinger, "Trajectory Generation and Control for Quadrotors", Ph.D. thesis, University of Pennsylvania, Philadelphia, USA, 2012.

[6] M. N. Lewellen, "Professional Development Short Course on: Unmanned Aircraft Systems and Applications. `http://www.aticourses.com/sampler/unmanned_aircraft_systems_coursesampler.pdf`, last visited on June 2014

[7] R. Austin, *Unmanned Aircraft Systems UAVS Design, Development and Deployment*, John Wiley & Sons Ltd, Great Britain, 2010.

[8] R. K. Barnhart, S. B. Hottman, D. M. Marshall, E. Shappee, *Introduction to Unmmaned Aircraft Systems*, CRC Press Taylor & Francis Group, USA, 2012.

[9] K. P. Valavanis, *Advances in Unmanned Aerial Vehicles State of the Art and the Road to Autonomy*, Springer, Netherlands, 2007.

[10] Ascending Technologies, `http://www.asctc.de/uav-applications/research/products/asctec-hummingbird/`, last visited on June 2014

[11] E. Altuğ, J. P. Ostrowski and R. Mahony, "Control of a Quadrotor Helicopter Using Visual Feedback", *Proceedings of the IEEE International Conference on Robotics & Automation*, Washington, DC, USA, May 2002, pp. 72-77.

[12] Ascending Technologies, `http://www.asctec.de/downloads/flyer/AscTec_RESEARCH_Catalogue.pdf`, last visited on June 2014

[13] AscTech Hummingbird quadrotor, https://pixhawk.ethz.ch/dev/non_public/hummingbird, last visited on June 2014

[14] P. Castillo, R. Lozano, A. E. Dzul, *Modelling and Control of Mini-Flying Machines*, Springer-Verlag London Limited, USA, 2005.

[15] Parrot AR. Drone, http://www.parrot.com/usa/, last visited on June 2014

[16] Draganfly Innovations Inc., http://www.draganfly.com/industrial/products.php, last visited on June 2014

[17] Syma Technology Co., Limited, http://www.symatoys.com/, last visited on June 2014

[18] Institute for Dynamic Systems and Control, ETH Zurich, http://www.idsc.ethz.ch/people/staff/hehn-m, last visited on June 2014

[19] University of California at Berkeley, STARMAC project, https://hybrid.eecs.berkeley.edu/, last visited on June 2014

[20] University of Pennsylvania, GRASP Lab., http://www.upenn.edu/spotlights/penn-quadrotors-ted, last visited on June 2014

[21] Massachusetts Institute of Technology, Aerospace Controls Laboratory, Variable-Pitch Quadrotor Project, http://acl.mit.edu/projects/vpitch_quad.html, last visited on June 2014

[22] G. M. Hoffmann, H. Huang, S. L. Waslander and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment", *Proceedings of the of the AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, USA, August 2007.

[23] M. Cutler, N. Kemal Ure, B. Michini and J. P. How, "Comparison of Fixed and Variable Pitch Actuators for Agile Quadrotors", *Proceedings of the AIAA Guidance, Navigation, and Control Conference(GNC)*, Portland, OR, USA, August 2011.

[24] M. J. Cutler, "Design and Control of an Autonomous Variable-Pitch Quadrotor Helicopter", M.Sc. thesis, Massachusetts Institute of Technology, Massachusetts, USA, 2012.

[25] S. Lupashin, A. Schollig, M. Sherback and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010, pp. 1642-1648.

[26] D. Mellinger, N. Michael and V. Kumar, ”Trajectory generation and control for precise aggressive maneuvers with quadrotors", *Proceedings of the International Symposium on Experimental Robotics*, New Delhi & Agra, India, December 2010.

[27] M. J. Cutler, "Design and Control of an Autonomous Variable-Pitch Quadrotor Helicopter", M.Sc. thesis, MA, USA, 2012.

[28] R. Xu and U. Ozguner, ”Sliding mode control of a quadrotor helicopter", *Proceedings of the IEEE 45th Conference on Decision & Control*, San Diego, CA, USA, December 2006, pp. 4957-4962.

[29] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying", Ph.D. thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, France, 2007.

[30] S. H. Lee, S. H. Kang and Y. Kim, ”Trajectory tracking control of Quadrotor UAV", *Proceedings of the ICCAS International Conference on Control, Automation and Systems*, Gyeonggi-do, Korea, October 2011, pp. 281-285.

[31] L. Derafa, T. Madani and A. Benallegue, ”Dynamic modelling and experimental identification of four rotors helicopter parameters ", *Proceedings of the IEEE International Conference on Industrial Technology*, Mumbai, India, December 2006, pp. 1834-1839.

[32] M. Bangura and R. Mahony, ”Nonlinear dynamic modeling for high performance control of a quadrotor ", *Proceedings of Australasian Conference on Robotics and Automation*, Wellington, New Zealand, December 2012, pp. 1-10.

[33] Ascending Technologies, AscTech Hummingbird `http://www.asctec.de/downloads/datasheets/AscTec-Hummingbird_Safetydatasheet.pdf`, last visited on June 2014

[34] T. Madani and A. Benallegue, ”Backstepping control for a quadrotor helicopter", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 3255-3260.

[35] D. Lee, C. Nataraj, T. C. Burg and D. M. Dawson, ”Adaptive tracking control of an underactuated aerial vehicle", *Proceedings of the American Control Conference*, San Francisco, CA, USA, June-July 2011, pp. 2326-2331.

[36] Y. Li and S. Song, ”A survey of control algorithms for quadrotor unmanned helicopter", *Proceedings of the IEEE fifth International Conference on Advanced Computational Intelligence(ICACI)*, Nanjing, Jiangsu, China, October 2012, pp. 365-369.

[37] Y. Li and S. Song, "A Survey of Control Algorithms for Quadrotor Unmanned Helicopter", *Proceedings of the IEEE International Conference on Advanced*

Computational Intelligence(ICACI), Nanjing, Jiangsu, China, October 2012, pp. 365-369.

[38] Y. M. Al-Younes, M. A. Al-Jarrah and A. A. Jhemi, "Linear vs. nonlinear control techniques for a quadrotor vehicle", *Proceedings of the ISMA10 International Symposium on Mechatronics and its Applications*, Sharjah, UAE, April 2010, pp. 1-10.

[39] M. Santos, V. Lopez and F. Morata, "Intelligent Fuzzy Controller of a Quadrotor", *Proceedings of the IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Hangzhou, China, November 2010, pp. 141-146.

[40] J.Su,P.Fan and K.Cai, "Attitude Control of Quadrotor Aircraft via Nonlinear PID", *.in Journal of Beijing University of Aeronautics and Astronautics*.37,9,pp.1054-1059,9.2011.

[41] M. Ö. Efe, "Neural Network Assisted Computationally Simple $PI^{\lambda}D^{\mu}$ Control of a Quadrotor UAV", *.in IEEE Transactions on Industrial Informatics*.Vol.7,No.2,May 2011.

[42] Q. L. Zhou, Y. Zhang, C. A. Rabbath and D. Theilliol, "Design of Feedback Linearization Control and Reconfigurable Control Allocation with Application to a Quadrotor UAV", *Proceedings of the Conference on Control and Fault Tolerant Systems*, Nice, France, October 2010, pp. 371-376.

[43] B. Panomrattanarug, K. Higuchi and F. Mora-Camino, "Attitude control of a quadrotor aircraft using LQR state feedback controller with full order state observer", *Proceedings of the SICE Annual Conference 2013*, Nagoya, Japan, September 2013, pp. 2041-2046.

[44] J. Berg, D. Wilkie, S. J. Guy, M. Niethammer and D. Manocha, "LQG-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty", *Proceedings of the IEEE International Conference on Robotics and Automation*, Saint Paul, Minnesota, USA, May 2012, pp. 346-353.

[45] L. D. Minh and C. Ha, "Modeling and Control of Quadrotor MAV Using Vision-based Measurement", *Proceedings of the International Forum on Strategic Technology*, Ulsan, Korea (South), October 2010, pp. 70-75.

[46] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara and J. Guichard, "LQR Control for a Quadrotor using Unit Quaternions: Modeling and Simulation", *Proceedings of the International Conference on Electronics, Communications and Computing (CONIELECOMP)*, Puebla, Mexico, March 2013, pp. 172-178.

[47] F. Hoffmann, N. Goddemeier and T. Bertram, "Attitude Estimation and Control of a Quadrocopter", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* , Taipei, Taiwan, October 2010, pp. 1072-1077.

[48] L. Tan, L. Lu and G. Jin, "Attitude Stabilization Control of a Quadrotor Helicopter using Integral Backstepping", *Proceedings of the International Conference on Automatic Control and Artificial Intelligence* , Xiamen, China, March 2012, pp. 573-577.

[49] T. Madani and A. Benallegue, "Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique", *Proceedings of the IEEE Conference on Decision & Control*, San Diego, CA, USA, December 2006, pp. 1515-1520.

[50] S. Nadda and A. Swarup, "Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique", *Proceedings of the IEEE International Colloquium on Signal Processing & its Applications (CSPA2014)*, Kuala Lumpur, Malaysia, March 2014, pp. 10-13.

[51] Z. Fang, Z. Zhi, L. Jun and W. Jian, "Linearization and Continuous Sliding Mode Control for a Quadrotor UAV", *Proceedings of the Chinese Control Conference*, Kunming, Yunnan, China, July 2008, pp. 349-353.

[52] Q. Geng, H. Shuai and Q. Hu, "Obstacle avoidance approaches for quadrotor UAV based on backstepping technique", *Proceedings of the Chinese Control and Decision Conference (CCDC)*, Guiyang, China, May 2013, pp. 1-10.

[53] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor" *.in IET Control Theory & Applications*. Vol. 4, Iss. 11, pp. 2343–2355,2010.

[54] S. A. Al-Hiddabi, "Quadrotor Control Using Feedback Linearization with Dynamic Extension", *Proceedings of the International Symposium on Mechatronics and its Applications (ISMA09)*, Sharjah, UAE, March 2009, pp. 1-3.

[55] Z. Fang and W. Gao, "Adaptive Integral Backstepping Control of a Micro-Quadrotor", *Proceedings of the International Conference on Intelligent Control and Information Processing* , Harbin, China, July 2011, pp. 910-915.

[56] Y. Al-Younes, M. A. Jarrah, "Attitude Stabilization of Quadrotor UAV Using Backstepping Fuzzy Logic & Backstepping Least-Mean-Square Controllers", *Proceedings of the International Symposium on Mechatronics and its Applications*, Amman, Jordan, May 2008.

[57] S. Bouabdallah and R. Siegwart, "Full Control of a Quadrotor", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, October-November 2007, pp. 153-158.

[58] G. V. Raffo, M. G. Ortega and F. R. Rubio, "Backstepping/Nonlinear H_∞ Control for Path Tracking of a QuadRotor Unmanned Aerial Vehicle", *Proceedings of the American Control Conference*, Seattle, Washington, USA, June 2008, pp. 3356-3361.

[59] T. Lee, M. Leok and N. H. McClamroch, "Geometric Tracking Control of a Quadrotor UAV on SE(3)", *Proceedings of the IEEE Conference on Decision and Control*, Atlanta, GA, USA, December 2010, pp. 5420-5425.

[60] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors", *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 2520-2525.

[61] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer and M. Pollefeys, "Autonomous Obstacle Avoidance and Maneuvering on a Vision-Guided MAV Using On-Board Processing", *Proceedings of the International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 2472-2477.

[62] M. Hehn and R. D'Andrea, "A Flying Inverted Pendulum", *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 763-770.

[63] S. Gupte, P. I. T. Mohandas and J. M. Conrad, "A Survey of Quadrotor Unmanned Aerial Vehicles", *Proceedings of the Southeastcon 2012 of IEEE*, Orlando, FL, USA, March 2012, pp. 1-6.

[64] D. Cabecinhas, R. Cunha and C. Silvestre, "Experimental Validation of a Globally Stabilizing Feedback Controller for a Quadrotor Aircraft with Wind Disturbance Rejection", *Proceedings of the American Control Conference(ACC)*, Washington, DC, USA, June 2013, pp. 1024-1029.

[65] B. Etkin, *Dynamics of Atmospheric Flight*, Dover Publications, New York, USA, 2005.

[66] M. Achtelik, "Nonlinear and adaptive control of a quadcopter", Dipl.-Ing. Dissertation, Lehrstuhl für Flugsystemdynamik, Technische Universität München, Garching, Germany, 2010.

[67] A. A. Shabana, *Dynamics of Multibody Systems*, Cambridge University Press, New York, USA, 2005.

[68] M. V. Cook, *Flight Dynamics Principles*, Butterworth-Heinemann publications, Great Britain, 2007.

[69] D. McLean, *Automatic Flight Control Systems*, Prentice Hall International (UK) Ltd, Cambridge, Great Britain, 1990.

[70] A. Tewari, *Atmospheric and Space Flight Dynamics: Modeling and Simulation with MATLAB and Simulink*, Birkhäuser Boston Springer Science+Business Media LLC, Birkhäuser Boston, 2007.

[71] G. D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling*, Blackwell Publishing, Washington DC, USA, 2007.

[72] M. J. Sidi, *Spacecraft Dynamics and Control:A Practical Engineering Approach*, Cambridge University Press, NY, USA, 1997.

[73] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, New Jersey, USA, 2002.

[74] E. C. Suicmez and A. T. Kutay, "Optimal path tracking control of a quadrotor UAV", *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA, May 2014, pp. 115-125.

[75] The MathWorks, Inc., `http://www.mathworks.com/help/control/ref/lqr.html`, last visited on June 2014