

CONTROL AND TRAJECTORY PLANNING OF A QUADROTOR WITH A
2-DOF ROBOTIC ARM FOR PRECISE TARGET ENGAGEMENT

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

BY

MEHMET ANIL SEL

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

JANUARY 2020

Approval of the thesis:

**CONTROL AND TRAJECTORY PLANNING OF A QUADROTOR WITH A
2-DOF ROBOTIC ARM FOR PRECISE TARGET ENGAGEMENT**

submitted by **MEHMET ANIL SEL** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. M. A. Sahir Arıkan
Head of Department, **Mechanical Engineering**

Assist. Prof. Dr. Ali Emre Turgut
Supervisor, **Mechanical Engineering, METU**

Assist. Prof. Dr. Kutluk Bilge Arıkan
Co-Supervisor, **Mechanical Engineering Dept., TEDU**

Examining Committee Members:

Assoc. Prof. Dr. Ulaş Yaman
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Ali Emre Turgut
Mechanical Engineering, METU

Assoc. Prof. Dr. Ender Yıldırım
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering Dept., TEDU

Prof. Dr. Kemal Özgören
Mechanical Engineering Dept., METU

Date: 27.01.2020

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, Surname: Mehmet Anıl Sel

Signature:

ABSTRACT

CONTROL AND TRAJECTORY PLANNING OF A QUADROTOR WITH A 2-DOF ROBOTIC ARM FOR PRECISE TARGET ENGAGEMENT

Sel, Mehmet Anıl
Master of Science, Mechanical Engineering
Supervisor: Assist. Prof. Dr. Ali Emre Turgut
Co-Supervisor: Assist. Prof. Dr. Kutluk Bilge Arıkan

January 2020, 160 pages

In this study, control and trajectory planning of a quadcopter system is presented for precise target engagement. Quadcopter system consists of a quadcopter body, a 2-DOF robotic arm mounted at the bottom and an object is held by the end-effector of the robotic arm. As for the dynamics of the quadcopter system is derived by using the kinematic relations of the system members. Equation of motion is obtained by using Lagrange-d'Alembert's Principle. Then, object-target engagement is investigated by considering an adjustable trajectory. Two mission parameters which are the relative distance of the target and the release angle of the object are established for shaping the trajectory. The forward kinematics algorithm is developed for finding the engagement states. Reference inputs of the quadcopter system are optimized by minimization of the control effort. The trajectory of the quadcopter system is planned for the initial to engagement state of the quadcopter system. Firstly, the cascaded PID controller is designed by linearizing the equation of motion of the quadcopter system. The controller is tested with the existence of the motor and the sensor subsystems of the simulation environment. An object throwing scenario is executed by generating the control commands with trial error method. Cascaded PID controller is also implemented in the real physical system. Then, hardware dependent algorithms are developed in order to improve the flight performance. In addition to that, quadcopter's moment of inertia is identified to have more realistic model of the system in the

simulation environment. Secondly, an infinite horizon LQR controller is developed for trajectory tracking. That controller is designed by considering the linearized equation of motion of the system. That controller structure is also tested in the same simulation environment. Precise target engagement is investigated while analyzing the energy consumption. All the proposed controller algorithms, kinematics and the dynamics of the quadcopter system are implemented in MATLAB/Simulink. Finally, the first controller structure is performed in the real physical system. However, both control algorithms are validated in simulation based experiments. In the framework, feasibility of the optimal trajectory with respect to both quadcopter system dynamics and the control inputs is guaranteed. Precise target engagement is achieved by the successive system performance.

Keywords: Precise Target Engagement, Quadcopter, Trajectory Optimization, Trajectory Planning, Aerial Manipulation, Cascaded PID Controller, LQR Controller, Moment of Inertia Identification, Quadcopter Physical Implementation, Throwing an Object

ÖZ

HASSAS NİŞAN İLE HEDEFLE BULUŞMA İÇİN 2 SERBESTLİK DERECELİ ROBOT KOLU İLE KUADKOPTER SİSTEMİNİN KONTROLÜ VE YÖRÜNGE PLANLAMASI

Sel, Mehmet Anıl
Yüksek Lisans, Makina Mühendisliği
Tez Danışmanı: Dr. Öğr. Üyesi Ali Emre Turgut
Ortak Tez Danışmanı: Dr. Öğr. Üyesi Kutluk Bilge Arıkan

Ocak 2020, 160 sayfa

Bu tez çalışmasında, bir dört pervaneli robot helikopter sisteminin yörünge planlaması ve kontrolünün hassas nişan ile hedefle buluşturulması sunulmuştur. Dört pervaneli robot helikopter sistemi, dört pervaneli robot helikopter gövdesi, alttan entegre 2 serbestlik dereceli bir robot kolu, uç elemanı ile bir objeden meydana gelmektedir. Bütünsel gövde dinamiği, sistem parçalarının kinematik ilişkilerinden türetilmiştir. Hareket denklemleri ise Lagrange-D'Alembert prensibinden elde edilmiştir. Ardından, obje-hedef buluşması biçimlendirilebilir yörünge özelliğinin dikkate alınması ile incelenmiştir. Yörünge biçimlendirme, iki değişkenden meydana gelen obje hızı ve objenin bırakılma açısı üzerinden olacak şekilde kurgulanmıştır. Bunun ardından, ileriye doğru kinematik algoritması geliştirilerek sistem atış durumları önceden belirlenmiş obje yörüngesi üzerinden elde edilmiştir. Bütünsel sistem referans girdileri kontrolcü girdileri üzerinden minimize edilecek şekilde optimize edilmiştir. Bütünsel sistem yörüngesi, sistemin ilklendirme durumundan, önceden belirlenmiş atış durumuna olacak şekilde planlanmıştır. Öncelikli olarak, temsili ardarda bağlanmış PID yapısı, doğrusallaştırılmış hareket denklemleri üzerinden bütünsel sistem için tasarlanmıştır. Kontrolcü ideal olmayan motor ve sensör modelleri ile simülasyon ortamında test edilmiştir. Deneme yanılma yöntemi ile türetilmiş kontrol komutlarıyla obje atma senaryosu uygulanmıştır. Ayrıca, ardarda

bağlanmış PID kontrolcüsü gerçek sisteme aktarılmıştır. Uçuş gürbüzlüğünü arttırabilmek için uçuş algoritması geliştirilmiştir. Simülasyon ortamını gerçekçileştirmek adına dört pervaneli robot helikopter ataleti tanımlanmıştır. Diğer deneysel parametreler de simülasyona uygulanmıştır. Sonrasında, yörünge takibi çalışması için bir sonsuz ufuk LQR kontrolcüsü geliştirilmiştir. Kontrolcü, doğrusallaştırılmış hareket denklemlerinden tasarlanmıştır. Bu kontrolcü yapısı da ideal olmayan motor ve sensör modelleri ile simülasyon ortamında test edilmiştir. Hassas hedefle buluşma konusu analiz edilirken enerji tüketimi incelenmiştir. Bütün önerilen algoritmalar ve sistem seviyesi uygulamalar MATLAB/Simulink ortamında tasarlanmıştır. Son olarak, ilk açıklanan kontrolcü yapısı gerçek sistemde test edilmiştir ve bununla birlikte iki kontrolcü yapısı, simülasyon çalışmaları ile test edilmiştir.

Anahtar Kelimeler: Hassas Hedef Buluşması, Kuadkopter, Yörünge Optimizasyonu, Yörünge Planlama, Hava Manipülasyonu, Ardarda bağlı PID kontrolcüsü, LQR kontrolcüsü, Atalet Momenti Tanımlama, Kuadkopter Fiziksel Uygulaması, Bir Objeye Fırlatma

To My Family and Continuation of Life

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my supervisor Assist. Prof. Dr. Ali Emre Turgut and co-supervisor Assist. Prof. Dr. Kutluk Bilge Arıkan, not only for their academic recommendation, but for their kindly and patient manner to keep my motivation alive and to continue working even in the tough times.

I would also like to thank to my friend Ahmet Turgut Başaranoğlu, for giving the all his patience and hard work to become a team that we built together. I am so grateful that, I couldn't even take the first step without him.

I would like to thank to my colleague and college friend Nebi Bulut for inspiring discussions that we shared and his best wishes to increase my acceleration throughout this study.

I would like to thank to my colleagues in ROKETSAN Rüştü Berk Gezer and Hakkı Ulaş Ateş for his invaluable support.

Finally, I wish to thank to my mother, my father, my sister and my lovely girlfriend for their support and encouragement throughout my study.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGEMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xvi
LIST OF FIGURES	xviii
LIST OF ABBREVIATIONS	xxiii
LIST OF SYMBOLS	xxiv
CHAPTERS	
1. INTRODUCTION	1
1.1. Motivation of the Thesis.....	1
1.2. Aim of the Thesis Study	3
1.3. Thesis Outline.....	3
2. LITERATURE REVIEW	5
2.1. Quadcopter with Manipulator Systems and Throwing an Object	5
2.2. Literature on Contributions	7
2.2.1. Studies Based on Quadcopters.....	7
2.2.2. Developments for Quadcopters with Their Flying Hands	8
2.2.3. Concentration on Throwing an Object.....	9
2.2.4. Studies Based on Trajectory Generation and Tracking for Quadcopters ..	10

2.3. Research Objectives and Summary of the Literature Survey	12
2.3.1. Contribution of the Thesis	13
3. MATHEMATICAL MODELLING AND CONTROLLER DESIGN	15
3.1. Kinematics of the Quadcopter System.....	15
3.1.1. Denavit-Hartenberg Parameters	16
3.1.1.1. Twist Angles	16
3.1.1.2. Joint Angles	17
3.1.1.3. Offsets.....	17
3.1.1.4. Link Lengths	17
3.1.2. Transformation Matrices	17
3.1.3. Position Analysis of the System Model	19
3.1.4. Velocity Analysis of the System Model.....	22
3.1.4.1. Linear Velocity Analysis of the System Members	22
3.1.4.2. Angular Velocity Analysis of the System Members	25
3.2. Dynamics of the Quadcopter System.....	29
3.2.1. System's Force and Torque Units	32
3.2.2. Input and Interaction Forces of the System.....	34
3.2.3. Motor Subsystem.....	38
3.2.4. Sensor Subsystem.....	39
3.3. Cascaded PID Controller of the Quadcopter System.....	40
3.3.1. Outer Loop Controller	41
3.3.2. Inner Loop Control	42
3.3.2.1. Altitude controller.....	42
3.3.2.2. Attitude Controller.....	43

3.3.3. Controller of the Robotic Manipulator	44
3.3.4. Multi-Objective Optimization	45
4. TRAJECTORY PLANNING AND PRECISE TARGET ENGAGEMENT	51
4.1. Problem Definition	51
4.2. Trajectory Analysis of the Ball	52
4.2.1. Forward Kinematics of the Ball Throwing Motion	52
4.2.2. Separation Velocity Vector and Release Angle Kinematic Equation.....	53
4.2.3. Forward Kinematics Algorithm for the Engagement State	54
4.3. Trajectory Generation of the Quadcopter System.....	57
4.3.1. Optimized Trajectory Generation	57
4.4. Controller Topology of the Optimal State Trajectory Tracking of the Quadcopter System.....	61
5. SIMULATION BASED EXPERIMENTS	65
5.1. Simulation of the Quadcopter System.....	65
5.2. Scenarios of the Simulation Based Experiment	66
5.2.1. Examining the Cascaded PID Controller Configuration for Ball Throwing	66
5.2.2. Examining the Trajectory Planning and LQR Controller Configuration for Ball Throwing	71
5.2.2.1. Trajectory Generation and Tracking Analysis for the Scenario-2 and the Scenario-3 Pair	71
5.2.2.2. Performance Limits of the Trajectory Optimization and Tracking....	84
6. PHYSICAL SYSTEM IMPLEMENTATION	95
6.1. System Hardware and Software Details	95
6.1.1. Moment of Inertia Identification.....	98

6.1.2. Robotic Arm	104
6.1.3. Quadcopter System Flight Controller.....	106
6.1.4. Indoor UWB Localization - Pozyx and Data Acquisition.....	107
6.1.5. Quadcopter System Signal Assistance Controller.....	111
6.1.6. Power Distribution	113
6.1.7. Detailed Information about the Other Subsystems.....	114
6.1.7.1. Battery.....	114
6.1.7.2. ESCs	115
6.1.7.3. Brushless DC Motors and Propellers.....	116
6.1.7.4. Voltage Regulator.....	117
6.1.8. System Software.....	118
6.1.8.1. Controller Architecture	118
6.1.8.2. Data Collection and Flight Replay	122
6.1.8.3. Flight Assistance Software	122
6.1.8.4. Ground Test for Bias Collection.....	122
6.2. Experiments	126
6.2.1. Experiments of the Quadcopter System without the Robotic Arm	128
6.2.1.1. Hover Test	128
6.2.1.2. 3-D Rectangular Path Tracking Test	134
7. DISCUSSION AND CONCLUSION	139
7.1. General Discussion	139
7.2. Conclusion	141
7.3. Future Work	143
REFERENCES	145

APPENDICES

Appendix A: 3-D Scenario Tracking with the Cascaded PID Controller Configuration.....	153
Appendix B: Experiment of the Quadcopter System.....	156
Appendix C: Implementation of 3-D LQR Controller for the Quadcopter System.....	159

LIST OF TABLES

TABLES

Table 3.1 DH Parameters of the Robotic Arm	17
Table 3.2 System Sensor's Model	39
Table 3.3 Pre-defined and Optimized PID Gains for Altitude Channel	46
Table 3.4 The Step Response Characteristics for the Altitude Dynamics	47
Table 3.5 Optimized Gains of the System for Cascaded PID	49
Table 5.1 System Parameter Set for the Simulation Model.....	65
Table 5.2 Scenarios of the Quadcopter System Simulation and Analysis.....	66
Table 5.3 Shooting Parameters of the Scenario-1	67
Table 5.4 Shooting Parameters obtained for Scenario-2	72
Table 5.5 Boundaries of the Path for Scenario-2.....	73
Table 5.6 Shooting Parameters obtained for Different Release Angle in Scenario-3.....	77
Table 5.7 Boundaries of the Path for Different Release Angle in Scenario-3.....	78
Table 5.8 Engagement State obtained for the Performance Index Analysis	85
Table 5.9 Boundaries of the Path for the First Layer of the Performance Index Analysis	86
Table 5.10 Boundaries of the Path for the Second Layer of the Performance Index Analysis	86
Table 6.1 Identified Inertia Values of the Quadcopter	102
Table 6.2 Inertia of the Links	103
Table 6.3 Link Length of the Robotic Arm	105
Table 6.4 Detailed Information about Robotic Arm's RC servo motors.....	105
Table 6.5 Quadcopter Flight Controller card performance specifications	107
Table 6.6 Naze 32 Rev 6 Technical Details	112
Table 6.7 Quadcopter System Energy Resource Unit	114
Table 6.8 Operating performance information of the Quadcopter's ESCs	115

Table 6.9 Technical Information of Motor and Propeller Used in the System	116
Table 6.10 Technical Specs of the Voltage Regulator	117
Table 6.11 Controller Parameters of the FCA	120
Table 6.12 FCA Filter Parameters	121
Table 6.13 Parameters and Conditions for Throwing the Ball.....	121
Table 6.14 Bias Values of Quadcopter System.....	125
Table 6.15 Hover Test Reference Input Coordinate in 3-D.....	129
Table 6.16 Reference Inputs of the 3-D Scenario Generation	134

LIST OF FIGURES

FIGURES

Figure 3.1 Kinematics of the Quadcopter System	16
Figure 3.2 Quadcopter Detailed FBD for Actuating Forces and Torques	32
Figure 3.3 Sensor Model.....	39
Figure 3.4 Closed Loop Controller Structure	41
Figure 3.5 Cascaded Altitude Controller Schematic	44
Figure 3.6 Reference Input and Closed Loop System Response for the Linearized Altitude Dynamics of the Quadcopter System	47
Figure 3.7 Sensitivity of the ITAE Compared with IAE and ISE for the Zero-Velocity Error in the Second Order System Dynamics [66]	48
Figure 4.1 Trajectory Planning and Precise Target Engagement	52
Figure 4.2 Ball Trajectory Kinematics	55
Figure 4.3 True Trajectory of the Ball.....	55
Figure 4.4 Forward Kinematics Algorithm	56
Figure 4.5 Simplified 3-DOF Quadcopter System	58
Figure 4.6 Optimal States of the Trajectory Generation.....	61
Figure 4.7 Implementation of the LQR controller.....	63
Figure 5.1 Linear Velocity of the Quadcopter System for Scenario-1	68
Figure 5.2 Angular Position of the System Parts for Scenario-1.....	68
Figure 5.3 Angular Velocity of the System Parts for Scenario-1	69
Figure 5.4 Rotational Speed of Rotors during the Flight Simulation for Scenario-1	69
Figure 5.5 Trajectory of the Ball Compared with the Achieved System Output for Scenario-1	70
Figure 5.6 Energy of the Quadcopter System in the Forward Kinematics Solution for Scenario-2	73
Figure 5.7 Linear Position Reference Input and System Response for Scenario-2... ..	74

Figure 5.8 Angular Position Reference Input and System Response for Scenario-2	74
Figure 5.9 Linear Velocity Reference Input and System Response for Scenario-2	75
Figure 5.10 Theta Angular Velocity Reference Input and System Response for Scenario-2	75
Figure 5.11 Theta1 Angular Velocity Reference Input and System Response for Scenario-2	76
Figure 5.12 Theta2 Angular Velocity Reference Input and System Response for Scenario-2	76
Figure 5.13 Energy of the Quadcopter System in the Forward Kinematics Solution for Different Release Angle in Scenario-3	78
Figure 5.14 Linear Position Reference Input and System Response for Different Release Angle in Scenario-3	79
Figure 5.15 Angular Position Reference Input and System Response for Different Release Angle in Scenario-3	79
Figure 5.16 Linear Velocity Reference Input and System Response for Different Release Angle in Scenario-3	80
Figure 5.17 Theta Angular Velocity Reference Input and System Response for Different Release Angle in Scenario-3	80
Figure 5.18 Theta1 Angular Velocity Reference Input and System Response for Different Release Angle in Scenario-3	81
Figure 5.19 Theta2 Angular Velocity Reference Input and System Response for Different Release Angle in Scenario-3	81
Figure 5.20 Comparison of the Achieved Trajectories with the Existence of an Obstacle	83
Figure 5.21 Energy of the Quadcopter System in the Forward Kinematics Solution for Scenario-4 and Scenario-5	85
Figure 5.22 Flight Path Comparison of the Quadcopter System for Scenario-4 and Scenario-5	86
Figure 5.23 Linear Position Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	87

Figure 5.24 Altitude Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	87
Figure 5.25 Euler Pitch Angle Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5.....	88
Figure 5.26 Link-1 Angular Position Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	89
Figure 5.27 Link-2 Angular Position Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	89
Figure 5.28 Linear Velocity in X-Direction Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	90
Figure 5.29 Linear Velocity in Z-Direction Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	90
Figure 5.30 Euler Pitch Rate Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5.....	91
Figure 5.31 Link-1 Angular Velocity Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	91
Figure 5.32 Link-2 Angular Velocity Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5	92
Figure 5.33 Trajectory of the Ball for the Performance Index Analysis	93
Figure 6.1 Physical System of the Ball Throwing Quadcopter	96
Figure 6.2 Physical System of the Ball Throwing Quadcopter on Duty	96
Figure 6.3 Software and Hardware Configuration	97
Figure 6.4 Bifilar Pendulum Test Setup	98
Figure 6.5 Collected IMU Signal output: Euler Yaw Angle	99
Figure 6.6 Collected IMU Signal output: Euler Yaw Angle	99
Figure 6.7 Differentiation of Euler Roll Angle	100
Figure 6.8 Differentiation of Euler Yaw Angle.....	100
Figure 6.9 Average Inertia for X and Y plane	101
Figure 6.10 Average Inertia for Z plane	101

Figure 6.11 Configuration of the Robotic Arm Mounted at the Bottom of the Quadcopter	104
Figure 6.12 Raspberry Pi 3 Hardware and GPIO Pinout Diagram [59]	106
Figure 6.13 Indoor UWB Localization System Unit Pozyx Tag [60].....	108
Figure 6.14 Indoor UWB Localization System Unit Pozyx Anchor [60].....	108
Figure 6.15 Flight Area Specified by the Locations of Anchors	109
Figure 6.16 Indoor UWB Localization Configuration Adjusted For Flight	110
Figure 6.17 Naze-32 Pin Diagram [40].....	111
Figure 6.18 Power Distribution Schematics of the Quadcopter System.....	113
Figure 6.19 Pro-Fuse battery type for the flight [40].....	114
Figure 6.20 ESCs of the Quadcopter System [40]	115
Figure 6.21 EMAX Brushless DC Motor [40].....	116
Figure 6.22 10” Propellers of the Quadcopter System [40].....	116
Figure 6.23 XY-3606 Voltage Regulator [40]	117
Figure 6.24 FCA Architecture	119
Figure 6.25 Mini Water Gage for Calibrating the Flight Assistance Card	123
Figure 6.26 Bias Calculation on Roll Channel.....	124
Figure 6.27 Bias Calculation on Pitch Channel	124
Figure 6.28 Bias Calculation on Yaw Channel.....	125
Figure 6.29 Test Area.....	126
Figure 6.30 The Quadcopter System without the Robotic Arm during a flight.....	127
Figure 6.31 The Quadcopter System.....	128
Figure 6.32 Altitude Controller in Hover Test.....	129
Figure 6.33 X Position Controller in Hover Test.....	130
Figure 6.34 Y Position Controller in Hover Test.....	130
Figure 6.35 Quadcopter Trajectory in XY Plane – Hover Test	131
Figure 6.36 Euler ϕ Controller in Hover Test	131
Figure 6.37 Euler θ Controller in Hover Test.....	132
Figure 6.38 Euler ψ Controller in Hover Test	132
Figure 6.39 ϕ Controller in Hover Test	133

Figure 6.40 θ Controller in Hover Test	133
Figure 6.41 X Position Controller Command Tracking Performance	134
Figure 6.42 Y Position Controller Command Tracking Performance	135
Figure 6.43 Quadcopter Trajectory in XY Plane - Position Test	135
Figure 6.44 ϕ Controller in Position Test.....	136
Figure 6.45 θ Controller in Position Test.....	136
Figure 6.46 ψ Controller in Position Test	137
Figure 6.47 $\dot{\phi}$ Controller in Position Test.....	138
Figure 6.48 $\dot{\theta}$ Controller in Position Test.....	138

LIST OF ABBREVIATIONS

ABBREVIATIONS

FCA: Flight Controller Algorithm

LQR: Linear Quadratic Regulator

DOF: Degree of Freedom

UAV: Unmanned Aerial Vehicle

UAM: Unmanned Air Manipulators

DC: Direct Current

CG: Center of Gravity

ESC: Electronic Speed Controller

RC: Radio Control

IMU: Inertial Measurement Unit

UWB: Ultra Wide Band

Li-Po: Lithium Polymer

PWM: Pulse Width Modulation

ITAE: Integral of the Time-Weighted Absolute Error

ITSE: Integral of the Time-Weighted Squared Error

IAE: Integral of the Absolute Value of the Error

ISE: Integral of the Squared Error

JE: Just Error

LIST OF SYMBOLS

SYMBOLS

ϕ : Euler roll angle

θ : Euler pitch angle

ψ : Euler yaw angle

$\bar{\Gamma}$: Column matrix of Euler Angles 1-2-3 respectively

O_i : Origin of the inertial reference frame

O_b : Origin of the body-fixed reference frame

O_1 : Origin of the link-1 fixed reference frame

O_2 : Origin of the link-2 fixed reference frame

O_e : Origin of the end-effector fixed reference frame

θ_1 : Relative angle between quadcopter body and link-1

θ_2 : Relative angle between link-1 and link-2

$\bar{\Omega}$: Column matrix of Joint Angles 1-2 respectively

b_k : k^{th} link offset

β_k : k^{th} twist angle

$\bar{u}_1^{(k)}, \bar{u}_2^{(k)}, \bar{u}_3^{(k)}$: Unit basis vectors of reference frame K

$\bar{u}_1, \bar{u}_2, \bar{u}_3$: Basic column matrices

$\hat{C}^{(b,1)}$: Transformation matrix from link-1 reference frame to body-fixed reference frame

$\hat{C}^{(1,2)}$: Transformation matrix from link-2 reference frame to link-1 reference frame

$\hat{C}^{(b,2)}$: Transformation matrix from link-2 reference frame to body-fixed reference frame

$\hat{C}^{(i,b)}$: Transformation matrix from body-fixed reference frame to inertial fixed reference frame

$\vec{p}_{1/q}$: Position vector of link-1 defined on CG wrt. quadcopter CG

$\vec{p}_{2/q}$: Position vector of link-2 defined on CG wrt. quadcopter CG

$\vec{p}_{e/q}$: Position vector of the center of end effector's gripper wrt. quadcopter CG

$\bar{p}_q^{(i)}$: Position of body wrt. inertial reference frame resolved in inertial reference frame

$\bar{p}_1^{(b)}$: Position of link-1 wrt. body-fixed reference frame resolved in body reference frame

$\bar{p}_2^{(b)}$: Position of link-2 wrt. body-fixed reference frame resolved in body reference frame

$\bar{p}_e^{(b)}$: Position of end-effector with respect to the body-fixed reference frame resolved in body reference frame

$\bar{\omega}_{q/i}^{(b)}$: Angular velocity of body-fixed reference frame wrt. inertial reference frame resolved in body reference frame

\hat{L} : Mapping Matrix

\hat{J} : Jacobian matrix

L: Lagrange operator

K: Kinetic energy

U: Potential energy

E_k : Energy consumption of k^{th} element

P_k : Power consumption of k^{th} element

\bar{u} : Column matrix of generalized input force

\bar{u}_{ext} : Column matrix of externally applied force on the end-effector

$m_t, m_b, m_{link1}, m_{link2}, m_{ball}$: Masses of the quadcopter system, quadcopter, link-1, link-2 and ball

g : Gravity constant

$V_{x_{ball}}, V_{z_{ball}}$: Velocity of the ball in planar space

$\theta_{release}$: Release angle of the ball resolved from the instant release velocity vector of the end effector

v_{req} : Required speed of the ball

$\omega_1, \omega_2, \omega_3, \omega_4$: Angular velocities of the rotors

c_T : Thrust factor

c_Q : Drag factor

σ_{noise} : Amplitude of the noise

K_p : Proportional gain of PID

K_i : Integral gain of PID

K_d : Derivative gain of PID

λ : Eigen value

CHAPTER 1

INTRODUCTION

1.1. Motivation of the Thesis

As the technological improvements are increasing, demand for the mobile technologies are to be supplied. Because of this matter of fact, the interest behind UAVs is increasing in the last decade. The grand potential on the UAVs is popped up. The developments in the design and the control of the UAVs are performed widely for making the community's life easier. Since they can be used not only within the research area but also within the industrial purposes.

UAVs are classified depending on their application area. The most important classification of the UAVs can be made on their flying principle and their propulsions. The motorized subclass of their propulsion is divided into 3 parts [61]. The first one, the fixed-wing UAVs are mostly used for long distance at high altitude missions. They have a remarkable place in the meteorological reconnaissance and the navigation purposes in the military applications. The second one, flapping-wing type of UAVs are still under development. This classification simulates the way birds or insects fly. They are low endurance and micro size. They have the capability of vertical take-off and landing. In the last one, rotary wings UAVs in other words Vertical Take-off and Landing rotorcrafts are mostly used in the missions that require hovering mode. They are superior in air turbulence compared with other classifications of UAVs.

Aerial robotics is a fast growing part of the rotary wings UAVs. The best example for this rapidly growing area in the community is the quadcopter systems. They are low cost and they can easily be modified for different configurations. Quadcopters also have vertical take-off and landing capabilities. In addition, they can be held stationary

in the air. Due to these properties, quadcopters are mostly used for military purposes, tactical surveillance or even in agriculture to increase the efficiency of cultivation.

In order to increase capabilities of quadcopters, robotic arms in other words, flying hands are assembled [62]. So, they can manipulate and carry any type of a payload in its physical limits. The most of the focus on these systems is the mathematical model and the control of the full system. Quadcopter systems have 4 rotors and the various numbers of RC servos depending on the degree of freedom of the robotic arm that is attached. The links of the robotic arm may be considered to be controlled by the RC servo. However, the quadcopter body has 6-DOF in despite of 4 rotors. There is a 2-DOF deficiency which defines this particular vehicle to have the underactuated nature. Representation of the quadcopter body is expressed by 3 Cartesian coordinates and 3 unique sequences of Euler angles. So, 6 states to be controlled by 4 rotors which are the quadcopter's control inputs. 2 state deficiency establishes several problems on stabilization and desired trajectory tracking. A hierarchical controlling architecture is mostly taken into an account. For this matter of fact, the 2 linear position and 2 angular position are coupled with the quadcopter body. The intermediate reference inputs are established for taking advantage of the underactuation. For example, the x-y position controller is coupled with the intermediate reference Euler pitch and roll angles, respectively.

In this thesis, a quadcopter system consists of a quadcopter body, a 2-DOF robotic arm mounted at the bottom and an object is held by the end-effector of the robotic arm is studied. The main focus of this thesis study is the trajectory planning-tracking of the quadcopter system for precise target engagement.

Briefly, the mathematical model of the quadcopter system is derived and controlled for the idea of precise target engagement. In this thesis, the developed control algorithms with the system specific applications are implemented in MATLAB/SIMULINK for understanding the system details while the quadcopter system on duty.

1.2. Aim of the Thesis Study

In this thesis, it is aimed to throw a ball by using a quadcopter with a 2-DOF serial manipulator. Adjustable trajectory generation from the steady condition to an aggressive state of precise throw is covered for minimum system effort. At first, basic decoupled controllers are used in the simulation and tested in a real experimental setup. Then, infinite horizon linear quadratic regulator controller is developed for reaching the required engagement states of the quadcopter system. Controller performance limits are tested in the simulation based experiments.

1.3. Thesis Outline

The thesis consists of six chapters. The first chapter is prepared for giving the general idea of the dissertation.

Chapter 2 details the literature related to the quadcopter types with their manipulators. Object throwing systems are also discussed for understanding different approaches. Mathematical modelling of the quadcopter system in the literature are analyzed. Developed control strategies for full system command tracking purposes are investigated. Experimental setups and hardware that are used in the quadcopter system are observed. Achieved performance index of the system hardware is observed. Trajectory generation for the quadcopter system or an object itself is also covered. Different types of scenario specific trajectory generation approaches are detailed accordingly.

Chapter 3 focuses on kinematic and dynamic model of the quadcopter system. Kinematic relations are obtained from the mechanism of the quadcopter system. Then, Denavith-Hartenberg Parameters are found for the 2-DOF robotic arm. The system depended rotation matrices are defined for each member. Then, the equation of motion of the quadcopter system is derived by using Lagrange-d'Alembert's Principle. Considering the decentralized approach, the quadcopter body and the robotic arm are controlled as two independent entities. After that, motor and sensor subsystems are developed to have more realistic modelling approach.

Chapter 4 is concentrated on throwing an object of a fixed target position. Engagement states of the quadcopter system are found by the forward kinematics algorithm. Minimum system effort trajectory optimization is studied from the initial state to engagement state of the quadcopter system. Then, LQR controller is developed for tracing the optimized trajectory.

Chapter 5 details simulation based results. Controller algorithms and the subsystems are implemented in the simulation. Ball throwing scenarios are tested on the simulation environment. The precise target engagement is analyzed by considering the cascaded PID controller and the LQR controller. Energy analysis is also performed. The results are discussed.

Chapter 6 presents the details of the hardware used in the real physical system. Controller cards and actuators are detailed. Cascaded PID controller algorithm is embedded to the controller card. Flight controller algorithm with the calibration steps are detailed. The data processing frequency of the sub-elements of the system's hardware is mentioned. Moment of inertia identification is experimentally found. Robotic arm details are presented. Experiments are performed.

Chapter 7 discuss and summarizes the achievements of the thesis study. Rough information is also given for the future work studies.

CHAPTER 2

LITERATURE REVIEW

2.1. Quadcopter with Manipulator Systems and Throwing an Object

Attention on rotary wings UAVs is increasing both in research and industrial purposes in the last decade. One of the most important subclass of the rotary wings UAVs is the quadcopters. Since the quadcopters have unique abilities such as high maneuverability, portable size and ease in the control. The most significant applications are emergency, search and rescue, military purposes like homeland security and search-and-destroy. In addition, they have the potential for earth sciences where they can give an opportunity in macroscale studies such as: climate changes, glacier dynamics and volcanic activities.

Dexterity of the quadcopters is increased by the assembly of the flying hands, in other words the manipulators. Quadcopters with manipulator systems can be defined as UAMs [62]. Four main elements constitute the UAMs. The first element is the base of the UAM which is the quadcopter body. The second element is the robotic arm configuration. The third element is the gripper of the robotic arm. The last element is sensors mounted on the system in order to enhance their dexterity. Combinations of these 4 main elements form the different types of quadcopter systems and their applications. Each element of the system is detailed according to the studies.

Aerodynamic effects on quadcopter flight regimes beyond hover position is studied [1]. Two main aerodynamic assumptions of the literature [61] are addressed as blade flapping and thrust force variation in the translational flight of the quadcopter. Both approaches are modelled. Generated thrust force and required power input for specific translational velocity is presented by considering the angle of attack of the quadcopter.

Equation of motion of the quadcopter is derived by considering the direction of the rotors. Specified unit of rotors are the control inputs for guiding the quadcopter to a required linear or angular position [2]. The control inputs determine the required moment to tilt the quadcopter while the total thrust force is separated out to each unit of rotor for the moment requirements.

Trajectory generation-tracking of this particular type of vehicle is covered by various types of control algorithms which are studied to overcome the underactuated nature of the quadcopter system. This problematic nature is constructed by the equation of motion of the quadcopter. Then, non-linear dynamics of the quadcopter system is controlled by simplified and linearized dynamics of quadcopter on the 3-D space [3]. Controllers are designed according to four main channels which are X,Y,Z and Euler yaw angle. These channels are investigated for their equilibrium points.

There are different types of manipulators used according to the task specific applications. In studies [4],[5], payloads are suspended via cable. The reason is that, the transportation requirement is the minimum swing and oscillation during the flight. Another type of the manipulation is achieved via the robotic arm configurations mounted to the bottom of the quadcopter. Two types of robotic arm is used in the quadcopter systems. One of them is the parallel manipulator which is preferred for maximizing the load capacity. Parallel manipulators are better in rejecting the disturbance while take-off and landing mode [6]. The other type is the serial robotic arm which can carry relatively lighter payloads in the stability margins [7],[8],[9]. An important distinction of the serial manipulators is the number of DOF. Even a 7-DOF serial manipulator is tested on the real physical system [10].

Capability of the quadcopter system is increased by the implementation of the various types of sensors. Quadcopter's altitude localization is obtained from a distance sensor mounted on the bottom, in which the direction of the sensor is looking downwards. There are also indoor motion tracking hardware for quadcopter localization [19].

Considering the throwing idea, literature is limited.

Throwing is covered in some specific studies. In a study, golf-swing motion is emulated from human behavior [11]. Industrial type of robots are also used for investigating the idea of throwing an object [32], [33], [34].

In this thesis, a 2-DOF serial type of robotic arm with a gripper is selected for holding and throwing purposes of the quadcopter system. Thus, the literature is limited to the throwing an object from a quadcopter system.

2.2. Literature on Contributions

In the literature, there are many different types of approaches which target the modelling and controlling the dynamics of the quadcopter system. In this part, studies based on quadcopters, UAM, throwing an object and trajectory optimization of UAMs topics are investigated in the literature.

2.2.1. Studies Based on Quadcopters

Raffo et. al [20], non-linear robust control strategy is developed for solving the full system states of reference input tracking problem due to the underactuated nature of quadcopters. Dynamic linearization technique is used for the system feedback information. Control strategy is generated on the idealized simulation environment.

Bellen et. al [12] concentrated on quadcopter stabilization with an unknown external inputs, that are the source of the unknown forces and moments. PID pose and PI wrench controller combination is designed in order to recover the unknown effects of interactions with environment. Scenarios are tested on the experimental setup. After this study, Fumagalli et. al [19], proposed a modified controller gain solution for the unknown disturbances and interactions of the quadcopter with the environment. Controller gains are tuned online for the unknown external force inputs. The system is also tested in the experimental setup. OptiTrack position tracking system is used for localization. Flight controller algorithm is embedded to Atom board.

Later, Ruggiero et. al [13] modeled the quadcopter by using the Newton-Euler equations. Hovering and path tracking tasks are implemented by means of an

unpredictable aerodynamic forces and external disturbances. These disturbances are included in the equation of motion of the quadcopter. External forces are estimated by using a 2nd order transfer function of the nominal ideal force inputs. Impedance controller is used for full system stabilization and command tracking. Algorithm is tested in the experimental setup. PD+Compensator, PD, PID configuration of impedance control is compared. ATOM CPU with UBUNTU Operating System is used with the data processing speed of 100Hz. Hardware communication is dealt with XBEE platform. Air ventilator disturbance is applied during the flight as the aerodynamic disturbance. In another experiment, position tracking performance is tested for the existence of an obstacle mounted on quadcopter. The weight of the obstacle is 110g. Similar scenario but different hardware is tested by Ruggiero et. al [15]. 150g of an obstacle mounted on quadcopter. Communication is covered with WiFi. The position feedback information is collected by OptiTrack motion-capture system.

Then, Tomic et. al [14] carried out external wrench estimation of tactile environment mapping. Environmental observers such as exteroceptive sensors are used for obstacle avoidance. External wrench is calculated algebraically by momentum based and acceleration based estimation, higher order terms of the transcription is filtered due to the noise ingredients of the IMU. Quadcopter is controlled in cascaded structure until reaching the interaction state. Admittance and impedance controller is implemented for the contact wrench. The system is tested in the experimental setup. Position loop is controlled at a frequency of 60 Hz and attitude loop is controlled at a frequency of 1 KHz.

2.2.2. Developments for Quadcopters with Their Flying Hands

Quadcopter with 2-DOF robotic arm is studied by Kim et. al [16]. Dynamics of the combined system is derived by d'Alambert's Lagrange principle. Adaptive sliding mode controller is designed for the centralized approach. The desired states are obtained by inverse kinematics. The system is tested in the experimental setup. An

object weights up to 450 g is successfully carried. Dynamixel MX-28 servomotors are used for the robotic arms. Servo angular position and rates are also measurable accordingly. Computation speed of the controller card is achieved up to 100 Hz. Then, Lee et. al [17] continued to UAV applications with a hexacopter configuration. Combined system is modeled in the same manner. Specifically an unknown mass is mounted to the manipulator, mass of the object is estimated during the flight by using parameter estimation. An augmented passivity based controller is designed for reference input tracking. In addition, unknown mass is estimated by using an adaptive sliding mode controller. Proposed and representative controller architecture are compared based on simulation and experimental results. 220 grams of an object is analyzed for the flight experiment. Same RC servos in [16] are used. Vicon indoor GPS with 100 Hz, XBEE communication for quadcopter reference inputs with 40 Hz and 1 KHz of controller card data processing speed is achieved. Robotic arm's control commands are sent externally via Bluetooth at 50 Hz. After that, the study is followed by Kim et al. [18]. Hexacopter with 3-DOF robotic arm is tested. Hexacopter system is guided by image-based visual servoing system to manipulate the end effector.

2.2.3. Concentration on Throwing an Object

Suzuki et. al [11] is concentrated on the motion of the swing of the Golf bar. The mathematical model of the Golf bar is derived by using the d'Alembert's Lagrange principle. The study mainly covers the swing velocity, angle and wrist angle with the interaction time. The motion and required torque input are investigated for the system boundaries. In another study, Frank et. al [21] developed a robotic structure that, robotic arm with its rotary axis to turn the robot like a human body and second rotary axis to throw objects like human with his arm in 3-D space. DH parameters of the structure is obtained. The position of the release point is found by forward kinematics. Trajectory of the ball is generated iteratively for the target engagement. Ideal air and non-ideal air subsystems are compared for the engagement performances. Mori et. al [22] studied on throwing 1-DOF disk while controlling the system variables. Three kinematic parameters of the system are set to be as the premises which are

translational-angular velocity and orientation. Throwing motion is modeled by using underactuated contact dynamics of the disk on the arm rotated by torque input. Joint torque generation is performed by iterating the system variables of the achieved linear velocity, the orientation and the angular velocity of the arm. Radial basis function, preliminary global search and simulated annealing methods are used for solution space analysis. System is both simulated and implemented in the physical system.

2.2.4. Studies Based on Trajectory Generation and Tracking for Quadcopters

In the study by Ritz et. al [23], quadcopters which are cooperatively catching and throwing the ball by using a net. Quadcopters are located with respect to the position of the ball. The radius of the net is also arranged in respect of quadcopters' positioning. Throwing trajectory is optimized for the minimization of the acceleration of the quadcopter maneuver with fixed duration. Catching trajectory is optimized for the net position or the net radius. Trajectories are traced by the finite horizon time varying LQR controller. The system is tested in the experimental set up. 6 grams of ping pong ball and 120 grams of net is used for the scenario. Infrared position measurement system and attitude measurement unit is used both with a feedback frequency of 200 Hz. Also, 67 Hz of command tracking frequency is achieved by quadcopters simultaneously.

In another study, Zhang et. al [24] offered bioinspired trajectory generation method for quadcopter perching from the hover or transition state. Perching is considered as the final state to be reached for the target location. Natural motion patterns of an animal is emulated for the trajectory generation. Quadcopter is guided to the target by Tau theory. Zero velocity and acceleration near contact is generated behind the theory.

Spedicato et. al [25] presented minimum time trajectories for quadrotors in environmental constraints. Trajectories are generated by considering the existence of obstacles on the path. Trajectory generation problem computed online. The projection operator newton method is performed with a barrier function approach for the trajectory optimization. Reference commands of the trajectory optimization algorithm

is controlled by regulation controller. Regulation controller calculates the thrust and moment required to be traced by the quadcopter. Controller output is fed to the motors virtually while neglecting the aerodynamic effects on the vehicle. Algorithm is tested in the experimental setup.

Foehn et. al [26] studied to the concept of quadcopter with cable suspended payloads. Fast trajectory optimization algorithm is proposed for the quadcopter system with a passive links. Mathematical program with complementarity constraints technique by sequential quadratic programming solver is used for that purpose. System is modelled for two modes. The first one, the cable transfers the forces between quadcopter and the payload. The second one, payload is on free-fall. Position on desired angular position and inner angular rate controller is used for trajectory tracking. The system is tested in the experimental setup. Scenarios performed in the experiments are basic waypoint tracking, obstacle avoidance and payload throwing. The PX4 controller card with its IMU is used for quadcopter system. Optitrack motion capture system with 200 Hz is achieved. Optimized trajectory reference commands are sent from a base computer via ROS platform.

Faust et. al [28] focused on the quadcopter system, which is completing the multi-waypoint scenario generation in the cluttered environment. Unknown payload is suspended under the quadcopter. Swing free trajectory is generated by machine learning approach. Approximate value iteration reinforcement learning algorithm is used for that purpose. Study is both tested in simulation and experimental setup.

Oleynikova et. al [29] implemented an online continuous time trajectory optimization for quadcopter. Quadcopter flight environment is designed to be as an unknown parameter. According to the obstacles on the environment, trajectory is rescheduled locally. Cost function for trajectory generation consists of the higher order derivative terms of the dynamics of the quadcopter. Unconstrained quadratic programming solver is used for minimization of the cost function. Algorithm and scenario generation is both tested in the simulation and experimental setup. Visual inertial stereo hardware

with 20 Hz of frequency is used for the state estimation and perception. According to the performance of the controller card, full map is updated at 5 Hz and replanning is achieved by 4 Hz. Closed loop online trajectory generation performance limit can go up to 25 Hz of computation frequency.

2.3. Research Objectives and Summary of the Literature Survey

In the thesis study, it assumed that, the target might be located in a harsh environment that may include many obstacles, while the path that the quadcopter system is travelling can be constrained. This includes control allocation behind the quadcopter system's dynamics and motion planning.

From 1-DOF [65] to 7-DOF [10] robotic arms are developed for the UAMs in the literature. In the thesis study, selection of the manipulator type for precise and measurable throwing states is decided as a 2-DOF serial robotic manipulator. Degree of freedom of the manipulator is selected for simple but an exact throw [7].

In the literature, there are two main system control approaches observed. The first one is to control the quadcopter and the robotic arm as the paired entity. This is the centralized approach [17]. Euler-Lagrangian formalism is mostly preferred for mathematically modelling the system. The second one is to control the robotic arm and the quadcopter as two different parts [31]. This is the decentralized approach [62]. Decentralized quadcopter systems are mostly modelled by Newton-Euler approach. In the thesis, two controller types are implemented for Euler-Lagrangian formalism.

Trajectory optimization of the quadcopters is performed for different achievements in the literature. Optimizing an achievement which is defined as the cost function minimization of the performance criteria [21]. Different types of scenario specific trajectory optimization studies are investigated. Optimization problem is concentrated for two common operational needs in the literature. The first one is time minimization of time while obtaining the path. This concept is analyzed by mostly for the system which is being pushed to its limit [25], [26]. The second one is the minimization of

the energy for the trajectory generation [35], [36]. However, second approach is studied for increasing the flight time and the possibility of more than one engagement.

2.3.1. Contribution of the Thesis

This is the first implementation of the quadcopter system involved in a precise target engagement in three dimensional space. The contribution of this study consists of threefold behind the combination of the unique methodology: Firstly, it is presented a quadcopter system composed of a quadcopter and a 2-DOF serial robotic manipulator which is holding an object at the end-effector. The second one is to find the engagement states according to the mission parameters. Forward kinematics algorithm is implemented to compute the engagement states according to the physical limits of the vehicle such as the joint and the body limits. Main purpose in this layer of the methodology is to satisfy the precise target engagement by the system when considering the engagement states. It is guaranteed that the target is hit by the thrown object if the engagement states are achieved. The third layer of the methodology is to generate the optimized trajectory of the system, while satisfying the engagement states in the middle of the operation. In this layer, the operation consists of take-off, throwing, stabilization and safe landing. The trajectory of the system is calculated for the minimum control effort. Path is constrained for the optimal states to be traced by the system at all samples. Minimum control effort optimal trajectory is obtained by the non-linear programming solver and the trajectory is traced by using the infinite horizon LQR controller. In the framework, feasibility of the optimal trajectory with respect to both quadcopter system dynamics and the control inputs (rotors and the RC servos of the robotic arm) bound is guaranteed.

CHAPTER 3

MATHEMATICAL MODELLING AND CONTROLLER DESIGN

3.1. Kinematics of the Quadcopter System

System is sketched in Figure 3.1. Kinematics of the system is derived according to the Denavit-Hartenberg Parameters of the robotic arm [38]. Then, transformation matrices between each frame are defined for the quadcopter system. Position and velocity vector of the system members are derived and Jacobian matrices are obtained. Generalized coordinates and velocities are defined for kinematic modelling of the system. Some assumptions are made during kinematic derivations. Quadcopter and links of robotic arm are assumed to be rigid. The base of the robotic arm is assumed to be mounted to the center of gravity of quadcopter. Mechanical offsets for the system parts are neglected. X-axis of the base of the robotic arm is oriented coaxial with the x-axis of the quadcopter body frame.

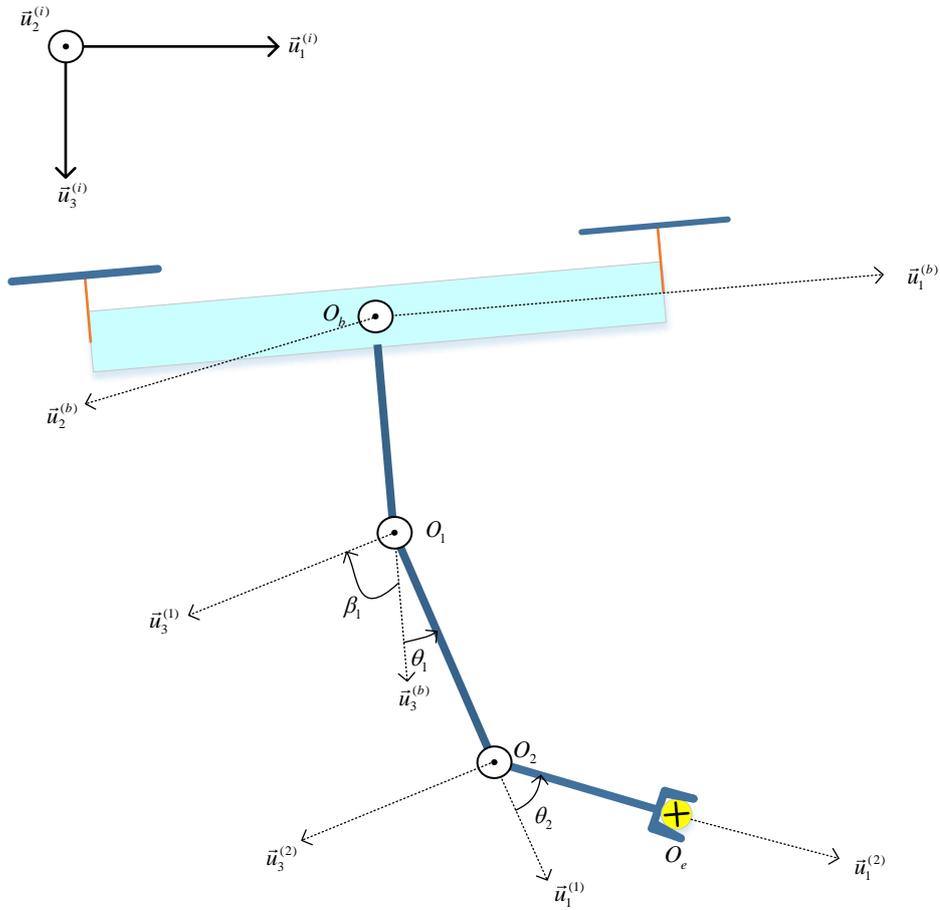


Figure 3.1 Kinematics of the Quadcopter System

3.1.1. Denavit-Hartenberg Parameters

Denavit-Hartenberg parameters are obtained for the robotic manipulator of the system [38].

3.1.1.1. Twist Angles

Angle between $\vec{u}_3^{(b)}$ and $\vec{u}_3^{(1)}$ about $\vec{u}_1^{(b)}$ is defined as β_1 .

$$\beta_1 = \frac{\pi}{2}$$

Angle between $\vec{u}_3^{(1)}$ and $\vec{u}_3^{(2)}$ about $\vec{u}_1^{(1)}$ is defined as β_2 .

$$\beta_2 = 0$$

3.1.1.2. Joint Angles

The angle between $\vec{u}_1^{(b)}$ and $\vec{u}_1^{(1)}$ about $\vec{u}_3^{(1)}$ is found as $\theta_1 + \frac{3\pi}{2}$.

The angle between $\vec{u}_1^{(1)}$ and $\vec{u}_1^{(2)}$ about $\vec{u}_3^{(2)}$ is found as θ_2 .

3.1.1.3. Offsets

The offset from O_b to O_1 is defined as b_0 .

3.1.1.4. Link Lengths

The link length from O_1 to O_2 is defined as b_1 .

The link length from O_2 to O_e is defined as b_2 .

Denavit-Hartenberg Parameters for the robotic arm of the quadcopter is formed in Table 3.1.

Table 3.1 DH Parameters of the Robotic Arm

	$k = 1$	$k = 2$
β_k	$-\frac{\pi}{2}$	0
θ_k	$\theta_1 + \frac{3\pi}{2}$	θ_2

3.1.2. Transformation Matrices

Transformation matrices are used to find the corresponding coordinate system from a specific coordinate system. A basic rotation matrix is defined in terms of basis vector direction and the rotation angle. Rodrigues formula which is obtained from Taylor series expansions is used [37].

Transformation matrix between quadcopter body-fixed reference frame to inertial-fixed frame is defined for Euler roll, pitch and yaw angles. Basic rotation matrices of each in exponential and matrix form is expressed as:

$$\begin{aligned}
\hat{R}_1(\phi) &= e^{\tilde{u}_1\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
\hat{R}_2(\theta) &= e^{\tilde{u}_2\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \\
\hat{R}_3(\psi) &= e^{\tilde{u}_3\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
\hat{C}^{(i,b)} &= \hat{R}_3(\psi)\hat{R}_2(\theta)\hat{R}_1(\phi) \tag{3.1}
\end{aligned}$$

Transformation matrix expressed in Eq (3.1) is expanded in Eq (3.2), “c” stands for cosine term and “s” stands for sine term.

$$\hat{C}^{(i,b)} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3.2}$$

Then, the transformation matrix between link-1 fixed frame and body-fixed reference frame in exponential and matrix form is found by using twist and joint angle [38] as:

$$\hat{C}^{(b,1)} = e^{\tilde{u}_1\beta_1} e^{\tilde{u}_3(\theta_1 + \frac{3\pi}{2})} \tag{3.3}$$

$$\hat{C}^{(b,1)} = e^{\tilde{u}_1(-\frac{\pi}{2})} e^{\tilde{u}_3(\theta_1 + \frac{3\pi}{2})} = e^{\tilde{u}_2\theta_1} e^{\tilde{u}_1(-\frac{\pi}{2})} e^{\tilde{u}_3(\frac{3\pi}{2})} \tag{3.4}$$

$$\hat{C}^{(b,1)} = \begin{bmatrix} s\theta_1 & c\theta_1 & 0 \\ 0 & 0 & 1 \\ c\theta_1 & -s\theta_1 & 0 \end{bmatrix} \tag{3.5}$$

The transformation matrix between link-2 fixed frame and body-fixed reference frame in exponential and matrix form is found by using twist and joint angle [38] as:

$$\hat{C}^{(1,2)} = e^{\tilde{u}_1\beta_2} e^{\tilde{u}_3\theta_2} = e^{\tilde{u}_3\theta_2} \tag{3.6}$$

$$\hat{C}^{(b,2)} = \hat{C}^{(b,1)}\hat{C}^{(1,2)} = e^{\tilde{u}_2\theta_1}e^{\tilde{u}_1(-\frac{\pi}{2})}e^{\tilde{u}_3(\frac{3\pi}{2})}e^{\tilde{u}_3\theta_2} \quad (3.7)$$

$$\hat{C}^{(b,2)} = e^{\tilde{u}_2(\theta_1+\theta_2+\frac{3\pi}{2})}e^{\tilde{u}_1(-\frac{\pi}{2})} \quad (3.8)$$

$$\hat{C}^{(b,2)} = \begin{bmatrix} s(\theta_1 + \theta_2) & c(\theta_1 + \theta_2) & 0 \\ 0 & 0 & 1 \\ c(\theta_1 + \theta_2) & -s(\theta_1 + \theta_2) & 0 \end{bmatrix} \quad (3.9)$$

3.1.3. Position Analysis of the System Model

Basic column matrices which are used in position and further velocity analysis [37].

They are shown as:

$$\bar{u}_1 = [1 \ 0 \ 0]^T$$

$$\bar{u}_2 = [0 \ 1 \ 0]^T$$

$$\bar{u}_3 = [0 \ 0 \ 1]^T$$

Euler roll, pitch and yaw angles of quadcopter body are defined in a compact form as:

$$\bar{\Gamma} = [\phi \ \theta \ \psi]$$

Joint angles of the robotic arm are also defined as:

$$\bar{\Omega} = [\theta_1 \ \theta_2]^T$$

Linear position of the quadcopter in vector representation is expressed as:

$$\vec{p}_q = \overrightarrow{O_t O_b}$$

Then, linear position vector of the quadcopter is resolved in the inertial-fixed reference frame and defined for position variables as:

$$\vec{p}_q^{(i)} = [x \ y \ z]^T$$

The position of link-1 with respect to body-fixed reference frame's base point is defined for the center of mass of the link and expressed as:

$$\vec{p}_{1/q} = b_0 \vec{u}_3^{(b)} + \frac{b_1}{2} \vec{u}_1^{(1)} \quad (3.10)$$

Then, the linear position vector is resolved in body-fixed reference frame,

$$\bar{p}_{1/q}^{(b)} = b_0 \bar{u}_3^{(b/b)} + \frac{b_1}{2} \hat{C}^{(b,1)} \bar{u}_1^{(1/1)} \quad (3.11)$$

$$\bar{p}_{1/q}^{(b)} = \begin{bmatrix} \frac{b_1}{2} s\theta_1 \\ 0 \\ b_0 + \frac{b_1}{2} c\theta_1 \end{bmatrix} \quad (3.12)$$

Linear position of link-1 with respect to the base point in the inertial-fixed reference frame is found as:

$$\bar{p}_1^{(i)} = \bar{p}_q^{(i)} + \hat{C}^{(i,b)} \bar{p}_{1/q}^{(b)} \quad (3.13)$$

Then, the position of link-2 with respect to body-fixed reference frame's base point is defined for the center of mass of the link and expressed as:

$$\vec{p}_{2/q} = \vec{p}_{2/1} + \vec{p}_{1/q} = b_0 \vec{u}_3^{(b)} + b_1 \vec{u}_1^{(1)} + \frac{b_2}{2} \vec{u}_1^{(2)} \quad (3.14)$$

Then, the linear position vector is resolved in body-fixed reference frame,

$$\bar{p}_{2/q}^{(b)} = b_0 \bar{u}_3^{(b/b)} + b_1 \hat{C}^{(b,1)} \bar{u}_1^{(1/1)} + \frac{b_2}{2} \hat{C}^{(b,2)} \bar{u}_1^{(2/2)} \quad (3.15)$$

$$\vec{p}_{2/q}^{(b)} = \begin{bmatrix} b_1 s \theta_1 + \frac{b_2}{2} s(\theta_1 + \theta_2) \\ 0 \\ b_0 + b_1 c \theta_1 + \frac{b_2}{2} c(\theta_1 + \theta_2) \end{bmatrix} \quad (3.16)$$

Linear position of the link-2 with respect to the base point in the inertial-fixed reference frame is found as:

$$\vec{p}_2^{(i)} = \vec{p}_q^{(i)} + \hat{C}^{(i,b)} \vec{p}_{2/q}^{(b)} \quad (3.17)$$

The linear position of end-effector with respect to body-fixed reference frame's base point is defined coincide with the center of mass of the ball and expressed as:

$$\vec{p}_{e/q} = \vec{p}_{e/2} + \vec{p}_{2/1} + \vec{p}_{1/q} = b_0 \vec{u}_3^{(b)} + b_1 \vec{u}_1^{(1)} + b_2 \vec{u}_1^{(2)} \quad (3.18)$$

Then, the linear position vector is resolved in body-fixed reference frame,

$$\vec{p}_{e/q}^{(b)} = b_0 \vec{u}_3^{(b/b)} + b_1 \hat{C}^{(b,1)} \vec{u}_1^{(1/1)} + b_2 \hat{C}^{(b,2)} \vec{u}_1^{(2/2)} \quad (3.19)$$

$$\vec{p}_{e/q}^{(b)} = \begin{bmatrix} b_1 s \theta_1 + b_2 s(\theta_1 + \theta_2) \\ 0 \\ b_0 + b_1 c \theta_1 + b_2 c(\theta_1 + \theta_2) \end{bmatrix} \quad (3.20)$$

Linear position of the end effector with respect to the base point in the inertial-fixed reference frame is found as:

$$\vec{p}_e^{(i)} = \vec{p}_q^{(i)} + \hat{C}^{(i,b)} \vec{p}_{e/q}^{(b)} \quad (3.21)$$

3.1.4. Velocity Analysis of the System Model

3.1.4.1. Linear Velocity Analysis of the System Members

The linear velocity of the system members are found by taking time derivative of the position information. Overhead dot “.” is used for expressing the derivative terms.

The linear velocity of the quadcopter resolved in the inertial-fixed reference frame as,

$$\dot{\vec{p}}_q^{(i)} = [\dot{x} \ \dot{y} \ \dot{z}]$$

The linear velocity of link-1 with respect to quadcopter body-fixed reference frame is expressed as.

$$\dot{\vec{p}}_{1/q}^{(b)} = \vec{v}_{1/q}^{(b)} = \begin{bmatrix} \frac{b_1}{2} c\theta_1 \\ 0 \\ -\frac{b_1}{2} s\theta_1 \end{bmatrix} \dot{\theta}_1 \quad (3.22)$$

Jacobian matrix of link-1 is defined in Eq (3.24) and matrices are obtained from Eq (3.22),

$$\vec{v}_{1/q}^{(b)} = \vec{v}_{11}^{(b)} \dot{\theta}_1 + \vec{v}_{12}^{(b)} \dot{\theta}_2 \quad (3.23)$$

$$\hat{J}_{1v} = \begin{bmatrix} \vec{v}_{11}^{(b)} & \vec{v}_{12}^{(b)} \end{bmatrix} \quad (3.24)$$

Obtained that,

$$\vec{v}_{11}^{(b)} = \begin{bmatrix} \frac{b_1}{2} c\theta_1 \\ 0 \\ -\frac{b_1}{2} s\theta_1 \end{bmatrix} \quad (3.25)$$

$$\vec{v}_{12}^{(b)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.26)$$

Linear velocity is redefined as,

$$\bar{v}_{1/q}^{(b)} = \dot{p}_{1/q}^{(b)} = \hat{J}_{1v} \dot{\bar{\Omega}} \quad (3.27)$$

Linear velocity of link-1 wrt. inertial reference frame is obtained as follows.

$$\dot{p}_1^{(i)} = \dot{p}_q^{(i)} + \hat{C}^{(i,b)} \bar{p}_{1/q}^{(b)} + \hat{C}^{(i,b)} \dot{p}_{1/q}^{(b)} \quad (3.28)$$

Eq (3.28) is expanded as.

$$\dot{p}_1^{(i)} = \dot{p}_q^{(i)} + \hat{C}^{(i,b)} \bar{\omega}_{q/i}^{(b)} \bar{p}_{1/q}^{(b)} + \hat{C}^{(i,b)} \hat{J}_{1v} \dot{\bar{\Omega}} \quad (3.29)$$

The same steps are followed for link-2 and procedure as follows.

$$\bar{v}_{2/q}^{(b)} = \dot{p}_{2/q}^{(b)} = \begin{bmatrix} \frac{b_2}{2} c(\theta_1 + \theta_2) + b_1 c \theta_1 \\ 0 \\ -\frac{b_2}{2} s(\theta_1 + \theta_2) - b_1 s \theta_1 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} \frac{b_2}{2} c(\theta_1 + \theta_2) \\ 0 \\ -\frac{b_2}{2} s(\theta_1 + \theta_2) \end{bmatrix} \dot{\theta}_2 \quad (3.30)$$

Then,

$$\bar{v}_{2/q}^{(b)} = \bar{v}_{21}^{(b)} \dot{\theta}_1 + \bar{v}_{22}^{(b)} \dot{\theta}_2 \quad (3.31)$$

$$\hat{J}_{2v} = \begin{bmatrix} \bar{v}_{21}^{(b)} & \bar{v}_{22}^{(b)} \end{bmatrix} \quad (3.32)$$

Obtained that,

$$\bar{v}_{21}^{(b)} = \begin{bmatrix} \frac{b_2}{2} c(\theta_1 + \theta_2) + b_1 c \theta_1 \\ 0 \\ -\frac{b_2}{2} s(\theta_1 + \theta_2) - b_1 s \theta_1 \end{bmatrix}$$

$$\bar{v}_{22}^{(b)} = \begin{bmatrix} \frac{b_2}{2} c(\theta_1 + \theta_2) \\ 0 \\ -\frac{b_2}{2} s(\theta_1 + \theta_2) \end{bmatrix}$$

Thus,

$$\bar{v}_{2/q}^{(b)} = \dot{\bar{p}}_{2/q}^{(b)} = \hat{J}_{2v} \dot{\bar{Q}} \quad (3.33)$$

Linear velocity of link-2 wrt. inertial reference frame is found as.

$$\dot{\bar{p}}_2^{(i)} = \dot{\bar{p}}_q^{(i)} + \hat{C}^{(i,b)} \bar{p}_{2/q}^{(b)} + \hat{C}^{(i,b)} \dot{\bar{p}}_{2/q}^{(b)} \quad (3.34)$$

$$\dot{\bar{p}}_2^{(i)} = \dot{\bar{p}}_q^{(i)} + \hat{C}^{(i,b)} \tilde{\omega}_{q/i}^{(b)} \bar{p}_{2/q}^{(b)} + \hat{C}^{(i,b)} \hat{J}_{2v} \dot{\bar{Q}} \quad (3.35)$$

In the final step, linear velocity of the end-effector is found as.

$$\bar{v}_{e/q}^{(b)} = \dot{\bar{p}}_{e/q}^{(b)} = \begin{bmatrix} b_2 c(\theta_1 + \theta_2) + b_1 c \theta_1 \\ 0 \\ -b_2 s(\theta_1 + \theta_2) - b_1 s \theta_1 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} b_2 c(\theta_1 + \theta_2) \\ 0 \\ -b_2 s(\theta_1 + \theta_2) \end{bmatrix} \dot{\theta}_2 \quad (3.36)$$

Then,

$$\bar{v}_{e/q}^{(b)} = \bar{v}_{e1}^{(b)} \dot{\theta}_1 + \bar{v}_{e2}^{(b)} \dot{\theta}_2 \quad (3.37)$$

$$\hat{J}_{ev} = \begin{bmatrix} \bar{v}_{e1}^{(b)} & \bar{v}_{e2}^{(b)} \end{bmatrix} \quad (3.38)$$

Obtained that,

$$\bar{v}_{e1}^{(b)} = \begin{bmatrix} b_2 c(\theta_1 + \theta_2) + b_1 c \theta_1 \\ 0 \\ -b_2 s(\theta_1 + \theta_2) - b_1 s \theta_1 \end{bmatrix}$$

$$\bar{v}_{e2}^{(b)} = \begin{bmatrix} b_2 c(\theta_1 + \theta_2) \\ 0 \\ -b_2 s(\theta_1 + \theta_2) \end{bmatrix}$$

Thus,

$$\bar{v}_{e/q}^{(b)} = \dot{\bar{p}}_{e/q}^{(b)} = \hat{J}_{ev} \dot{\bar{Q}} \quad (3.39)$$

Linear velocity of end-effector wrt. inertial reference frame is found as.

$$\dot{\bar{p}}_e^{(i)} = \dot{\bar{p}}_q^{(i)} + \hat{C}^{(i,b)} \bar{p}_{e/q}^{(b)} + \hat{C}^{(i,b)} \dot{\bar{p}}_{e/q}^{(b)} \quad (3.40)$$

$$\dot{\bar{p}}_e^{(i)} = \dot{\bar{p}}_q^{(i)} + \hat{C}^{(i,b)} \tilde{\omega}_{q/i}^{(b)} \bar{p}_{e/q}^{(b)} + \hat{C}^{(i,b)} \hat{J}_{ev} \dot{\bar{Q}} \quad (3.41)$$

3.1.4.2. Angular Velocity Analysis of the System Members

Angular velocity of the quadcopter in body-fixed reference frame is defined as.

$$\bar{\omega}_{q/i}^{(b)} = [p \ q \ r]^T \quad (3.42)$$

Then, the equality in Eq (3.42) and Euler angle rates have the following relation [37].

$$\bar{\omega}_{q/i}^{(b)} = \hat{L}\dot{\Gamma}$$

\hat{L} matrix maps the Euler rates to angular velocity of the quadcopter resolved in body-fixed reference frame and expressed as [37].

$$\hat{L} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix}$$

Where,

$$\det(\hat{L}) = \cos\theta$$

There exists singularity point for mapping matrix. Euler pitch angle should not be near “ $\mp \frac{\pi}{2}$ ” in order to find the Euler rates from the body angular velocities.

Angular velocity of the quadcopter in the inertial-fixed reference frame is found as.

$$\bar{\omega}_{q/i}^{(i)} = \hat{C}^{(i,b)}\bar{\omega}_{q/i}^{(b)} \quad (3.43)$$

Then, Eq (3.43) is expanded as.

$$\bar{\omega}_{q/i}^{(i)} = \hat{C}^{(i,b)}\hat{L}\dot{\Gamma}$$

Where, it is defined that,

$$\hat{T} = \hat{C}^{(i,b)}\hat{L}$$

“ \hat{T} ” matrix maps the Euler rates to angular velocity of the quadcopter resolved in inertial-fixed reference frame.

After that, the angular velocity of members of the robotic arm is obtained by using the form that includes transformation matrix representation to angular velocity representation expressed as [38],

$$\hat{C}^{(a,b)} = e^{\tilde{n}_1 \theta_1} e^{\tilde{n}_2 \theta_2} \dots \quad (3.44)$$

$$\bar{\omega}_{b/a}^{(a)} = \dot{\theta}_1 \bar{n}_1 + \dot{\theta}_2 e^{\tilde{n}_1 \theta_1} \bar{n}_2 \quad (3.45)$$

Angular velocity of link-1 is found by using the formulation obtained in Eq (3.44), (3.45) and Eq (3.3), shown as

$$\bar{\omega}_{1/q}^{(b)} = \dot{\theta}_1 \bar{u}_2$$

Then,

$$\bar{\omega}_{1/q}^{(b)} = \bar{\omega}_{11}^{(b)} \dot{\theta}_1 + \bar{\omega}_{12}^{(b)} \dot{\theta}_2 \quad (3.46)$$

$$\hat{J}_{1\omega} = \begin{bmatrix} \bar{\omega}_{11}^{(b)} & \bar{\omega}_{12}^{(b)} \end{bmatrix} \quad (3.47)$$

Obtained that,

$$\bar{\omega}_{11}^{(b)} = [0 \ 1 \ 0]^T$$

$$\bar{\omega}_{12}^{(b)} = [0 \ 0 \ 0]^T$$

Thus,

$$\bar{\omega}_{1/q}^{(b)} = \hat{J}_{1\omega} \dot{\bar{Q}} \quad (3.48)$$

Angular velocity of link-1 wrt. inertial reference frame is found as.

$$\bar{\omega}_1^{(i)} = \bar{\omega}_q^{(i)} + \hat{C}^{(i,b)} \bar{\omega}_{1/q}^{(b)} \quad (3.49)$$

Eq (3.49) is expanded as,

$$\bar{\omega}_1^{(i)} = \bar{\omega}_q^{(i)} + \hat{C}^{(i,b)} \hat{J}_{1\omega} \dot{\bar{Q}} \quad (3.50)$$

Then, angular velocity of link-2 is found by using the formulation obtained in Eq (3.44), (3.45) and Eq (3.8), shown as

$$\bar{\omega}_{2/q}^{(b)} = (\dot{\theta}_1 + \dot{\theta}_2) \bar{u}_2$$

Thus,

$$\bar{\omega}_{2/q}^{(b)} = \bar{\omega}_{21}^{(b)} \dot{\theta}_1 + \bar{\omega}_{22}^{(b)} \dot{\theta}_2 \quad (3.51)$$

$$\hat{J}_{2\omega} = \begin{bmatrix} \bar{\omega}_{21}^{(b)} & \bar{\omega}_{22}^{(b)} \end{bmatrix} \quad (3.52)$$

Obtained that,

$$\bar{\omega}_{21}^{(b)} = [0 \ 1 \ 0]^T$$

$$\bar{\omega}_{22}^{(b)} = [0 \ 1 \ 0]^T$$

Thus,

$$\bar{\omega}_{2/q}^{(b)} = \hat{J}_{2\omega} \dot{\bar{Q}} \quad (3.53)$$

Angular velocity of link-2 wrt. inertial reference frame is found as.

$$\bar{\omega}_2^{(i)} = \bar{\omega}_q^{(i)} + \hat{C}^{(i,b)} \bar{\omega}_{2/q}^{(b)} \quad (3.54)$$

Eq (3.54) is expanded as,

$$\bar{\omega}_2^{(i)} = \bar{\omega}_q^{(i)} + \hat{C}^{(i,b)} \hat{J}_{2\omega} \dot{\bar{Q}} \quad (3.55)$$

Similarly, angular velocity of the end effector is found by using the formulation obtained in Eq (3.44), (3.45) , Eq (3.8) and shown as

$$\bar{\omega}_{e/q}^{(b)} = (\dot{\theta}_1 + \dot{\theta}_2) \bar{u}_2$$

Then,

$$\bar{\omega}_{e/q}^{(b)} = \bar{\omega}_{e1}^{(b)} \dot{\theta}_1 + \bar{\omega}_{e2}^{(b)} \dot{\theta}_2 \quad (3.56)$$

$$\hat{J}_{e\omega} = \begin{bmatrix} \bar{\omega}_{e1}^{(b)} & \bar{\omega}_{e2}^{(b)} \end{bmatrix} \quad (3.57)$$

Obtained that,

$$\bar{\omega}_{e1}^{(b)} = [0 \ 1 \ 0]^T$$

$$\bar{\omega}_{e2}^{(b)} = [0 \ 1 \ 0]^T$$

Thus,

$$\bar{\omega}_{e/q}^{(b)} = \hat{J}_{e\omega} \dot{\bar{Q}} \quad (3.58)$$

Angular velocity of end-effector wrt. inertial reference frame is found as.

$$\bar{\omega}_e^{(i)} = \bar{\omega}_q^{(i)} + \hat{C}^{(i,b)} \bar{\omega}_{e/q}^{(b)} \quad (3.59)$$

Eq (3.40) is expanded as,

$$\bar{\omega}_e^{(i)} = \bar{\omega}_q^{(i)} + \hat{C}^{(i,b)} \hat{J}_{e\omega} \dot{\bar{Q}} \quad (3.60)$$

Required kinematic relations are obtained for the linear and the angular terms. Then, quadcopter system model is defined in terms of generalized coordinates “ \bar{q} ” and velocities “ $\dot{\bar{q}}$ “,

$$\bar{q} = [x \ y \ z \ \phi \ \theta \ \psi \ \theta_1 \ \theta_2] \quad (3.61)$$

$$\dot{\bar{q}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi} \ \dot{\theta}_1 \ \dot{\theta}_2]$$

Linear and angular velocity of the each member of the system in terms of generalized velocities are defined in the following form [38],

$$\begin{aligned} \bar{v} &= \sum_{i=1}^8 \bar{V}_i \dot{\bar{q}}_i \\ \bar{\omega} &= \sum_{j=1}^8 \bar{W}_j \dot{\bar{q}}_j \end{aligned} \quad (3.62)$$

“ \bar{V} ” and “ \bar{W} ” are defined respectively as the linear and angular velocity influence coefficients which are expressed in Eq (3.62). After that, linear and angular velocity terms are reorganized. Subscripts define the dependent members in the following expressions.

$$\dot{\vec{p}}_q^{(i)} = [\hat{I}_{3x3} \quad \hat{0}_{3x5}] \dot{\vec{q}} = \hat{V}_q \dot{\vec{q}} \quad (3.63)$$

$$\dot{\vec{\omega}}_q^{(i)} = [\hat{0}_{3x3} \quad \hat{T} \quad \hat{0}_{3x2}] \dot{\vec{q}} = \hat{W}_q \dot{\vec{q}} \quad (3.64)$$

$$\dot{\vec{p}}_1^{(i)} = [\hat{I}_{3x3} \quad -\hat{C}^{(i,b)} \hat{p}_1^{(b)} \hat{L} \quad \hat{C}^{(i,b)} \hat{f}_{1v}] \dot{\vec{q}} = \hat{V}_1 \dot{\vec{q}} \quad (3.65)$$

$$\dot{\vec{\omega}}_1^{(i)} = [\hat{0}_{3x3} \quad \hat{T} \quad \hat{C}^{(i,b)} \hat{f}_{1\omega}] \dot{\vec{q}} = \hat{W}_1 \dot{\vec{q}} \quad (3.66)$$

$$\dot{\vec{p}}_2^{(i)} = [\hat{I}_{3x3} \quad -\hat{C}^{(i,b)} \hat{p}_2^{(b)} \hat{L} \quad \hat{C}^{(i,b)} \hat{f}_{2v}] \dot{\vec{q}} = \hat{V}_2 \dot{\vec{q}} \quad (3.67)$$

$$\dot{\vec{\omega}}_2^{(i)} = [\hat{0}_{3x3} \quad \hat{T} \quad \hat{C}^{(i,b)} \hat{f}_{2\omega}] \dot{\vec{q}} = \hat{W}_2 \dot{\vec{q}} \quad (3.68)$$

3.2. Dynamics of the Quadcopter System

Equation of motion quadcopter system is derived according to the Lagrange-d’Alambert Formula expressed as,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\vec{q}}} \right) - \frac{\partial L}{\partial \vec{q}} = \bar{u} + \bar{u}_{ext} \quad (3.69)$$

$$L = K - U$$

Kinetic and potential energies of the system parts are calculated, then equation of motion of 6-DOF Quadcopter System is generated.

Total kinetic energy of the system is the combination of the energies on the members. These members are as follows: Quadcopter, link-1, link-2.

$$K_{total} = K_Q + K_{link_1} + K_{link_2} \quad (3.70)$$

“ \hat{I} “ stands for the inertia matrix and the subscripts represent the dependent member.

Each of the kinetic energy terms are calculated as follows:

$$K_Q = \frac{1}{2} \dot{\bar{p}}_q^{(i)T} m_q \dot{\bar{p}}_q^{(i)} + \frac{1}{2} \bar{\omega}_q^{(i)T} \hat{C}^{(i,b)} \hat{I}_q \hat{C}^{(b,i)} \bar{\omega}_q^{(i)} \quad (3.71)$$

$$K_{link_1} = \frac{1}{2} \dot{\bar{p}}_1^{(i)T} m_1 \dot{\bar{p}}_1^{(i)} + \frac{1}{2} \bar{\omega}_1^{(i)T} \hat{C}^{(i,1)} \hat{I}_1 \hat{C}^{(1,i)} \bar{\omega}_1^{(i)} \quad (3.72)$$

$$K_{link_2} = \frac{1}{2} \dot{\bar{p}}_2^{(i)T} m_2 \dot{\bar{p}}_2^{(i)} + \frac{1}{2} \bar{\omega}_2^{(i)T} \hat{C}^{(i,2)} \hat{I}_2 \hat{C}^{(2,i)} \bar{\omega}_2^{(i)} \quad (3.73)$$

Total potential energy of the system is the combination of members. These members are as follows: Quadcopter, link-1, link-2.

$$U_{total} = U_Q + U_{link_1} + U_{link_2} \quad (3.74)$$

Each of the potential energy terms are calculated as:

$$U_Q = m_q g \bar{u}_3^T \bar{p}_q^{(i)} \quad (3.75)$$

$$U_{link_1} = m_{link_1} g \bar{u}_3^T \bar{p}_1^{(i)} \quad (3.76)$$

$$U_{link_2} = m_{link_2} g \bar{u}_3^T \bar{p}_2^{(i)} \quad (3.77)$$

The equation of motion of the system is written in the following form [37].

$$\hat{M}(\bar{q}) \ddot{\bar{q}} + \hat{C}(\bar{q}, \dot{\bar{q}}) \dot{\bar{q}} + \hat{G}(\bar{q}) = \bar{u} + \bar{u}_{ext} \quad (3.78)$$

“ \hat{M} “ defined in Eq (3.78) is the inertia matrix of the system, that is positive definite. Inertia matrix is expressed by the following form [36],[37] and calculated by using Eq (3.63) to (3.68).

$$K = \frac{1}{2} \dot{\bar{q}}^T \hat{M}(\bar{q}) \dot{\bar{q}} \quad (3.79)$$

$$\begin{aligned}\hat{M}(\bar{q}) &= \hat{V}_q^T m_q \hat{V}_q + \hat{W}_q^T \hat{C}^{(i,b)} \hat{I}_q \hat{C}^{(b,i)} \hat{W}_q \\ &+ \sum_{j=1}^2 [\hat{V}_j^T m_j \hat{V}_j + \hat{W}_j^T \hat{C}^{(i,j)} \hat{I}_j \hat{C}^{(j,i)} \hat{W}_j]\end{aligned}\quad (3.80)$$

“ \hat{C} ” defined in Eq (3.78) includes the Centripetal, Coriolis and Gyroscopic terms of the system, that is obtained by using the inertia matrix indices [36], found in Eq (3.80).

$$c_{a,b} = \sum_{j=1}^8 \frac{1}{2} \left[\frac{\partial m_{a,b}}{\partial q_j} + \frac{\partial m_{a,j}}{\partial q_b} - \frac{\partial m_{j,b}}{\partial q_a} \right] \quad (3.81)$$

Where “ \hat{C} ” matrix is built up by the following element matching.

$$\hat{C}(a,b) = c_{a,b} \quad (3.82)$$

“ \hat{G} ” defined in Eq (3.78) is the gravity matrix of the system, that is found by partially differentiating the potential energy by the generalized coordinates of the system.

$$\hat{G}(\bar{q}) = \frac{\partial U}{\partial \bar{q}} \quad (3.83)$$

3.2.1. System's Force and Torque Units

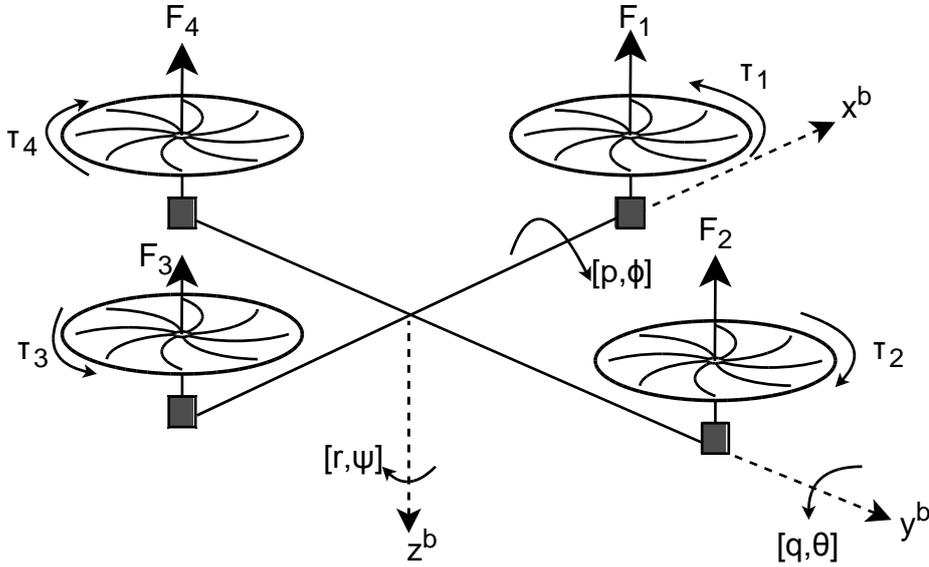


Figure 3.2 Quadcopter Detailed FBD for Actuating Forces and Torques

Some assumptions are made for the thrust force and the torque generation. Thrust force and moments on quadcopter body are assumed to be proportional to the square of the rotors' rotational speed. Moreover, induced drag and frictions are neglected. The thrust force and the torque generated by the rotors' have the following relation [46].

$$f_i = c_T \omega_i^2 \quad (3.84)$$

$$\tau_i = c_Q \omega_i^2 \quad (3.85)$$

Where $i=1,2,3$ and 4 , represents the selected rotor.

“ ω_i ” is the rotor's rotational speed. “ c_T ” is the lumped thrust coefficient of the rotor and “ c_Q ” is the lumped torque coefficient of the rotor. These coefficients are consists of the rotor geometry-profile and density of the air. They can be obtained experimentally. Aerodynamic coefficient values are taken from [39].

Then, the thrust force generated on quadcopter body fixed reference frame is expressed as,

$$\bar{f}_q^{(b)} = \begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix} \quad (3.86)$$

Then, total thrust force is found by the following expression,

$$f_z = -(f_1 + f_2 + f_3 + f_4)$$

Then, the torque generated on quadcopter body fixed reference frame is expressed as,

$$\bar{\tau}_q^{(b)} = \begin{bmatrix} \tau_{q1} \\ \tau_{q2} \\ \tau_{q3} \end{bmatrix} = \begin{bmatrix} d(f_4 - f_2) \\ d(f_1 - f_3) \\ \tau_2 - \tau_1 + \tau_4 - \tau_3 \end{bmatrix} \quad (3.87)$$

“ d ” is the distance between rotor to center of mass of the quadcopter. It is assumed to be equal for 4 rotors. Then, rotational speed can be converted into force and moment generated on the system [47].

$$\begin{bmatrix} f_z \\ \tau_{q1} \\ \tau_{q2} \\ \tau_{q3} \end{bmatrix} = \begin{bmatrix} -c_T & -c_T & -c_T & -c_T \\ 0 & -dc_T & 0 & dc_T \\ dc_T & 0 & -dc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (3.88)$$

Torque generated by the RC servo motors of the robotic arm is defined as,

$$\bar{\tau}_{12} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (3.89)$$

3.2.2. Input and Interaction Forces of the System

Virtual work principle is used to obtain the generalized forces “ \bar{u} ” on the quadcopter system. Virtual work done is derived for the generated thrust force “ \vec{f}_q ” and moment “ $\vec{\tau}_q$ ” by rotors, as follows [37].

$$\delta W = \vec{f}_q \delta(\vec{p}_q)_{O_b/O_i} + \vec{\tau}_q \delta(\vec{\Psi})_{F_b/F_i} \quad (3.90)$$

Eq (3.81) resolved in body-fixed reference frame and converted into column matrix representation as,

$$\delta W = (\bar{f}_q^{(b)})^T (\hat{C}^{(i,b)})^T \delta \bar{p}_q^{(i)} + (\bar{\tau}_q^{(b)})^T \delta \bar{\Psi}^{(b)} \quad (3.91)$$

Then, the following expression is defined in terms of Euler angles,

$$\delta \bar{\Psi}^{(b)} = \hat{L} \delta \bar{\Gamma} \quad (3.92)$$

Then, Eq (3.82) is expanded by using Eq (3.92) as follows [37],

$$\delta W = [\hat{C}^{(i,b)} \bar{f}_q^{(b)}] \delta \bar{p}_q^{(i)} + [\hat{L}^T \bar{\tau}_q^{(b)}] \delta \bar{\Gamma} \quad (3.93)$$

It is defined that, generalized forces of $\delta \bar{p}_q^{(i)}$ and $\delta \bar{\Psi}^{(b)}$ are shown respectively as \bar{Q}_p and \bar{Q}_Ψ .

Then, Eq (3.93) is reorganized as,

$$\delta W = \bar{Q}_p^T \delta \bar{p}_q^{(i)} + \bar{Q}_\Psi \cdot \delta \bar{\Gamma} \quad (3.94)$$

Virtual work defined for the robotic manipulator is defined as follows [37],

$$\delta W = \tau_1 \delta \theta_1 + \tau_2 \delta \theta_2 \quad (3.95)$$

Input force generated by the system is written by using Eq (3.95) and Eq (3.93),

$$\bar{u} = \begin{bmatrix} \hat{C}^{(i,b)} & \hat{O}_{3 \times 3} & \hat{O}_{3 \times 2} \\ \hat{O}_{3 \times 3} & (\hat{L})^T & \hat{O}_{3 \times 2} \\ \hat{O}_{2 \times 3} & \hat{O}_{2 \times 3} & \hat{I}_{2 \times 2} \end{bmatrix} \begin{bmatrix} \bar{f}_q^{(b)} \\ \bar{\tau}_q^{(b)} \\ \bar{\tau}_{12} \end{bmatrix} \quad (3.96)$$

Then, “ \bar{u} ” is 8x1 column matrix. Eq (3.96) is converted into the following expression for the further observation of the system characteristics,

$$\bar{u} = \hat{S} \bar{u}_{sys} \quad (3.97)$$

Where, the determinant of “ \hat{S} ” is calculated as,

$$\det(\hat{S}) = \cos(\theta)$$

It is defined that,

$$\theta \neq \frac{k\pi}{2}$$

Where,

$$k = \pm 1, \pm 2, \dots$$

Physical limits of this study should satisfy for the application of the virtual work principle. The range of application is established as in the region of $\left[-\frac{\pi}{3}, \frac{\pi}{3}\right]$.

External forces and moments applied to the system is defined as [41],

$$\bar{P} = [F_1 \ F_2 \ F_3 \ M_1 \ M_2 \ M_3]^T \quad (3.98)$$

External force is created by the existence of the ball on the system. Ball is assumed to be located at the end-effector.

$$F_3 = m_{ball}g$$

Then, “ \bar{u}_{ext} ” is defined in terms of generalized coordinates,

$$\bar{u}_{ext} = \hat{H}\bar{P} \quad (3.99)$$

Where, “ \hat{H} ” is the 8x6 conversion matrix.

The group of concentrated external forces defined for the generalized coordinates of $[x \ y \ z]^T$ is expressed in Eq (3.100).

$$\begin{bmatrix} u_{ext,1} \\ u_{ext,2} \\ u_{ext,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \quad (3.100)$$

The group of concentrated external forces defined for the generalized coordinates of the $[\phi \ \theta \ \psi]^T$ is expressed in Eq (3.101).

$$\begin{bmatrix} u_{ext,4} \\ u_{ext,5} \\ u_{ext,6} \end{bmatrix} = \vec{p}_{e/q} x \vec{F} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \quad (3.101)$$

$$\begin{bmatrix} u_{ext,4} \\ u_{ext,5} \\ u_{ext,6} \end{bmatrix} = skew(\vec{p}_{e/q}) \bar{F} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \quad (3.102)$$

$$\begin{bmatrix} u_{ext,4} \\ u_{ext,5} \\ u_{ext,6} \end{bmatrix} = \begin{bmatrix} 0 & -\bar{p}_{e/q}^{(b)}(3) & \bar{p}_{e/q}^{(b)}(2) \\ \bar{p}_{e/q}^{(b)}(3) & 0 & -\bar{p}_{e/q}^{(b)}(1) \\ \bar{p}_{e/q}^{(b)}(2) & \bar{p}_{e/q}^{(b)}(1) & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \quad (3.103)$$

The group of concentrated external forces defined for the generalized coordinates of the $[\theta_1 \ \theta_2]^T$ is expressed in Eq (3.104).

$$\begin{bmatrix} u_{ext,7} \\ u_{ext,8} \end{bmatrix} = \hat{f}_e^{(b)} \bar{P} \quad (3.104)$$

Where, the Jacobian matrix of the end effector “ $\hat{f}_e^{(b)}$ ” is formed by using Eq (3.38) and (3.57).

$$\hat{f}^{(b)} = [\hat{f}_{ev}^T \hat{f}_{e\omega}^T] \quad (3.105)$$

Eq (3.104) is expanded by using Eq (3.98) and Eq (3.105) as,

$$\begin{bmatrix} u_{ext,7} \\ u_{ext,8} \end{bmatrix} = \begin{bmatrix} v_{e11} & v_{e12} & v_{e13} & \omega_{e11} & \omega_{e12} & \omega_{e13} \\ v_{e21} & v_{e22} & v_{e23} & \omega_{e21} & \omega_{e22} & \omega_{e23} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} \quad (3.106)$$

Following relation obtained from Eq (3.99) is formed by using Eq (3.100),(3.103) and (3.106)

$$\bar{u}_{ext} = \begin{bmatrix} \hat{I}_{3x3} & \hat{O}_{3x3} \\ \tilde{p}_{e/q}^{(b)} & \hat{I}_{3x3} \\ \hat{f}_{ev}^T & \hat{f}_{e\omega}^T \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix}$$

3.2.3. Motor Subsystem

Non-ideal actuation model is implemented to the 6-DOF quadcopter system's simulation. Model consists of quadcopter's DC motors and the RC servos of the robotic arm.

Representation of the DC motor is written for the rotational speed of the rotors, the transfer function in the s-domain as follows,

$$G_{DC}(s) = \frac{\omega_{response}(s)}{\omega_{command}(s)} = \frac{0.98}{0.062s + 1} \quad (3.108)$$

Identified DC motor transfer function is taken from [39]. Eq (3.108) stands for the DC motor command and response transfer function. The input and output relation is constructed according to the rotational speed of the rotors.

Then, representation of the RC servo motor model in s-domain as follows,

$$G_{servo}(s) = \frac{\tau_{response}(s)}{\tau_{command}(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2} \quad (3.109)$$

Eq (3.109) stands for the RC servo motor, command and response transfer function. The input and output relation is constructed according to the torque generation. It is assumed to be as a second order low pass filter with 25 Hz of cut-off frequency and " $\frac{\sqrt{2}}{2}$ " of damping ratio [41].

3.2.4. Sensor Subsystem

System simulation model is matched with the real physical system's experimental output. To have a better system model, noise and sensor dynamics of the measurement units are implemented. These measurement units are the accelerometer and gyroscope, the subsystem is shown in Figure 3.3.

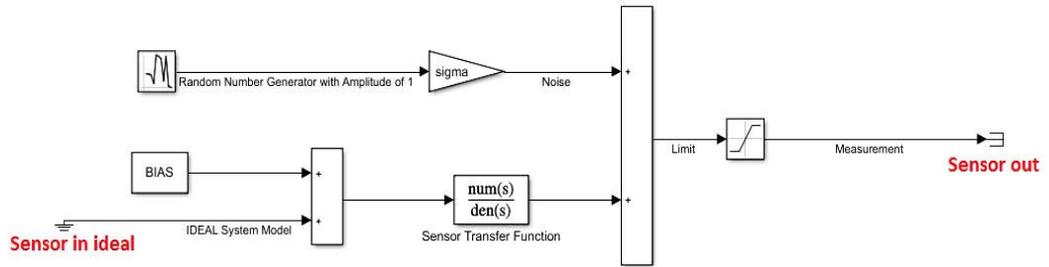


Figure 3.3 Sensor Model

Noise and the dynamics of the sensor subsystem is shown in Table 3.2 [63],

Table 3.2 System Sensor's Model

	$w_{n,sensor}$	ξ_{sensor}	σ_{noise}	Bias	Limit
Accelerometer	25 Hz	1	50 mg	50 mg	± 4 g
Gyroscope	25 Hz	1	5 deg/s	-	± 250 deg/s

Where, dynamics of the sensor is implemented as a second order transfer function [64],

$$\frac{G_{measure}(s)}{G_{ideal}(s)} = \frac{w_{n,sensor}^2}{s^2 + 2\xi_{sensor}w_{n,sensor}s + w_{n,sensor}^2}$$

The amplitude of the white noise signal is considered as the sigma value. Each sigma value of sensor input for the noise model is shown in Table 3.2. White noise is added to the input. Measurement limit is saturated according to the sensor limits. In order to simulate the sensor dynamics with the system simulation, cut-off frequencies and sigma values of the noise character of the sensors are found by trial and error of simulation generation.

3.3. Cascaded PID Controller of the Quadcopter System

In this part, cascaded PID controlling technique is developed. This technique is implemented in order to control the robotic arm with a small disturbance generated by the existence of the ball during the flight generation. Controllers of the cascaded PID technique are designed according to the linearized equation of motion of the quadcopter system. Decoupled controller channels are constructed for the quadcopter system. Linear x and y position controllers represent the outer loop of the structure. There also exists an inner loop control and two of them are coupled with the outer loop controller. For example, quadcopter should tilt in order to move in inertial x and y direction, so there is a cascaded intermediate inner angular position and inner angular rate controller structure. Coupled inner loop elements which are desired Euler roll and pitch angles. These angles are obtained from the output of the outer loop controller. The rest of the inner loop elements which are decoupled, defined as altitude, attitude and joint angles controllers. Closed loop controller structure is expressed in Figure 3.4.

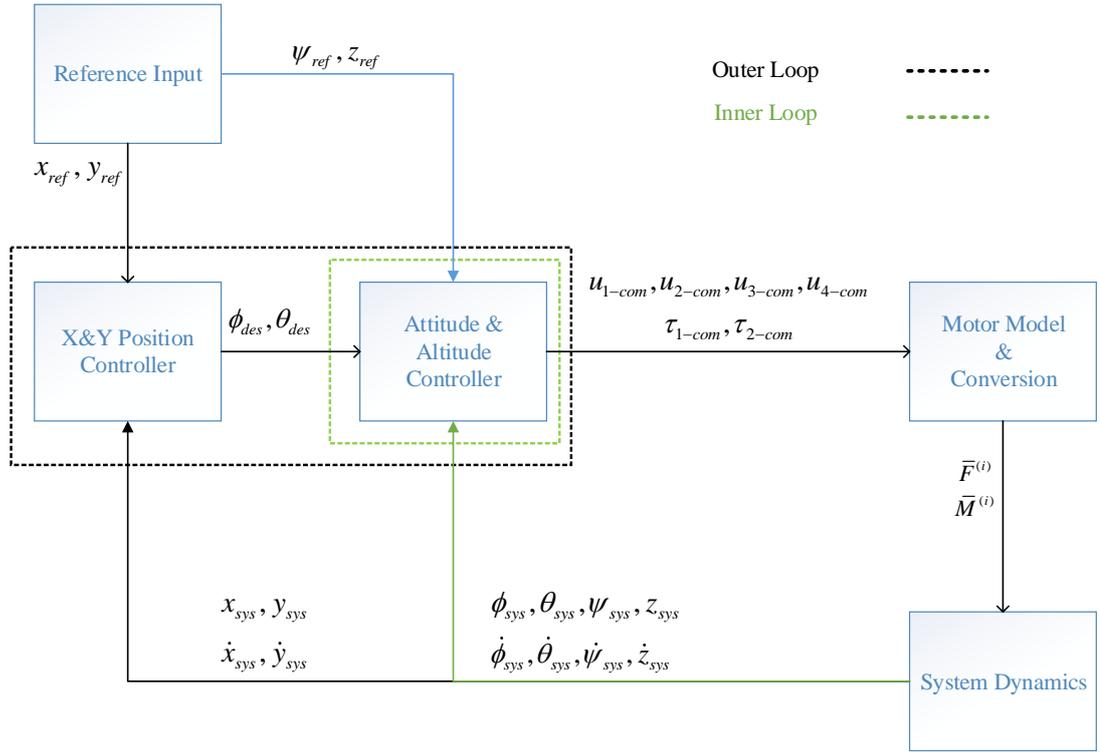


Figure 3.4 Closed Loop Controller Structure

3.3.1. Outer Loop Controller

The difference between the reference and the system position indicates the error. In the outer loop controller, the desired intermediate Euler pitch and roll angles are generated by x and y position errors. Designed controller is constructed according to Eq (3.96), equation of motion is expressed as follows.

$$\begin{aligned} u_1 &= f_z(c\phi s\theta c\psi + s\phi s\psi) \\ u_2 &= f_z(c\phi s\theta s\psi - s\phi c\psi) \end{aligned} \quad (3.110)$$

Then, Eq (3.110) is rewritten by using the small angle assumption for Euler roll and pitch angles. Euler yaw angle is assumed to be near zero for the outer loop control.

$$\begin{aligned} u_1 &= f_z \theta = m \ddot{x} \\ u_2 &= -f_z \phi = m \ddot{y} \end{aligned} \quad (3.111)$$

By considering Eq (3.111), “ m “ is the total mass of the quadcopter system. Desired position of the quadcopter system in the x-y direction is obtained by two chain integrators. The error between desired position and actual position of the quadcopter system generates desired Euler angle as follows,

$$\begin{aligned} \theta_d &= \frac{m}{f_z} \left(\ddot{x}_{required} + K_{p,x} e_x + K_{i,x} \int_0^t e_x(\tau) d\tau + K_{d,x} \dot{e}_x \right) \\ \phi_d &= -\frac{m}{f_z} \left(\ddot{y}_{required} + K_{p,y} e_y + K_{i,y} \int_0^t e_y(\tau) d\tau + K_{d,y} \dot{e}_y \right) \end{aligned} \quad (3.112)$$

Required acceleration on x and y is neglected due to noise ingredient of the system.

3.3.2. Inner Loop Control

Altitude and attitude controller of quadcopter, the angular position controller of the robotic arm is covered by the inner loop control.

3.3.2.1. Altitude controller

Equation of motion for altitude channel is also linearized by considering the Eq (3.96) as follows.

$$u_3 = m \ddot{z} = f_z + mg \quad (3.113)$$

Gravitational acceleration of the quadcopter system is considered as a constant disturbance during the flight simulation. Inertial z position of the quadcopter is controlled by the following PID structure,

$$f_z = m \left(\ddot{z}_{required} - g + K_{p,z} e_z + K_{i,z} \int_0^t e_z(\tau) d\tau + K_{d,z} \dot{e}_z \right) \quad (3.114)$$

The same approach in outer loop is also followed for the acceleration of the required inertial z position. Here, it is taken to be as zero.

3.3.2.2. Attitude Controller

Euler angular position of the quadcopter is stabilized by the reference angular position generated by the system. Angular rate controller loop is also performed to increase the performance of stabilization. Bandwidth of the attitude controller is expanded. External disturbances are eliminated by the fast internal loop. Position control is relatively slow when compared with the rate loop. This controller is constructed by the equation of motion. Attitude dynamics of the quadcopter is obtained by using Eq (3.96). Then by using Eq (3.87), the equation of motion is linearized and to be used for PID control as follows [47],

$$u_4 = I_{q,x} \ddot{\phi} = \tau_{q1} \quad (3.115)$$

$$u_5 = I_{q,y} \ddot{\theta} = \tau_{q2} \quad (3.116)$$

$$u_6 = I_{q,z} \ddot{\psi} = \tau_{q3} \quad (3.117)$$

Linearized equations are represented by means of the second order derivative of Euler angles and the inertia terms of the quadcopter. Inertial contributions of the extra equipment of the quadcopter system is not considered. Then, the following control law is rewritten for the attitude dynamics as.

$$\tau_{q1} = \{K_{p,\phi} e_\phi - \dot{\phi}\} \left(K_{p,\phi} e_\phi + K_{i,\phi} \int_0^t e_\phi(\tau) d\tau + K_{d,\phi} \dot{e}_\phi \right) \quad (3.118)$$

$$\tau_{q2} = \{K_{p,\theta}e_{\theta} - \dot{\theta}\} \left(K_{p,\dot{\theta}}e_{\dot{\theta}} + K_{i,\dot{\theta}} \int_0^t e_{\dot{\theta}}(\tau)d\tau + K_{d,\dot{\theta}}\dot{e}_{\dot{\theta}} \right) \quad (3.119)$$

$$\tau_{q3} = \{K_{p,\psi}e_{\psi} - \dot{\psi}\} \left(K_{p,\dot{\psi}}e_{\dot{\psi}} + K_{i,\dot{\psi}} \int_0^t e_{\dot{\psi}}(\tau)d\tau + K_{d,\dot{\psi}}\dot{e}_{\dot{\psi}} \right) \quad (3.120)$$

Angular position loop is controlled by P gain, the inner rate loop is stabilized by the PID controller. Figure 3.5 shows the cascaded attitude controller of the Euler roll and pitch angles.

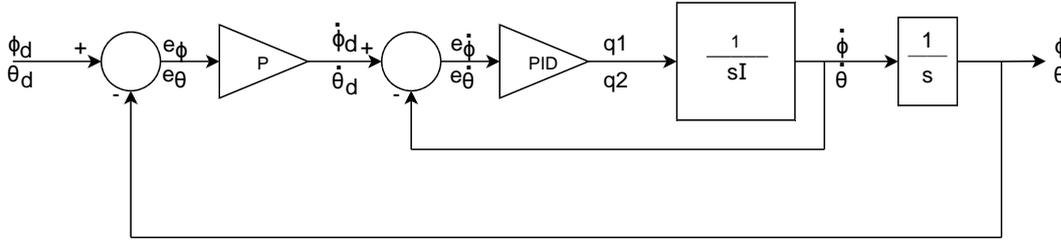


Figure 3.5 Cascaded Attitude Controller Schematic

3.3.3. Controller of the Robotic Manipulator

Equation of motion of the robotic manipulator is expressed as,

$$I_{12,y}\ddot{\theta}_1 = \tau_1 - m_{12ball}g\sin(\theta_1) \left(\frac{b_1 + b_2}{2} \right) \quad (3.121)$$

$$I_{2,y}\ddot{\theta}_2 = \tau_2 - m_{2ball}g\sin(\theta_2) \left(\frac{b_2}{2} \right) \quad (3.122)$$

In Eq (3.121) and (3.122), the inertia term is the combination of the moment of inertia of the link-1 and link-2. However, mass terms are the combination of link-1, link-2 and the ball included as the linear time varying disturbance on the system. Required torque generation by the controller is designed by feedback linearization as follows.

$$\tau_1 = \tau_{\theta_1} + m_{12ball}g\sin(\theta_1)\left(\frac{b_1 + b_2}{2}\right) \quad (3.123)$$

$$\tau_2 = \tau_{\theta_2} + m_{2ball}g\sin(\theta_2)\left(\frac{b_2}{2}\right) \quad (3.124)$$

Where, it is defined that,

$$\tau_{\theta_1} = K_{p,\theta_1}e_{\theta_1} + K_{i,\theta_1}\int_0^t e_{\theta_1}(\tau)d\tau + K_{d,\theta_1}\dot{e}_{\theta_1} \quad (3.125)$$

$$\tau_{\theta_2} = K_{p,\theta_2}e_{\theta_2} + K_{i,\theta_2}\int_0^t e_{\theta_2}(\tau)d\tau + K_{d,\theta_2}\dot{e}_{\theta_2} \quad (3.126)$$

In the torque obtained in Eq (3.123) and (3.124) is satisfied by using Eq (3.125) and (3.126).

3.3.4. Multi-Objective Optimization

In order to enhance the controller performance, controller gains are needed to be initialized for the multi-objective optimization. To do so, the initial controller gains are set to be determined by the PID tuner of MATLAB for the Cascaded PID configuration. In this way, the obtained gain set are optimized for the highly non-linear simulation environment which includes motor and sensor subsystem. It is aimed to find the controller gain set that provides the closed loop system response of minimum percentage of overshoot and settling time. ITAE is chosen to be the cost function for solving the multi objective optimization problem.

For the sake of generality, the performance indices are defined as, ITAE (Integral of the Time-Weighted Absolute Error), ITSE (Integral of the Time-Weighted Squared Error), IAE (Integral of the Absolute Value of the Error), ISE (Integral of the Squared Error), JE (Just Error)

$$JE = \sum_{k=t_0}^{t_f} e_k^2$$

$$ISE = \int_{t_0}^{t_f} e^2(t)dt$$

$$IAE = \int_{t_0}^{t_f} |e(t)|dt$$

$$ITSE = \int_{t_0}^{t_f} t e^2(t)dt$$

$$ITAE = \int_{t_0}^{t_f} t|e(t)|dt$$

Where “ t_0, t_f ” are the beginning and the final time of the analysis, respectively. A demonstration for analyzing the performance indices is made for the linearized quadcopter dynamics in the altitude channel. Initial controller gains are set by using PID Tuner of MATLAB. Pre-defined controller gains are optimized for each of the performance indices. The optimized and pre-defined PID gains are expressed in Table 3.3.

Table 3.3 Pre-defined and Optimized PID Gains for Altitude Channel

Method	P	I	D
PID Initialization	5.2871	1.2893	5.1171
JE	0.0012202	0.00011543	9995.9
ISE	5.6624	3.9677	36.595
IAE	1.75	1.4398	16.72
ITSE	4.3067	0.82039	14.077
ITAE	0.79519	0.84444	13.972

Reference input and the closed loop system response according to the each of the PID gain set are expressed in Figure 3.6.

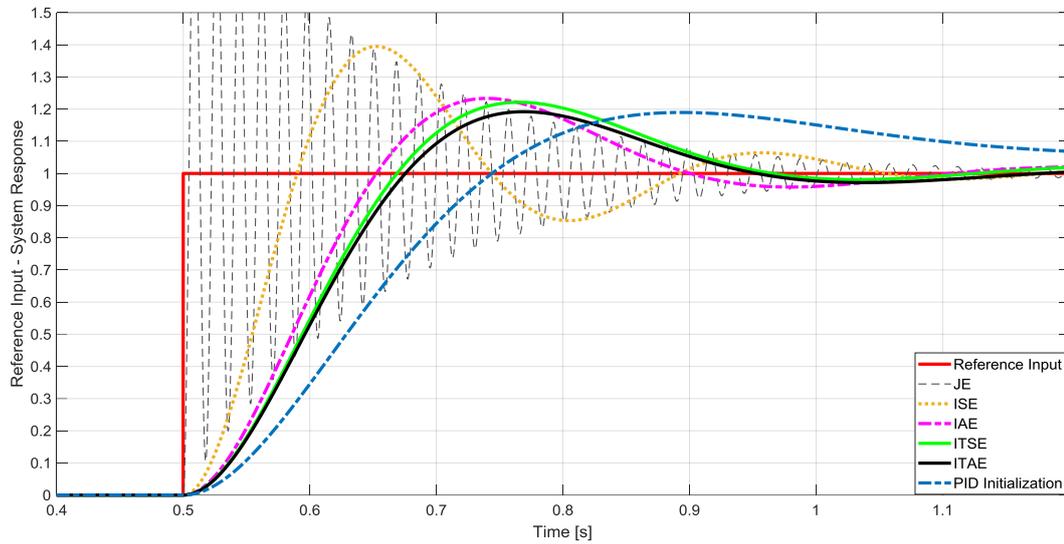


Figure 3.6 Reference Input and Closed Loop System Response for the Linearized Altitude Dynamics of the Quadcopter System

The step response characteristics for the closed loop altitude dynamics are expressed in Table 3.4.

Table 3.4 The Step Response Characteristics for the Altitude Dynamics

Method	Settling Time [s]	Rise Time [s]	Overshoot [%]	The Minimized Cost Value
PID Initialization	2.5386	0.2277	19.647	-
JE	0.62077	0.0029804	94.582	2.2104×10^{-6}
ISE	0.52926	0.060508	39.393	3.5875×10^{-9}
IAE	0.74758	0.10356	23.314	8.1414×10^{-7}
ITSE	0.94741	0.11469	22.124	1.9242×10^{-7}
ITAE	0.5978	0.11898	19.182	3.7052×10^{-6}

ITAE criterion penalizes the error with the time that persist up until the settling occurs. In general, the ITAE usually results in the most conservative controller settings which provide less overshoot and settling time [66]. By contrast, the ISE criterion provides

the most aggressive settings, while the IAE and ITSE criteria tend to produce controller settings that are between those for the ITAE and ISE criteria [66], as shown in Figure 3.6.

As expressed in Table 3.4 that, ITAE provides the minimum percentage of overshoot with the fastest settling time compared with the other performance indices. Also, ITAE has a remarkable sensitivity on the cost function in the zero-velocity and the zero-acceleration error for the second order system dynamics which has the damping ratio in the range of 0.4 to 1 [67]. Sensitivity of the ITAE compared with the ISE and the IAE is shown in Figure 3.7.

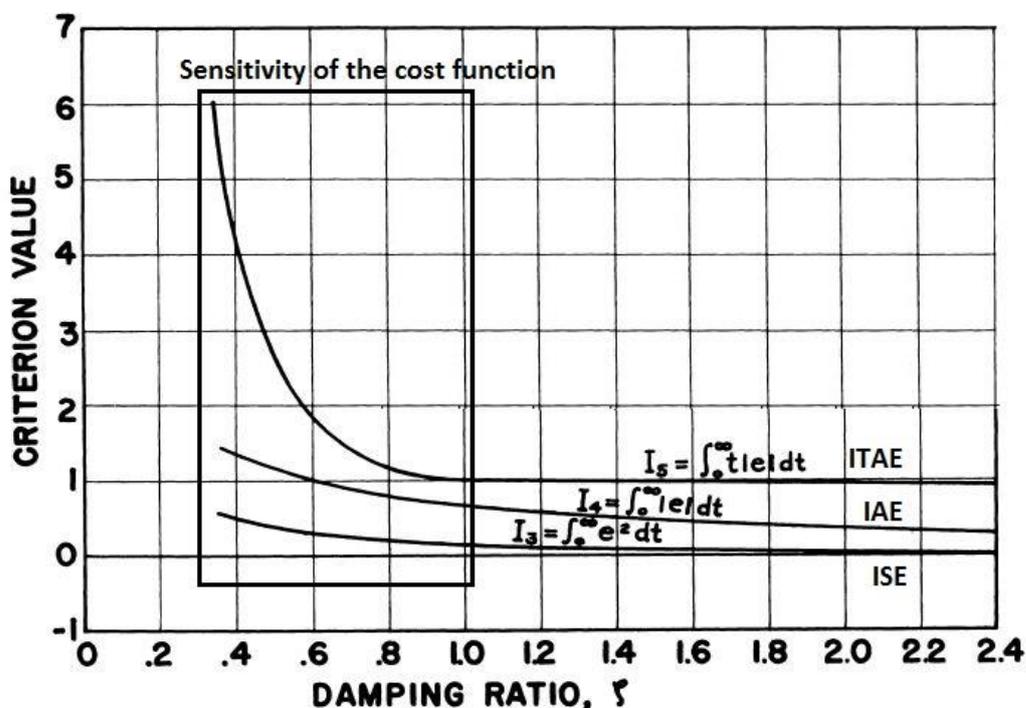


Figure 3.7 Sensitivity of the ITAE Compared with IAE and ISE for the Zero-Velocity Error in the Second Order System Dynamics [66]

It is reasonable to consider that, the minimization of ITAE may deliver an output that, the percentage of overshoot and the settling time of the system is reduced for the reference input. MATLAB Optimization Toolbox is used for the multi-objective optimization. Single objective transcription is defined as,

$$f_1 = \int_{t_0}^{t_f} t|e(t)|dt \quad (3.127)$$

In Eq (3.127), where “ t_0, t_f ” are the beginning and the final time of the single objective of the optimization, respectively. “ e ” is the error, which is calculated for the single objective in the region of the reference input and system’s predefined settling time.

The performance index of the multi-objective optimization problem - ITAE is minimized by using Nonlinear Least Square Curve Fitting Algorithm of MATLAB. The number of objectives involved in the optimization problem is expressed by the subscript “ k ”. “ N ” dimensional multi-objective optimization is implemented as,

$$\min \sum_{k=1}^N |w_k f_k|^2$$

Where, “ w_k ” is the weighting of the objective which is optimized.

Iterated gain set of the cascaded PID controller for minimum ITAE is expressed in Table 3.5.

Table 3.5 Optimized Gains of the System for Cascaded PID

	x	y	z	ϕ	θ	ψ	θ_1	θ_2
K_p	4.9158	4.9158	4.5467	2.5636	2.5636	1.0434	4680	12410
K_i	0.2432	0.2432	2.525	1.0303	1.0303	3.9229	0.2432	2543
K_d	1.8772	1.8772	1.382	0.0022	0.0022	0.0011	204	237
K_{p2}	-	-	-	6.9568	6.9568	3.0431	-	-

CHAPTER 4

TRAJECTORY PLANNING AND PRECISE TARGET ENGAGEMENT

4.1. Problem Definition

Precise target engagement depends on two mission parameters. These two parameters are the position information of the target and the throwing angle information that is needed in order not to engage with the environmental constraints. These environmental constraints may be the blockages for protecting the target from a wide-open engagement. Because of that reason, the target location and throwing angle are assumed to be known premises. According to the relative position between the end effector - the target location and the throwing angle, engagement state of the quadcopter system is calculated by forward kinematics. Then, trajectory of the quadcopter is obtained for the initial and the final states of the quadcopter system in 2-D.

Trajectory optimization is performed to minimize the input forces to the quadcopter system, which are the thrust forces of the rotors and the torque inputs of the links of the robotic arm. Trajectory of the quadcopter is tracked by the infinite horizon LQR controller. Variation on the final velocity vector of the ball expresses the controller performance.

Trajectory of the quadcopter system is generated with different scenarios and trajectories in 2-D for the engagement with the target. The trajectory of the quadcopter from the initial state to the final throwing state will be followed by the LQR controller. As a result, target engagement is performed at the end of trajectory planning and tracking stages.

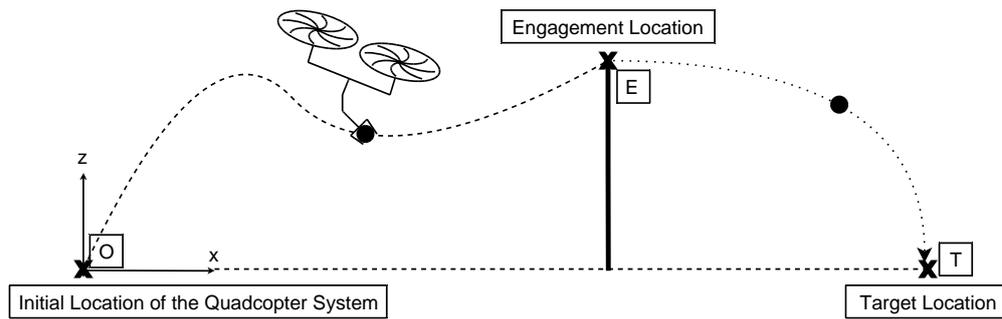


Figure 4.1 Trajectory Planning and Precise Target Engagement

As presented in Figure 4.1, Point O is the initial position of the quadcopter system, Point E is the engagement location and Point T is the fixed target position. Chapter 4.2 gives the details about the Trajectory Analysis of the Ball, where it is indicated in the figure as point E to T. Chapter 4.3 presents the Trajectory Generation of the Quadcopter System, where it is expressed as point O to E. In the final part of this chapter, LQR controller is designed to track the optimized trajectory from point O to point E.

4.2. Trajectory Analysis of the Ball

Quadcopter system consists of a quadcopter body, a 2-DOF robotic arm and a ball held by the end-effector of the robotic arm. Trajectory analysis is performed by considering the quadcopter system kinematics in 2-D planar space. This approach is made for the sake of simplicity.

4.2.1. Forward Kinematics of the Ball Throwing Motion

In this part, engagement state of the quadcopter system is calculated. The engagement state is the required generalized positions and velocities of the quadcopter system for precise hit. The total energy of the final configuration is calculated in detail. Then, the required velocity vector of the ball is derived from the mission parameters. In the last step, the required ball trajectory is found for the quadcopter system. Forward kinematics algorithm for engagement state is developed for that purpose.

4.2.1.1.1. Total Energy of the Quadcopter System

The total energy of the quadcopter system is found in Chapter 3.2. However, calculation of the total energy of the quadcopter system is modified by including the motion of the ball. Then, it is defined as,

$$E_{total} = K_{total} + U_{total} \quad (4.1)$$

Total kinetic energy and potential energy of the system are calculated by considering the effect of the ball as,

$$K_{total} = K_Q + K_{ball} \quad (4.2)$$

$$U_{total} = U_Q + U_{ball} \quad (4.3)$$

$$K_{ball} = \frac{1}{2} \dot{\vec{p}}_e^{(i)T} m_{ball} \dot{\vec{p}}_e^{(i)} \quad (4.4)$$

$$U_{ball} = m_{ball} g \bar{u}_3^T \vec{p}_e^{(i)} \quad (4.5)$$

Total energy consumed by the system expressed in Eq (4.1) is as a function generalized positions and velocities.

$$E_{total} = f(\dot{x}, \dot{z}, \dot{\theta}, \dot{\theta}_1, \dot{\theta}_2, x, z, \theta, \theta_1, \theta_2) \quad (4.6)$$

4.2.2. Separation Velocity Vector and Release Angle Kinematic Equation

Final position of the ball can be considered for target engagement requirement while the speed of the ball can be derived accordingly. Projectile motion of thrown ball is calculated in planar space as follows:

Vertical Motion:

$$z_f = z_0 + v_{req} \sin(\theta_{release}) t + \frac{1}{2} (-g + a_z) t^2 \quad (4.7)$$

There is not any acceleration exerted on the ball by the robotic arm of the quadcopter system before shooting in the z direction, Eq (4.7) can be simplified as “ $a_z = 0$ ”. Vertical relative displacement is expressed as,

$$\Delta z = z_f - z_0 \quad (4.8)$$

Horizontal Motion:

$$x_f = x_0 + v_{req} \cos(\theta_{release}) t + \frac{1}{2} (a_x) t^2 \quad (4.9)$$

There is not any acceleration exerted on the ball by the robotic arm of the quadcopter system before shooting in the x direction, Eq (4.9) can be simplified as “ $a_x = 0$ ”. Horizontal relative displacement is expressed as,

$$\Delta x = x_f - x_0 \quad (4.10)$$

According to the known release angle “ $\theta_{release}$ ”, total required throwing speed “ v_{req} ” is found generically from Eq (4.7) to (4.10).

$$v_{req} = \frac{\Delta x}{\cos(\theta_{release})} \sqrt{\frac{g}{2(\Delta x \tan(\theta_{release}) - \Delta z)}} \quad (4.11)$$

4.2.3. Forward Kinematics Algorithm for the Engagement State

Main purpose is to specify the true while iterating the achieved trajectory of the ball. In order to satisfy the true trajectory by the achieved trajectory, initial position and velocity of the ball must be known.

The assumptions made for both calculation of the true and the achieved trajectory of the ball are:

- There is not any acceleration exerted in the x and z direction.
- There is not any mass change on the ball during the scenario.

- The only acceleration exerted to the ball is the gravitational acceleration.
- Initial position and velocity vector of the ball equals to the position and the velocity vector of the end-effector at the engagement state.

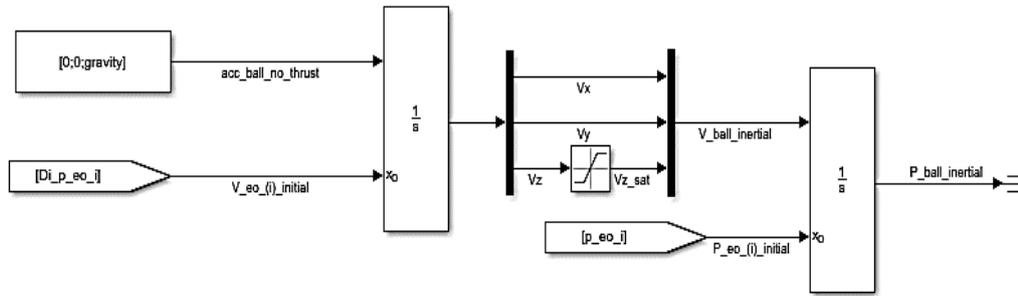


Figure 4.2 Ball Trajectory Kinematics

Position of the target and the release angle are assumed to be the known premises. Total required throwing speed is found from Eq (4.11). Then, true trajectory of the ball is obtained by the initial position and the initial velocity vector calculated for the known parameters. However, achieved trajectory of the ball is obtained by the position and the velocity vector of the end effector at the engagements state. True and achieved trajectories are implemented in MATLAB/Simulink. The velocity and the position trajectory of the ball is calculated by the implementation presented in Figure 4.2.

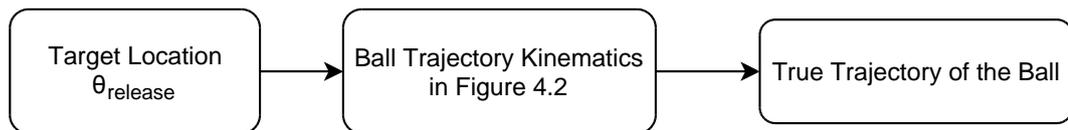


Figure 4.3 True Trajectory of the Ball

Evaluation of the true and the achieved trajectory of the ball is presented in Figure 4.3 and Figure 4.4, respectively.

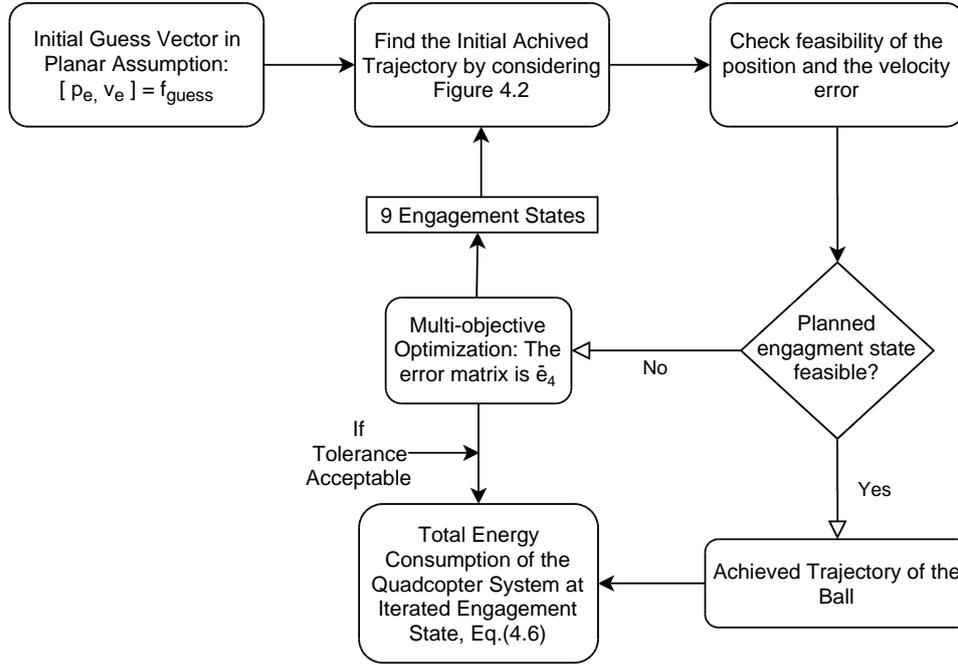


Figure 4.4 Forward Kinematics Algorithm

Figure 4.4 shows the flow chart of the Forward Kinematics Algorithm. Kinematics of the end-effector of the quadcopter system is obtained from Eq (3.21) and Eq (3.41). Energy analysis is also taken into considerations by using Eq (4.1). The iterated position - velocity vectors and total energy consumption of the quadcopter system at engagement state are implemented. Kinematic relations and the subsystem explained in Figure 4.2 are implemented to the algorithm, which is defined as the Forward Kinematics Algorithm. It is considered that the achieved position and the velocity trajectory of the ball depends on nonlinear equations obtained from the kinematics of the quadcopter system, which are iterated in forward kinematics for satisfying the true position and velocity trajectory of the ball.

Initial guess vector in planar assumption consists of 9 generalized positions and velocities of the quadcopter system, that are iterated in the forward kinematics algorithm and expressed as,

$$[\theta_{throw}, V_{throw}] = f_{guess}(\theta, \theta_1, \theta_2, \dot{\theta}, \dot{\theta}_1, \dot{\theta}_2, z, \dot{x}, \dot{z}) \quad (4.12)$$

Then, each trial of the generalized positions and velocities of the quadcopter system are iterated by considering Chapter 3.3.4. Here, “ e ” is defined for the error between true position and velocity trajectory with respect to achieved position and velocity trajectories in x-z direction, which are minimized. Nonlinear least-squares curve fitting algorithm’s boundaries are defined for the quadcopter and joint limits. In the each iteration of the solver, the weighting of the generalized positions and velocities of the robotic manipulator is set twice of the generalized positions and velocities of the quadcopter body. These limits consist of the kinematic variables in Eq (4.12). The position and the velocity terms are set according to the different scenarios and trajectories in 2-D.

The acceptable tolerance is defined as a constant value, which depends on the scenario. ITAE of the multi-objective optimization problem is the compared variable for the logic shown in Figure 4.4. If the ITAE of the iteration is less than the pre-specified ITAE tolerance, acceptable group of iterations are displayed for the scenarios.

4.3. Trajectory Generation of the Quadcopter System

Trajectory of the quadcopter system is obtained for the minimum input generated by the rotors and the RC servos. A proposed controller structure is designed in order to track the optimized trajectory. Infinite horizon LQR controller is used for the altitude and the attitude stabilization. Gains of the controller is set for the optimized trajectory tracking. These two topic will be detailed in this part.

4.3.1. Optimized Trajectory Generation

Trajectory is generated according to linearization of the equation of motion of the quadcopter system in planar space. Since the system is underactuated, motion in the x direction is generated by tilting the quadcopter. Gravity is assumed to be in the opposite direction of the z-axis. Freebody diagram of the simplified system is shown in Figure 4.5.

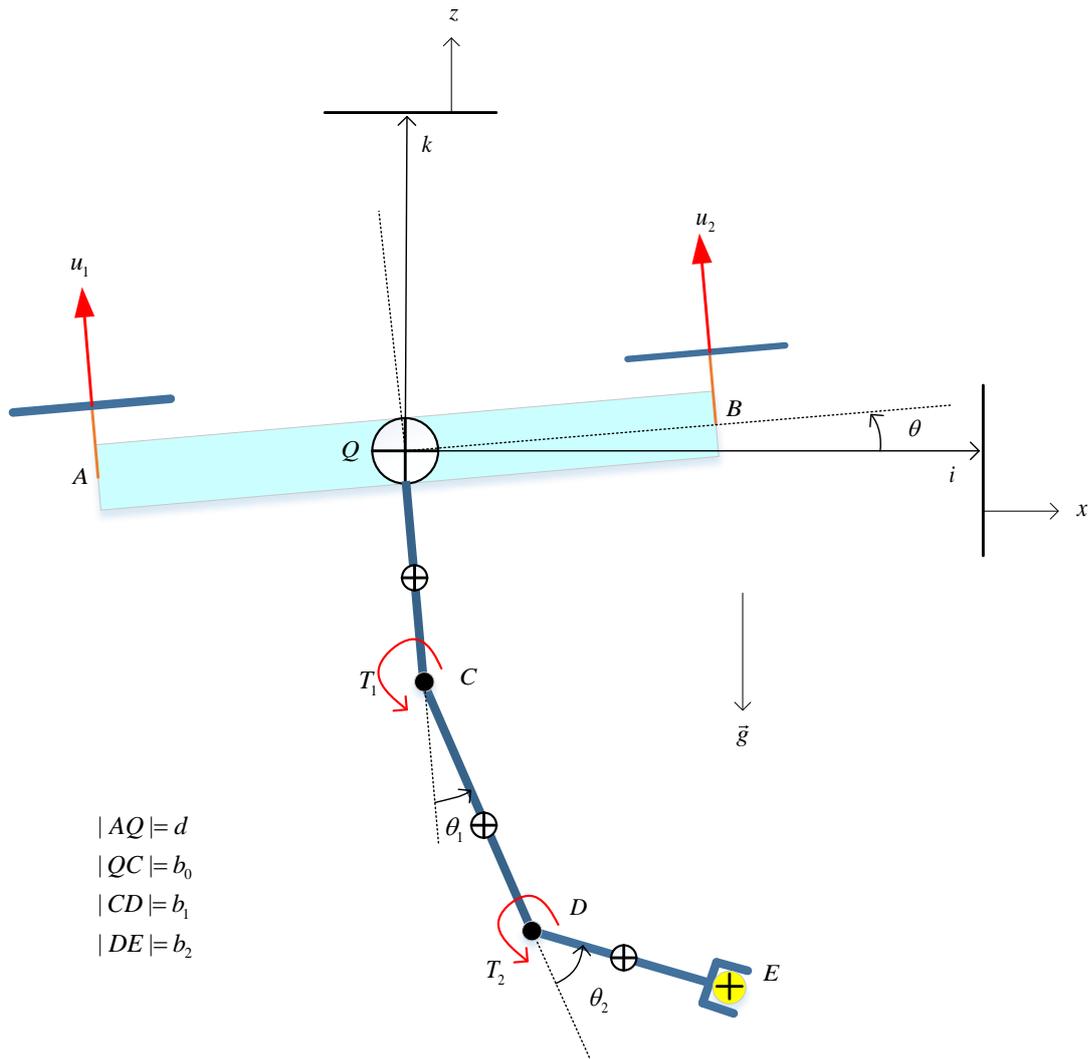


Figure 4.5 Simplified 3-DOF Quadcopter System

Thrust force and the linearized attitude angle produce the acceleration in the x direction. The links are assumed to be perfectly rigid and interaction forces of the links are neglected. However, the reaction moments of the links are included for the linearized quadcopter body dynamics.

Generalized coordinates of the quadcopter system consists of 5 elements for the planar assumption, they are as follows.

$$\bar{q} = [x \quad z \quad \theta \quad \theta_1 \quad \theta_2]^T$$

Force and moment generated in the quadcopter system is presented by considering the planar assumption as,

$$f_z = u_1 + u_2$$

$$\tau_{q2} = (u_1 - u_2)d$$

Then, the system dynamics is defined in Eq (4.13)

$$\ddot{\bar{q}} = M^{-1}f_{sys} \quad (4.13)$$

Eq (4.13) is expanded as,

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -\sin\theta \frac{f_z}{m_t} \\ \cos\theta \frac{f_z}{m_t} - g \\ -\frac{\tau_{q2}}{m_t d^2} \\ \tau_1 - \frac{b_1}{2I_{1,y}} g \sin\theta_1 (m_{12ball}) \\ \tau_2 - \frac{b_2}{2I_{2,y}} g \sin\theta_2 (m_{2ball}) \end{bmatrix}$$

“M” is the mass-inertia matrix of the quadcopter system in the planar assumption. ”M” is a function of “h” expressed in Eq (4.14). “ f_{sys} ” is the internal force matrix of the system and it is a function of “f” shown in Eq (4.15).

$$M = h(m_q, m_1, m_2, m_{ball}, I_{yy}, I_{y1}, I_{y2}) \quad (4.14)$$

$$f_{sys} = f(u_1, u_2, t_1, t_2, \theta, \theta_1, \theta_2, b_0, b_1, b_2, d, g) \quad (4.15)$$

Force and moment inputs of the system are defined as:

$$\bar{u} = \begin{bmatrix} u_1 \\ u_2 \\ t_1 \\ t_2 \end{bmatrix}$$

- u_1 and u_2 are the thrust forces generated by rotors.
- t_1 and t_2 are the torque generated by RC servo motors on the joints of the links.

States that are required to be solved for the optimal trajectory is the collection of the position and the velocity terms of the generalized coordinates [51].

$$\bar{v} = [\bar{q} \dot{\bar{q}}]^T$$

\bar{v} can be expanded as.

$$\bar{v} = [x \quad z \quad \theta \quad \theta_1 \quad \theta_2 \quad \dot{x} \quad \dot{z} \quad \dot{\theta} \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T$$

Force input is defined as a cost function in order to get the minimum energy consumption while trajectory is executed. Cost function is expressed as.

$$J_{cost} = \int_{t_0}^{t_f} u^2(\tau) d\tau$$

Subjected to pre-specified system parameters, which are mass, length, force and torque characteristics involved in the trajectory of the quadcopter system;

Time Boundary:

$$t^* \in [t_0, t_f]$$

Initial State Boundary is fixed for the initial state vector as:

$$v(t_0) \in [v_{min}, v_{0,max}] \rightarrow v(t_0) = v_0$$

Final State Boundary is fixed for the final state vector as:

$$v(t_f) \in [v_{f,min}, v_{f,max}] \rightarrow v(t_f) = v_f$$

System Dynamics is also generically expressed in Eq (4.13):

$$\dot{v} = f(t, v(t), u(t))$$

State Boundary on Continuous Time such as: Joint limits and quadcopter linear region.

$$v_{min} \leq v(t^*) \leq v_{max}$$

Control Boundary on Continuous Time such as: Thrust force and joint torque limits

$$u_{min} \leq u(t^*) \leq u_{max}$$

Constrained Nonlinear Programming Solver for Multivariable Function of MATLAB is used to get the set of optimal states on the path. Trapezoid integrator of MATLAB [50] is set for solving the high order derivative terms. Trajectory is divided into segments by using MATLAB Optimization Toolbox. The number of segment that the trajectory is divided into is set for specified grids for pre-specified flight duration. Quadratic Spline Interpolation of MATLAB [49] is used for finding the off grid points of the trajectory [52]. Optimization is iterated for each segments of the trajectory. The general principle of the trajectory optimization is illustrated in Figure 4.6.

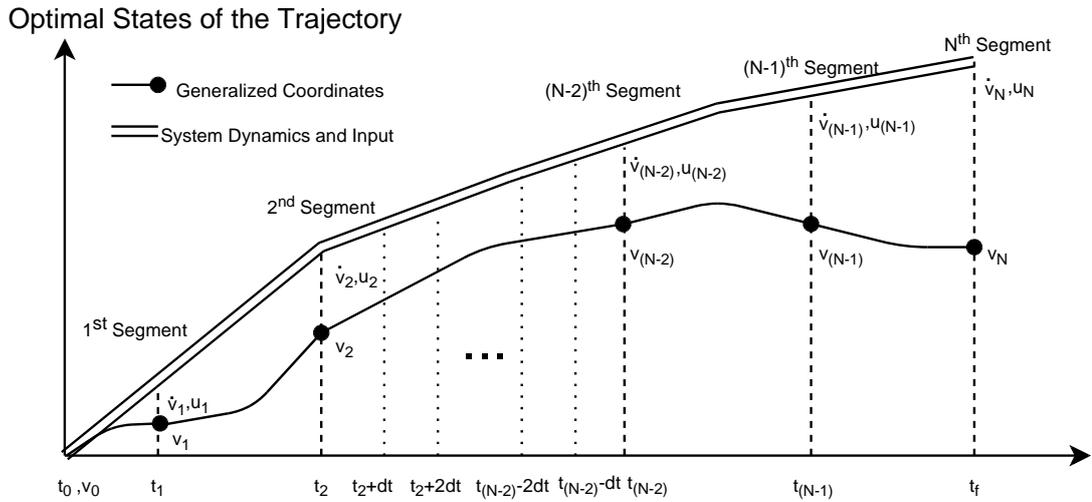


Figure 4.6 Optimal States of the Trajectory Generation

4.4. Controller Topology of the Optimal State Trajectory Tracking of the Quadcopter System

Optimal state trajectory information is set to the reference inputs of the quadcopter system. A controller structure is implemented in order to track the reference inputs

purely. Reference input is traced by LQR controller designed for unified quadcopter system.

The altitude and the attitude dynamics is included for the LQR controller structure [43]. Since the system model is designed for the linear region, small angle assumption satisfies X position and velocity requirement in quadcopter pitch dynamics. By considering Eq (4.13),

The infinite horizon, LQR is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ J &= \int_0^{\infty} (x^T Q x + u^T R u) dt \end{aligned} \quad (4.16)$$

Let's define the control input as.

$$u = -R^{-1}B^T P x$$

Fix gain set “K” is defined as.

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

Algebraic Ricatti equation is solved for P as follows.

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

States of the LQR controller are defined as.

$$x = [\dot{z} \ \dot{\theta} \ \dot{\theta}_1 \ \dot{\theta}_2 \ z \ \theta \ \theta_1 \ \theta_2]^T$$

LQR controller follows the desired trajectory by control law expressed in Eq (4.17), [43].

$$u = -Kx \quad (4.17)$$

“A” matrix of the system is 8x8 according to the equation of motion defined for attitude and altitude dynamics. “B” matrix is defined by 8x4 according to the control inputs. Control inputs are the thrust force and the pitch moment generated by the rotors and the torque inputs from the RC Servo motors, so that “u” is 4x1 matrix. “K” gain matrix has the size of 4x8. Desired states of the trajectory is found in the previous section and are followed by the controller. Implementation of the controller architecture is illustrated in Figure 4.7.

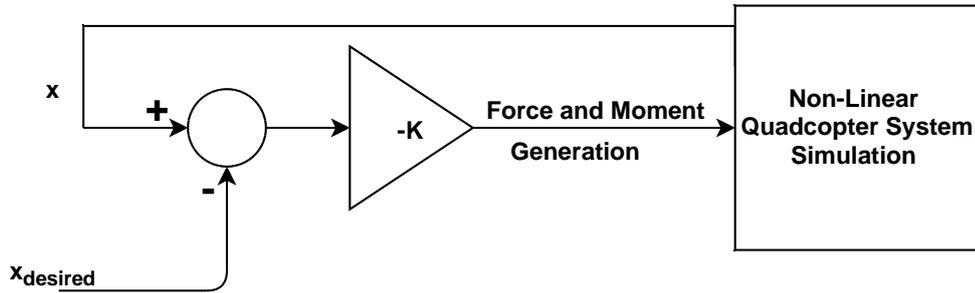


Figure 4.7 Implementation of the LQR controller

The nominal gains of the LQR controller are implemented by considering the weights Q and R in Eq (4.16). Q weights on states and R weights on control input in cost function, which are further improved by considering Chapter 3.3.4. Here, “ \bar{e}_k ” is defined for the error between optimal states of trajectory with respect to the achieved states of the quadcopter system.

Error minimization between reference input and system response for finding the LQR weightings are obtained for two scenario pairs generated on 6-DOF nonlinear model and they are expressed in a row matrix. The first and the second LQR weightings are obtained for the scenarios detailed in Chapter 5.2.2.1 and Chapter 5.2.2.2, respectively. Each elements of the row matrix represents the diagonal elements of “Q” and “R” matrices which are 8x8 and 4x4 respectively.

$$Q_1 = [3434 \ 5 \ 0.787 \ 3.6 \times 10^{-7} \ 3735 \ 537 \ 122 \ 100]$$

$$R_1 = [0.2 \ 0.4 \ 2.5 \ 1]$$

$$Q_2 = [3434 \ 5 \ 0.5 \ 0.01 \ 3735 \ 537 \ 700 \ 100]$$

$$R_2 = [0.2 \ 0.4 \ 8 \ 5]$$

Eigenvalues of the closed loop A matrix is found as.

$$\lambda_i = \text{eig}(A - BK)$$

Where, it is defined that $i = 1,2,3,4,5,6,7$ and 8 .

It is found by using Control System Toolbox of MATLAB. Eigenvalues of the closed loop A matrix are found as.

$$\bar{\lambda}_1 = \begin{bmatrix} -93.5 \\ -711.6 \\ -1.04 \\ -169.2 + 169.2j \\ -169.2 - 169.2j \\ -171.3 \\ -12.5 \\ -10.4 \end{bmatrix}$$

$$\bar{\lambda}_2 = \begin{bmatrix} -93.5 \\ -1.04 \\ -230.8 \\ -171.3 \\ -110.9 \\ -10.4 \\ -314.9 \\ -37.7 \end{bmatrix}$$

CHAPTER 5

SIMULATION BASED EXPERIMENTS

5.1. Simulation of the Quadcopter System

The nonlinear coupled quadcopter system dynamics and kinematics are modeled. The quadcopter system model is implemented in MATLAB/Simulink. Control algorithms, motor and sensor subsystems of the quadcopter system are performed in the same simulation environment. Then, control algorithms are tested with the nonlinear quadcopter system simulation. In order to generate the model realistically, physical limits are defined as: maximum rotational speed of the propellers is set to 2000 rad/sec, torque limit for RC servo command is set according to the values listed in Table 6.4, rotation limit of the links of the robotic arm is set to be within 25 degrees. Solver of the simulation model is set for 1 millisecond of step size with the ODE-4 solver. Aerodynamic thrust and torque coefficients are taken from [39]. In addition, energy analysis is also presented.

Table 5.1 System Parameter Set for the Simulation Model

m_q [kg]	1.240
m_1 [kg]	0.06
m_2 [kg]	0.08
m_{ball} [kg]	0.022
$[I_{q,x} \ I_{q,y} \ I_{q,z}]$ [kg.m ²]	[0.0206 0.0206 0.017181]
$[I_{12b,x} \ I_{12b,y} \ I_{12b,z}]$ [kg.m ²]	[0.675 1.97 1.97]x10 ⁻⁴
$[I_{2b,x} \ I_{2b,y} \ I_{2b,z}]$ [kg.m ²]	[0.425 1.44 1.44]x10 ⁻⁴
d [m]	0.225
b_0 [m]	0.10
b_1 [m]	0.11
b_2 [m]	0.12
$c_{1,2}$ [m]	0.05
$d_{1,2}$ [m]	0.05
Gravity [m/s ²]	9.81

c_T [N/(rad/s) ²]	2.74×10^{-5}
c_Q [Nm/(rad/s) ²]	0.047×10^{-5}

5.2. Scenarios of the Simulation Based Experiment

In this part, two controller configurations are tested in the simulation environment. According to the scenarios, the simulation environment is modified by enabling the motor and sensor subsystems. Details of the scenarios are presented in Table 5.2. Scenarios are analyzed by considering two approaches. The first approach is to throw the ball for precise target engagement which is common for the each scenario. The second approach is analyzed for Scenario-4 and Scenario-5, which is tested for the throwing-stabilization-landing modes. Trajectories of the ball for each scenario are also analyzed.

Table 5.2 Scenarios of the Quadcopter System Simulation and Analysis

Scenario Number	Controller	Motor Subsystem	Sensor Subsystem	Scenario Property	Release Angle	Target Relative Distance
Scenario-1	Cascaded PID	+	+	Throw	39 °	(1.7 , - 0.7) m
Scenario-2	LQR	+	-	Throw	51 °	(1.7 , - 0.7) m
Scenario-3	LQR	+	+	Throw	45 °	(1.7 , - 0.7) m
Scenario-4	LQR	+	-	Throw and Land	45 °	(5 , - 0.7) m
Scenario-5	LQR	+	+	Throw and Land	45 °	(5 , - 0.7) m

A 3-D trajectory tracking is tested for the cascaded PID controller configuration. Results are presented in Appendix-A.

5.2.1. Examining the Cascaded PID Controller Configuration for Ball Throwing

In the first scenario, X position controller is bypassed by the Euler pitch controller. Other controller channels are remained the same for closed loop stabilization. Mission parameters are defined as: release angle is set to 39 degrees, target position is relatively located at (1.7 , - 0.7) m. Engagement states of the quadcopter system are calculated

in Forward Kinematics Algorithm and expressed in Table 5.3. Reference inputs for the quadcopter system are generated by trial and error. However, the engagement states of the quadcopter system are achieved.

Table 5.3 Shooting Parameters of the Scenario-1

Parameter	Value
θ	2.9 deg
θ_1	9 deg
θ_2	9 deg
$\dot{\theta}$	21 deg/s
$\dot{\theta}_1$	20 deg/s
$\dot{\theta}_2$	20 deg/s
Δx	1.74 m
Δz	0.7072 m
x_q	-0.07m
z_q	1.02m
\dot{x}	2.39 m/s
\dot{z}	2.05 m/s
V_{xball}	2.61 m/s
V_{zball}	2.11 m/s
Energy of the motion	22.06 J

Shoot signal is also shown in the following the figures. Engagement states of the quadcopter system is achieved in 2.23 second of simulation time. Figure 5.1 presents the time evolution of linear velocity of the quadcopter system. Linear velocity of the quadcopter system in the x direction is increased by Euler pitch reference input. As shown in the Figure 5.2, 10 degrees of constant Euler pitch reference input is fed to the attitude controller for satisfying the linear velocity requirement of shooting.

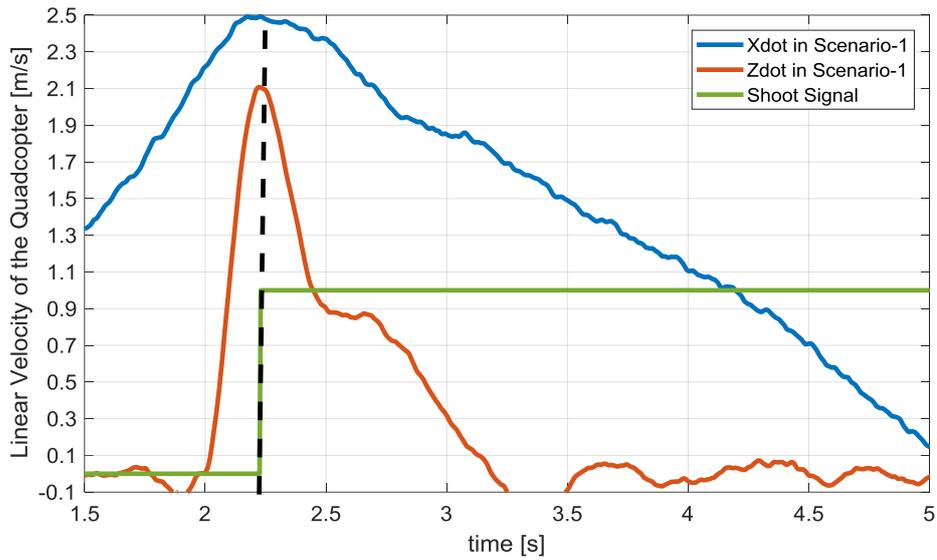


Figure 5.1 Linear Velocity of the Quadcopter System for Scenario-1

Time evolution of the arm's joint angles and rates are shown in the Figure 5.2 and 5.3, respectively.

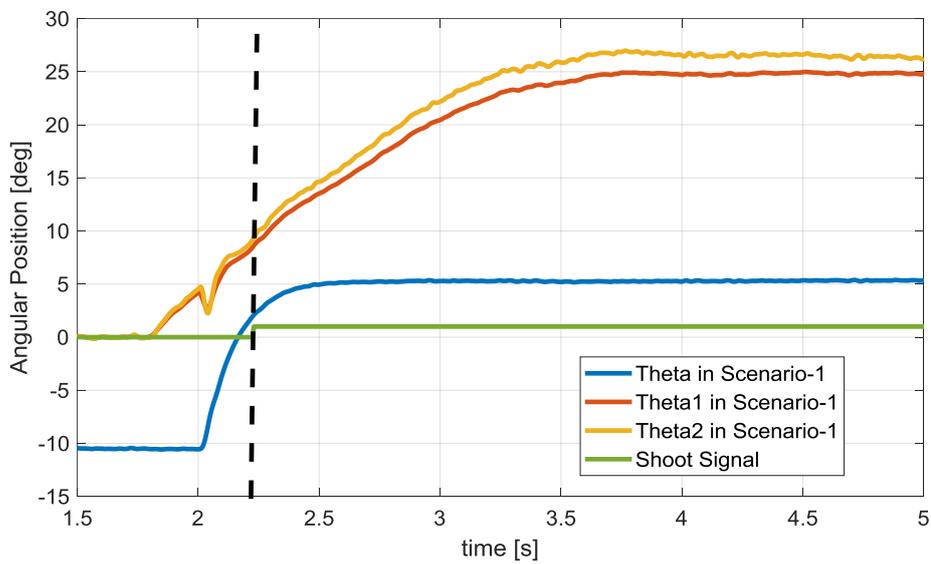


Figure 5.2 Angular Position of the System Parts for Scenario-1

Figure 5.3 presents the time evolution of the inner rate loop controller. Required angular rate of the quadcopter system is obtained by generation of the 15 degrees of Euler pitch angle reference input in the opposite direction, as shown in Figure 5.2.

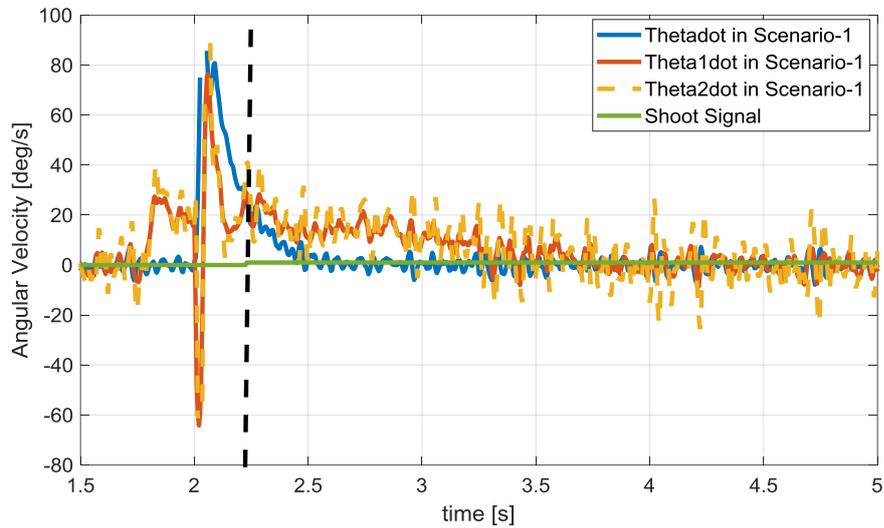


Figure 5.3 Angular Velocity of the System Parts for Scenario-1

There is a change in the nominal trend of the rotational speed of the rotors before shooting the ball. As seen in Figure 5.4 that, the nominal rotational speed is increased by 250 rpm.

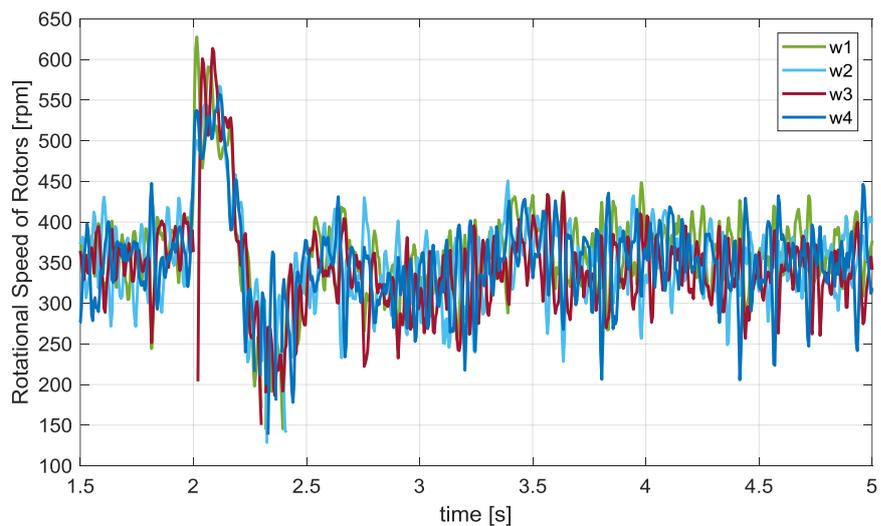


Figure 5.4 Rotational Speed of Rotors during the Flight Simulation for Scenario-1

In Figure 5.5, trajectories of the ball are compared with the ideal case and the simulation output. There is less than 10 cm of difference on hit location.

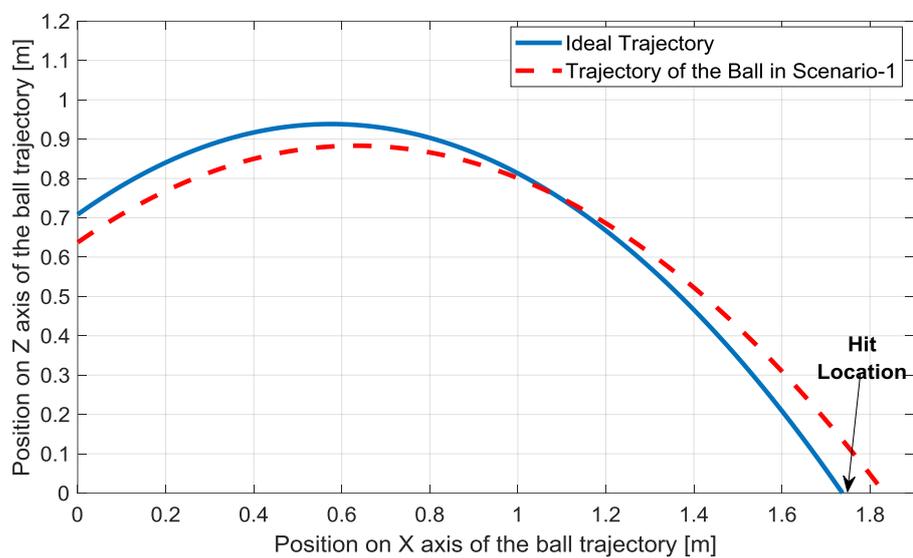


Figure 5.5 Trajectory of the Ball Compared with the Achieved System Output for Scenario-1

5.2.2. Examining the Trajectory Planning and LQR Controller Configuration for Ball Throwing

The 6-DOF nonlinear quadcopter simulation is used for demonstrations. LQR controller configuration is implemented in the same simulation environment. Planar control commands are generated and fed into the LQR controller. Y position, Euler roll angle and rate reference inputs, Euler Yaw angle and rate reference inputs are set to be zero.

Two topics are investigated for the optimal state trajectory generation and controller tracking. In the first part, two scenarios are generated for the same delta distance on the target location with different release angles. These are the Scenario-2 and Scenario-3, which are expressed in Table 5.2. In the second part, performance limits of the controller and the trajectory generation are analyzed. These are the Scenario-4 and Scenario-5, which are expressed in Table 5.2 Throwing-stabilization-landing modes are performed. The external disturbance of the ball is removed in the stabilization-landing phase.

Mission parameters are set for the Scenario 2, 3 and 4-5. Engagement states are obtained by using forward kinematics algorithm. Generalized positions and velocities of the engagement states of the quadcopter system are the final state of the optimal state trajectory. Initial states of the trajectory are set to zero matrix. In the last part of the analysis, achieved energy consumption analysis are investigated according to the trajectory tracking performances of the controllers.

5.2.2.1. Trajectory Generation and Tracking Analysis for the Scenario-2 and the Scenario-3 Pair

Trajectory generation is obtained for 3 seconds of flight analysis. Path of the trajectory is divided into 10 segments. The illustration of the segments of the trajectory is shown in Figure 4.6.

5.2.2.1.1. Trajectory Optimization and Tracking for the Engagement State in Scenario-2

Mission parameters are defined as: release angle is set to 39 degrees, target position is relatively located at (1.7 , - 0.7) m. Engagement states of the quadcopter system are calculated in Forward Kinematics Algorithm and expressed in Table 5.4.

Table 5.4 Shooting Parameters obtained for Scenario-2

Parameter	Value
θ	1.35 deg
θ_1	8.47 deg
θ_2	6.77 deg
$\dot{\theta}$	28.68 deg/s
$\dot{\theta}_1$	41.73 deg/s
$\dot{\theta}_2$	41.26 deg/s
Δx	1.704 m
Δz	0.72 m
x_q	-0.05 m
z_q	1.047m
\dot{x}	2.19 m/s
\dot{z}	2.02 m/s
$V_{x_{ball}}$	2.599 m/s
$V_{z_{ball}}$	2.11 m/s
Energy of the motion	21.6 J
ITAE	0.0004056

Energy of the quadcopter system in the forward kinematics algorithm is expressed in Figure 5.10. Error tolerance of ITAE is chosen to be relaxed in order to observe the energy changes on the iteration steps. The engagement states are found in the 90th iteration step.

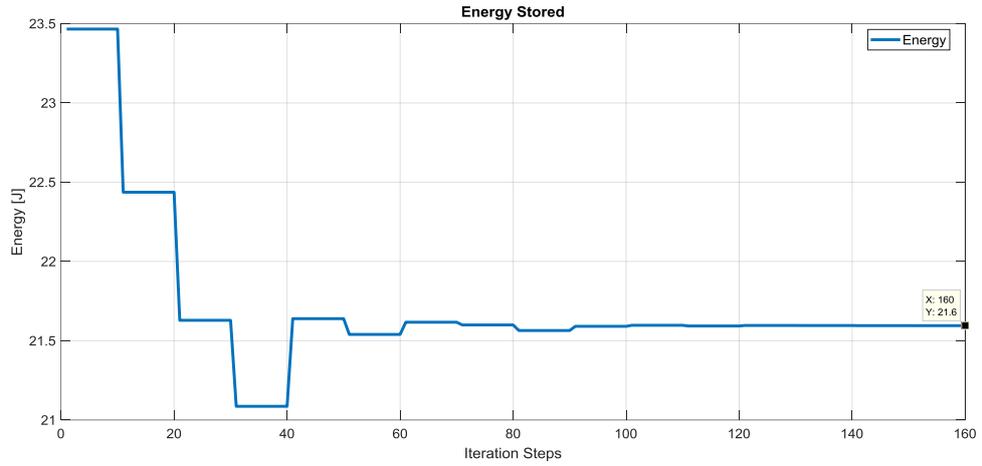


Figure 5.6 Energy of the Quadcopter System in the Forward Kinematics Solution for Scenario-2

Boundaries for the path generation is shown in Table 5.5.

Table 5.5 Boundaries of the Path for Scenario-2

	x	z	θ	θ_1	θ_2	\dot{x}	\dot{z}	$\dot{\theta}$	$\dot{\theta}_1$	$\dot{\theta}_2$
v_{min}	0	0	-15°	0°	-5°	-2	-2	-50°	0°	0°
v_{max}	4	2	15°	15°	15°	3	3	50°	60°	60

Figure 5.8 presents the time evolution of the angular position tracking. Since the angular position tracking is satisfactory in the linear region of the quadcopter system, there is almost zero error in the x-position compared with the reference position in the x direction of the generated trajectory, as in Figure 5.7. The time evolution of the linear velocity in the x direction is shown in Figure 5.9.

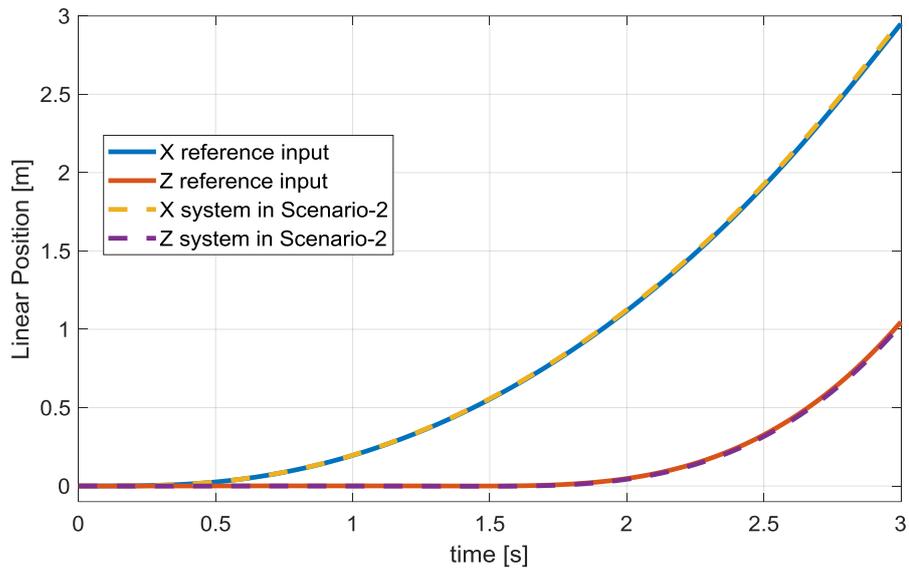


Figure 5.7 Linear Position Reference Input and System Response for Scenario-2

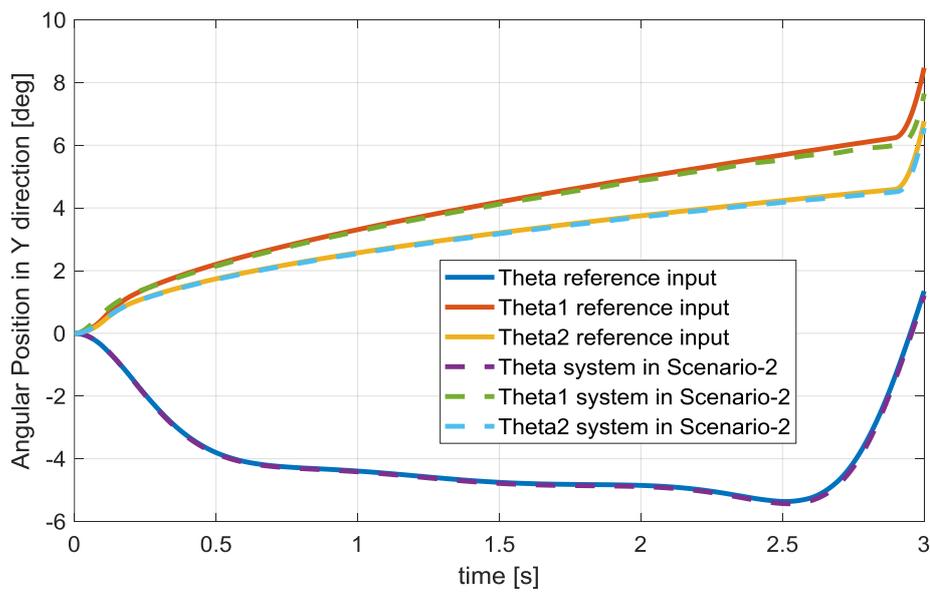


Figure 5.8 Angular Position Reference Input and System Response for Scenario-2

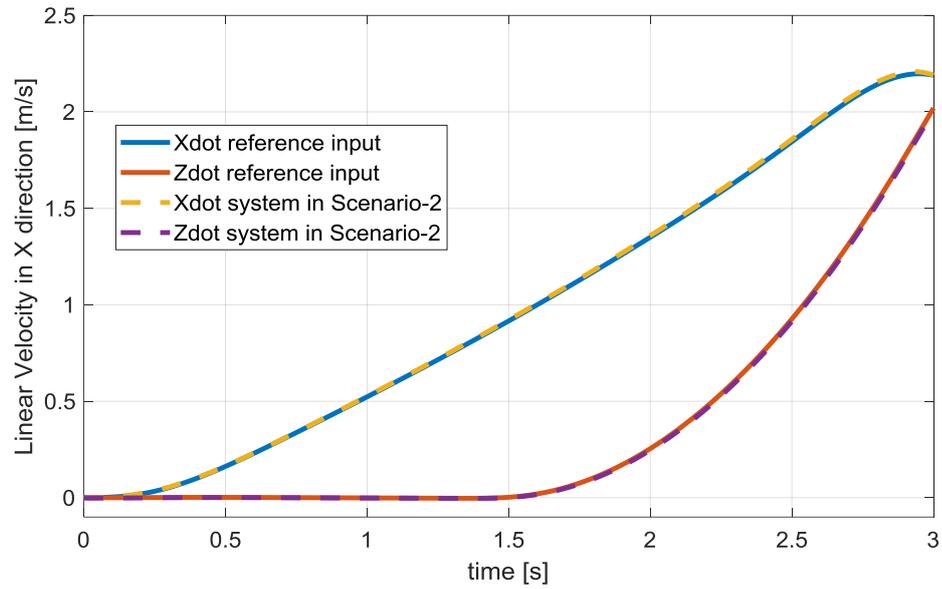


Figure 5.9 Linear Velocity Reference Input and System Response for Scenario-2

The time evolution of the Euler pitch rate, link-1 and link-2 angular rates are presented in Figure 5.10 to 5.12, respectively.

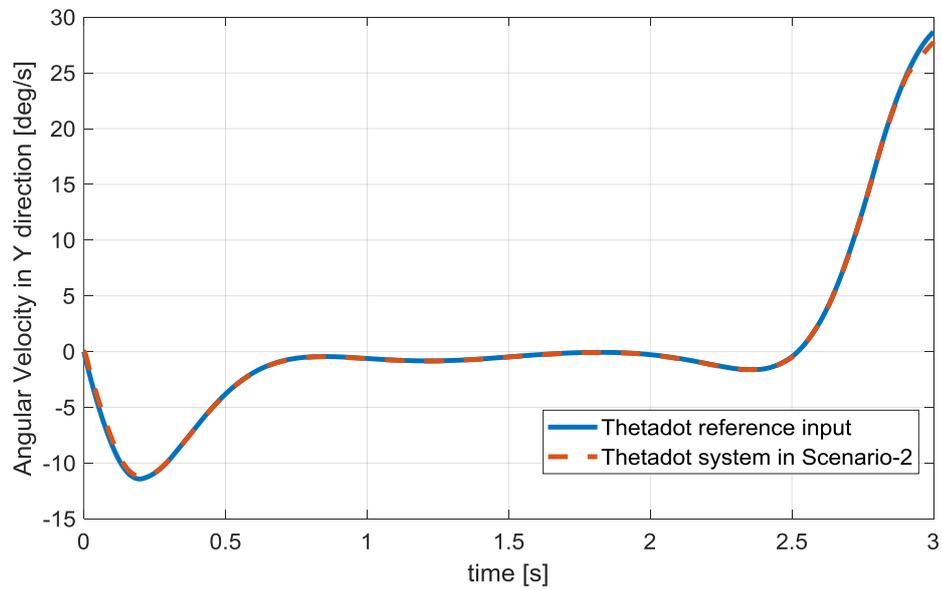


Figure 5.10 Theta Angular Velocity Reference Input and System Response for Scenario-2

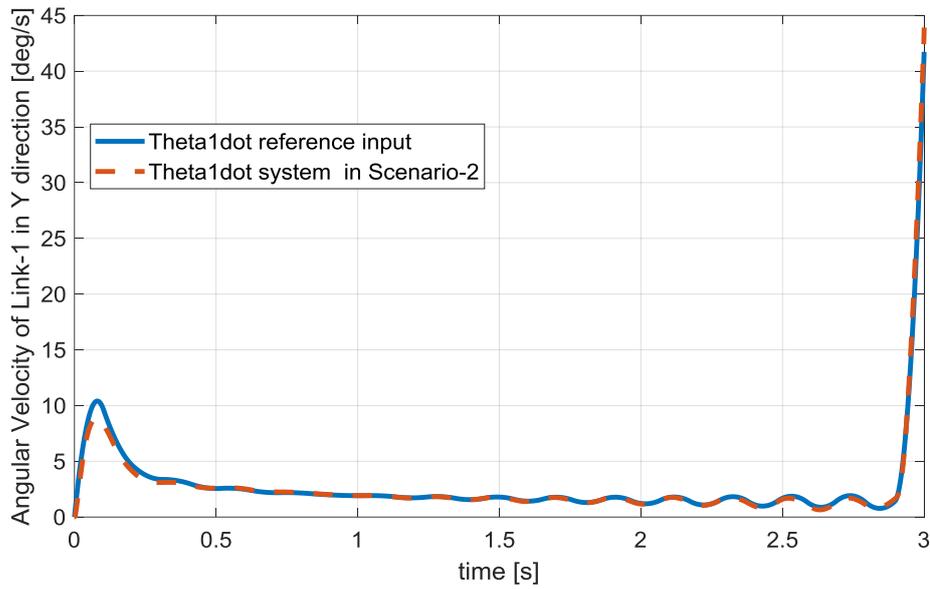


Figure 5.11 Theta1 Angular Velocity Reference Input and System Response for Scenario-2

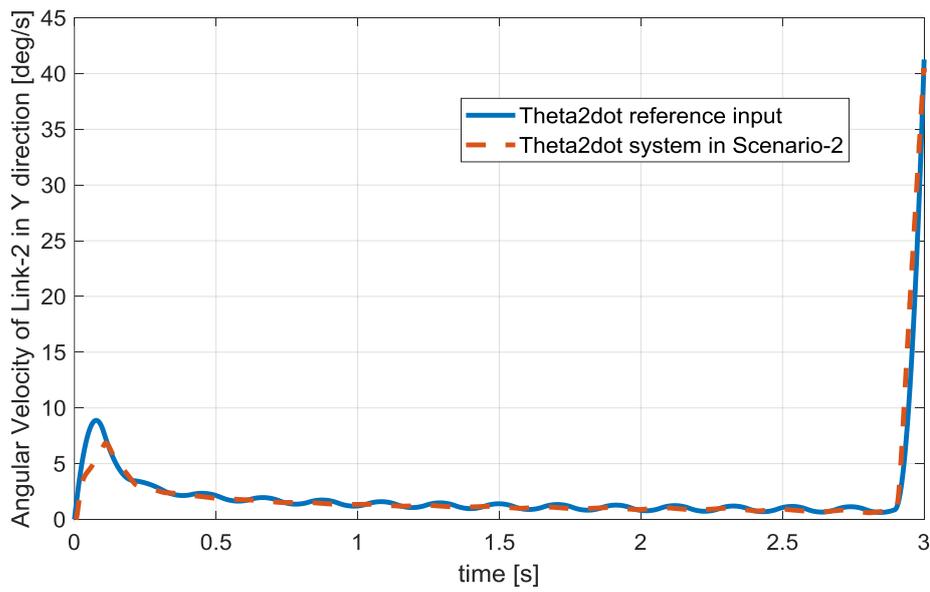


Figure 5.12 Theta2 Angular Velocity Reference Input and System Response for Scenario-2

5.2.2.1.2. Different Release Angle Trajectory Optimization and Tracking for the Engagement State in Scenario-3

Different release angle demonstration is analyzed. The same hit location considered in Scenario-2 with 45 degrees of different release angle is solved by forward kinematics algorithm. The same steps are followed. The obtained parameter set is expressed in Table 5.6,

Table 5.6 Shooting Parameters obtained for Different Release Angle in Scenario-3

Parameter	Value
θ	10.56 deg
θ_1	12.06 deg
θ_2	12.06 deg
$\dot{\theta}$	20.80 deg/s
$\dot{\theta}_1$	33.44 deg/s
$\dot{\theta}_2$	33.65 deg/s
Δz	0.75 m
Δx	1.704 m
x_q	-0.13 m
z_q	1.04m
\dot{x}	2.13 m/s
\dot{z}	2.26 m/s
V_{xball}	2.42 m/s
V_{zball}	2.42 m/s
Energy of the motion	22.07 J
ITAE	0.00047

Energy of the quadcopter system in the forward kinematics algorithm is expressed in Figure 5.13. The engagement states are found in the 100th iteration step.

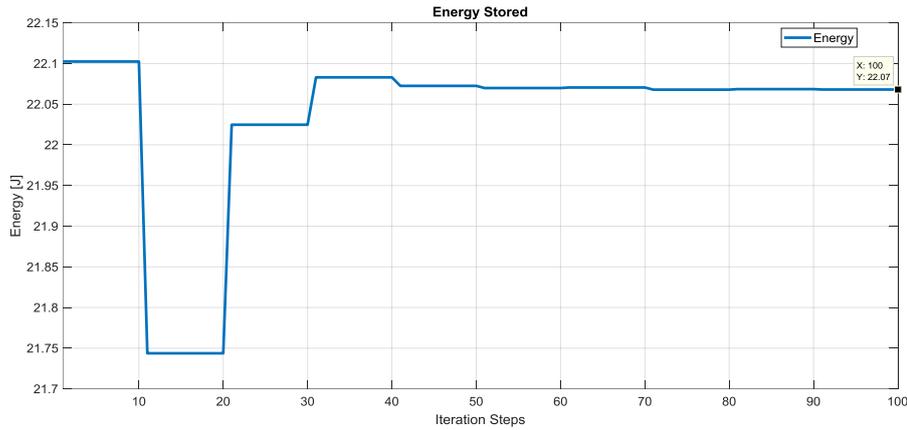


Figure 5.13 Energy of the Quadcopter System in the Forward Kinematics Solution for Different Release Angle in Scenario-3

Boundaries for the path generation is expressed in Table 5.7.

Table 5.7 Boundaries of the Path for Different Release Angle in Scenario-3

	x	z	θ	θ_1	θ_2	\dot{x}	\dot{z}	$\dot{\theta}$	$\dot{\theta}_1$	$\dot{\theta}_2$
v_{min}	0	0	-15°	0°	-5°	-2	-2	-50°	0°	0°
v_{max}	4	2	15°	15°	15°	3	3	50°	60°	60°

Quadcopter system behaves similar as in Scenario-2. The minimum point of the Euler pitch position reference input is around -10 degrees. The only difference compared with the Scenario-2 is the linear velocity error in the x direction, as shown in Figure 5.16.

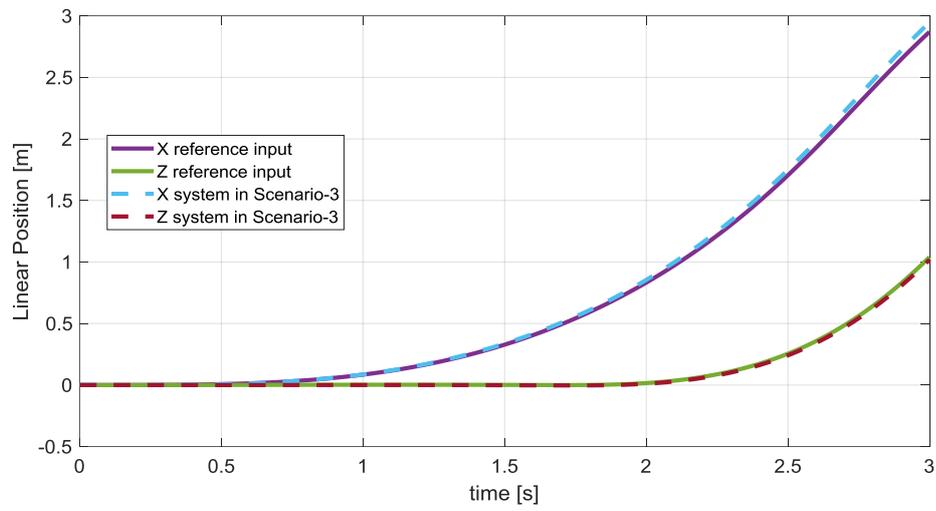


Figure 5.14 Linear Position Reference Input and System Response for Different Release Angle in Scenario-3

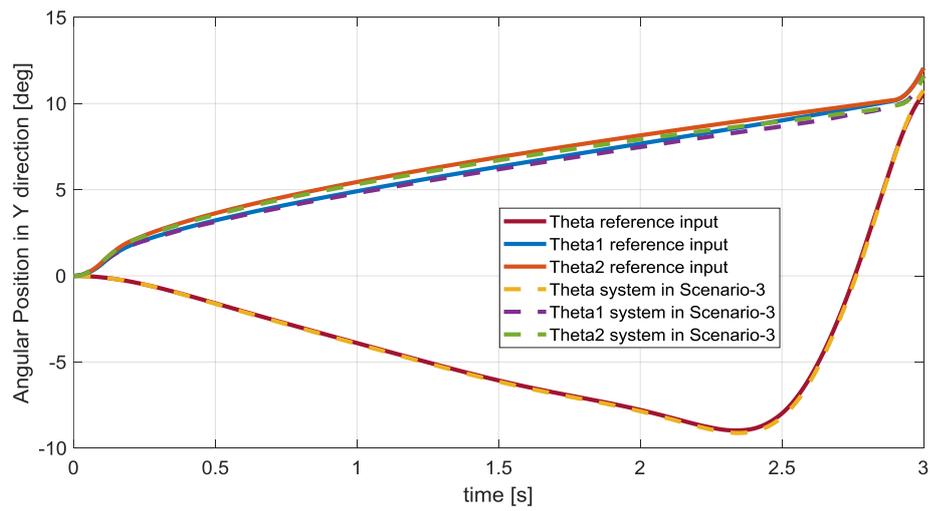


Figure 5.15 Angular Position Reference Input and System Response for Different Release Angle in Scenario-3

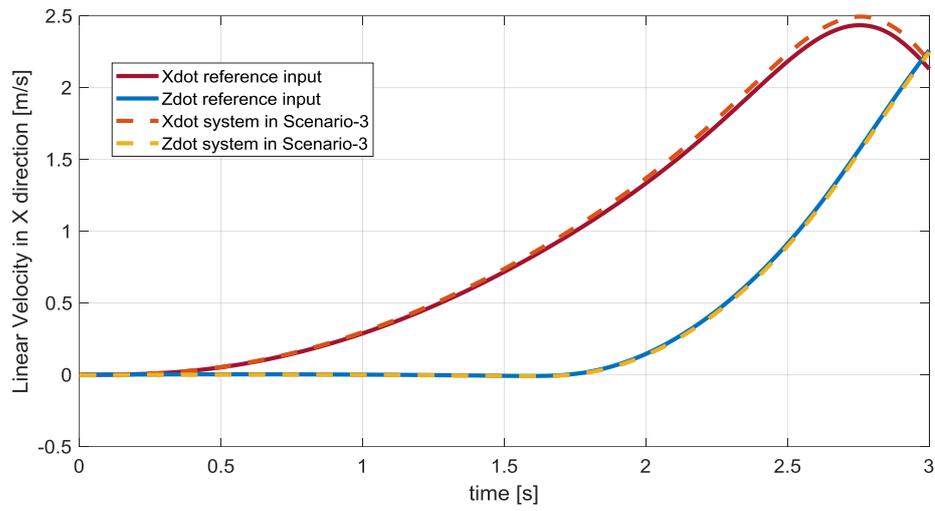


Figure 5.16 Linear Velocity Reference Input and System Response for Different Release Angle in Scenario-3

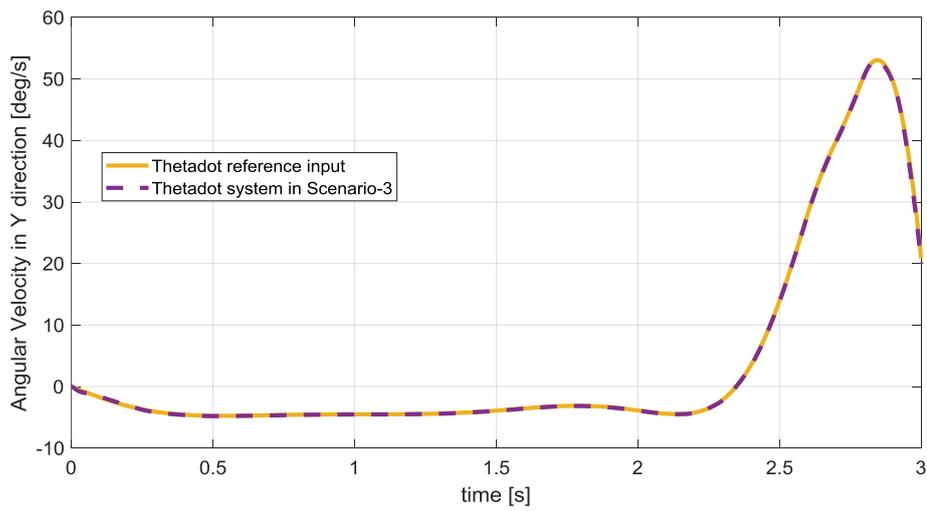


Figure 5.17 Theta Angular Velocity Reference Input and System Response for Different Release Angle in Scenario-3

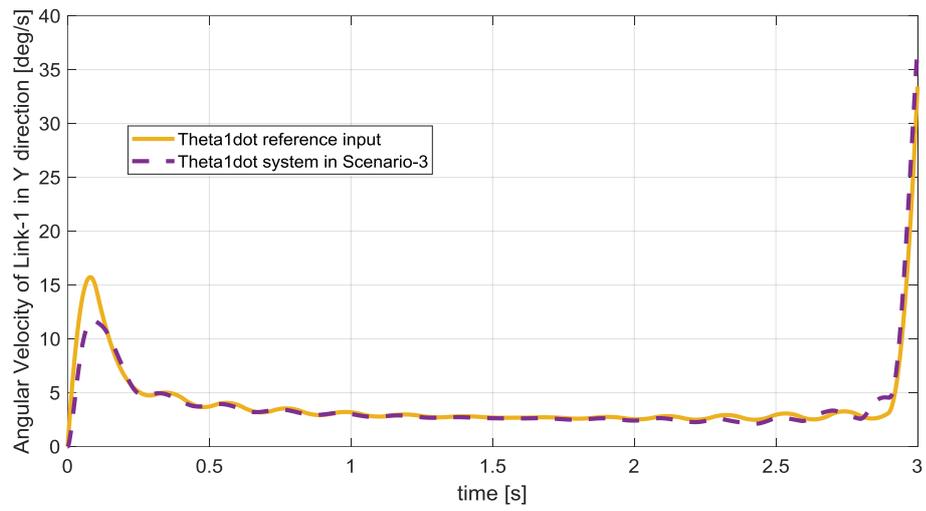


Figure 5.18 Theta1 Angular Velocity Reference Input and System Response for Different Release Angle in Scenario-3

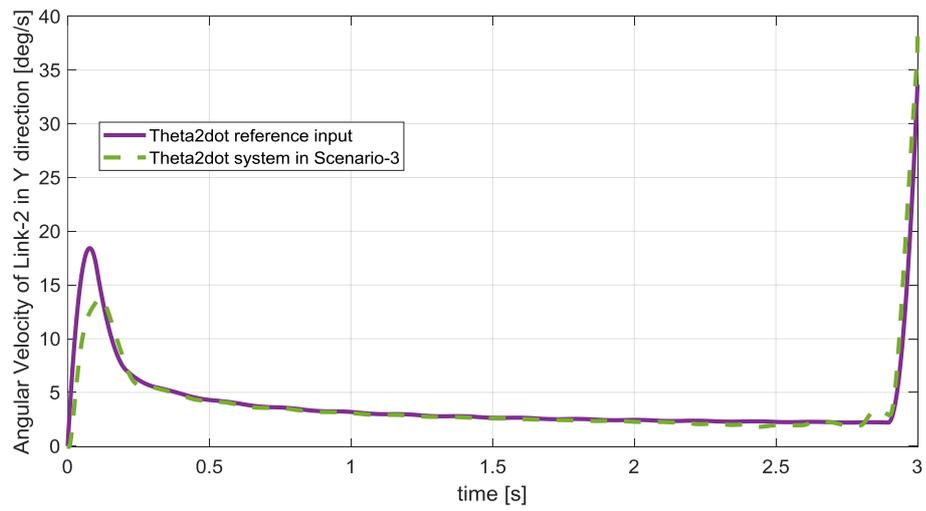


Figure 5.19 Theta2 Angular Velocity Reference Input and System Response for Different Release Angle in Scenario-3

5.2.2.1.3. Analysis Performed for the Energy Consumption and Trajectory of the Ball

Power consumption of the quadcopter system consists of rotors and RC servo motors. Power consumption of the actuators are defined in Eq (5.1) and (5.2), the unit of the equality is Joule/s.

$$P_{links} = \sum_{i=1}^2 t_i \dot{\theta}_i \quad (5.1)$$

$$P_{rotors} = \sum_{i=1}^4 \tau_i \omega_i \quad (5.2)$$

Energy consumption is the integral of the power consumption. The integral is assigned for the simulation stopping time. Energy consumed by actuators are defined in Eq (5.3) and (5.4), the unit of this operation is in Joule.

$$E_{links} = \int_0^{t_{stop}} P_{links} dt \quad (5.3)$$

$$E_{rotors} = \int_0^{t_{stop}} P_{rotors} dt \quad (5.4)$$

Energy consumption for the whole scenario generation consists of throwing energy and energy consumed to reach the engagement states.

$$E_{total} = E_{trajectory} + E_{throw}$$

Let's define that,

$$E_{trajectory} = E_{quadcopter} + E_{link}$$

Achieved total energy consumed according to the simulation output which is assigned for Scenario-2, obtained as.

$$E_{quad-1} = 291.6 \text{ J}$$

$$E_{Links-1} = 2.231 \times 10^{-4} \text{ J}$$

$$E_{throw-1} = 20.79 J$$

$$E_{total-1} = 312.39 J$$

Achieved total energy consumed according to the simulation output which is assigned for Scenario-3, obtained as.

$$E_{quad-2} = 296.6 J$$

$$E_{Links-2} = 4.693 \times 10^{-3} J$$

$$E_{throw-2} = 21.23 J$$

$$E_{total-2} = 317.83 J$$

Figure 5.20 compares the trajectories of the ball in Scenario-2 and Scenario-3 with the ideal trajectory. There is a less than 30 cm of shift in the hit location. Different release angle consideration in Scenario-3 shapes the trajectory for the same hit location. If there exists any type of an obstacle in between the quadcopter and the target, the trajectory shaping is solution for that sense.

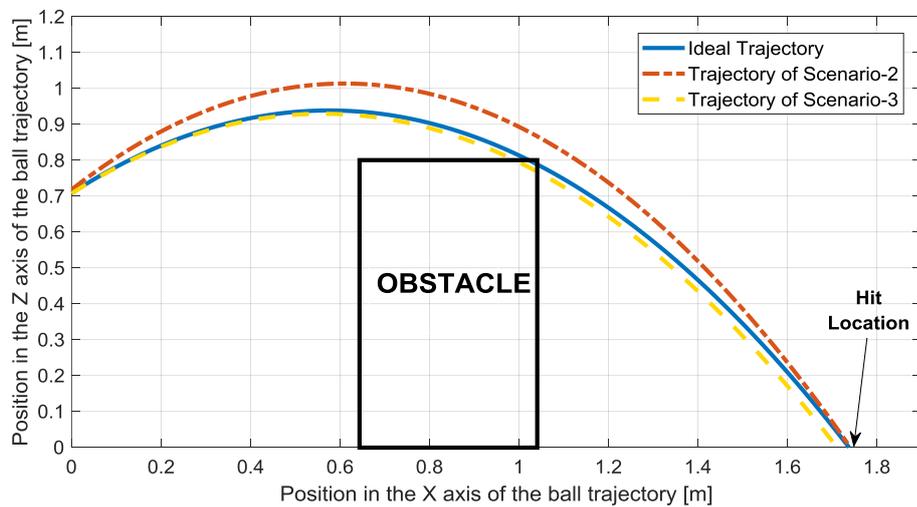


Figure 5.20 Comparison of the Achieved Trajectories with the Existence of an Obstacle

5.2.2.2. Performance Limits of the Trajectory Optimization and Tracking

Scenario-4 and Scenario-5 are both generated in this part. This part is analyzed by two-layered approach. The first layer is to take-off and throw. The second layer is the stabilization and landing. Trajectory is optimized for non-linear region of the angular position of the quadcopter system. Flight duration is set for 8 and 4 seconds for the first and second layer of the trajectory optimization, respectively. Both paths are divided into 10 segments for the first and the second layer of each scenario. A multi-variable logic is implemented for the generation of the shoot signal, which is defined as the tolerance band to each engagement state's nominal value. When the kinematics of the quadcopter system is ready for throwing, the shoot signal is activated. Besides of the engagement states, the shoot signal is also defined in between 7.5th to 8.2nd seconds of the simulation time for the Scenario-5. However, shoot signal is generated in the 8th of the simulation time for the Scenario-4. After the shooting process is completed, mass of the ball is eliminated both in the scenarios. Landing is performed in the final part of the analysis.

5.2.2.2.1. Trajectory Optimization for the Performance Limits of the Quadcopter System in Scenario-4 and Scenario-5

Mission parameters are defined for both Scenario-4 and Scenario-5 as: release angle is set to 45 degrees, target position is relatively located at (5 , - 0.7) m. Engagement states of the quadcopter system are calculated in the Forward Kinematics Algorithm and expressed in Table 5.8.

Table 5.8 Engagement State obtained for the Performance Index Analysis

Parameter	Value
θ	5.3 deg
θ_1	5.2 deg
θ_2	5.2 deg
$\dot{\theta}$	17.2 deg/s
$\dot{\theta}_1$	17.2 deg/s
$\dot{\theta}_2$	17.2 deg/s
Δx	5 m
Δz	0.7072 m
x_q	-0.0015 m
z_q	1.03 m
\dot{x}	4.43 m/s
\dot{z}	4.58 m/s
V_{xball}	4.64 m/s
V_{zball}	4.64 m/s
Energy of the motion	46.45 J
ITAE	0.00037

Energy of the quadcopter system in the forward kinematics algorithm is expressed in Figure 5.21. Error tolerance of ITAE is chosen to be strict in order to have a precise target engagement. The engagement states are found in the 6th iteration of the acceptable error tolerance step.

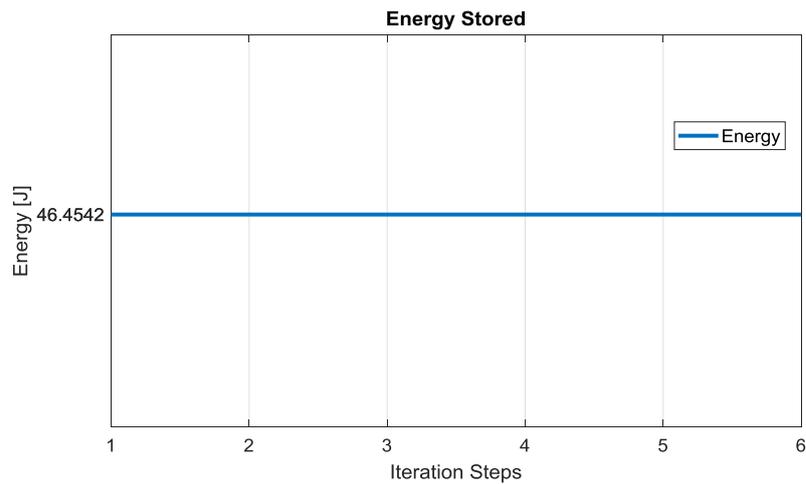


Figure 5.21 Energy of the Quadcopter System in the Forward Kinematics Solution for Scenario-4 and Scenario-5

Trajectory reference inputs of the Scenario-4 and Scenario-5 are generated according to the boundaries of the path which are expressed in Table 5.9 and Table 5.10. Two layers of the trajectory is approached by a different set of path boundaries.

Table 5.9 Boundaries of the Path for the First Layer of the Performance Index Analysis

	x	z	θ	θ_1	θ_2	\dot{x}	\dot{z}	$\dot{\theta}$	$\dot{\theta}_1$	$\dot{\theta}_2$
v_{min}	0	0	-70°	0°	0°	-1	-2	-120°	0°	0°
v_{max}	4	2	70°	45°	45°	5	5	120°	60°	60

Table 5.10 Boundaries of the Path for the Second Layer of the Performance Index Analysis

	x	z	θ	θ_1	θ_2	\dot{x}	\dot{z}	$\dot{\theta}$	$\dot{\theta}_1$	$\dot{\theta}_2$
v_{min}	0	0	-70°	0°	0°	-3	-3	-120°	-60°	-60°
v_{max}	4	2	70°	45°	45°	5	5	120°	60°	60

5.2.2.2.2. Controller Performance

Controller performances are compared for the same reference inputs, however the simulation environment in Scenario-5 is modified by enabling the sensor subsystem to observe the controller performance limits.

Trajectory of the quadcopter system in the x-z plane is presented in Figure 5.22.

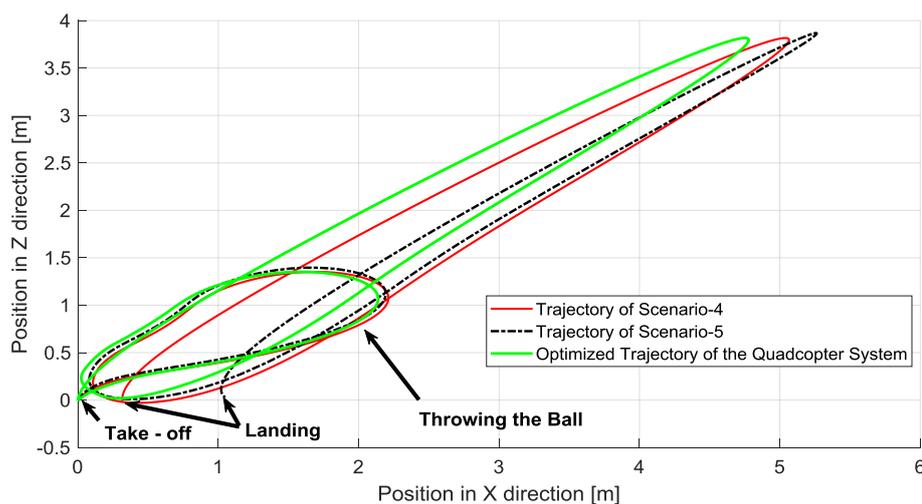


Figure 5.22 Flight Path Comparison of the Quadcopter System for Scenario-4 and Scenario-5

Comparison of the generated trajectory and the achieved x-position of the quadcopter system is shown in Figure 5.23. There is a shift in the landing position of the Scenario-4 and the Scenario-5, which are 0.4m and 1m, respectively.

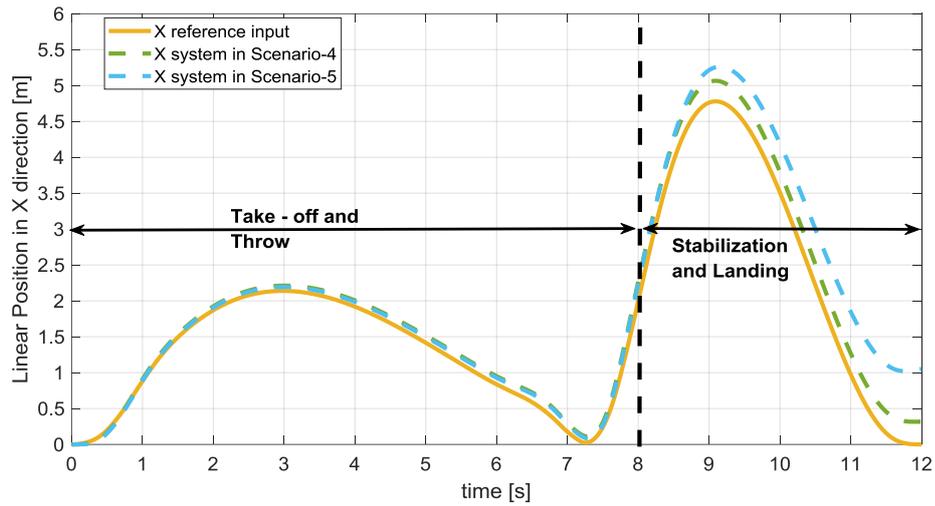


Figure 5.23 Linear Position Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

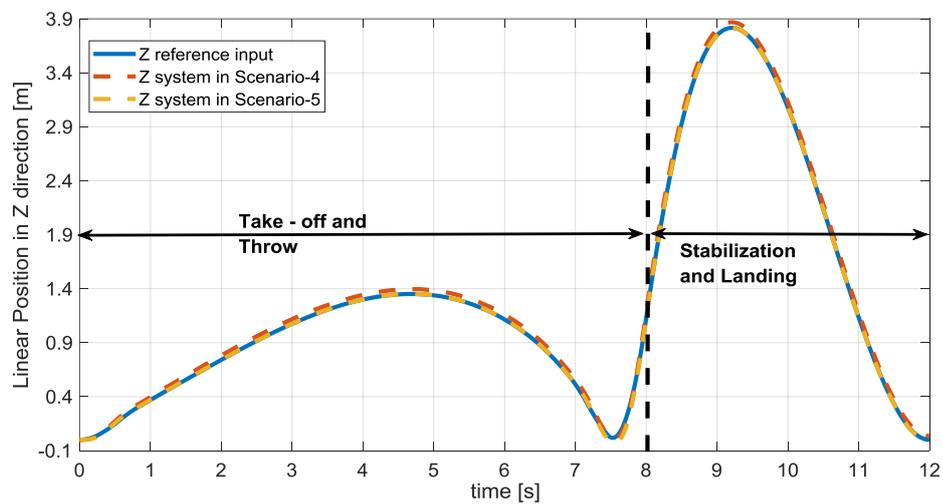


Figure 5.24 Altitude Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

Figure 5.25 presents the time evolution of the angular position tracking. Trajectory is generated in the region of -40 degrees to 45 degrees of Euler pitch reference input. Quadcopter system traces the Euler pitch reference input satisfactorily. However, Euler pitch rate tracking is limited for Scenario-4 and Scenario-5, as in Figure 5.30. Reference angular rate for link-1 and link-2 are presented in Figure 5.31 and 5.32, respectively. The angular rate control in the Scenario-5 is marginally stable because of the existence of the sensor subsystem. In addition, Scenario-4 is limited to track the angular rate reference input.

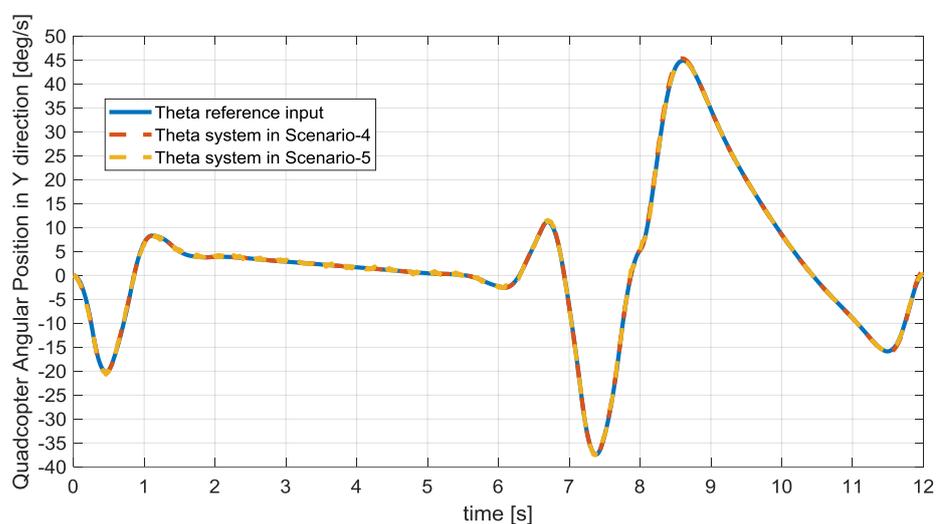


Figure 5.25 Euler Pitch Angle Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

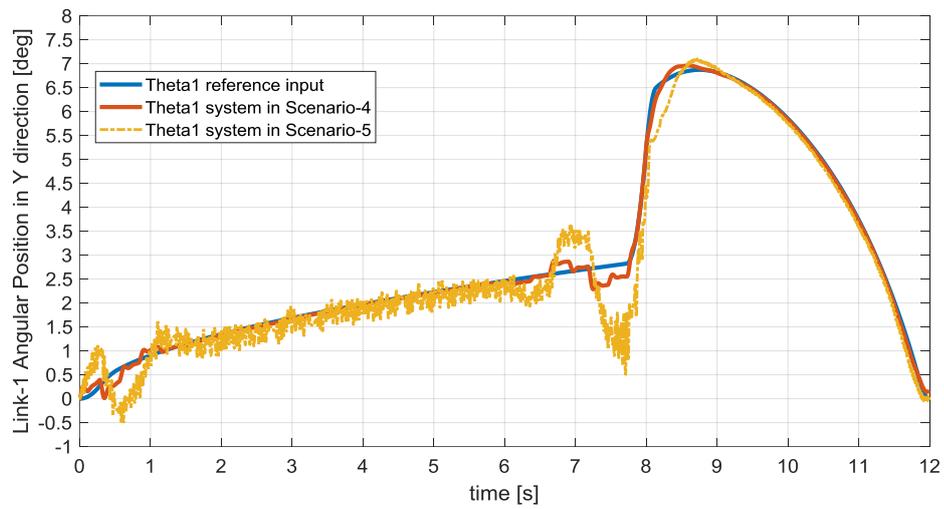


Figure 5.26 Link-1 Angular Position Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

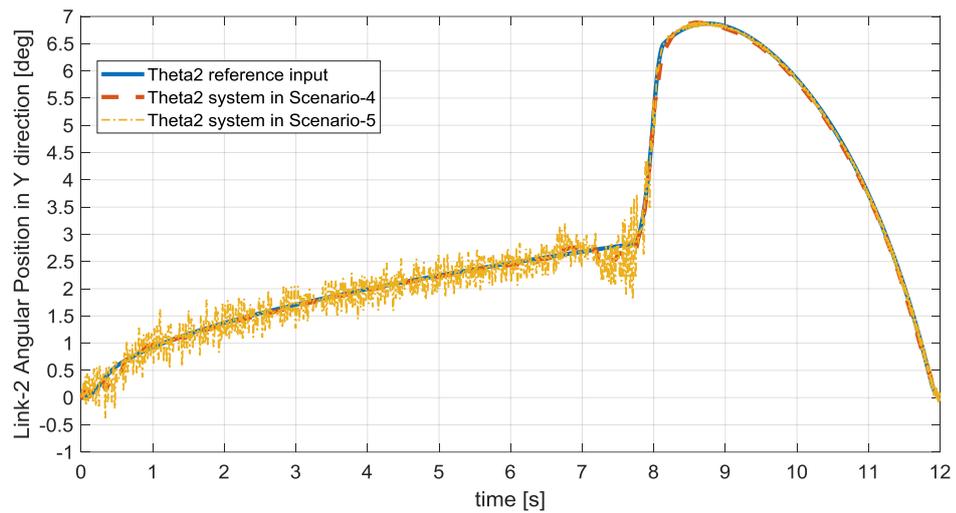


Figure 5.27 Link-2 Angular Position Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

Linear velocity reference input tracking performances of the controllers are presented in Figure 5.28 and Figure 5.29.

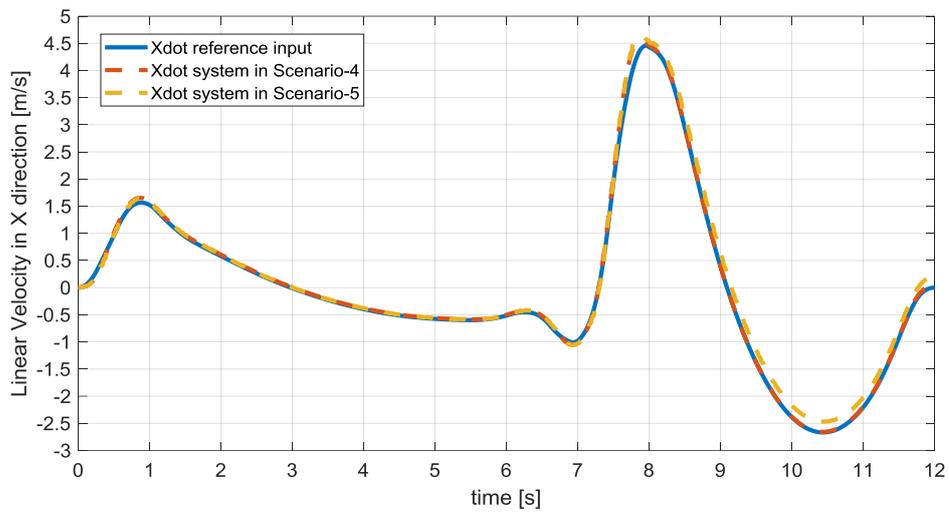


Figure 5.28 Linear Velocity in X-Direction Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

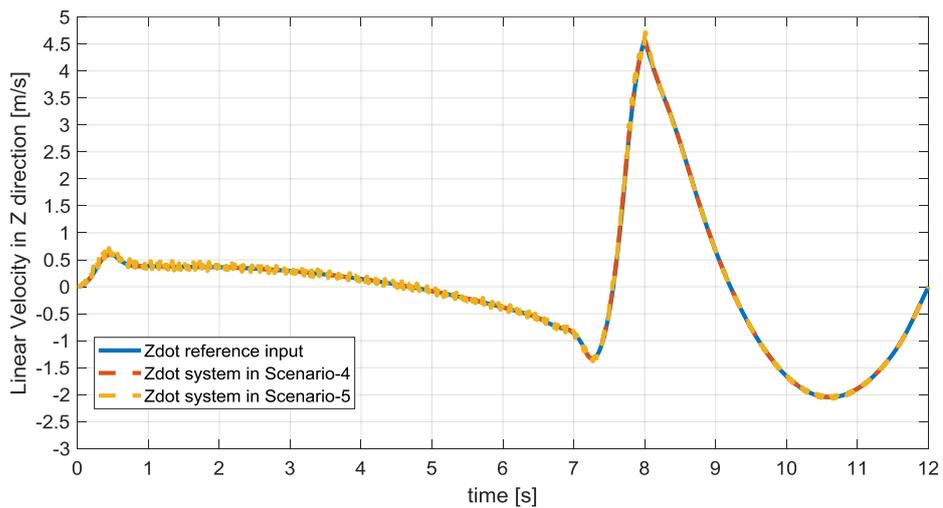


Figure 5.29 Linear Velocity in Z-Direction Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

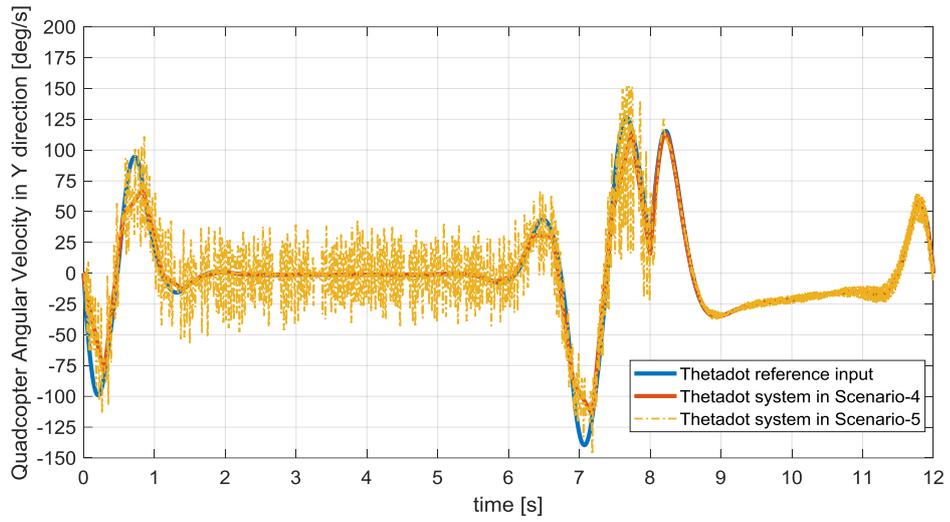


Figure 5.30 Euler Pitch Rate Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

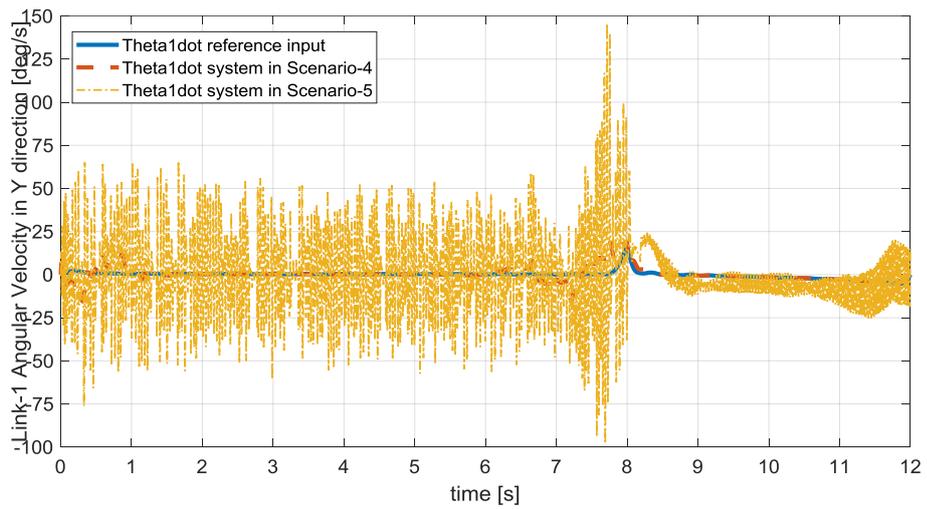


Figure 5.31 Link-1 Angular Velocity Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

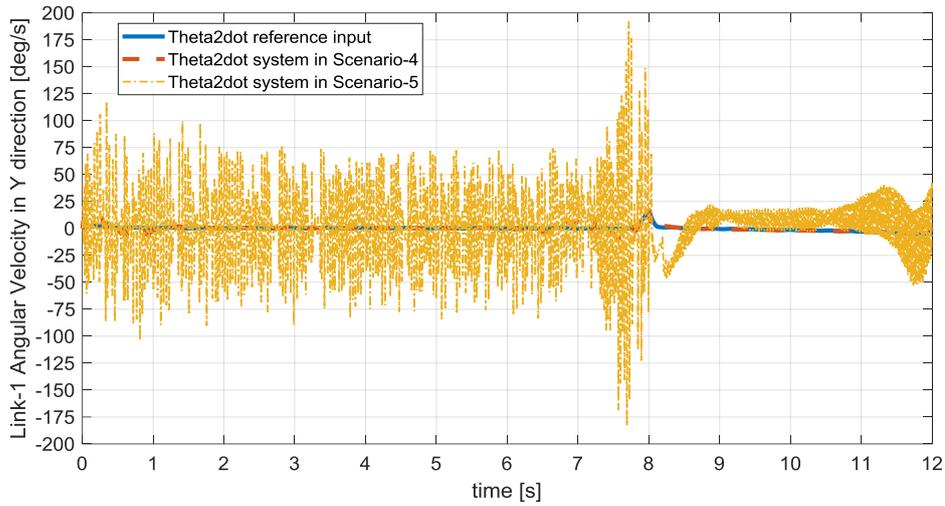


Figure 5.32 Link-2 Angular Velocity Reference Input and System Response for Performance Index Analysis in Scenario-4 and Scenario-5

5.2.2.2.3. Analysis Performed for the Energy Consumption and Trajectory of the Ball in Performance Limits

Same steps in Chapter 5.2.2.1.3 are followed for the energy consumption comparison analysis.

Achieved total energy consumption according to the results obtained in Scenario-4 as.

$$E_{quad-1} = 1112 \text{ J}$$

$$E_{Links-1} = 9.63 \times 10^{-4} \text{ J}$$

$$E_{throw-1} = 43.86 \text{ J}$$

$$E_{total-1} = 1155.86 \text{ J}$$

Achieved total energy consumption according to the results obtained in Scenario-5 as.

$$E_{quad-2} = 1912 \text{ J}$$

$$E_{Links-2} = 1.489 \times 10^{-1} \text{ J}$$

$$E_{throw-2} = 46.96 J$$

$$E_{total-2} = 1959.109 J$$

There is a remarkable difference of the total energy consumption in Scenario-4 and Scenario-5. Trajectories of the ball in Scenario-4 and Scenario-5 are compared with the ideal trajectory in Figure 5.33. System is pushed to the limits. However, there is less than 1 meter of shift on the hit location for the Scenario-5. In order to release the ball with a correct timing, the multi-variable logic is used for satisfying the engagement state. The inertial position of the throwing directly effects on hit location. However, ball is travelled almost 5 meters in the x direction.

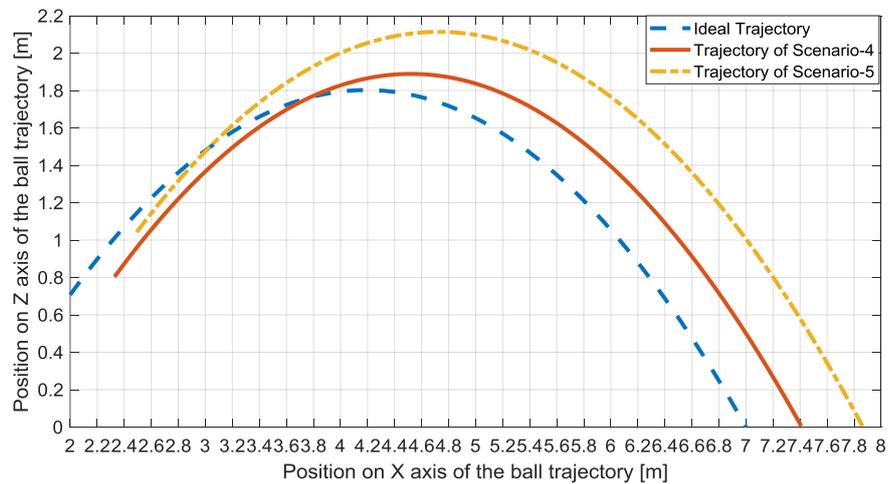


Figure 5.33 Trajectory of the Ball for the Performance Index Analysis

CHAPTER 6

PHYSICAL SYSTEM IMPLEMENTATION

6.1. System Hardware and Software Details

In this section, the system hardware and software implementation are explained. The system consists of controller cards, mechanical subsystems and the other system operational units which as follows:

- Flight Controller Card
- Flight Assistance Card
- Indoor UWB Localization
- Robotic Arm
- Voltage Regulators
- Electronic Speed Controllers (ESC)
- RC servo Motors
- Brushless DC Motors
- Propellers
- Li-Po Battery
- Power Distribution Card

Physical system of the ball throwing quadcopter is shown in Figure 6.1 and Figure 6.2.

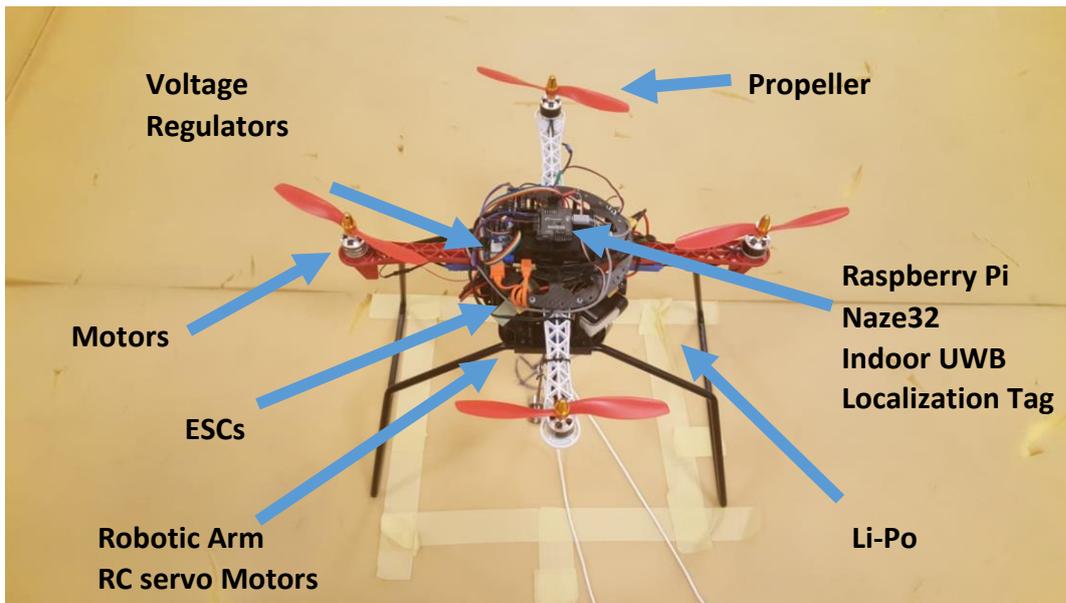


Figure 6.1 Physical System of the Ball Throwing Quadcopter

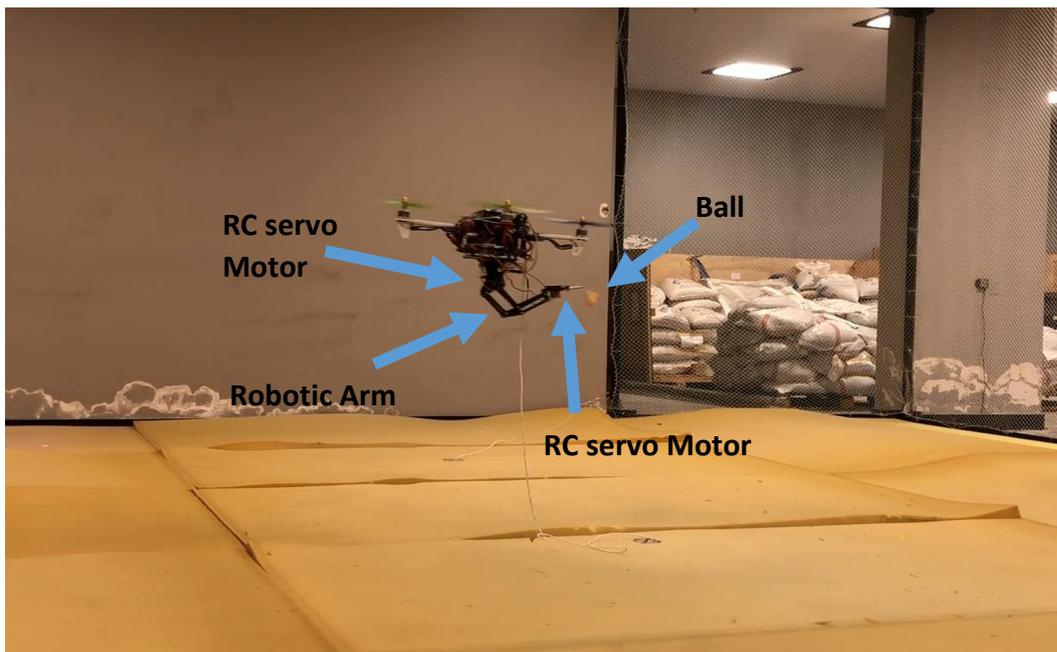


Figure 6.2 Physical System of the Ball Throwing Quadcopter on Duty

System is controlled by a Raspberry Pi 3 controller. It is a Linux based system and computer can access via wireless connection to the Raspberry Pi 3 controller. Computer connection between Raspberry Pi 3 controller enhances the software adjustments by online.

FCA is embedded into the Raspberry Pi 3 controller. FCA control signals are sent to Naze32 assistance controller. I/O ports of Naze 32 assistance controller are assisted to ESCs for driving brushless DC motors properly. Naze32 assistance controller parameters are tuned on Multiwii Software Platform [40]. This platform provides the emergency and RC output signals where the assistance modes are enabled. The robotic arm is controlled by FCA in real time.

Also, signals generated for system control are stored and data is logged after each flight. Then, data logging is used for post process-flight replay and further FCA parameter improvement applications.

Li-Po battery provides system voltage with a sufficient nominal value. Li-Po nominal voltage is regulated to operational voltage requirement of subsystems. These subsystems can be considered as RC servo motors of the robotic arm and the electronic cards used in the system.

Software and hardware configuration is illustrated on Figure 6.3.

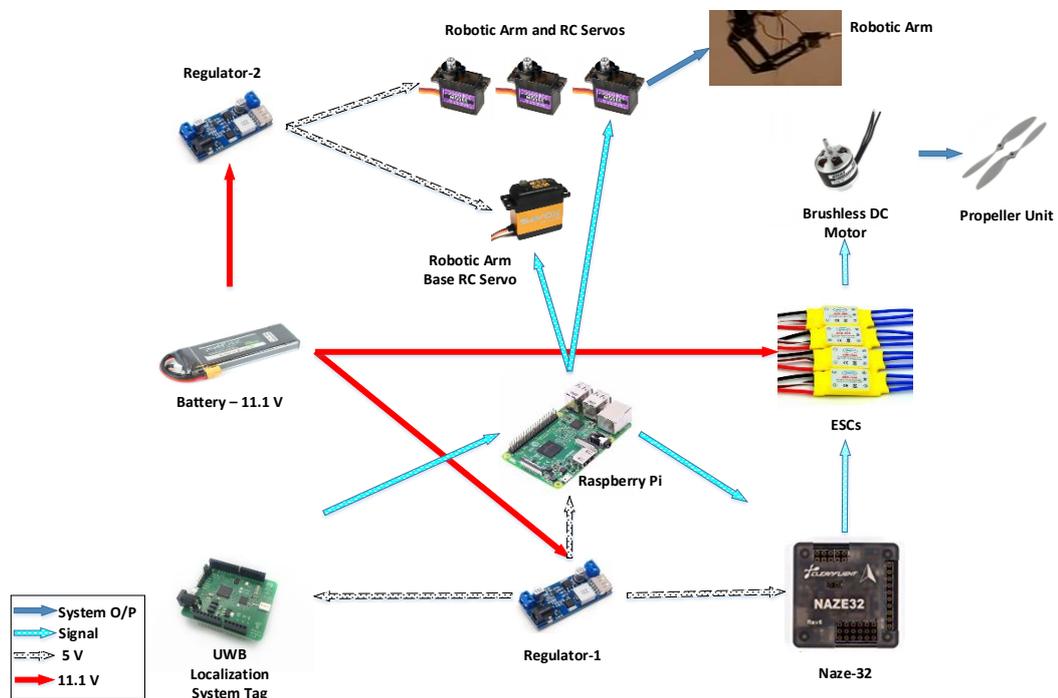


Figure 6.3 Software and Hardware Configuration

6.1.1. Moment of Inertia Identification

The real physical system characteristic is identified in this part. Bifilar pendulum test environment is prepared. Important parameters considered in the test environment is the rope length and distance from each hang position. Quadcopter is mounted to the dock by tied ropes. The system is shown in Figure 6.4.

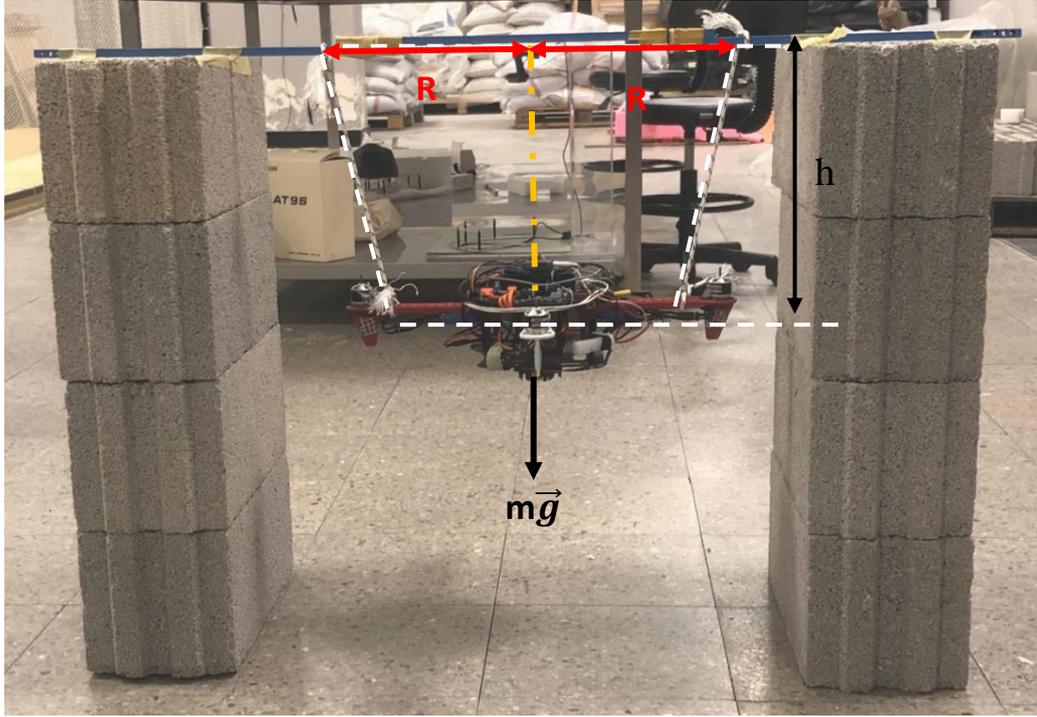


Figure 6.4 Bifilar Pendulum Test Setup

Inertia identification is done using Eq (5.1). The test setup and the configuration for inertia estimates are developed from [42]. Euler roll and yaw angles are collected by IMU of Naze32 in the experiment, the signals are shown in Figure 6.5 and Figure 6.6. Frequency of each sinusoidal Euler angles is defined as “ f_i ”, “ m ” is the mass of the quadcopter, “ d ” is the distance between two ropes, “ g ” represents magnitude of gravitational acceleration.

$$Inertia = \frac{mgd^2}{16h\pi^2 f_i^2} \quad (5.1)$$

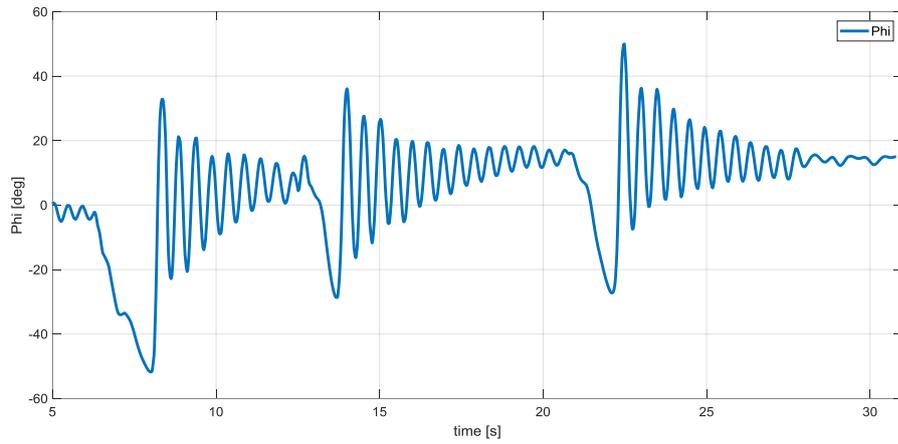


Figure 6.5 Collected IMU Signal output: Euler Yaw Angle

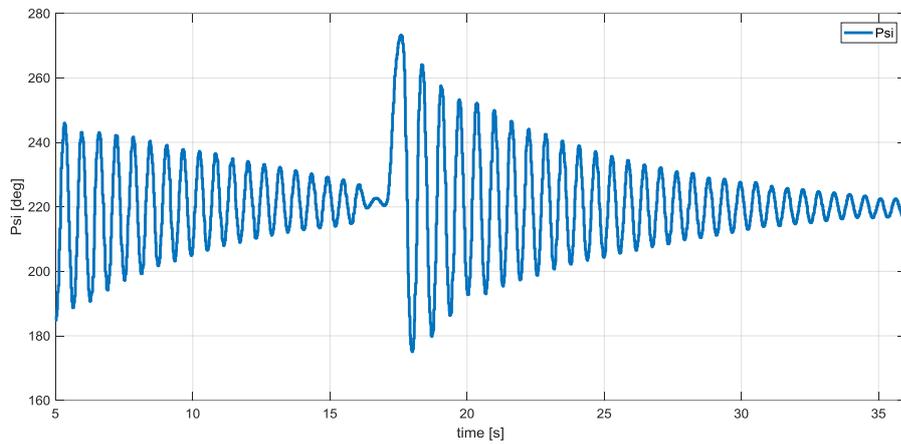


Figure 6.6 Collected IMU Signal output: Euler Yaw Angle

In order to calculate the frequency of the sinusoids while eliminating the bias shift on the IMU, Euler angles are differentiated. Frequency of each sinusoidal Euler angles are found by differentiated input signals expressed in Figure 6.7 and Figure 6.8. Frequency calculation is made for each period. Average inertia values are also shown in Figure 6.9 and Figure 6.10.

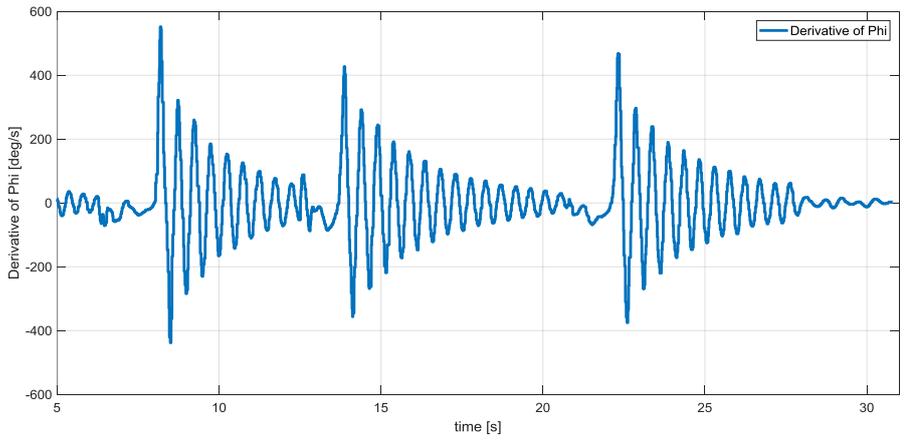


Figure 6.7 Differentiation of Euler Roll Angle

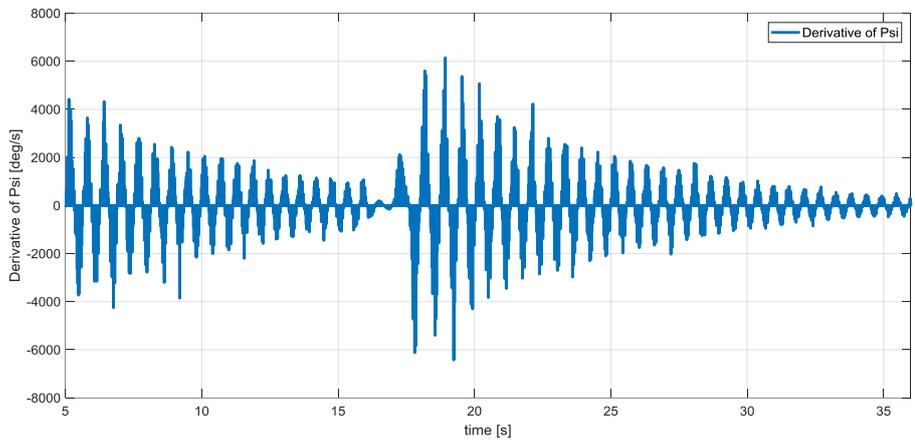


Figure 6.8 Differentiation of Euler Yaw Angle

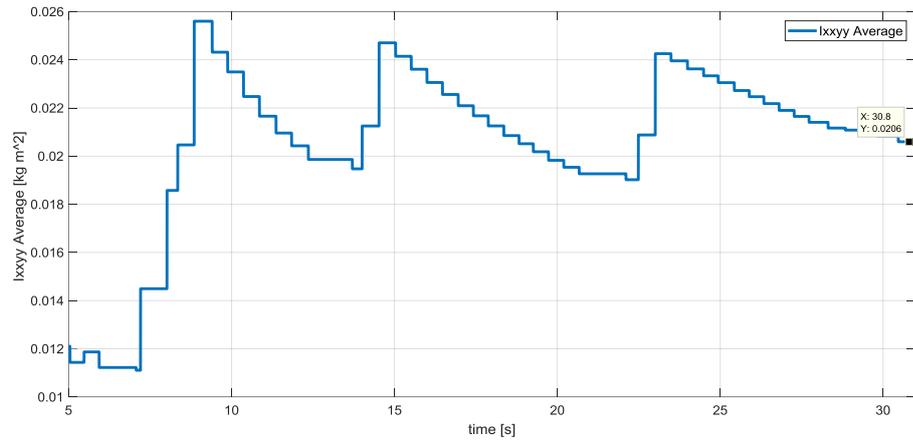


Figure 6.9 Average Inertia for X and Y plane

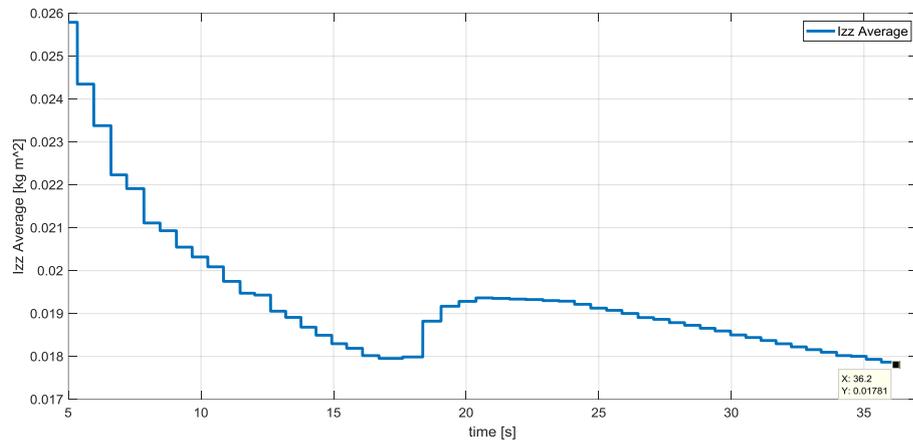


Figure 6.10 Average Inertia for Z plane

Quadcopter x and y plane inertia values are considered to be as equal to each other due to the symmetric body of the quadcopter. The inertia matrix of the real system is assumed to be diagonal. After all, the last value of the averaged signal data shown in Figure 6.9 and Figure 6.10, are considered to be as the system inertia values which are expressed in Table 6.1.

Table 6.1 Identified Inertia Values of the Quadcopter

Identified Inertia Values	[kg.m ²]
$I_{q,x}$	0.0206
$I_{q,y}$	0.0206
$I_{q,z}$	0.01781

Inertia of the links are approximated as rectangular prism and inertia matrices of each link are found by using the following formula. Table 6.3 is used for that purpose.

$$I_{i,x} = \frac{1}{12} m_i (c_i^2 + d_i^2)$$

$$I_{i,y} = \frac{1}{12} m_i (b_i^2 + d_i^2)$$

$$I_{i,z} = \frac{1}{12} m_i (b_i^2 + c_i^2)$$

Where $i = 1$ and 2 .

The edge of the prism is assumed to be as square. “ c_i ” which is width of the i^{th} link. The length of the links are shown as “ b_i ”.

$$d_i = c_i$$

The inertia matrices of the links are assumed to be diagonal and expressed in the following form.

$$\hat{I}_i = \begin{bmatrix} I_{i,x} & 0 & 0 \\ 0 & I_{i,y} & 0 \\ 0 & 0 & I_{i,z} \end{bmatrix}$$

Table 6.2 Inertia of the Links

Inertia Values	$[kg.m^2]$
$I_{1,x}$	2.5×10^{-5}
$I_{1,y}$	7.3×10^{-5}
$I_{1,z}$	7.3×10^{-5}
$I_{2,x}$	3.33×10^{-5}
$I_{2,y}$	1.13×10^{-4}
$I_{2,z}$	1.13×10^{-4}

6.1.2. Robotic Arm

2-DOF Robotic arm is integrated for the pitch axis of the quadcopter. Base of the robotic arm is driven by mini RC servo mounted to the quadcopter bottom end. A micro RC servo is mounted to the robotic arm mechanism and another micro RC servo is assembled for the motion of the end effector with a ‘C’ type RC servo bracket. Robotic arm configuration is driven by 1 mini and 2 micro RC servos.

The integrated physical system is detailed in Figure 6.11.

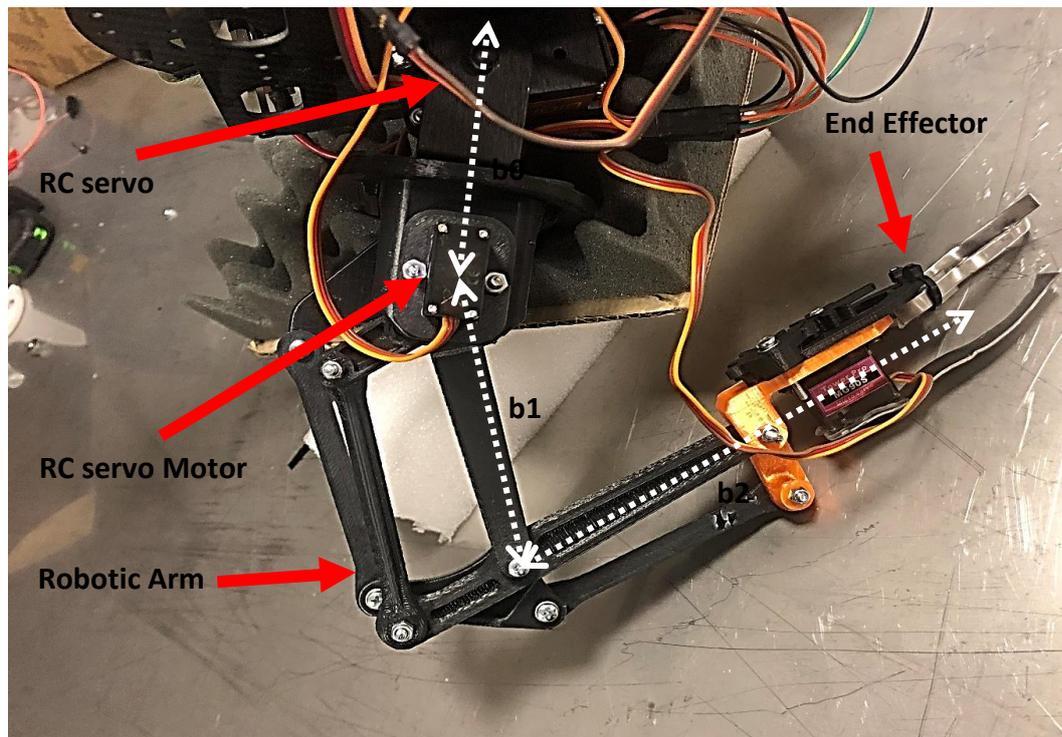


Figure 6.11 Configuration of the Robotic Arm Mounted at the Bottom of the Quadcopter

Dimensions of 2-DOF robotic arm is given in Table 6.3. Robotic arm links are assumed to be as rectangular prism with a square edge. “ b_{number} ” in Figure 6.11 represents the length of the indexed link of the arm.

Table 6.3 Link Length of the Robotic Arm

	Length [m]	Width [m]
Link-0	0.1	0.05
Link-1	0.11	0.05
Link-2	0.12	0.05

Characteristics of the RC servo motors used in the robotic arm is given in Table 6.4.

Table 6.4 Detailed Information about Robotic Arm's RC servo motors

RC servo Model	Savöx SH-1290MG	TowerPro MG90-S
Usage	Main Pitch Motion Generator on the Base	Arm's pitch motion & End effector mechanism
Operating Voltage	5V	5V
$\omega_{operating}$	0.05 second / 60°	0.1 second / 60°
$T_{supplied}$	0.049 kg.m	0.025 kg.m
Size	4.03 x 2.02 x 3.72 cm	2.31 x 1.22 x 2.9 cm
Weight	56.1 g	14 g

6.1.3. Quadcopter System Flight Controller

General usage of the flight controller card divided into 2 parts. The first part is the embedding the FCA designed on MATLAB/Simulink. The second part is the quadcopter subsystems communication protocol which is written on python script [40]. Two of the parts explained are in real time and synchronized. Flight controller hardware is shown in Figure 6.12.

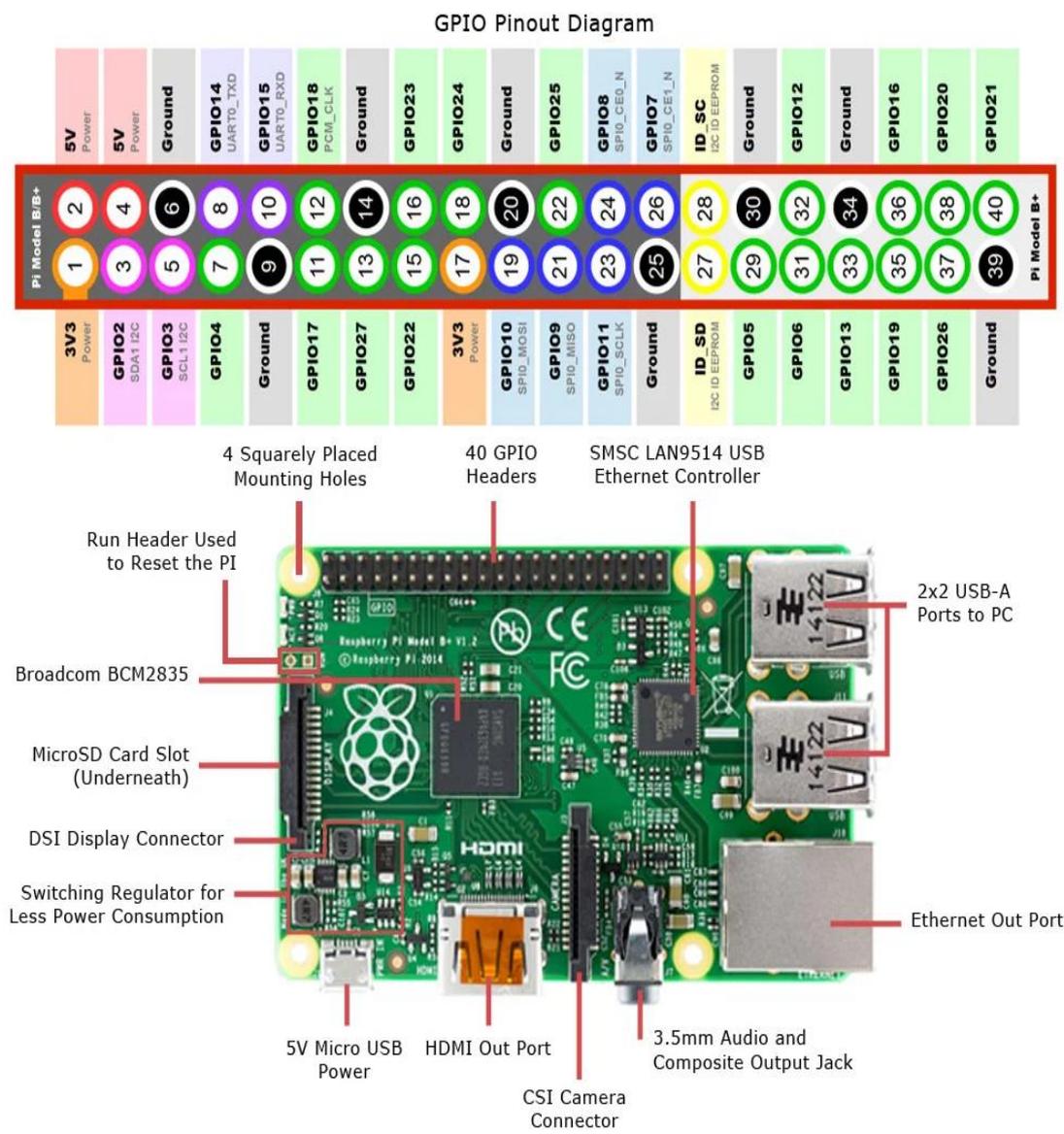


Figure 6.12 Raspberry Pi 3 Hardware and GPIO Pinout Diagram [59]

Controller card performance specifications are detailed in Table 6.5

Table 6.5 Quadcopter Flight Controller card performance specifications

CPU	QuadCortex A53 1.2 Ghz
Instruction Set Architecture	ARMv8-A
GPU	400 MHz - VideoCore 4
RAM	1GB – SDRAM
Storage	MicroSD – 16 GB is used
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
PINs	40 – Specific ones with PWM signal output
Other Output Units	Video and Audio
Data Processing Frequency	200 Hz is achieved

6.1.4. Indoor UWB Localization - Pozyx and Data Acquisition

Naze32 - IMU and accelerometer units include great amount of bias and noise level. That brings the system to uncontrollable flight condition. In order to enhance the flight control performance and the scenario generation, main feedback unit for the FCA is chosen as Pozyx hardware also known as indoor UWB localization hardware.

Indoor UWB localization system collects the linear position, Euler angles and Euler rate information all together. System consists of 2 elements:

- Pozyx Tag: Euler angles and Euler rate information
- Pozyx Anchor: Linear position information

Pozyx Indoor UWB localization system units; tag and anchor are shown in Figure 6.13 and Figure 6.14.



Figure 6.13 Indoor UWB Localization System Unit Pozyx Tag [60]



Figure 6.14 Indoor UWB Localization System Unit Pozyx Anchor [60]

Position data collection is handled by 4 Pozyx Anchors mounted on the wall. To get the best position information, anchors are mounted as if there are not forming a plane specifically. Flight area is set according to the locations of the anchors. Configuration is shown in Figure 6.15.

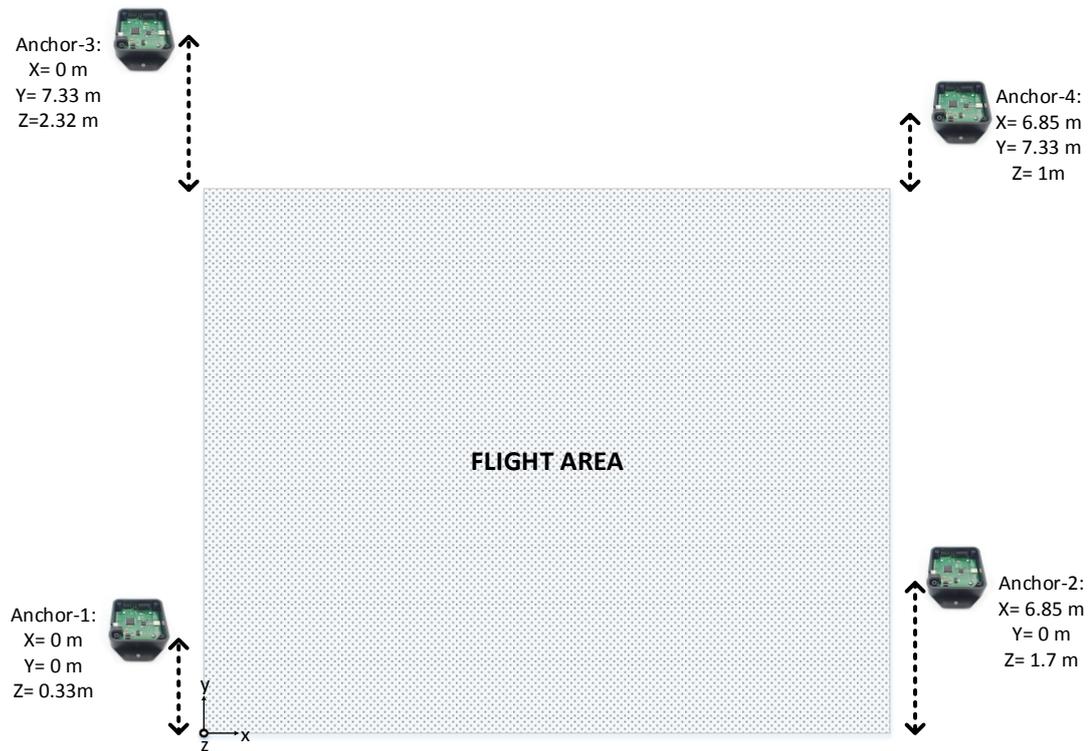


Figure 6.15 Flight Area Specified by the Locations of Anchors

Pozyx Tag is the main onboard IMU connected via USB to the flight controller card and collects the feedback information for FCA. Pozyx system is using Ultra Wideband technology with an adjustable filtering options and machine learning methods. Tag communicates with the anchors and collects the position information. However, angular position-rate information is collected by Tag using its sensor configuration.

This system is also compatible with Linux based controller cards such as Raspberry Pi 3. The available manufacturer's Python Library makes data exchange protocol with the Flight Controller Card and Indoor UWB localization unit easier.

Anchor communication speed and data accuracy directly related with the type of filter used during flight. System can provide position information with a FIR filter in the order up to 10th degree. This is adjusted offline by a computer. Required type of configuration is handled by the connection between computer with tag and anchors via USB. Pozyx Device Configurator is shown in Figure 6.16.

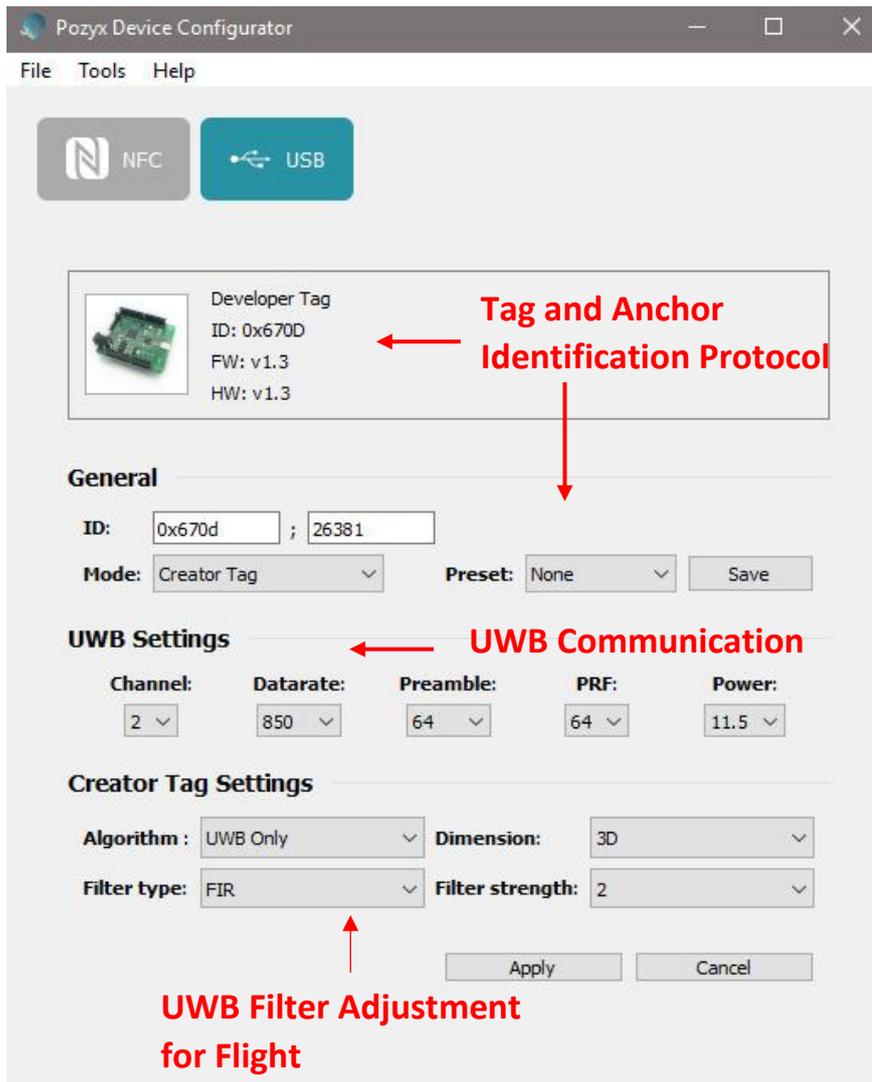


Figure 6.16 Indoor UWB Localization Configuration Adjusted For Flight

The configuration shown in the figure implies such a performance index that, Indoor UWB Localization system has a position error in the range of 10cm in the x - y direction and 20cm in the z direction. That error amplitude is satisfactory for indoor quadcopter applications. In addition, that amount of error is observed by 65 to 100 Hz of frequency in linear position feedback. Data is collected for the same configuration selected for Anchors. Furthermore, angular position-rate information is obtained around 200 Hz.

6.1.5. Quadcopter System Signal Assistance Controller

Flight control signals generated by Raspberry Pi 3 controller are sent to the assistance controller card via USB. In this study, Naze 32 controller is used as an assistance controller card for driving the rotors. In addition, assistance controller helps for safe landing if emergency situation is occurred during flight.

Naze 32 controller outputs the generated roll, pitch, yaw and altitude channel motor control commands by main controller card to the ESC for driving the rotors. Pin diagram is shown in Figure 6.17.

Naze 32 Revision 6 Pinout

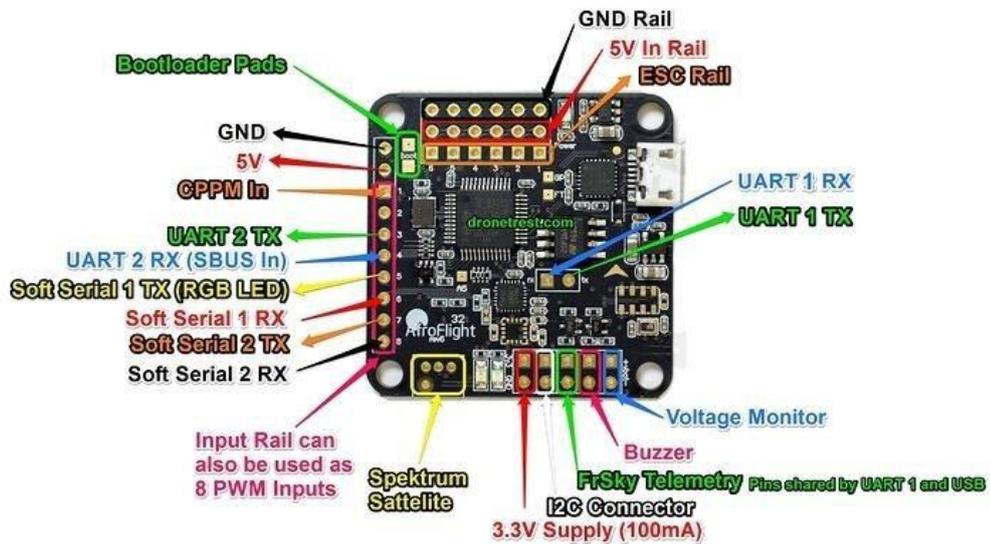


Figure 6.17 Naze-32 Pin Diagram [40]

Technical capabilities of the flight assistance controller card are given in Table 6.6.

Table 6.6 Naze 32 Rev 6 Technical Details

CPU	32 – bit Arm Cortes M3 72 MHz
IMU	MPU 6050 - MEMS
Barometer	BMP280
Weight	7.3 g
Output Pins	PWM supported, SBUS inverter
Size	3.6 x 3.6 cm
Data Processing Frequency	300 Hz

6.1.6. Power Distribution

Power distribution is resourced by 11.1 Volts Li-Po battery during flight. Direct Li-Po voltage output is parallel distributed to ESCs. However, Flight Controller Card, Quadcopter Signal Assistance Controller Card, Indoor UWB Localization Pozyx Card and RC servo Motors are fed with 5 volts by using two voltage regulators. Simple power distribution schematics is shown in Figure 6.18.

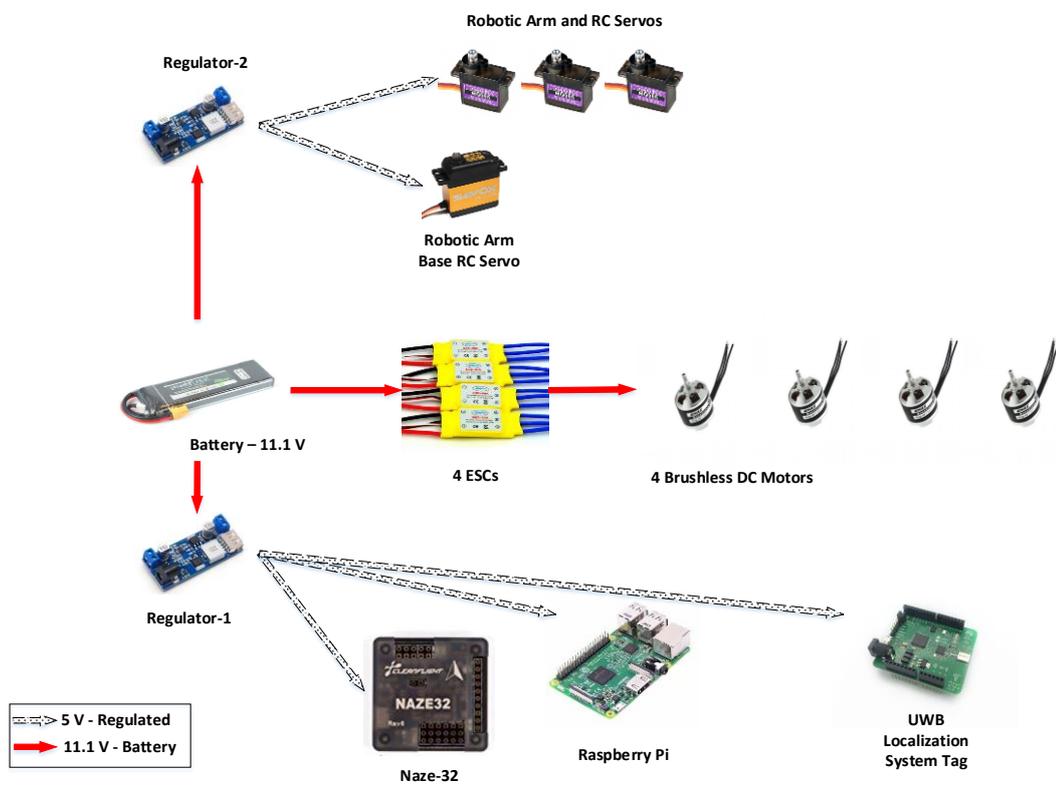


Figure 6.18 Power Distribution Schematics of the Quadcopter System

6.1.7. Detailed Information about the Other Subsystems

In this section, the other subsystems are detailed. Each elements are selected for the minimum weight and maximum agility characteristics of the quadcopter system for the flight.

6.1.7.1. Battery

Quadcopter main energy resource is the battery. Li-Po type battery is chosen due to compact usage. Voltage and the current need for the actuators and the controller cards are fed with the 11.1 Volts version. Battery details are given in Table 6.7.

Table 6.7 Quadcopter System Energy Resource Unit

Capacity	3400 mAh
Voltage	11.1 V
Max Continuous Current Supply	25 Ampere
Weight	240 g
Dimensions	13.5 x 4.4 x 1.9 cm

Picture of the selected Li-Po battery is shown in Figure 6.19.



Figure 6.19 Pro-Fuse battery type for the flight [40]

6.1.7.2. ESCs

Each Brushless DC motor is driven by ESC on the system. Selected type of ESC is in the operating limits of the brushless DC motors. Four ESCs are used in the Quadcopter system. Operating performance information of ESC is expressed in Table 6.8.

Table 6.8 Operating performance information of the Quadcopter's ESCs

Voltage Input Range	8.7 – 12.5 V
Instant Current Output	40 A for 10 seconds
Operating Current Output	30 A
ESC External Output	1 A / 5 V
Weight	25 g
Size	4.5 x 2.4 x 1.1 cm

Picture of the selected ESC is shown in Figure 6.20.



Figure 6.20 ESCs of the Quadcopter System [40]

6.1.7.3. Brushless DC Motors and Propellers

Proper Brushless DC is selected according to the voltage and current suitability of the battery and ESCs. Motors generate thrust force based on propeller size. Single motor - propeller information is detailed in Table 6.9.

Table 6.9 Technical Information of Motor and Propeller Used in the System

Voltage	11.1 V
Propeller Size	10 inches (25.4 cm)
Maximum Current	15.1 A
Maximum Thrust Force	0.88 kg
Power Requirement	181.2 W
Efficiency	4.9 g/W
RPM	6960

Picture of the selected motor and propeller is shown in Figure 6.21 and Figure 6.22.



Figure 6.21 EMAX Brushless DC Motor [40]



Figure 6.22 10" Propellers of the Quadcopter System [40]

6.1.7.4. Voltage Regulator

Voltage output of the Li-Po battery is fed parallel to the voltage regulators. Voltage - Current requirements of Flight Controller Card, Quadcopter Signal Assistance Controller Card, Indoor UWB Localization Pozyx Card and RC servo Motors are overcome directly by XY-3606 regulator. Two regulators are used for the system due to the current output limit of single unit.

Voltage regulator of quadcopter system is shown in Figure 6.23 and technical specs are expressed in Table 6.10.

Table 6.10 Technical Specs of the Voltage Regulator

Voltage Input	11.1 V
Output Voltage	5 V
Maximum Current Output	6 A
Weight	22 g

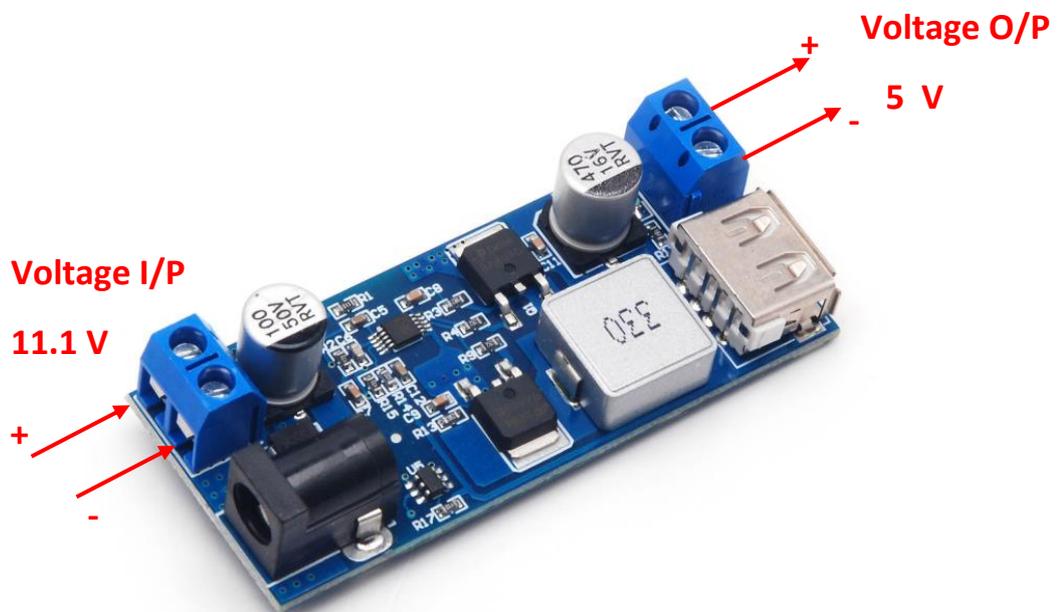


Figure 6.23 XY-3606 Voltage Regulator [40]

6.1.8. System Software

Real physical quadcopter system consists of many subsystems. Main controller which is Raspberry Pi 3 controller calculates the control commands. Then the control signal package is sent to the Naze-32 controller. Flight assistance card sends motor signal to the ESC. Then, ESC directly amplifies the input signals to output signals in order to drive the motors.

Robotic arm's motion is carried out by the "Shoot" signal generation. This signal is sent directly from the flight controller card to RC servo motors.

In this part, control signal generation and the other algorithm sub-elements for a robust flight are detailed.

6.1.8.1. Controller Architecture

The controller of the quadcopter system presented in Chapter 3.3 is implemented to the real system as FCA. Controller architecture is implemented on Matlab/Simulink. FCA is designed for autonomous flight with the capability of 200Hz processing speed using a solver type of ode2 - Heun. The algorithm is embedded to Raspberry Pi 3 controller. Controller card also collects the feedback information which is needed to track the control commands. Additionally, FCA has the capability of user interaction part. This is designed for making modifications on controller inputs and precautionary actions.

Control structure of the FCA is sketched in Figure 6.24. Controller parameters of FCA are presented in Table 6.11.

Table 6.11 Controller Parameters of the FCA

	P gain	I gain	D gain	Controller output limit
Altitude Controller	3	0.01	0.1	\pm %2.5 PWM/Duty
X Position Controller	3	1	0.005	8deg
Y Position Controller	4.5	1	0.1	13deg
Phi Angle Controller	16	0.05	-	70 deg/s
Theta Angle Controller	16	0.05	-	70 deg/s
Psi Angle Controller	4	0.01	-	25deg/s
Phi Rate Controller	0.3	0.01	-	\pm %7.5 PWM/Duty
Theta Rate Controller	0.3	0.01	-	\pm %7.5 PWM/Duty
Psi Rate Controller	0.3	0.01	-	\pm %7.5 PWM/Duty

As shown in Figure 6.24, x-y position, altitude and attitude controllers output in each channel is the period of the PWM signal. Duty cycle is proportional with the calculated period of the PWM signal.

Nominal thrust value of the altitude controller is modified by an input transfer function. Energy of the battery decreases by time, the altitude controller is fed by an open loop PWM recovery input in order not to lose the nominal altitude.

Filtering is used in 3 different parts of the FCA. These filters are used for motor command, initial thrust and voltage recovery filter. Filter parameters of motor command filter, initial thrust generator and PWM/Duty recovery input transfer functions are expressed in Table 6.12.

Table 6.12 FCA Filter Parameters

	Natural Frequency	Damping Ratio	Nominal Value
Motor Command Filter	150 Hz	1	Nominal PWM/Duty
Initial Thrust Generation	15 Hz	1	% 60 PWM/Duty
PWM Recovery External Input	2 Hz	1	% 2.5 PWM/Duty

One of the reason behind the motor command filtering is to ignore the instant reverse directional motor rotation command. That protects the motor from warming up. Open Loop Thrust Transfer Function is implied in order to hold the quadcopter altitude while battery is decreasing and losing its effectiveness.

FCA also includes shooting logic for throwing the ball. Dependent variables of the logic are V_x , V_z and quadcopter Euler pitch angle. Velocity of the quadcopter is obtained by using a second order derivative filter. Natural frequency of the derivative filter is the same with the initial thrust generation filter expressed in Table 6.12.

Shooting logic in FCA can be expressed by an example designed for shooting condition which is given in Table 6.13. If the quadcopter states are appropriate for shooting, then PWM/Duty signal is generated for the RC servo motors.

Table 6.13 Parameters and Conditions for Throwing the Ball

Parameter	Condition
V_x	Greater than 2 m/s
V_z	Greater than 2 m/s
θ	Greater than 2 degrees

X-y position, attitude and altitude controller output PWM/Duty signals are sent from Raspberry Pi 3 controller to NAZE 32 controller. When the acceptable shooting logic is satisfied, pre-specified PWM/Duty signal is sent from Raspberry Pi 3 controller to the RC servo motors directly for throwing the ball. NAZE 32 controller listens the PWM/Duty input signal and assists to input signals. Then NAZE 32 controller sends the output signals to ESCs. NAZE 32 controller has also internal precautionary protection mode for safe-landing. In safe landing mode, PWM/Duty output from NAZE 32 controller is cut after something wrong during a flight.

6.1.8.2. Data Collection and Flight Replay

Post process steps is explained in this part. After a scenario generation is completed, then the flight data is collected by the flight controller card. Collected flight data detail is expressed below:

- Linear position and rotational position-rate feedback information, that is listened from Pozyx Tag
- On board calculated control signals
- Flight Assistance IMU data

This information is sent online to the base-computer. Then the logged data is post processed and monitored.

6.1.8.3. Flight Assistance Software

Cleanflight software platform is installed on flight assistance card. Installation and implementation steps are followed [40]. The output pins for PWM signal generation are specified for the ESCs input. SBUS option for communication with flight controller card is set for system direct control from FCA.

6.1.8.4. Ground Test for Bias Collection

The FCA is calibrated before every flight. The calibration of the flight assistance card is made on Cleanflight environment. The flight assistance card is calibrated by holding

the quadcopter with zero offset attitude. That is made by using mini water gage shown in Figure 6.25.



Figure 6.25 Mini Water Gage for Calibrating the Flight Assistance Card

Attitude bias calculation is made in the FCA is handled by taking the average value of the collected data from the Indoor UWB Localization and Naze 32's IMU in a specific period of time. After bias calculation is finished, all the calculated bias values are used during the flight. Attitude bias occurs due to the assembly of the Pozyx tag on the chassis or any type of hard landing – crash afterwards. The correct attitude feedback is obtained by subtracting the calculated bias value from the feedback information. The bias is simply calculated as follows,

$$\textit{Attitude Bias} = \textit{Attitude}_{\textit{Pozyx}} - \textit{Attitude}_{\textit{Naze32}}$$

The position bias is found from the reference position of the tag with respect to the reference base point specified to the indoor UWB Localization system. A sample of a calibration step for bias values of the linear position is calculated as follows,

$$\textit{Position Bias} = \textit{Indoor UWB Localization Output} - \textit{Tape Measure}$$

The bias values of the position vector is observed to be as consistent in x-y plane. However, bias is unpredictable in z direction due to the performance of the 4 anchors for flying robotic applications. Increase in the number of anchors used in the indoor test area might increase the sensor accuracy in the z direction. The bias value on x-y plane is specified and fed to the FCA as a constant tolerance value after calibration. Outputs are shown in Figure 6.26 to 6.28.

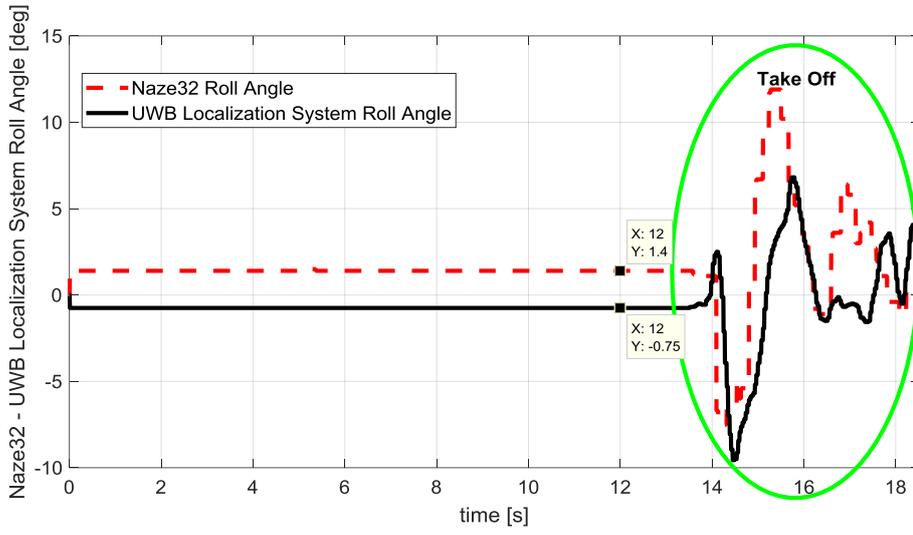


Figure 6.26 Bias Calculation on Roll Channel

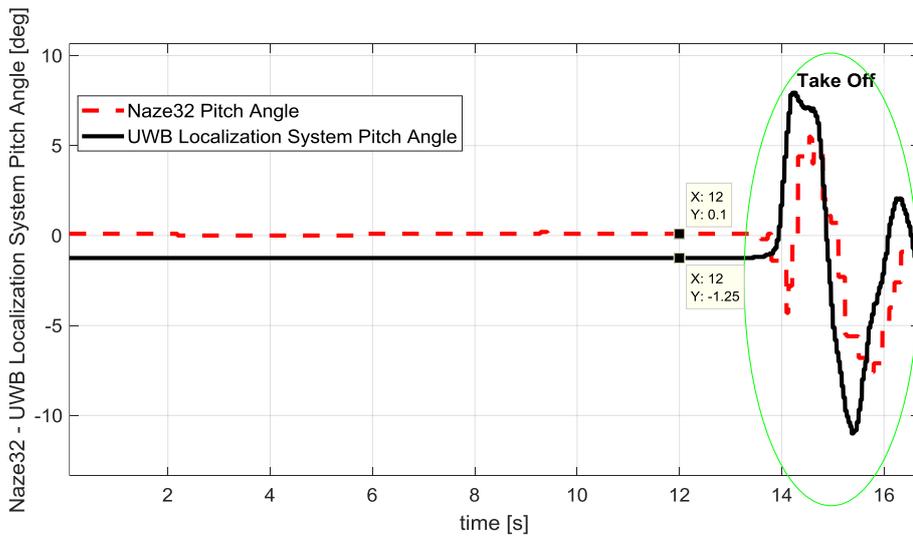


Figure 6.27 Bias Calculation on Pitch Channel

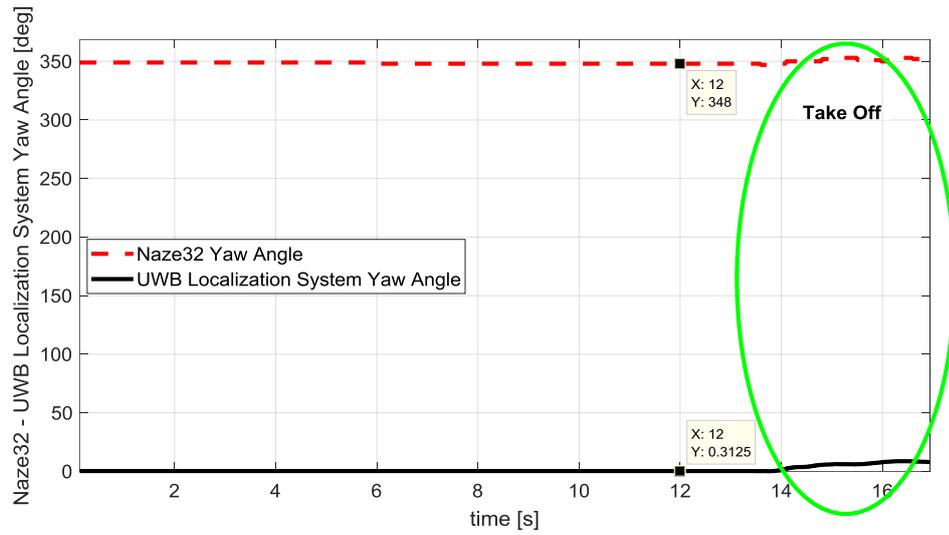


Figure 6.28 Bias Calculation on Yaw Channel

The linear and angular position bias values are written on Table 6.12.

Table 6.14 Bias Values of Quadcopter System

Bias in linear X position	11 cm
Bias in linear Y position	15 cm
Bias in angular X position	-2.15 deg
Bias in angular Y position	-1.35 deg
Bias in angular Z position	12.31 deg

6.2. Experiments

FCA performance is examined in this section. FCA is used for the autonomous tests. Specified test area covered by anchors is shown in Figure 6.29. The quadcopter system during a typical flight is shown in Figure 6.30.

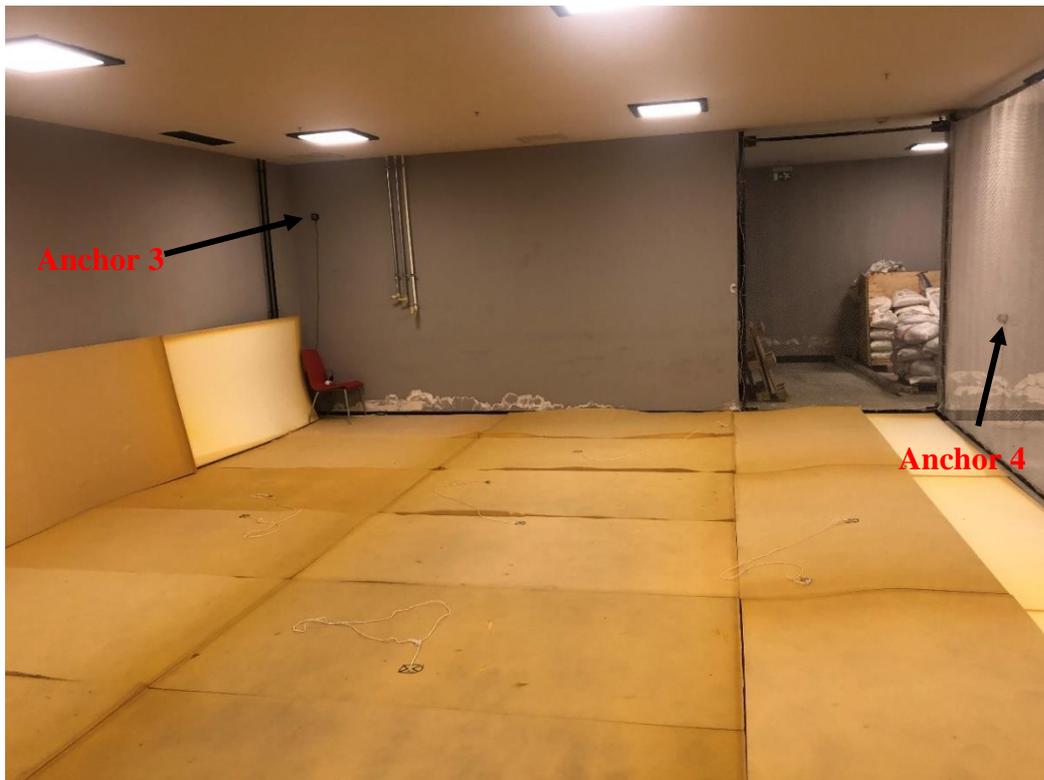


Figure 6.29 Test Area

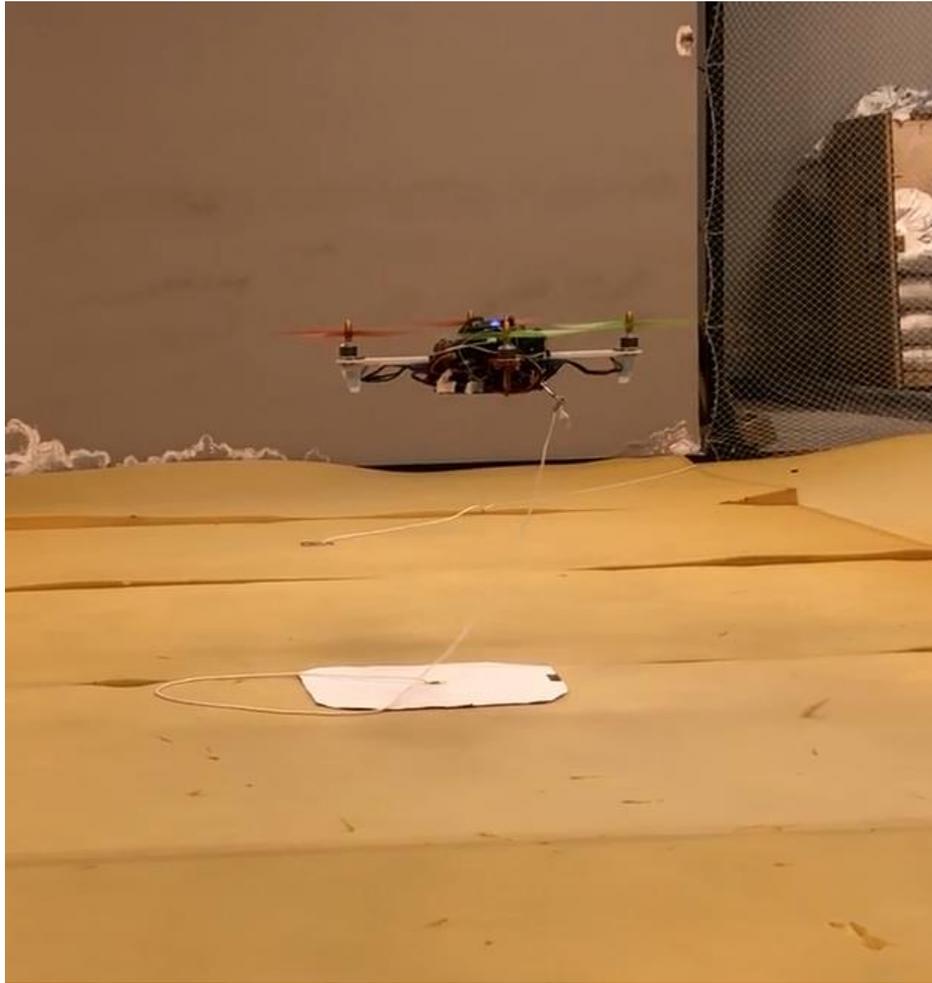


Figure 6.30 The Quadcopter System without the Robotic Arm during a flight

In this part, three different types of experiments are performed. Firstly, the hover performance of the quadcopter system without the robotic arm is examined with low thrust. Secondly, 3-D rectangular path tracking is experimented with the same configuration. In Appendix-B, the robotic arm of the quadcopter system is mounted and tested for the target engagement. The quadcopter system is shown in Figure 6.31.



Figure 6.31 The Quadcopter System

6.2.1. Experiments of the Quadcopter System without the Robotic Arm

6.2.1.1. Hover Test

Altitude control command is fed far from the ground effect region. Difficult part of that test configuration is that, change in the generated thrust force is directly affecting the x-y position controller and Euler yaw controller. A fixed coordinate in 3-D is commanded to the FCA in order to examine the hover performance of the quadcopter. Then, the IMU feedback measure, the control commands of the inner and the outer loop of FCA are examined in the hover configuration.

X-y position error must be in a tolerable level and the quadcopter Euler angles should stay in the linear region, which is around 12 degrees in absolute. Angular rates of the quadcopter must also be held in the lowest possible amplitude.

Controller reference inputs are expressed in Table 6.15.

Table 6.15 Hover Test Reference Input Coordinate in 3-D

X linear position	3.3 m
Y linear position	3.6 m
Z linear position	60 cm
Z angular position	0 deg

Bias in each channel is collected at the beginning of the flight. Altitude controller performance is shown in Figure 6.32. Change in the nominal PWM value express the position error in the z-direction. The altitude measure is changing with a large amount due to insufficiency on the number of anchors used. Because of that reason, proportional term of the altitude controller is chosen to be as small as possible. Due to the noisy nature of position measurements in z-direction, derivative term is not used in the altitude controller.

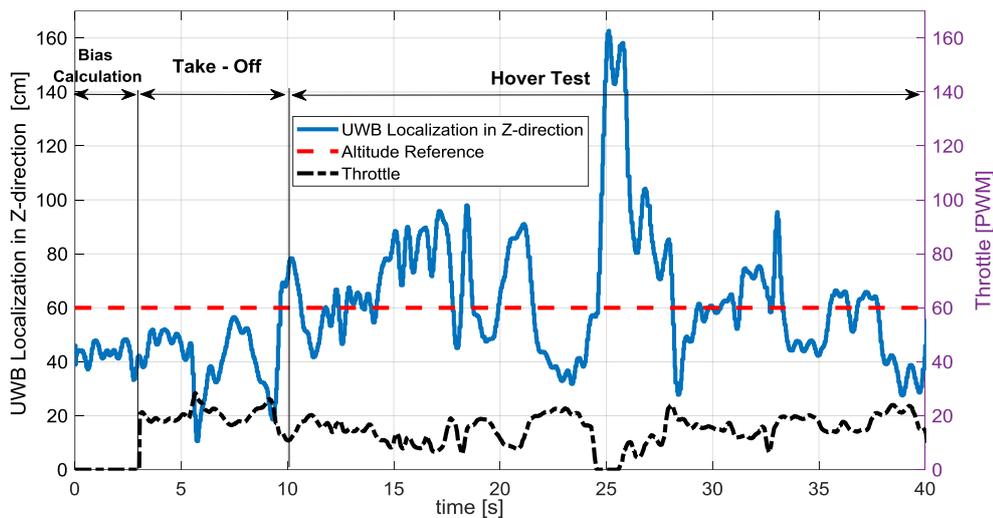


Figure 6.32 Altitude Controller in Hover Test

As shown in Figure 6.33 and Figure 6.34 that, the main problem with position control is the take off. Rotors are not generating the same amount of thrust force for the same duty cycle during take-off generation signal. Thrust force offset creates moment on the body. For that reason, there is a natural position overshoot after take-off. However, overshoot in x direction is greater than y direction. The reason is the battery orientation

creates asymmetry. Trajectory of the quadcopter in the x-y plane is shown in Figure 6.35.

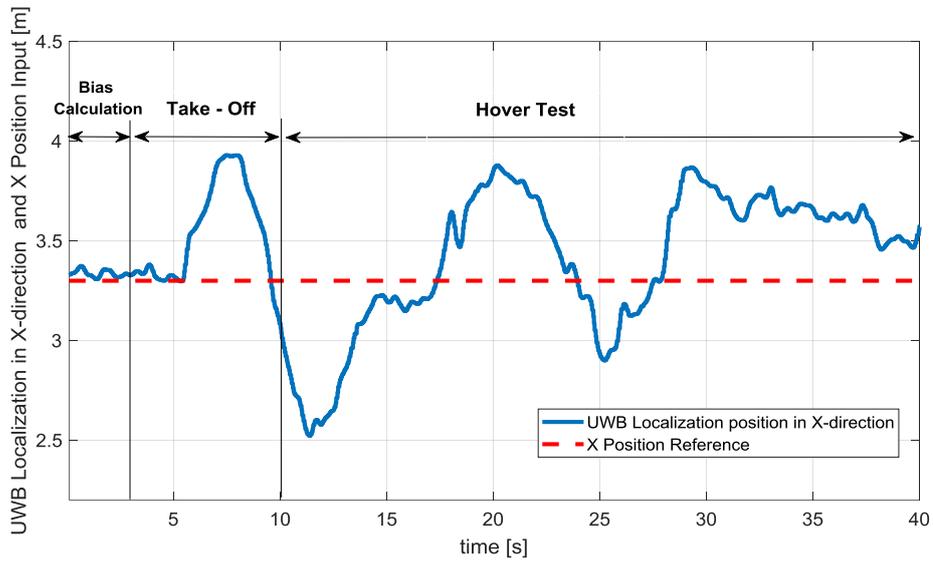


Figure 6.33 X Position Controller in Hover Test

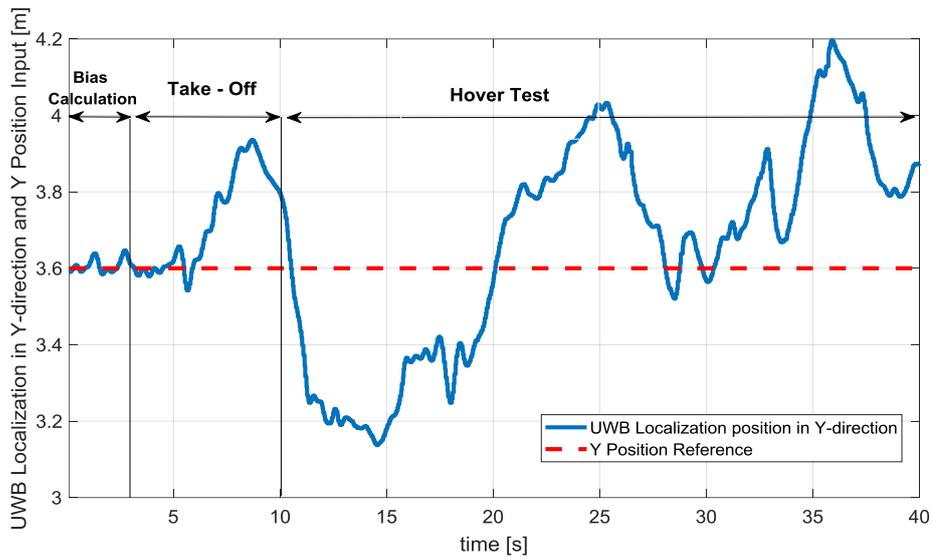


Figure 6.34 Y Position Controller in Hover Test

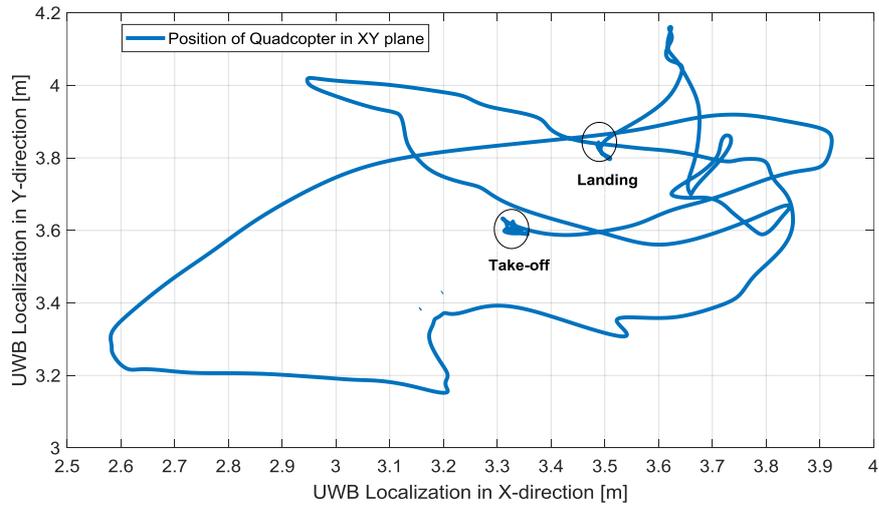


Figure 6.35 Quadcopter Trajectory in XY Plane – Hover Test

Reference intermediate Euler pitch and roll angles are calculated by the error in the x-y position. Intermediate Euler pitch and roll angles are the reference input of the inner loop controller. The angular position reference input and the system response is presented in Figure 6.36 and Figure 6.37. The IMU feedback measures of the Euler roll and pitch angles are in the range of 3 degrees in absolute.

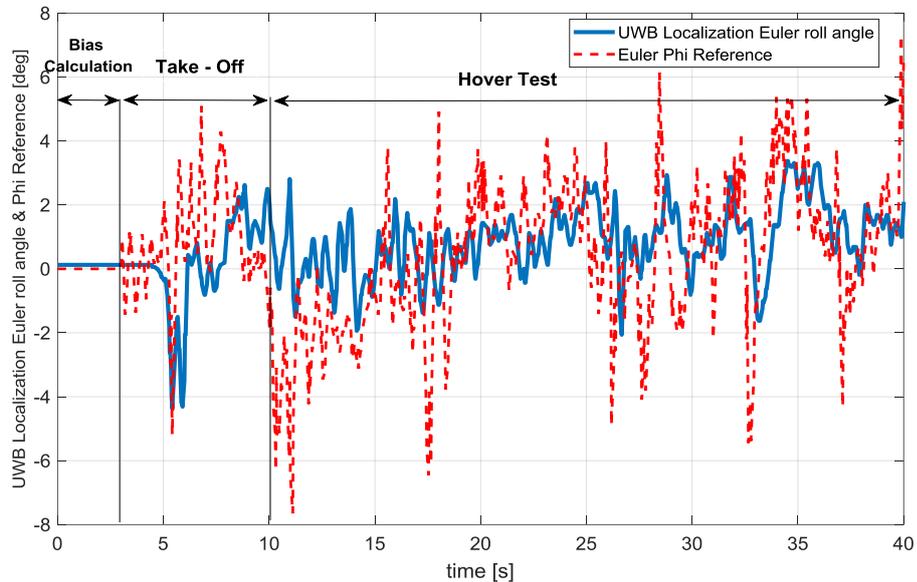


Figure 6.36 Euler ϕ Controller in Hover Test

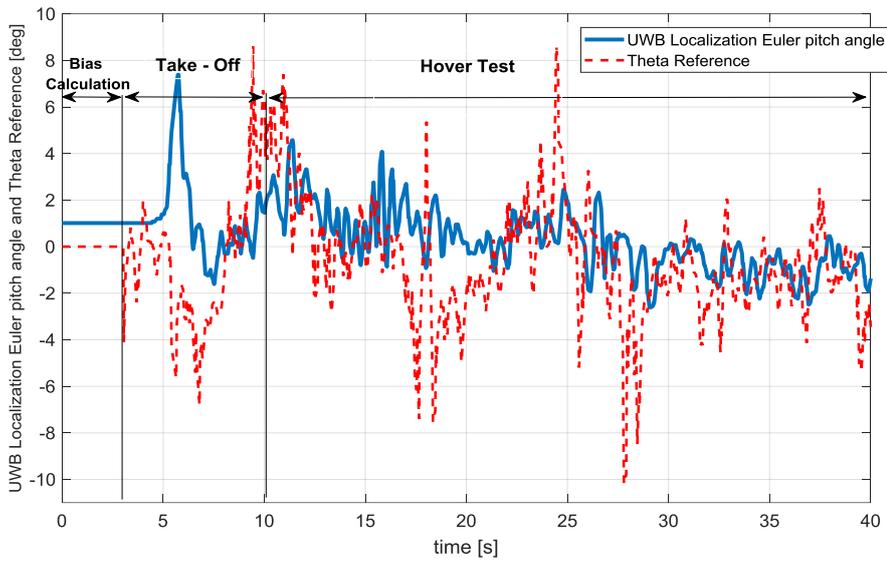


Figure 6.37 Euler θ Controller in Hover Test

Time evolution of Euler yaw angle is shown in Figure 6.38. Initially there exists 5 degrees of Euler yaw angle offset. Then, 10 degrees of difference observed during take-off. In the application area, oscillation in Euler yaw angle is below 3 degrees during hover test.

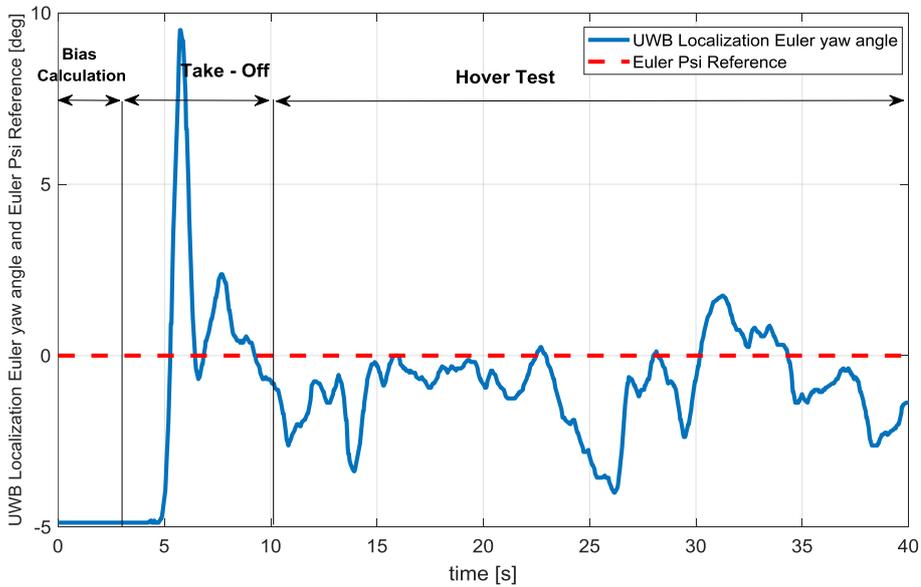


Figure 6.38 Euler ψ Controller in Hover Test

$\dot{\phi}$ and $\dot{\theta}$ angular rates of the quadcopter are controlled in the inner loop. The time evolution of the angular rate loop is shown in Figure 6.39 and Figure 6.40. Quadcopter Euler rates are in the region of 30 deg/s.

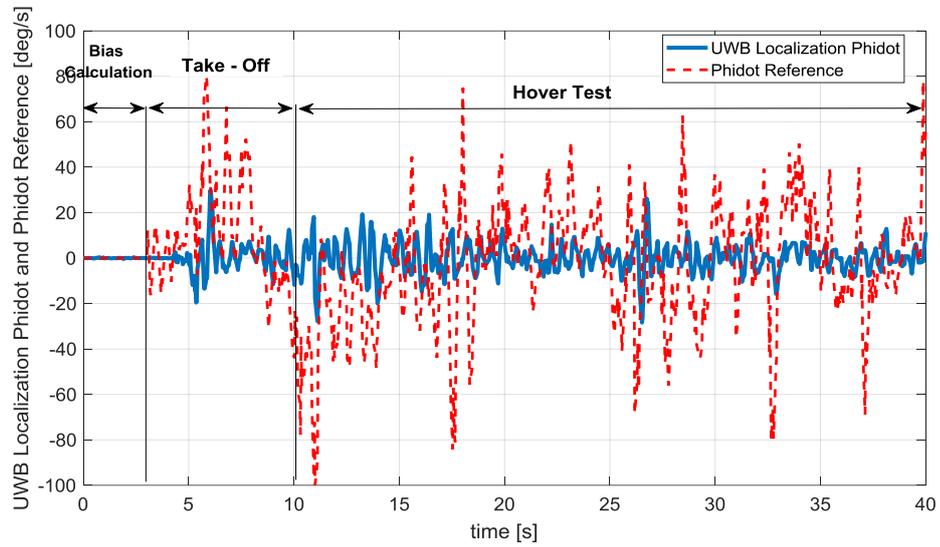


Figure 6.39 $\dot{\phi}$ Controller in Hover Test

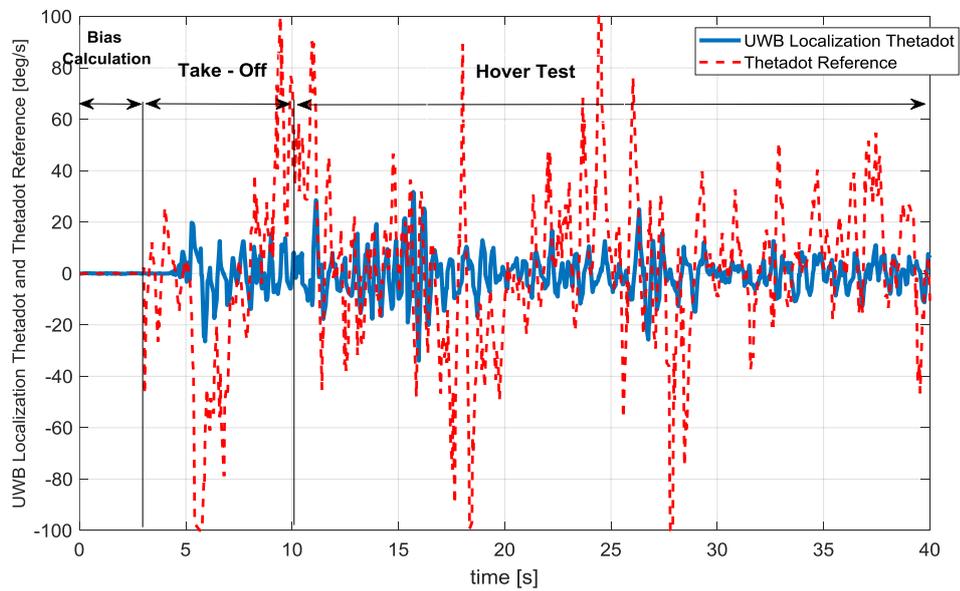


Figure 6.40 $\dot{\theta}$ Controller in Hover Test

6.2.1.2. 3-D Rectangular Path Tracking Test

Rectangular position reference command on x-y plane is generated in order to examine the whole controller channels' performance. Coupled position command is fed to the system. At the end, quadcopter safely landed.

Controller reference inputs are expressed in Table 6.16. The outer loop reference control inputs are fed by considering the change in the nominal position. A second order filter with a 3 Hz of natural frequency is used for reference input change.

Table 6.16 Reference Inputs of the 3-D Scenario Generation

Reference input	Nominal Position	Input-1	Time-1	Input-2	Time-2
X-direction	3.3 m	-1 m	15.74 s	1.5 m	35 s
Y-direction	3.5 m	-0.85 m	24 s	0.7 m	45 s

X-y position reference input and the system response are depicted in Figure 6.41 and Figure 6.42. System response on y position control has a faster response than the x position control for similar decoupled command inputs. The filtered position of the quadcopter in x-y plane is illustrated in Figure 6.43.

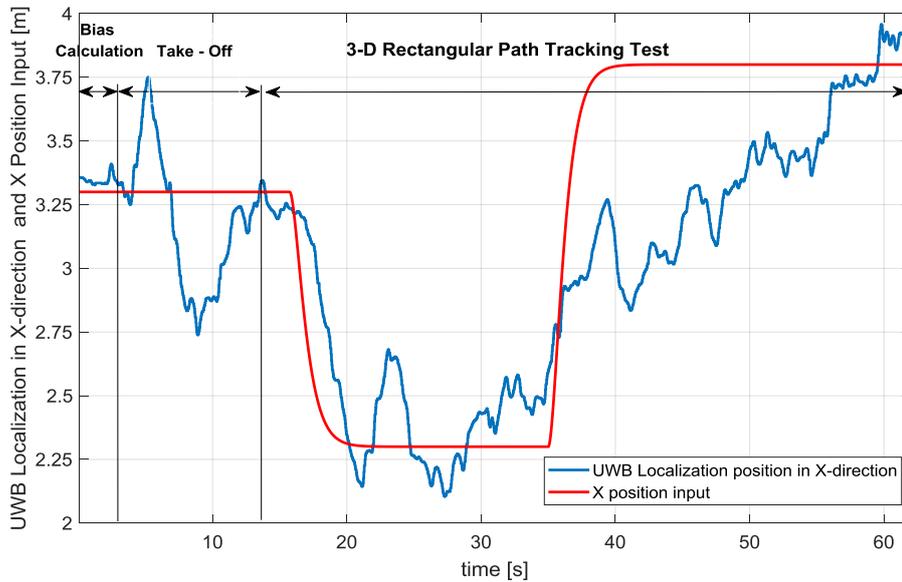


Figure 6.41 X Position Controller Command Tracking Performance

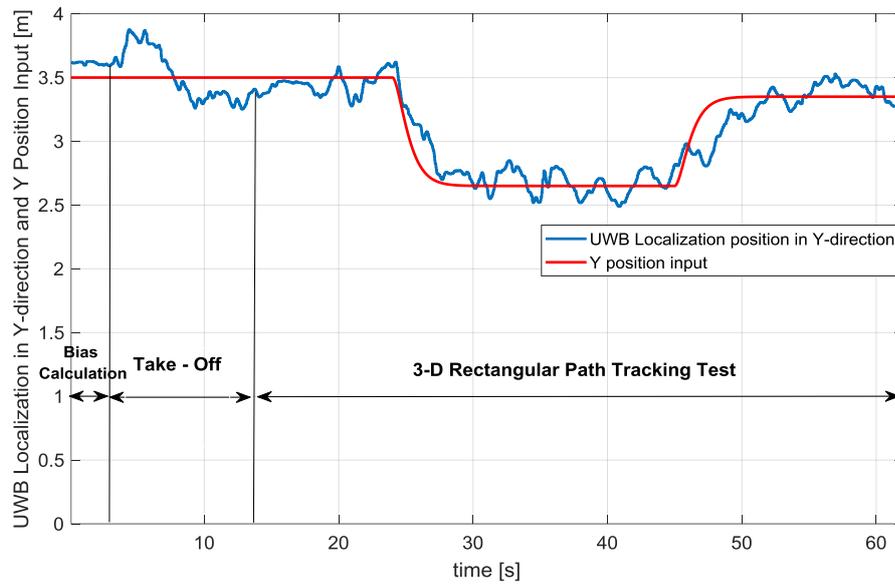


Figure 6.42 Y Position Controller Command Tracking Performance

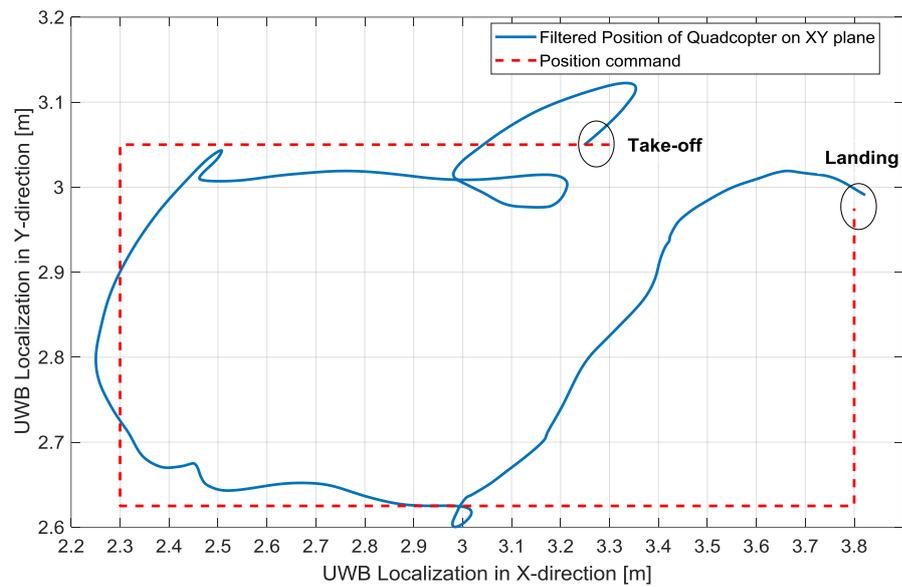


Figure 6.43 Quadcopter Trajectory in XY Plane - Position Test

Euler roll and pitch angle controller performance are expressed in Figure 6.44 and Figure 6.45. Oscillations on Euler pitch channel has greater amplitude than the Euler

roll channel. Inertial effects on Y plane is more dominant than X plane. However, angular position oscillations stay in the linear region.

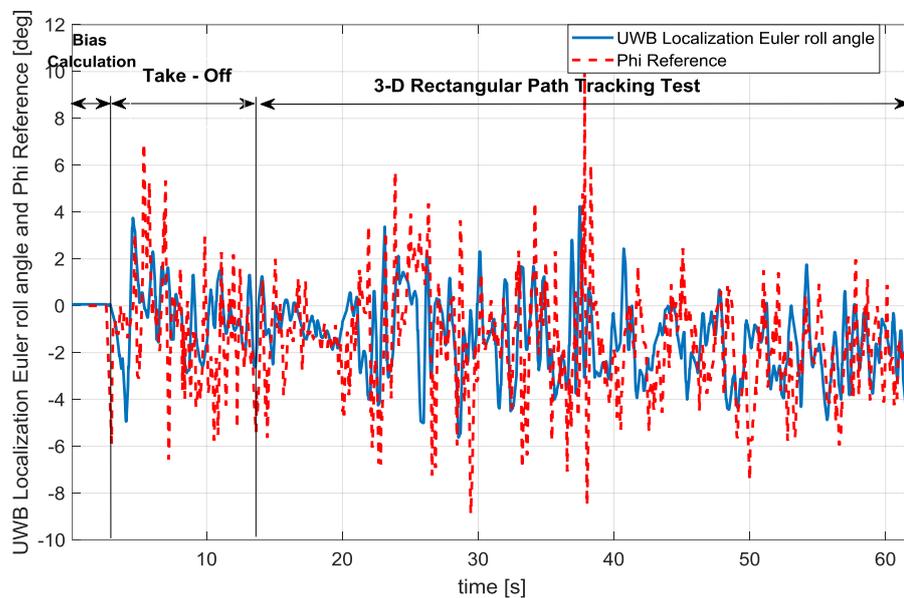


Figure 6.44 ϕ Controller in Position Test

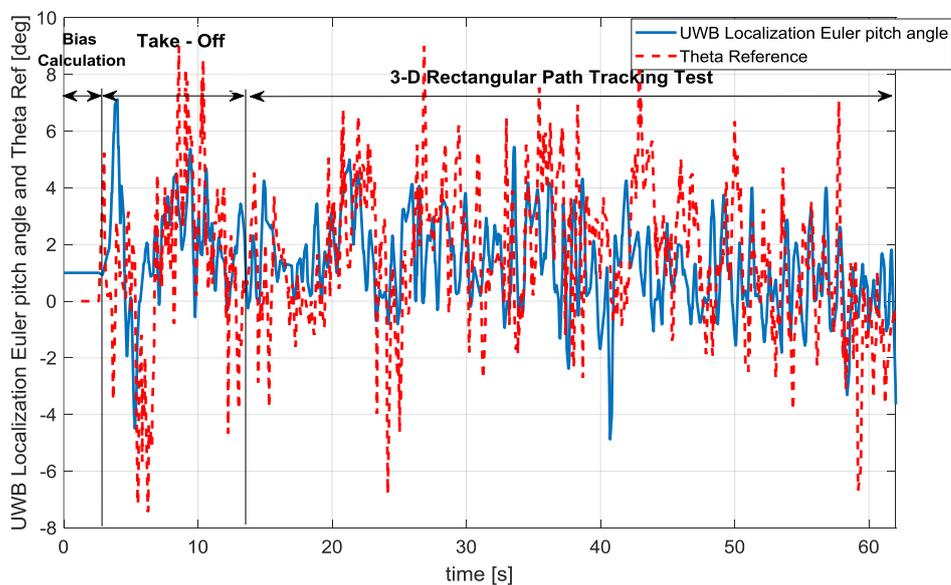


Figure 6.45 θ Controller in Position Test

Time evolution of Euler yaw controller is expressed in Figure 6.46. It is required that the Euler yaw angle must be held in the minimum level during the flight in order to observe the whole system performance clearly. Coupled position reference input and the yaw angle performance is in the tolerable level during the test. Angular error observed on the yaw channel is around 4 degrees. However, take off disturbance peaks the yaw channel error value but the controller compensates the error accumulation immediately.

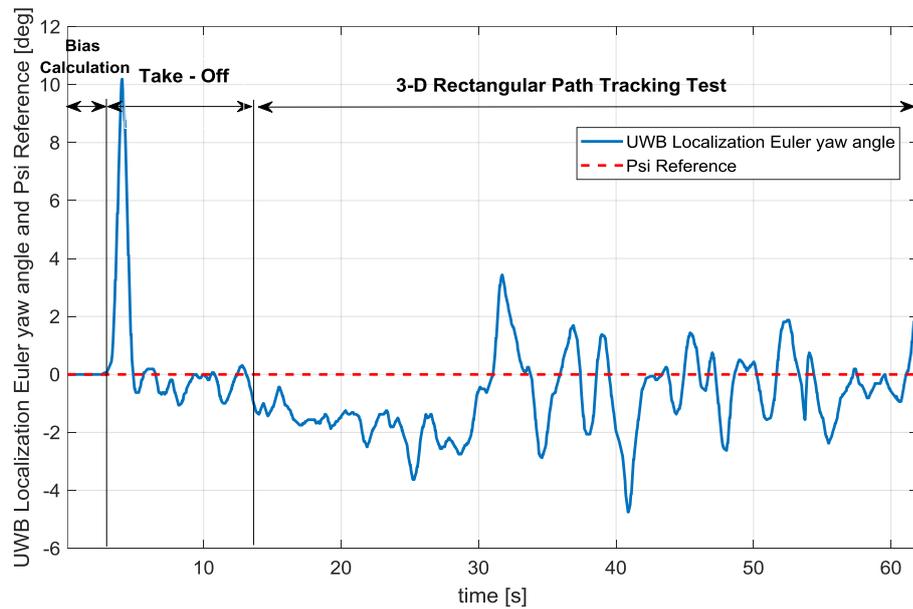


Figure 6.46 ψ Controller in Position Test

Euler roll and pitch rate loop command input and the sensor feedback values are plotted in Figure 6.47 and Figure 6.48. Euler rates of the quadcopter is oscillating below 40 deg/s during the flight.

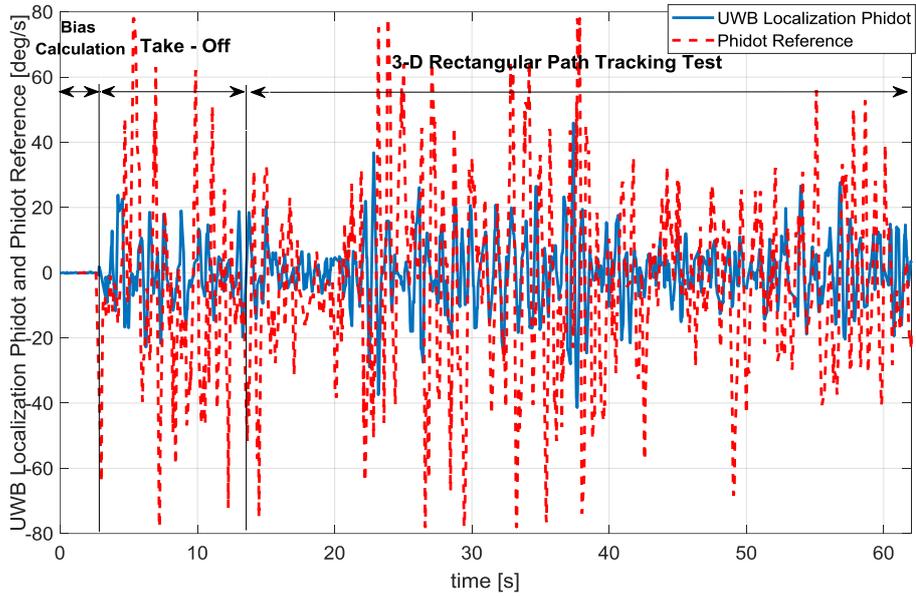


Figure 6.47 $\dot{\phi}$ Controller in Position Test

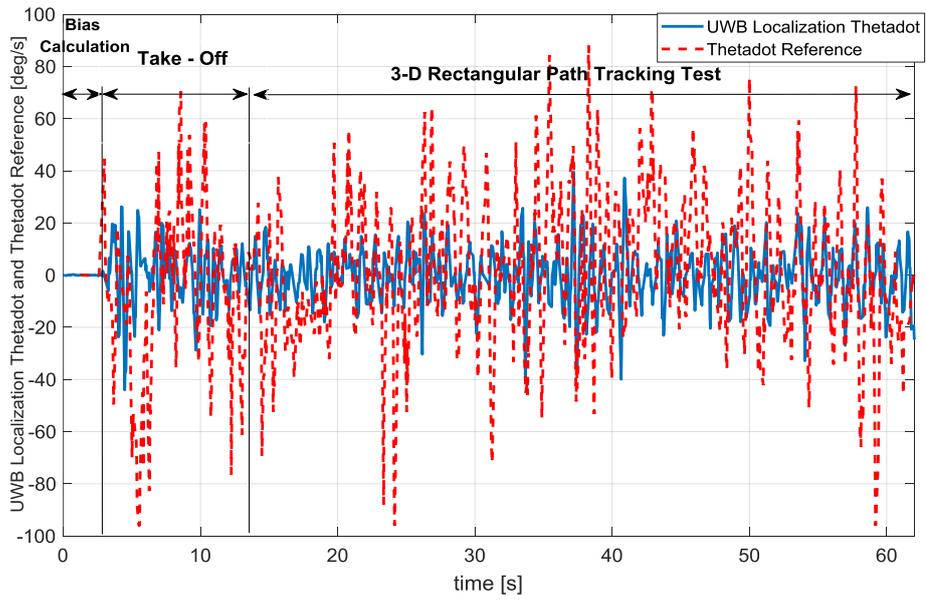


Figure 6.48 $\dot{\theta}$ Controller in Position Test

CHAPTER 7

DISCUSSION AND CONCLUSION

7.1. General Discussion

In this part, simulation and experimental based result are investigated. In the experimental based results, there are two scenarios performed which are the hover and 3-D rectangular path tracking tests. It is observed in the test results that the altitude feedback information of the indoor UWB Localization unit is quite noisy. The altitude measure is changing with a large amount due to insufficiency on the number of anchors used. The noisy measures on the altitude feedback information can be eliminated. Increasing the order of the FIR filter in the built-in settings of the Indoor UWB Localization system, noisy trend on the position feedback becomes a smooth and considerable trend. However, delay is generated by increasing the order of the FIR filter. Processing frequency is also naturally lowered for the same issue under heavy calculations done by Tag on the feedback information.

The reference intermediate Euler roll and pitch angles have peaks in the time evolution. It is suspected that, the sampling frequency of Tag and the Anchors are changing. The main reason may be the decrease in the battery voltage. Angular position control is faster than the linear position control. Angular position controller waits for the position update and reacts aggressively. However, the system response is satisfactory according to the rise time of the rotors to compensate for the peaks observed in the reference input. This approves the system to behave as it is required.

The orientation and the position of each hardware creates asymmetry on the quadcopter body. It is observed that, the inertial effects in Y plane is more dominant than the X plane. For that reason, system response in y-position control has a faster response than the x-position control for similar decoupled command inputs.

In the simulation based experiments, there are 5 Scenarios performed for the analysis of the precise engagement. In Scenario-1, cascaded PID controller is analyzed. Tracking performance of the cascaded PID controller is satisfactory. However reaching the engagement states is difficult with that controller. Engagement states are tried to be achieved by tuning the reference inputs of each controller channel for correct timing, since the cascaded PID controller is the decentralized approach for the quadcopter system.

In Scenario-2 to 5, the optimized trajectory reference input is fed to the LQR controller. This is the centralized approach. Controller tracking performance is satisfactory. Centralized approach is much better than the decentralized approach for reaching the engagement states of the quadcopter system. Different release angle, mission parameter is set to Scenario-3. As the release angle increase in Scenario-3 for the same relative distance of the target in Scenario-2, there is a small amount of increase in the energy consumption.

In Scenario-4 and Scenario-5, trajectory is generated for the non-linear regions while the object is thrown for a long distance. Then, the quadcopter system is commanded back to its initial state. Error accumulation is observed in the landing position of the quadcopter system. The reason behind is that the non-linear effects are dominant for the system.

Energy consumption increases by enabling the sensor subsystem. Another energy dependent observation is that, as the required displacement on the thrown object increase, kinetic energy requirement directly increases by the demand on the linear velocity of the system. Trajectories of the thrown object by different configuration and scenarios are compared. Required lateral displacement of the thrown object is achieved quite precisely. However, precise target engagement directly depends on the controller performance. Reaching the release point with the engagement states is quite challenging for that type of an underactuated air vehicles. The difference between

ideal and achieved release point reflects directly on the hit location. That causes the miss distance.

7.2. Conclusion

A quadcopter system with a 2-DOF robotic system is investigated. The mathematical model of the system is derived. Kinematics of the quadcopter system is found. DH are is formed into a table for the details of the serial manipulator. Equation of motion is formulated for the quadcopter system by considering the system's kinematics. Lagrange-d'Alambert approach is chosen for that purpose. A cascaded PID controller structure is designed. Equation of motion is linearized according to the system dynamics. Three dimensional positions - Euler yaw angle of the quadcopter body, link angles of the robotic arm are controlled independently. Since the system is underactuated, Euler roll and pitch angles are implemented as the intermediate control commands for the body attitude. Y-X position control is coupled with those of intermediate Euler angles respectively. Gains of the cascaded PID controller are found. Then, the gain set is optimized with the existence of the motor and sensor subsystems in the highly non-linear simulation model. Non-linear multi-objective optimization problem is improved by using MATLAB Optimization Toolbox. The global performance criteria in stability which is ITAE, is assigned as the cost function of the optimization problem. Nonlinear least squares solver of MATLAB is assigned for gain iteration purposes.

Then, cascaded PID controller structure is modified for hardware implementation. System specific performance improvements are performed. Such as, the battery recovery open-loop transfer function and the motor command signals are filtered for system level flight robustness.

Throwing an object of a specific location is studied. Precise target - object engagement problem is analyzed in the reversed order. Target location is assumed to be fixed in

the space. It is considered that, the target may be hiding. Because of that reason, the mission parameters are set for the engagement.

Velocity and position trajectories of the object are found from the mission parameters, then the forward kinematics algorithm is developed. Engagement state is found for the generalized positions and velocities of the quadcopter system.

Trajectory of the quadcopter system is optimized for the minimum control effort. Path to be traced might have some harsh conditions to be paid an attention. Path conditions are defined. Trajectory planning is developed from initial system state to engagement state while considering the environmental dependency. Flight duration is considered as a boundary condition. Path generation is generated by Constrained Nonlinear Programming Solver for Multivariable Function of MATLAB. Equation of motion of the system is simplified for planar assumption and linearized in order to converge a local solution fast. Trapezoid integrator of MATLAB is set for solving the high order derivative terms. Trajectory is divided into segments by using MATLAB Optimization Toolbox. The number of segment that the trajectory is divided into is set for specified grids for pre-specified flight duration. Quadratic Spline Interpolation of MATLAB is used for finding the off grid points of the trajectory. Optimization is iterated for each segment of the trajectory. Then, infinite horizon LQR controller is developed for tracing the optimized trajectory. States of the controller is defined. X position control is bypassed.

The gain set of the LQR controller is pre-specified in the planar LTI simulation model by using the weighting factors of the states. Then, gains are again iterated by using MATLAB Optimization Toolbox on the 6-DOF executable simulation model.

An appropriate quadcopter system parts are assembled. Prepared experimental setup is arranged for the hardware implementation. Robotic manipulator is inserted into the system. Inertia of the quadcopter is identified. Controller designed for the mathematical model of the system is embedded to the controller card as the Flight Controller Algorithm. Calibration steps for initial hardware conditions such as the

orientation and position of the system are fed into the controller architecture. Calibration constants are diminished for a robust flight. Data processing frequency of the sub-elements of the system's hardware are tried for their limits by making modifications on the communication protocols and the solver types. Then, experiments are executed.

Simulation based experiments are executed under two topics, the first one is to examine the cascaded PID controller. The second one is the trajectory planning and tracking by LQR controller. Scenarios are analyzed for cascaded PID controller which is tested on the non-linear simulation model.

In the simulation output considered for trajectory planning and tracking by LQR controller is examined under two topics. First topic is generated for different release angle trajectory generation and tracking. Trajectory of the ball is shaped for the possible existence of an object. In the second topic, controller performance limits are tested. Energy consumption of each simulation based experiment is analyzed.

7.3. Future Work

Hardware set-up may be revisited. So, the version of the Pozxy Indoor Localization system might be updated in that sense. Raspberry Pi 3 Linux Based controller card is limited to embedding the different varieties of MATLAB features. Controller card might also be revisited for that matter.

The quadcopter body is symmetric. However, the quadcopter system has a greater moment of inertia on the pitch plane compared with the roll plane due to the inertial contribution of the serial manipulator. So that, the system has a slower response in the pitch plane, in the contrary roll plane performance is remarkably faster for position command tracking. In order to compensate that much of a performance difference in different control axes. The design of the robotic arm may be revisited as one of the revolute joint actuated in the yaw plane.

Aerodynamic effects on the ball may be included and revisited for the engagement states. Trajectory generation and tracking by the LQR controller of precise engagement may be validated for the real physical system.

REFERENCES

- [1] H. Huang, G.M Hoffmann, S.L. Waslander, “Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering”, 2009 IEEE International Conference on Robotics and Automation, Kobe International Conference Center
- [2] E. Kuantama, R.Tarca, “Quadcopter Attitude and Thrust Simulation Based on Simulink Platform” *Robotica & Management* . Dec2015, Vol. 20 Issue 2, p40-44. 5p.
- [3] A.L. Salih, H.A.F. Mohamed, K.S. Gaeid, “Flight PID controller design for a UAV quadrotor” *Scientific Research and Essays* Vol. 5(23), pp. 3660-3667
- [4] K. Sreenath, V. Kumar, “Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots” *Robotics: Science and Systems IX*, 2013.
- [5] F.A. Goodarzi, D. Lee, T. Lee, “Geometric Control of a Quadrotor UAV Transporting a Payload Connected via Flexible Cable” *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486-1498, 2015
- [6] T. W. Danko, K. P. Chaney, and P. Y. Oh, “A parallel manipulator for mobile manipulating UAVs,” 2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), 2015.
- [7] A. Khalifa, M.Fanni, A.Ramadan,A.Abo-Ismael, “New Quadrotor Manipulation System: Inverse Kinematics, Identification And Ric-Based Control” *International Journal of Recent advances in Mechanical Engineering (IJMECH)* Vol.4, No.3,2015
- [8] J. U. Álvarez-Muñoz, Nicolas Marchand, Fermi Guerrero-Castellanos, Sylvain Durand, A. E. Lopez-Luna. “Improving control of quadrotors carrying a manipulator arm”. XVI Congreso Latinoamericano de Control Automático. 6 p, (CLCA), 2014

- [9] C. Korplea, M. Orsag, M. Pekala, P. Oh, “Dynamic Stability of a Mobile Manipulating Unmanned Aerial Vehicle” 2013 IEEE International Conference on Robotics and Automation (ICRA), 2013
- [10] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, “Control of a multirotor outdoor aerial manipulator,” 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014.
- [11] S. Suzuki, H. Inooka, “Golf-Swing Robot Emulating a Human Motion”, IEEE International Workshop on Robot and Human Communication, 1997.
- [12] S. Bellens, J.D. Schutter, H. Bruyninckx, “A Hybrid Pose / Wrench Control Framework for Quadrotor Helicopters” 2012 IEEE International Conference on Robotics and Automation, 2012
- [13] F. Ruggiero, J. Cacace, H. Sadeghian, V. Lippiello, “Impedance Control of VTOL UAVs with a Momentum-based External Generalized Forces Estimator” IEEE International Conference on Robotics & Automation (ICRA), 2014
- [14] T. Tomic, S. Haddadin, “A Unified Framework for External Wrench Estimation, Interaction Control and Collision Reflexes for Flying Robots” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014
- [15] F. Ruggiero, J. Cacace, H. Sadeghian, V. Lippiello, “Passivity-based Control of VTOL UAVs with a Momentum-based Estimator of External Wrench and Unmodeled Dynamics” Robotics and Autonomous Systems, 2015
- [16] S. Kim, S. Choi, and H. J. Kim, “Aerial manipulation using a quadrotor with a two DOF robotic arm,” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.
- [17] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, “Adaptive control for UAVs equipped with a robotic arm,” IFAC Proceedings Volumes, vol. 47, no. 3, pp. 11049–11054, 2014.

- [18] H. Lee, S. Kim, and H. J. Kim, "Control of an aerial manipulator using on-line parameter estimator for an unknown payload," 2015 IEEE International Conference on Automation Science and Engineering (CASE), 2015.
- [19] M. Fumagalli, R. Carloni, "A Modified Impedance Control for Physical Interaction of UAVs" IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013
- [20] G.V. Raffo, M.G. Ortega, F. R. Rubio, "Path Tracking of a UAV via an Underactuated H_∞ Control Strategy" European Journal of Control 2:194–213, EUCA, 2011
- [21] H. Frank, T. Frank, A. Mittnacht, "A Bioinspired 2-DOF Throwing Robot" IEEE Africon - The Falls Resort and Conference Centre, 2011
- [22] Wataru Mori, Jun Ueda, and Tsukasa Ogasawara, "1-DOF Dynamic Pitching Robot that Independently Controls Velocity, Angular Velocity, and Direction of a Ball: Contact Models and Motion Planning" IEEE International Conference on Robotics and Automation, Kobe International Conference Center, 2009
- [23] R. Ritz, M.W. Müller, M. Hehn, R. D'Andrea, "Cooperative Quadcopter Ball Throwing and Catching" IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012
- [24] Z. Zhang, P. Xie, O. Ma, "Bio-inspired Trajectory Generation for UAV Perching Movement Based on Tau Theory", International Journal of Advanced Robotic Systems 11(9):1 ,2014
- [25] S. Spedicato, and G. Notarstefano , "Minimum-time trajectory generation for quadrotors in constrained environments", IEEE Transactions on Control Systems Technology ,Volume: 26 , Issue: 4 , 2018
- [26] P. Foehn, D. Falanga, N. Kuppaswamy, R. Tedrake, D. Scaramuzza, "Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload", Robotics: Science and Systems (RSS), 2017

- [27] B. Zhou, F. Gao, L. Wang, C. Liu and S. Shen, “Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight”, IEEE ROBOTICS AND AUTOMATION, 2019
- [28] A. Faust, I. Palunko, P. Cru, R. Fierro, L. Tapia, “Learning Swing-free Trajectories for UAV s with a Suspended Load”, IEEE International Conference on Robotics and Automation (ICRA), 2013
- [29] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, E. Galceran, “Continuous-Time Trajectory Optimization for Online UAV Replanning”, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016
- [30] J. Villagómez, M. Vargas, M. Ortega, and F. Rubio, “Modeling and Control of the tPVTOL**This work was partially supported by the spanish Ministry of Education (MECD) under national research projects DPI2012 — 37580 — C02 — 02 and DPI2013 — 44135 — R.,” IFAC-PapersOnLine, vol. 48, no. 9, pp. 150–155, 2015.
- [31] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismael, “Modeling and control of a new quadrotor manipulation system,” 2012 First International Conference on Innovative Engineering Systems, 2012.
- [32] Yi. Gai, Y. Kobayashi, Y. Hoshino, T. Emaru, “Motion Control of a Ball Throwing Robot with a Flexible Robotic Arm”, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:7, No:7, 2013
- [33] C. H. Kim, S. Sugano, “Executing optimized throwing motion on robot arm with free joint”, ADVANCED ROBOTICS, VOL. 30, NO. 24, 1571–1578, , 2016
- [34] F. Lombai, G. Szederkenyi, “Throwing motion generation using nonlinear optimization on a 6-degree-of-freedom robot manipulator”, IEEE International Conference on Mechatronics, 2019

- [35] Y. A. Sambo, P. V. Klaine, J. P. B. Nadas, M. A. Imran, “Energy Minimization UAV Trajectory Design for Delay-Tolerant Emergency Communication”, IEEE International Conference on Communications Workshops (ICC Workshops), 2019
- [36] F. Morbidi , R. Cano, D. Lara, “Minimum-Energy Path Generation for a Quadrotor UAV”, IEEE International Conference on Robotics and Automation (ICRA), 2016
- [37] M. K. Ozgoren, "Lecture Notes for ME 502 (Advanced Dynamics)", Department of Mechanical Engineering, METU, 2016.
- [38] M. K. Ozgoren, "Lecture Notes for ME 522 (Principles of Robotics)", Department of Mechanical Engineering, METU, 2017.
- [39] M. Yıldız, “Effects of Active Landing Gear On The Attitude Dynamics Of A Quadrotor (thesis),” 2015.
- [40] A. T. Basaranoglu, “Design Of Control Systems For An Aerial Manipulator (thesis)”, 2019
- [41] N. Bulut, “Modeling, Simulation, And Control Of A Quadrotor Having A 2-Dof Robotic Arm (thesis)”, 2019
- [42] Matija Krznar, Denis Kotarski, Petar Piljek and Danijel Pavkovic, “On-line Inertia Measurement of Unmanned Aerial Vehicles Using On-Board Sensors and Bifilar Pendulum”, Interdisciplinary Description of Complex Systems 16(1), 149-161, 2018
- [43] R. M. Murray, “Control and Dynamical Systems Lecture Notes– LQR Control”, California Institute Of Technology, 2006
- [44] E. C. Suicmez, “Trajectory Tracking Of A Quadrotor Unmanned Aerial Vehicle (Uav) Via Attitude And Position Control (thesis)”, 2014

- [45] R. Mahony, V. Kumar, and P. Corke, “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [46] M. Chipofya, D. J. Lee, and K. T. Chong, “Trajectory Tracking and Stabilization of a Quadrotor Using Model Predictive Control of Laguerre Functions,” *Abstract and Applied Analysis*, vol. 2015, pp. 1–11, 2015.
- [47] D. Kotarski, Z. Benic, and M. Krzmar, “Control Design for Unmanned Aerial Vehicles with Four Rotors,” *Interdisciplinary Description of Complex Systems*, vol. 14, no. 2, pp. 236–245, 2016.
- [48] K. Ogata, *Modern control engineering*. Delhi: Pearson, 2016.
- [49] M. Kelly, “OptimTraj -- Trajectory Optimization Library”, GitHub. Retrieved, 2019.
- [50] J. T. Betts, “Practical Methods for Optimal Control and Estimation Using Nonlinear Programming”, Siam, Philadelphia, PA, 2010.
- [51] M. Kelly, “An Introduction to Trajectory Optimization: How to do your own Direct Collocation”, SIAM, 2017
- [52] M. Kelly, “Transcription Methods for Trajectory Optimization”, Tutorial, Cornell University, 2015
- [53] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Berlin: Springer, 2008.
- [54] M. W. Spong, *Robot Modeling and Control*. John Wiley & Sons, 2012.
- [55] T. Luukkonen, “Modelling and control of quadcopter,” Independent research project in applied mathematics. Aalto University, 2011.

- [56] M. F. Khan, R. ul Islam, and J. Iqbal, "Control Strategies for Robotic Manipulators," IEEE International Conference of Robotics and Artificial Intelligence, 2012.
- [57] B. Randal, "Quadrotor Dynamics and Control Rev 0.1". All Faculty Publications. 1325., 2008.
- [58] F. L. Lewis, C. T. Abdallah, D. M. Dawson, and F. L. Lewis, Robot manipulator control: theory and practice. Milton Keynes UK: Lightning Source UK Ltd., 2018.
- [59]<https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html>
- [60] <https://www.pozyx.io/shop>
- [61] S.N. Ghazbi, Y. Aghli, M. Alimohammadi, A. A. Akbari, "Quadrotors Unmanned Aerial Vehicles: A Review", International Journal On Smart Sensing And Intelligent Systems Vol. 9, No. 1, 2016
- [62] F. Ruggiero, V. Lippiello, A. Ollero, "Aerial Manipulation: A Literature Review", IEEE Robotics and Automation, 2018
- [63] InvenSense Inc, "MPU-6000 and MPU-6050 Product Specification Revision 3.4", 2013
- [64] A. El-Osery, K. Wedeward, "EE570: Location and Navigation – Aided INS Lecture Notes", 2016
- [65] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodriguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Impedance Control of an aerial-manipulator: Preliminary results," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3848–3853, 2016.
- [66] Seborg, Edgar, Mellichamp and Doyle, "Process Dynamics and Control", Wiley, 4th Edition

[67] D. Graham and R.C. Lathrop, "The Synthesis of Optimum Response: Criteria and Standard Forms, Part 2", Transactions of the AIEE 72, Nov. 1953, pp. 273-288

APPENDICES

Appendix A: 3-D Scenario Tracking with the Cascaded PID Controller Configuration

In this part, closed loop angular and linear position control performance of the quadcopter system with the ball on the simulation environment is observed.

Scenario position reference inputs are expressed in Table A.1. Final time of the simulation is set to 10 seconds.

Table A.1 Details of the 3-D Scenario Tracking

States	t_{start} [s]	t_{stop} [s]	r_{input}
x	2	7	3.5 m
y	2	7	0.5 m
z	0	1.5	1
ψ	1	3	-5°
θ_1	7	9	25°
θ_2	7	9	15°

Reference input and the system response is shown in figures.

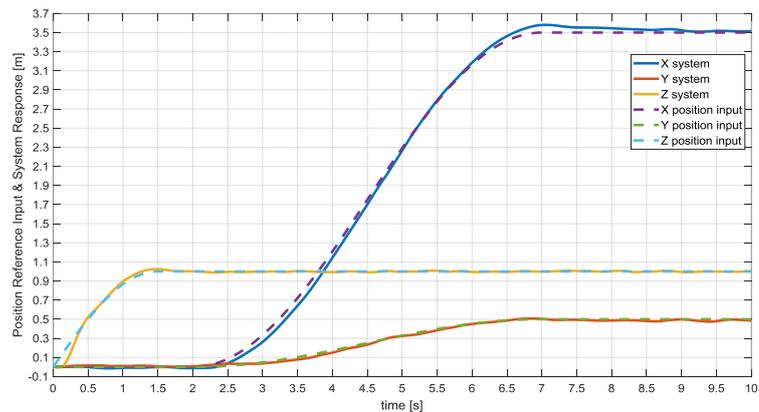


Figure A.0.1 Linear Position Reference Input and System Response

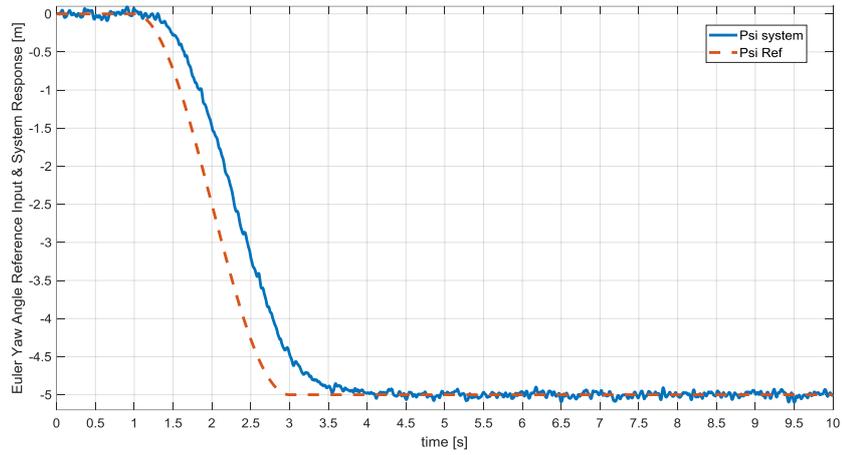


Figure A.0.2 Euler Yaw Angle Reference Input and System Response

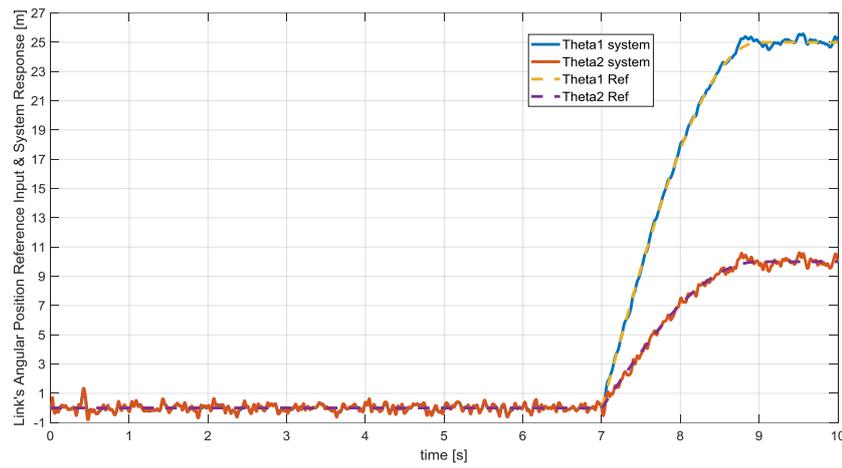


Figure A.0.3 Link's Angular Position Reference Input and System Response

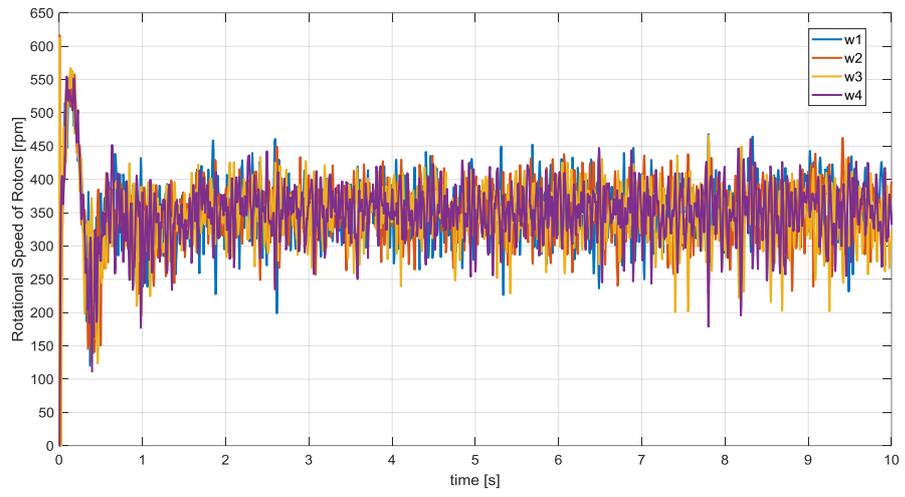


Figure A.0.4 Rotational Speed of Rotors during the Flight Simulation

Appendix B: Experiment of the Quadcopter System

In this part, robotic arm of the quadcopter system is mounted to the body. The quadcopter system is firstly took-off and oriented to the target location. The system is navigated to the target location. Then, the shoot signal is generated after the appropriate conditions are satisfied. Logic of the shooting conditions are expressed in Table 6.13. Generated PWM/Duty signal on Raspberry Pi 3 controller is sent to the RC servo of the robotic arm. Scenario sequence is presented in Figure B.1.

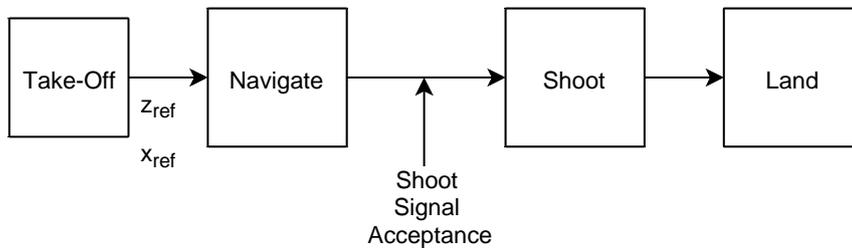


Figure B.1 Scenario Schematics of the Quadcopter System

Details of the scenario is available in Table B.1.

Table B.1 Scenario Sequence of the Quadcopter System

Reference input	Nominal Position	Input-1	Time-1	Natural Frequency of the input transfer function
X-direction	2.5 m	2 m	21 s	2.5 Hz
Z-direction	0.85 m	0.65 m	21 s	2.5 Hz

Reference position of the quadcopter system is fed by using a second order transfer function. Natural frequency of the transfer function is shown in Table B.1. Position change of the quadcopter system is shown in Figure B.2. Linear velocity of the quadcopter system which is obtained by using a second order derivative filter of the position feedback, which is shown in Figure B.3.

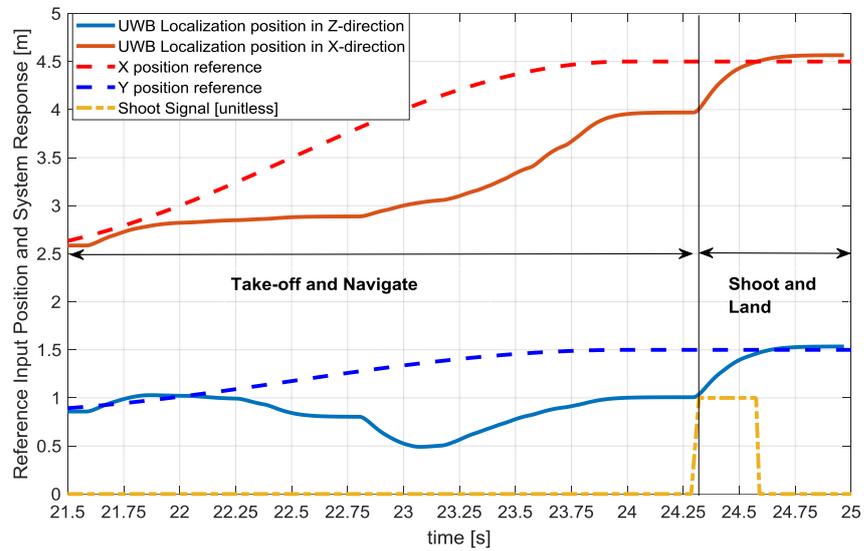


Figure B.2 Position of the Quadcopter

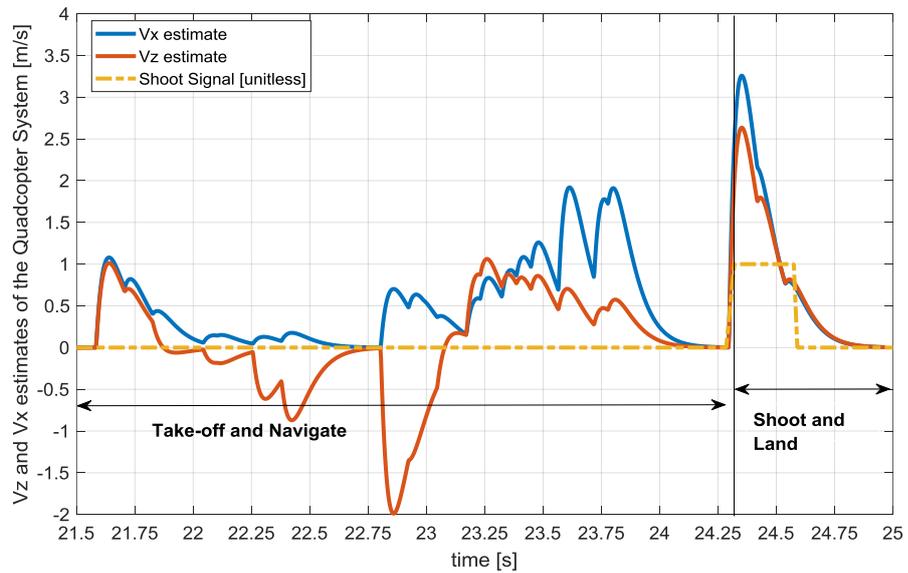


Figure B.3 Calculated Velocity of the Quadcopter

Quadcopter Euler angles are displayed in Figure B.4. During the test, 13 degrees of change is observed in the Euler pitch angle of the quadcopter system after shooting the ball. Then, landing is completed right after the 25th second of the experiment.

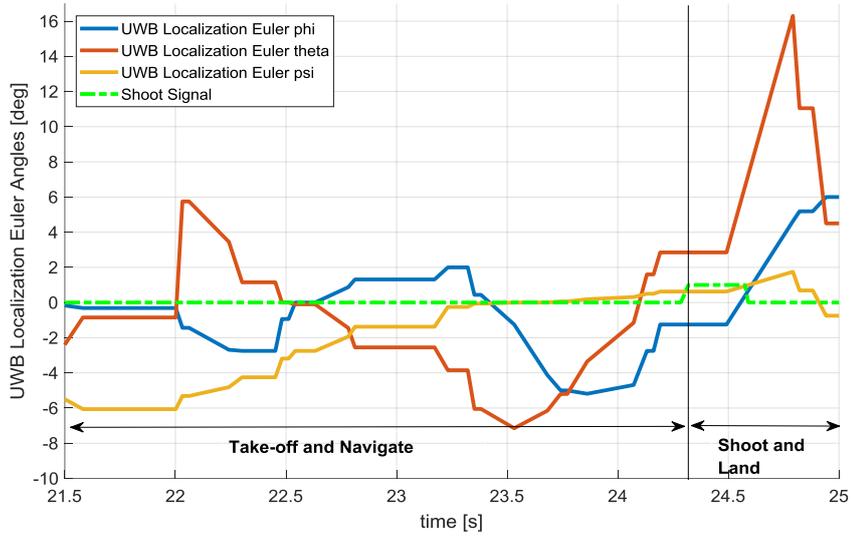


Figure B.4 Quadcopter Euler Angles

Quadcopter Euler rates are expressed in Figure B.5. Disturbance generated by the motion of the robotic arm on pitch plane is also observed on angular rates. Euler pitch rate is prepared for generating the shooting signal. Achieved Euler pitch rate is around 20 deg/s. Euler roll rate observed before shoot is around 30 deg/s. However, disturbance generated by arm's motion brings the pitch rate up to 70 deg/s.

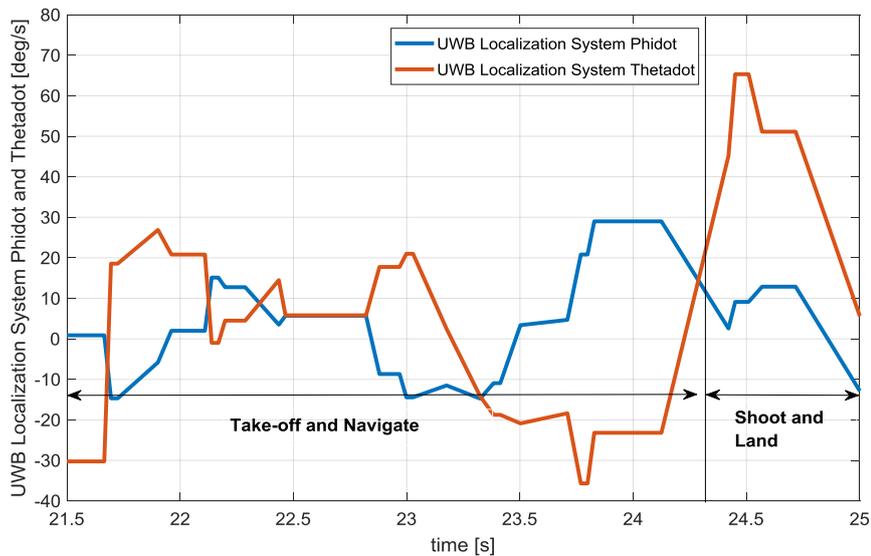


Figure B.5 Euler Rates of the Quadcopter

Appendix C: Implementation of 3-D LQR Controller for the Quadcopter System

In order to examine the quadcopter system in 3-D space, Euler yaw and roll controllers are implemented additionally. The main reason is to suppress the nonlinear effect occurred in the Euler yaw and roll channels of the quadcopter system. Zero reference input is fed to the attitude controller. State space representation of the attitude dynamics as,

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

$$\begin{bmatrix} \ddot{\psi} \\ \ddot{\phi} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \phi \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{q,z}} & 0 \\ 0 & \frac{1}{I_{q,x}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} [\tau_{q3} \ \tau_{q1}]$$

The infinite horizon, LQR is given by,

$$J_{attitude} = \int_0^{\infty} (x^T Q_a x + u^T R_a u) dt$$

Let's define the control input as.

$$u = -R_a^{-1} B^T P x$$

Fix gain set "K" is defined as.

$$K = R_a^{-1} B^T P$$

Algebraic Ricatti equation is solved for P as follows.

$$0 = PA + A^T P - PBR_a^{-1} B^T P + Q_a$$

LQR controller follows the desired trajectory by control law expressed as,

$$u = -Kx$$

“ Q_a ” weights on states and “ R_a ” weights on control input in the cost function of the attitude channel. Each elements of the row matrix represents the diagonal elements of “ Q_a ” and “ R_a ” matrices, which are 4x4 and 2x2 respectively.

$$Q_a = [1000 \quad 2100 \quad 15 \quad 20]$$

$$R_a = [1 \quad 1.5]$$