



Faculty of Engineering and Materials Science  
German University in Cairo

# **Analysis and Implementation of a Model-Predictive-Control (MPC) for QuadCopter Control**

A thesis submitted in partial fulfilment of the requirements for the degree of  
Bachelor of Science (B.Sc.) in Mechatronics Engineering

By  
**Mahmoud Saeed Hussein Sayed**

Supervised by  
**Prof. Elsayed Ibrahim Imam Morgan**  
**M.Sc. Omar Mahmoud Mohamed Shehata**  
**M.Sc. Catherine Malak Elias**

Mechatronics Department  
Engineering and Materials Science Faculty  
German University in Cairo

2018



This is to certify that:

- (i) the thesis comprises only my original work toward the Bachelor of Science (B.Sc.) at the German University in Cairo (GUC),
- (ii) due acknowledgement has been made in the text to all other material used

---

Mahmoud Saeed Hussein Sayed

31 May, 2018

# Acknowledgements

Working on this project was kind of challenge to me, it took a lot of time and work along with working on weekends, trying to catch up with my family and friends, and trying to have some break. Yet, I managed to finally submit my Bachelor Thesis on the due date after a hectic semester full of ups and downs.

Thank God for what you gave and prepared for making me pass this semester after all this hard work, I was, am, and will always be grateful for Allah's mercy.

I would like to thank M.Sc. Omar Mahmoud Mohamed Shehata and M.Sc. Catherine Malak Elias and my professor, Prof. Dr. EL-Sayed Imam Morgan, for his ultimate support and guidance throughout the term, and for this amazing chance he gave me to work with him.

Secondly, my family, my everything, my dad and mom, Mariam, Moataz and Lina may Allah keep you safe for me. I love you and thank you for always being beside me.

Not forgetting to mention, my friends and my lab mates thanks for always being there and never giving up on me. I would like to express my special thanks of gratitude to Tassneem for her great support especially in the hardest times.

Last but not least, my team mates; specially Ahmed Nassim and Abdel Hamed, and lab mates; specially mentioning the mobile robot team and vehicle team , thanks for this wonderful semester, I enjoyed working with you all.

# List of Publications

The following publications have been published while in conduct of this study:

- Mohamed N. Ahmed, Mostafa A. Mostafa, Mahmoud S. Hussein, Ahmed S. Salem, Catherine M. Elias, Omar M. Shehata, and Elsayed I. Morgan. A comparitive study on the control of UAVs for Trajectory tracking by MPC, SMC, Backstepping, and Fuzzy Logic controllers. In *International Conference on Vehicular Electronics and Safety(ICVES), 2018 International Conference*, IEEE, (Under Review)

# Abstract

This thesis investigates the control and dynamics of a quadrotor Unmanned Aerial Vehicle (UAV) using the Model Predictive Control (MPC) approach. The dynamic model is of high fidelity and nonlinear. The control strategy is developed primarily based on MPC to track distinct reference trajectories ranging from simple ones such as circular helix to complex helical trajectories. In this control technique, a linearized model is derived and the receding horizon technique is applied to generate the optimal control sequence. Although MPC is computer expensive, it is rather effective to deal with the distinct types of nonlinearities and constraints such as actuators saturation and model uncertainties. The MPC parameters (control, prediction horizons and weights) are chosen by trial-and-error approach. Several simulation scenarios are performed to validate and evaluate the performance of the proposed control strategy using MATLAB and Simulink environment specifically MATLAB MPC Designer Toolbox which this thesis used to perform all the simulations. Simulation results show that this control approach is exceptionally effective to track a given reference trajectory.

# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>List of Publications</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Contents</b>	<b>vi</b>
<b>List of Abbreviations</b>	<b>viii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Locomotion . . . . .	1
1.1.1 Stationary Robots . . . . .	2
1.1.2 Wheeled Robots . . . . .	4
1.1.3 Legged Robots . . . . .	5
1.1.4 Swimming Robots . . . . .	5
1.1.5 Aerial Robots . . . . .	6
1.1.6 Swarm Robots . . . . .	6
1.1.7 Modular and Soft Robots . . . . .	6
1.2 Motivation . . . . .	6
1.3 Background And History . . . . .	7
1.3.1 Background on UAVs . . . . .	7
1.3.2 A Glimpse of History . . . . .	7
1.4 Types of UAVs . . . . .	9
1.5 Applications of UAVs . . . . .	11
1.6 Challenges That Faces UAVs . . . . .	12
1.7 Problem Statement . . . . .	12
1.8 Thesis Structure . . . . .	12

<b>2</b>	<b>Literature Review</b>	<b>14</b>
2.1	Thesis Proposition . . . . .	24
<b>3</b>	<b>Methodology</b>	<b>25</b>
3.1	System Modelling . . . . .	25
3.1.1	Basic Concepts . . . . .	25
3.1.1.1	Throttle (U1) . . . . .	27
3.1.1.2	Roll Input (U2) . . . . .	28
3.1.1.3	Pitch Input (U3) . . . . .	28
3.1.1.4	Yaw Input (U4) . . . . .	28
3.1.2	Newton-Euler Method . . . . .	29
3.1.3	Kinematics . . . . .	30
3.1.4	Dynamics Model . . . . .	32
3.1.4.1	Rotational Equations of Motion . . . . .	32
3.1.4.2	Translational Equations of Motion . . . . .	35
3.1.5	Aerodynamic Effects . . . . .	36
3.1.5.1	Drag Forces . . . . .	36
3.1.5.2	Drag Moments . . . . .	36
3.1.6	State-Space Model . . . . .	37
3.1.7	Linearized Model . . . . .	39
3.2	Controller Design . . . . .	41
3.2.1	Introduction . . . . .	41
3.2.2	Predictive Model . . . . .	43
3.2.3	Optimizer and Cost Function . . . . .	45
3.2.4	Receding Horizon Control . . . . .	47
<b>4</b>	<b>Results</b>	<b>49</b>
4.1	Open Loop Response without the aerodynamic drag . . . . .	49
4.2	Model Predictive Control for point tracking . . . . .	54
4.2.1	Point Tracking With Unconstrained Control Inputs . . . . .	55
4.2.2	Point Tracking With Constrained Control Inputs . . . . .	58
4.3	Model Predictive Control For Trajectory Tracking . . . . .	61
4.3.1	Model Predictive Control for helix tracking . . . . .	61
4.3.2	Model Predictive Control For Complex Helix Tracking . . . . .	65
4.3.3	Model Predictive Control For Square Trajectory . . . . .	68
<b>5</b>	<b>Conclusion and Future Work</b>	<b>73</b>
5.1	Conclusion . . . . .	73
5.2	Future Work . . . . .	73
<b>References</b>		<b>74</b>

# List of Abbreviations

<b>UAV</b>	Unmanned Aerial Vehicle
<b>MAV</b>	Micro Aerial Vehicle
<b>MPC</b>	Model Predictive Control
<b>LMPC</b>	Linear Model Predictive Control
<b>NMPC</b>	Nonlinear Model Predictive Control
<b>PID</b>	Proportional Integral Derivative
<b>LQR</b>	Linear Quadratic Regulation
<b>FPID</b>	Fuzzy Logic Based Proportional Integral Derivative
<b>MRS</b>	Multi-Robot System
<b>SRS</b>	Single-Robot System
<b>ODE</b>	Open Dynamics Engine
<b>IBC</b>	Intelligent Backstepping Control
<b>DOF</b>	Degree Of Freedom
<b>RHC</b>	Receding Horizon Control
<b>GA</b>	Genetic Algorithm
<b>IMU</b>	Inertial Measurement Unit
<b>RMSE</b>	Root-Mean-Square Error
<b>MAE</b>	Mean Absolute Error

# List of Figures

1.1	Stationary Robots . . . . .	3
1.2	Wheeled Robots . . . . .	4
1.3	Legged Robots . . . . .	5
1.4	Examples of UAVs . . . . .	9
1.5	Classification of Aircrafts Depending on The Flying Principle and Propulsion Mode . . . . .	10
1.6	Classification of UAVs . . . . .	10
1.7	Classification of Aerial Robotics Based on Their Endurance and Maneuverability Properties. . . . .	11
2.1	Cascade PID Control Structure . . . . .	15
2.2	OS4 Test-Bench Block Diagram . . . . .	15
2.3	Control Scheme . . . . .	16
2.4	Control Architecture . . . . .	16
2.5	Simulator Framework . . . . .	17
2.6	Attitude Stabilization Controller Block Diagram Using FPID . . . . .	17
2.7	FPID Controller Block Diagram . . . . .	18
2.8	Structure of The Intelligent Backstepping Controller . . . . .	19
2.9	Structure of The Fuzzy Inference Block . . . . .	19
2.10	The Closed-Loop Simulation Scheme of The NMPC Strategy With Constraint Satisfactions For The Quadcopter . . . . .	21
2.11	The Overall Closed-Loop System . . . . .	22
2.12	Block Diagram of The Controllers Structure . . . . .	23
2.13	Block Diagram of The Proposed MPC . . . . .	23
2.14	Block Diagram of The Proposed Control Structure . . . . .	24
3.1	PLUS Quadcopter . . . . .	26
3.2	X Quadcopter . . . . .	26
3.3	Inertial Frame and Body Frame Quadcopter . . . . .	27
3.4	Quadcopter Movements . . . . .	29
3.5	Quadcopter Reference Frames . . . . .	30

3.6	Model Predictive Control Block Diagram . . . . .	42
4.1	Constructed Quadcopter . . . . .	50
4.2	Open Loop Response for Thrust . . . . .	51
4.3	Open Loop Response for Thrust and Roll . . . . .	52
4.4	Open Loop Response for Thrust and Pitch . . . . .	53
4.5	Open Loop Response for Thrust and Yaw . . . . .	54
4.6	Closed Loop Response for Point Tracking With Unconstrained Control Inputs .	57
4.7	Closed Loop Response for Point Tracking With Constrained Control Inputs .	60
4.8	Quadcopter Tracking a Helix Trajectory . . . . .	61
4.9	Closed Loop Response for Helix Trajectory Tracking . . . . .	64
4.10	Error Between The Desired And Actual Trajectories . . . . .	64
4.11	Quadcopter Tracking a Complex Helix Trajectory . . . . .	65
4.12	Closed Loop Response for Complex Helix Trajectory Tracking . . . . .	68
4.13	Error Between The Desired And Actual Trajectories . . . . .	68
4.14	Quadcopter Tracking a Square Trajectory . . . . .	69
4.15	Closed Loop Response for Square Trajectory Tracking . . . . .	72

# List of Tables

4.1	Parameters Of The Quadcopter . . . . .	49
4.2	RMSE For Helix And Complex Helix . . . . .	72
4.3	MAE For Helix And Complex Helix . . . . .	72

# **Chapter 1**

## **Introduction**

Robots are widely used now a days in many fields and becoming a field of interest for many of the researchers. Starting in the late 1980s, many researchers started investigating and researching in many aspects concerning this field of interest. These investigations and researches always target issues, problem solving and algorithms concerning this robotics field. This field shows so much interests due to its high outcome efficiency which come with so many other advantages from saving time, eliminating human errors as well as higher efficiency.

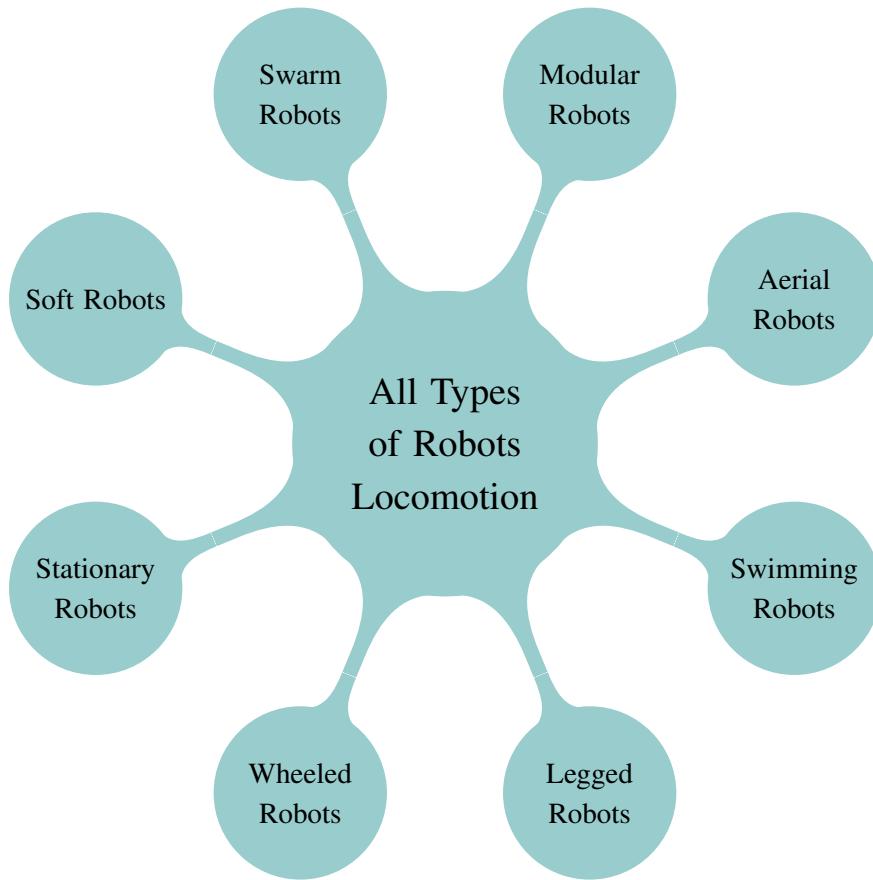
This robotic field is divided into two systems whether it is a Multi-Robot System (MRS) or it is a Single-Robot System (SRS).

### **1.1 Locomotion**

Locomotion is the use of various methods such that the robot move from one point to another. it depends on the physical interaction between the vehicle and its environment and it is concerned with the interaction forces, along with the mechanisms and actuators that generate them. we encounter some problems while dealing with locomotion.those problems are listed in 3 main points:

- Stability
  - Number of contact points
  - Center of gravity
  - Static versus Dynamic stabilization
- Environment
  - Medium

- Structure
- Contact
  - Contact point or area
  - Angle of contact
  - Friction



### 1.1.1 Stationary Robots

Stationary Robots are robots that work while their base doesn't change its position(base fixed in position). Referring to it by stationary doesn't mean that the whole robot isn't actually moving. These kind of robots generally manipulate their environment by controlling the position and orientation of an end-effector. Stationary robot category includes:

- Cartesian/Gantry Robots
- Cylindrical Robots
- Spherical Robots

- SCARA Robots
- Robotic Arms - (Articulated Robots )
- Parallel Robots
- Others



(a) Cartesian Robot



(b) Cylindrical Robot



(c) Spherical Robot



(d) SCARA Robot



(e) Robotic Arm



(f) Parallel Robot

Figure 1.1: Stationary Robots

### 1.1.2 Wheeled Robots

Wheeled robots are robots that can change their position by the means of their wheels which also can change its speed and direction of rotation. Mechanically wise wheeled motion is the easiest and the cheapest to implement and achieve. Moreover control of wheeled motion is easier. The reasons stated above makes wheeled robots one of the most frequently seen robots.

Wheeled robots category includes:

- Single Wheel (Ball) Robots
- Two-Wheeled Robots
- Three Wheeled Robots
- Four Wheeled Robots
- Multi Wheeled Robots
- Tracked Robots
- Others



(a) Single Wheel Robots



(b) Two-Wheeled Robots



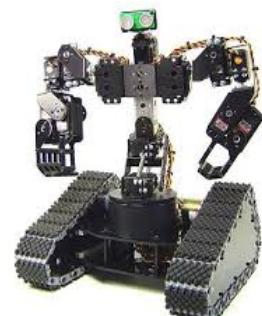
(c) Three Wheeled Robots



(d) Four Wheeled Robots



(e) Multi Wheeled Robots



(f) Tracked Robots

Figure 1.2: Wheeled Robots

### 1.1.3 Legged Robots

Legged robots have so much in common with wheeled robots, however compared to their wheeled counterparts they are more complicated and sophisticated. Legged robots control their locomotion through their legs and they achieve a much high performance than wheeled robots on irregular terrain. Despite the fact that their cost and complexity of production is high for these robots, they have a great advantages on uneven terrain which makes these robots crucial for most applications. Legged robots category includes:

- One Legged Robots
- Two Legged Bipedal Robots (Humanoids)
- Three Legged Tripedal Robots
- Four Legged Quadrupedal Robots
- Six Legged Robots (6 Legged Hexapod)
- Others



(a) One Legged Robots



(b) Two Legged - (Humanoids)



(c) Three Legged Tripedal Robots



(d) Four Legged Quadrupedal Robots



(e) Six Legged Robots - (Hexapod)

Figure 1.3: Legged Robots

### 1.1.4 Swimming Robots

Swimming robots are robots which move underwater. These robots are generally inspired by fish and they use their fin-like actuators to maneuver in water.

### 1.1.5 Aerial Robots

Aerial robots are robots that float and maneuver on air by the mean of their plane-like or bird/insect-like wings, propellers or balloons. Aerial robots category includes airplane robots, bird/insect inspired wing flapping robots, propeller based multicopters and balloon robots. Aerial robots is challenging in terms of control, however it can navigate and reach areas that can't be reached by other robots.

### 1.1.6 Swarm Robots

Swarm robots consists of multiple small robots.the structure of modular robots does not create a single united robot, but operates as their robot modules operate cooperatively. as well as modular robotic systems, elements of swarm robots have much less functionality and herd configurations does not create new robots.

### 1.1.7 Modular and Soft Robots

Modular robots have much in common with swarm robots, they have multiple robots in their configurations. Modules of these systems are more functional compared to a robotic herd. For example a single module of a modular robotic system can have self-mobility and operates alone. Modular robots are very versatile in its configurations. The power of modular robotics comes from its versatility in its configurations. By changing the configurations of the modules we can achieve different system modules of a modular robotic system and different configurations gives distinct abilities. Soft robots are newly introduced to the field of robotics.These robots are generally inspired from biological environments . Most applications are inspired from squids or inchworms both structurally and functionally.

## 1.2 Motivation

In the recent years, a great interest has been directed to robotics, as several Industries (automotive, medical, manufacturing, space, etc.) require robots to replace men in dangerous, boring and repetitive situations.Therefore a wide area of the research is dedicated to aerial platform. At the present time, the robotics research community interest has increased rapidly in the field of Unmanned Aerial Vehicle (UAV). This interest can be verified by the fact that Unmanned Aerial Vehicles have low complexity, high versatility and eliminate need for high skilled pilots. Nowadays, more than 70 countries are investing in Unmanned Aerial Vehicles technology.

## 1.3 Background And History

### 1.3.1 Background on UAVs

The field of aerial robotics encompasses a broad class of flying vehicles that recently often possess the perception capabilities and decisional autonomy to accomplish complex tasks without the need for any direct human interference. Historically and within the aerospace jargon, robotic flying machines are commonly referred to as UAVs, while the entire infrastructures, systems and human-machine interfaces required for autonomous operation are often called unmanned aerial systems (UAS). Aerial robotic technologies are currently on the cutting edge of aerospace and robotic research. Breakthrough contributions take place in various fields such as design, estimation, perception, control, and planning, paving the way for a historical change on how flying systems are operated and what application challenges they fulfill. A UAV is defined as "an aircraft which is designed or modified, not to carry a human pilot and is operated through electronic input initiated by the flight controller or by an onboard autonomous flight management control system that does not require flight controller intervention."

As is generally the case in robotics, aerial robots tend to become more and more complex systems as a result of the effort to achieve advanced decision making and planning capabilities based on its on-board perception of the environment and a set of relatively abstract mission goals.

Aerial robots possess the unique capability to gently fly over terrain that other robots struggle to roll or crawl over. The price to be paid is related with the advanced challenges in terms of system design, propulsion, perception, control, and navigation. Autonomous flight requires handling of all six degrees of freedom and advanced cognition capabilities within challenging environments. In that sense, perception and navigation complexity drastically increase, while payload and available power consumption for processing tends to be limited, especially as scale decreases. The design of aerial robots requires increased attention and thorough selection, or even combination, of one or more existing or new flying concepts, electronic components and algorithms. The design engineer has to assess specific optimization challenges and trade-offs as important desired goals like decreased weight and modularity typically contradict each other.

### 1.3.2 A Glimpse of History

Aerial robotics is a field of active research and promising perspectives, yet it already accumulates more than a century of developments. Figure 1.4 [1] depicts some historical as well as recent examples of UAVs in the military and civilian sector. Starting as conceptual designs in the context of the human efforts to develop flying machines, aerial robots soon proved their

extensive potential and have already created their own legacy. As was also the case for manned aviation, aerial robotic technologies accelerated within the framework of the 20th century world conflicts. Within World War I, HewittSperry developed an automatic plane that acted as a flying torpedo, carrying onboard intelligence to autonomously sustain flight over long periods of time. This page-turning success was achieved through the integration of (Sperrys self-made) gyroscopes which were then mechanically connected to the control surfaces and therefore established the necessary feedback control loop. During World War II, the German armed forces deployed one of the first successful cruise missiles, the V1. Despite the fact that V1 had limited success rate it did incorporate most of the elementary components, estimation algorithms and control loops that can allow autonomous navigation and reference tracking. Military applications kept being, and still are, the main driving force of aerial robotics research and the newest developments in the area change and shape the modern warfare. With the introduction of global positioning systems (GPSs), aerial robots managed to achieve the first completely autonomous surveillance missions. As information and intelligence gathering became one of the most important aspects of the worlds open or silent conflicts, military research around the 1970s led to systems equipped with cameras and other sensory systems, giving birth to the UAV prototype the way we know it today. However, civil applications are currently emerging at a very fast pace and the majority of market predictions converge to the conclusion that this area will take dominant characteristics, and most importantly, will become an equally important if not more innovation drive.

Within this framework, the advancements within the field of microprocessors, sensing, in addition to actuator efficiency and down-scaling significantly accelerated the field of aerial robots and paved the way for the exquisite achievements we look at nowadays. Aerial robot shave superior to a state in which sophisticated sensor modules for onboard state estimation and environmental perception, powerful embedded processors running sophisticated navigation algorithms, potentially several communication interfaces, in addition to high-end-mission oriented payloads that enable the execution of challenging tasks in various environments, can be tightly integrated.



Figure 1.4: Examples of UAVs

By taking a look at the time line of the aerial vehicles, in 1922 ,First Launch of an unmanned aircraft (RAE 1921 Target) from an aircraft carrier (HMS Argus). also 3 September 1924 was the first successful flight by a radio controlled unmanned aircraft without a safety pilot on-board; performed by the British RAE 1921 Target 1921, which flew 39 minutes. on 1933 was the first use of unmanned aircraft as a target drone. 12 June 1944 was the first combat using an unmanned aircraft (German Fi-103 V-I) in the cruise missile role. April 1946 was the first time to use unmanned aircraft for scientific research performed by a converted Northrop P-61 Black Widow for flights into thunderstorms by the U.S. Weather Bureau to collect meteorological data. in 1955 was the first flight of an unmanned aircraft designed for reconnaissance performed by the Northrop Radioplane SD-1 Falconer/Observer, later fielded by the U.S. and British armies. on 20-21 August 1998, was the first trans-Atlantic crossing by an unmanned aircraft, performed by the Insitu Groups Aerosonde Laima between Bell Island, Newfoundland, and Benbecula, Outer Hebrides, Scotland.

## 1.4 Types of UAVs

Through out the years, we have seen the exquisite advancements in the field of aerial robotics. Depending on the flying principle and propulsion mode, one can classify the aerial vehicles in

multiple categories.

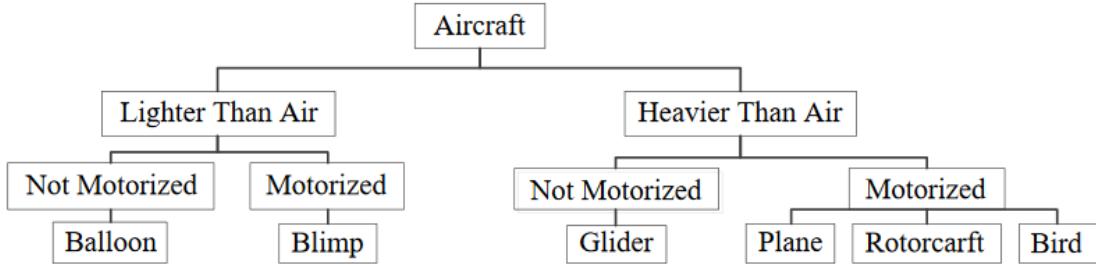


Figure 1.5: Classification of Aircrafts Depending on The Flying Principle and Propulsion Mode

In the motorized class, the bird-like Micro Aerial Vehicle (MAV) can be considered as a good example for fast area navigation. Additionally Vertical Take-Off and Landing (VTOL) and UAV fall under the same category as the MAVs. However, UAVs themselves can be classified.

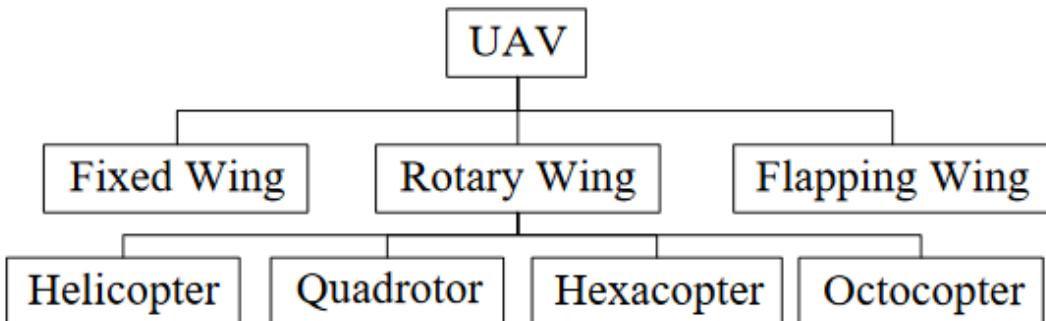


Figure 1.6: Classification of UAVs

Compared to the categorization of manned aviation, aerial robots classification is more complex, as the term currently refers to a very wide variety of systems of different scale, mechanical configuration, and actuation principles. In their vast majority, aerial robots correspond, in one way or another, to miniaturized versions of manned aircraft designs. Relatively classical fixed-wing unmanned aerial systems (FW-UAS) designs and rotary-wing unmanned aerial systems (RW-UAS) such as those shown in Figure 1.6 [2] are common vehicle configurations one may encounter in most applications, including those of surveillance, monitoring, inspection, mapping, or payload transportation. However, even within these relatively traditional concepts, several design aspects differ from those chosen for manned systems. This reflects the fact that for different scales, the variation of the physical properties behavior, along with the search for optimized designs, will naturally lead to modified and novel design considerations.

Apart from lighter-than-air systems (LtA-UAS), Fixed Wing - Unmanned Aerial Systems (FW-UAS) tend to be the most power efficient flying principle, while Rotary Wing - Unmanned Aerial Systems (RW-UAS) are tailored to increased maneuverability as well as the ability of stationary vertical flight (hovering). This general classification is then further complicated with the relatively large class of convertible designs (such as tilt-rotors or cruise-flight-enabled ducted fans). This first attempt for aerial robots classification has then to be further augmented to account for the biologically inspired concepts, and especially the emerging field of flapping-wing UAS (Fl-UAS). Figure 1.7 [1] provides an abstract yet incomplete overview of the vehicle classes one may encounter in most of the application fields. As shown, a large diversity is observed as a result of the engineering efforts to propose designs with optimized endurance, agility, controllability, or even simplicity in a very wide scale range.

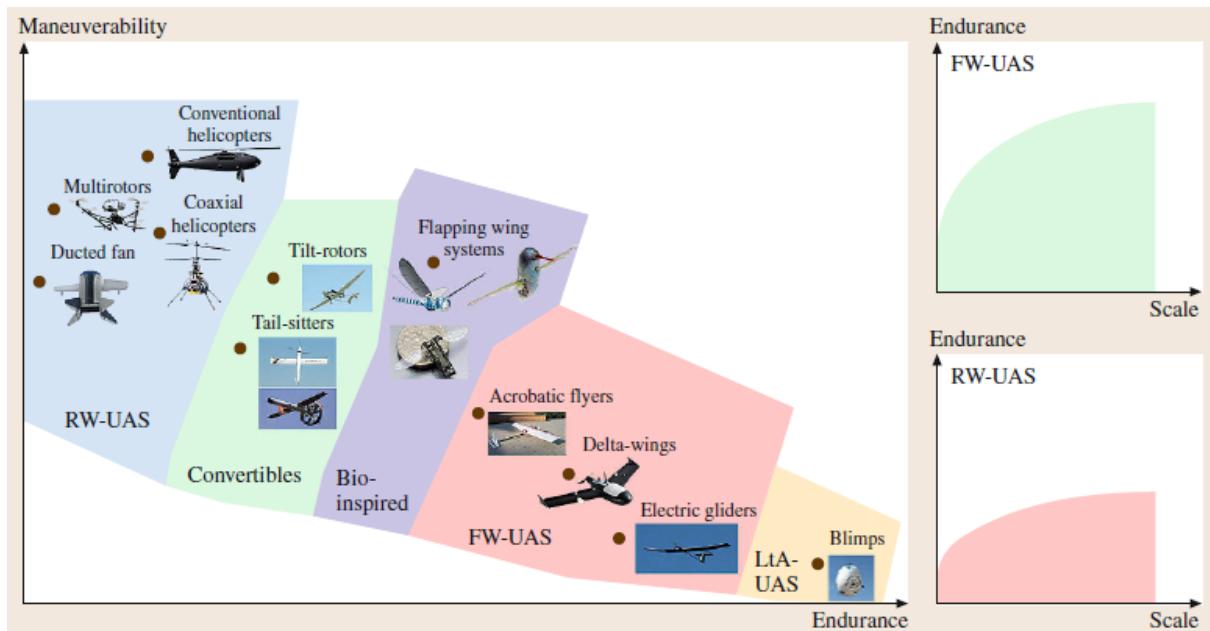


Figure 1.7: Classification of Aerial Robotics Based on Their Endurance and Maneuverability Properties.

## 1.5 Applications of UAVs

Unmanned aerial vehicles serve in most of our applications to ease our life and relieve us from the boring duties.

1. UAVs in Military and law enforcement: Quadcopter unmanned aerial vehicles are used for surveillance and reconnaissance by military and law enforcement agencies, as well as search and rescue missions in urban environments.

2. UAVs in Photography : The largest use of UAVs in the USA has been in the field of aerial imagery. UAVs are suitable for this job because of their autonomous nature and huge cost savings
3. UAVs in Journalism In 2014 The Guardian reported that major media outlets have started to put serious effort into exploring the use of drones for reporting and verifying news on events that include floods, protests and wars.
4. UAVs in Delivery In December 2013, the Deutsche Post gathered international media attention with the project Parcelcopter, in which the company tested the shipment of medical products by drone delivery. Also amazon uses drones to deliver its products.
5. Others

## 1.6 Challenges That Faces UAVs

Unmanned air vehicles especially quadcopters face many challenges. First of all the inability to recognize and avoid other aircraft and airborne objects in a manner similar to manned aircraft. Also a lack of technological and operational standards needed to guide safe and consistent performance of UAVs. last but not least Vulnerabilities in the command and control of UAV operations such as (GPS-Jamming, hacking and the potential for cyberterrorism).

## 1.7 Problem Statement

Quadcopters are unstable systems by nature. in order to navigate unknown areas with unknown disturbances and track a certain trajectory, control is needed to maintain the system stability and the trajectory all the time. controlling a quadcopter to do some applications is complicated and requires alot of computation time. the aim of this thesis is to apply control to the quadcopter dynamic model in order to achieve stability and reference tracking.

## 1.8 Thesis Structure

The thesis discusses these objectives in the next chapters. The chapters are categorized as follows:

- Chapter 1 presents an introduction to different types of locomotions. in addition to a brief introduction about UAVs and its types and applications.
- Chapter 2 reviews the literature on UAVs control focusing on rotary wing platforms.

- Chapter 3 discusses the Methodology.
- Chapter 4 presents the results acquired during tuning of the control algorithm parameters. Further more, the chapter presents the results of the trajectory tracking.
- Chapter 5 discusses the conclusion reached after reviewing the results and recommends future improvements to the project.

# **Chapter 2**

## **Literature Review**

From what we have discussed in the previous section 1, one task can be handled by more than one approach. Our main focus is quadcopter control. We will get through some researches and investigations targeting this point of concern.

In 2016, a study [3] was conducted by Pengcheng Wang et. al. . a cascade Proportional Integral Derivative (PID) feedback control algorithm is proposed to stabilize the attitude of a quadcopter so that the balancing state can be ensured regardless the disturbances. A mathematical model of quadcopter dynamics is formulated by applying Newton-Euler method. It shows the exact relationships among all the variables involved. Both linear and nonlinear state-space equations are derived afterwards, which are mandatory for the controller design and further development. The simulations are also carried out to demonstrate the effectiveness and robustness of the cascade PID algorithm in contrast with classic PID control scheme. A robust cascade PID control algorithm has been implemented based on the dynamic model of a quadcopter in this paper. The main advantage of the cascade PID control scheme is its strong robustness regarding the external disturbances. Furthermore, classic and cascade PID control schemes have been compared to determine the effectiveness of the designed controller.

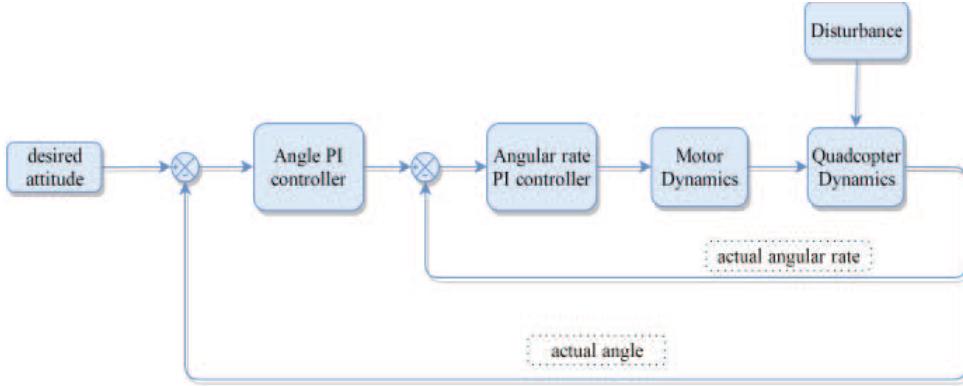


Figure 2.1: Cascade PID Control Structure

In 2005, another study [4] was conducted by Samir Bouabdallah et. al. . In this paper the results of two nonlinear control techniques applied to an autonomous quadcopter are discussed. A backstepping and a sliding-mode techniques. this study executes various simulations in open and closed loop and implements several experiments on the test-bench to validate the control laws. Finally, the results of each approach is discussed. the two nonlinear control techniques are tested for OS4 configuration stabilization. the sliding mode approach shows that it provides average results as a result of the switching nature of the controller which introduces high frequency, low amplitude vibrations provoking the sensor to drift. in contrast, the backstepping controller shows the ability to control the orientation angles in the presence of a approximately high perturbations.

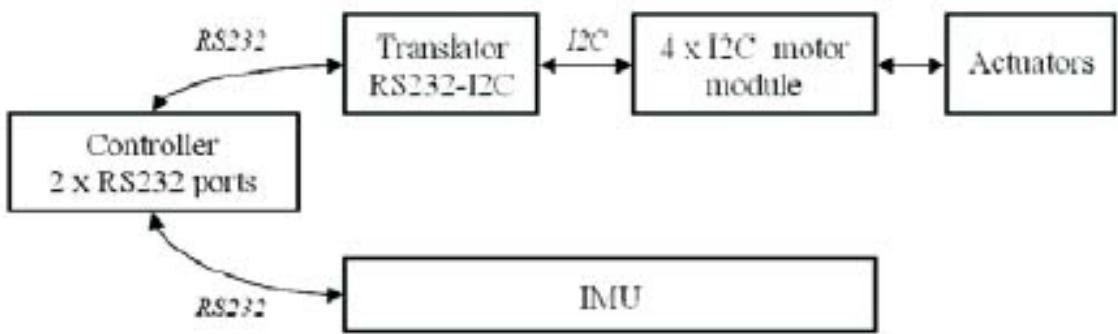


Figure 2.2: OS4 Test-Bench Block Diagram

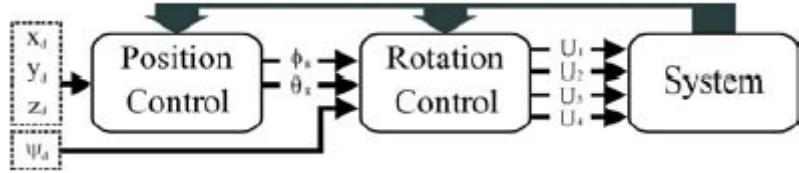


Figure 2.3: Control Scheme

In 2013, a study [5] was conducted by Amr Nagaty et. al. . This paper presents the development of a nonlinear quadcopter simulation framework collectively with a nonlinear controller. The quadcopter stabilization and navigation issues are tackled using a nested loops control architecture. A nonlinear Backstepping controller is applied for the inner stabilization loop. It asymptotically tracks reference attitude, altitude and heading trajectories. The outer loop controller generates the reference trajectories for the inner loop controller to attain the desired set-point. To ensure boundedness of the reference trajectories, a PD controller with a saturation function is used for the outer loop. Due to the complexity involved in controller development and testing, a simulation framework has been developed. It is primarily based on the Gazebo 3D robotics simulator and the Open Dynamics Engine (ODE) library.

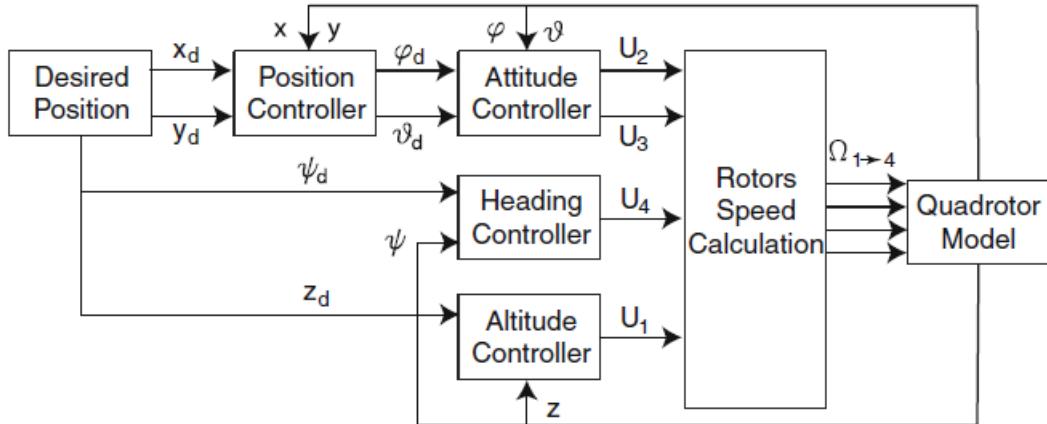


Figure 2.4: Control Architecture

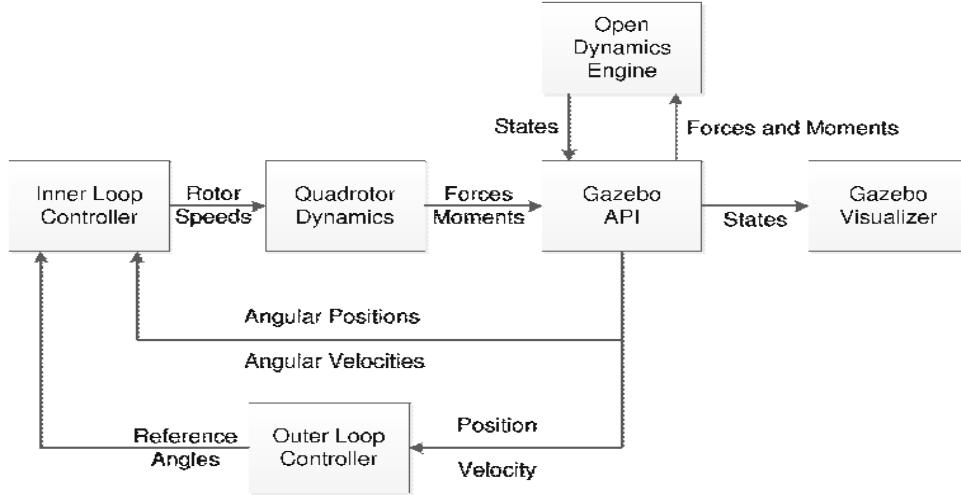


Figure 2.5: Simulator Framework

In 2015, another study [6] was conducted by S. Faiz Ahmed et al. . In this paper a Fuzzy Logic Based Proportional Integral Derivative (FPID) for attitude stabilization of quadcopter Unmanned Arial Vehicle (UAV) is presented. The Fuzzy logic controller keeps updating the PID controller gains in an efficient way such that it stabilizes the attitude of quadcopter. This technique enhances the capabilities of conventional PID Controller to a dynamic PID Controller. The proposed FPID control approach for attitude controlling of quadcopter was tested on developed prototype of quadcopter for experimental purpose. Experimental results proved that proposed Fuzzy Logic based PID (FPID) controller quickly stabilized quadcopter and its response time and settling time is reasonably precise in attitude stabilization control application.

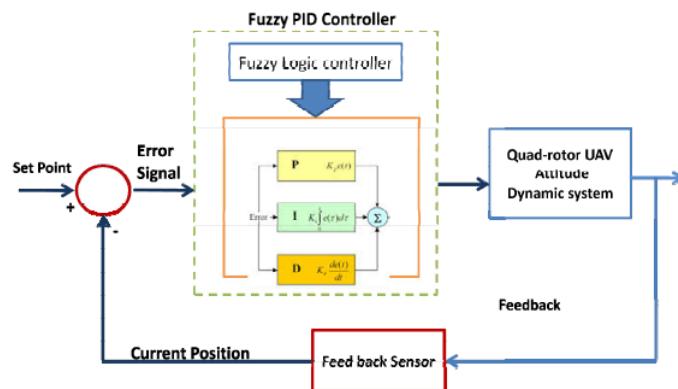


Figure 2.6: Attitude Stabilization Controller Block Diagram Using FPID

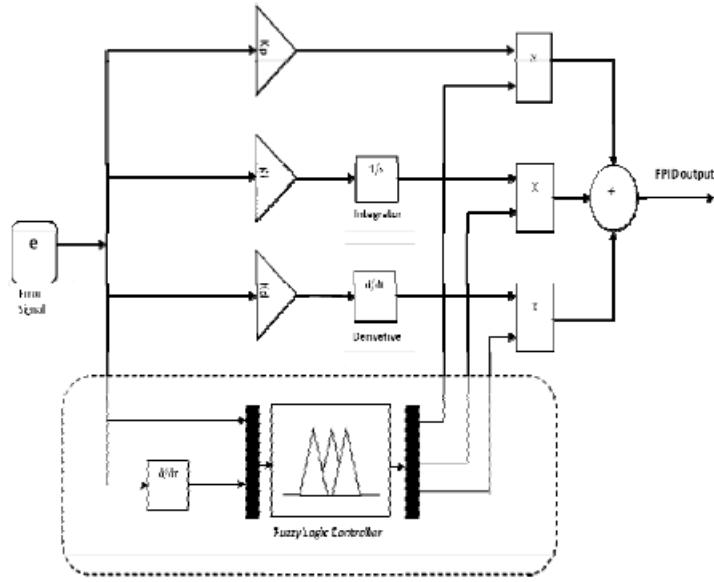


Figure 2.7: FPID Controller Block Diagram

In 2015, another study [7] was conducted by Ramy Rashad et. al. . This paper provides a trajectory tracking controller based totally on the backstepping technique. To avoid the increasing complexity of analytically calculating the derivatives of the virtual control signals in general backstepping control, a command filtered backstepping method is used. The command filtered backstepping controller is modified to include integral action to enhance robustness against external disturbances and unmodeled dynamics. The stability proof of the command filtered integral backstepping method is introduced primarily based on Lyapunov's theorem. The controller is applied on a quadcopter in simulation and compared to a general integral backstepping controller. Simulation results exhibit that the introduced controller yields an enhancement in the tracking overall performance of the quadcopter in the presence of steady disturbances and unmodeled actuator dynamics with lower control effort.

In 2014, a study [8] was conducted by Mohd Ariffanan et. al. . an Intelligent Backstepping Control (IBC) is designed for the quadcopter altitude and attitude stabilization in the existence of external disturbances and measurement noise. The designed controller consists of a backstepping controller which can automatically choose its parameters online by means of a fuzzy supervisory mechanism. The stability criterion for the stabilization of the quadcopter is validated by means of the Lyapunov theorem. Several numerical simulations using the dynamic model of a four Degree Of Freedom (DOF) quadcopter exhibit the effectiveness of the approach. Besides, the simulation results point out that the proposed design techniques can stabilize the quadcopter helicopter with exceptional performance than established linear design techniques. Several simulation results show that high-precision transient response can be achieved by using the proposed control system.

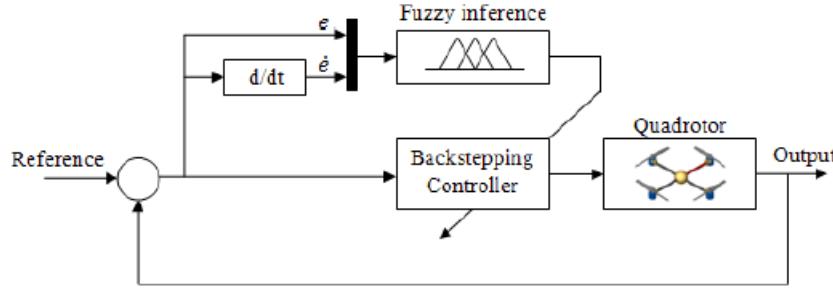


Figure 2.8: Structure of The Intelligent Backstepping Controller

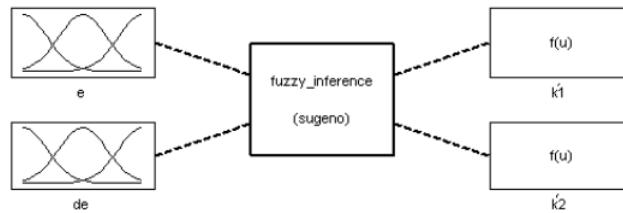


Figure 2.9: Structure of The Fuzzy Inference Block

In 2017, another study [9] was conducted by Maidul Islam et. al. . This study investigates the dynamics and control of a quadcopter using the Model Predictive Control (MPC) approach. The dynamic model is of high fidelity and nonlinear, with six degrees of freedom that encompass disturbances and model uncertainties. The control strategy is developed primarily based on MPC to track different reference trajectories ranging from simple ones such as circular to complex helical trajectories. In this control technique, a linearized model is derived and the Receding Horizon Control (RHC) is utilized to generate the optimal control sequence. Although MPC is computer expensive, it is exceptionally effective to deal with the different types of nonlinearities and constraints such as actuators saturation and model uncertainties. The MPC parameters (control and prediction horizons) are chosen through trial-and-error approach. Several simulation scenarios are carried out to study and evaluate the overall performance of the proposed control approach using MATLAB and Simulink environment. Simulation results show that this control method is exceptionally effective to track a given reference trajectory.

In 2017, a study [10] was conducted by Maidul Islam et. al. . This paper discuss a comparative evaluation of performance of two distinct controllers i.e. Proportional-Derivative Controller (PD) and Linear Quadratic Regulation (LQR) in quadcopter dynamic system that is under-actuated with high nonlinearity. As only four states can be controlled at the same time in the quadcopter, the trajectories are designed on the basis of the 4 states whereas three dimensional position and rotation along an axis, known as yaw motion are considered. In this work, both the PD controller and LQR control method are used for quadcopter nonlinear model to

track the trajectories. LQR control strategy for nonlinear model is designed on the basis of a linear model of the quadcopter due to the fact that the overall performance of linear model and nonlinear model around certain nominal point is nearly similar. LQR control method shows robustness and generates very low steady-state error however preserving update twelve states at the same time may additionally create a transitional delay whereas the fast response is essential during flight. However, PD controller can provide faster response and overcome steady-state error through adding an extra integral part in controller however it failed to give robust performance like LQR control approach. The controllers could not provide exact performances as these are supposed to follow the trajectories because of not tuning proper gains while it is the toughest phase of designing controllers perfectly. The gains for PD controllers are tuned with the aid of Genetic Algorithm (GA). As it is difficult to manage ten gains for GA optimization simultaneously, the optimizer tuned the best feasible gains as it could. However, if the optimization can be performed iteratively, it is viable to tune the ideal gains. On the other hand, Q and R matrices are chosen arbitrarily to calculate gains for LQR control method and it also can be improved by using several techniques like Brysons rule or optimization etc. . If the suitable gains can be tuned properly, the settling time and rise time can be decreased more.

In 2017, a study [11] was conducted by Ye Wang et. al. . This paper provides a Nonlinear Model Predictive Control (NMPC) approach combined with constraint satisfactions for a quadcopter. The full dynamics of the quadcopter describing the attitude and position are nonlinear, which are pretty sensitive to changes of inputs and disturbances. By means of constraint satisfactions, partial nonlinearities and modeling errors of the control-oriented model of full dynamics can be converted into the inequality constraints. Subsequently, the quadcopter can be controlled by using an NMPC controller with the updated constraints generated through constraint satisfactions. Finally, the simulation results applied to a quadcopter simulator are provided to show the effectiveness of the proposed strategy. From the simulation results, the NMPC controller is able to grant pretty accurate tracking results and the inherent robustness of the NMPC controller is able to deal with uncertainties. On the other hand, the nonlinear optimization problem is no longer easy to be solved compared with the linear one requiring more time to find a solution. Taking into account the continuous hardware and software development in the computer science field, the nonlinear programming approach is expected to find the answer that satisfies the required real-time constraints permitting the real implementation.

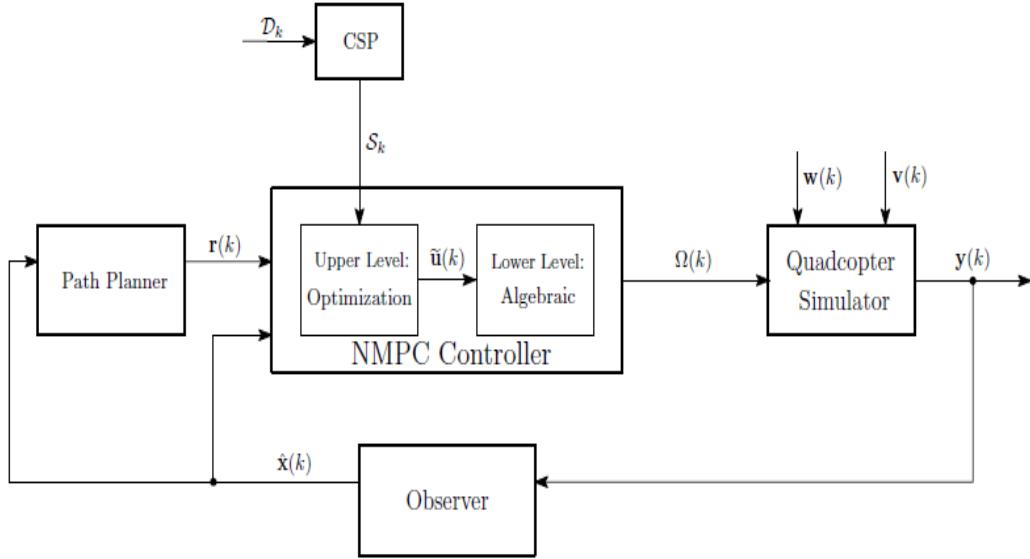


Figure 2.10: The Closed-Loop Simulation Scheme of The NMPC Strategy With Constraint Satisfaction For The Quadcopter

In 2006, a study [12] was conducted by A. Benallegue et. al. . A feedback linearization-based controller with a high order sliding mode observer running parallel is applied to a quadcopter. The high order sliding mode observer works as an observer and estimator of the impact of the external disturbances such as wind and noise. The entire observer-estimator-control law constitutes an original method to the vehicle regulation with minimal number of sensors. so the principal motivation of this paper are Feedback linearization controller of the quadcopter needs the third derivatives of measured states in order to reconstruct tilt angles and to fulfill the controller requirement, when quadcopter is subjected to external disturbances, it would be suitable to compensate them through an observer primarily based controller and The observers should be robust with respect to external perturbations (wind and noise). A feedback linearization controller using high order sliding mode observer has been applied to a quadcopter. Although the behavior of the quadcopter, affected by aerodynamic forces and moments, is non linear and highly coupled, the feedback linearization coupled to HOSM observer and applied to the UAV, turns out to be a proper starting point to avoid complex nonlinear control solutions and immoderate chattering. However, in the presence of nonlinear disturbances the system after linearization stays nonlinear. The observer used here overcomes effortlessly this nonlinearities with the aid of an internal estimation of the external disturbances to impose desired stability and robustness properties on the global closed loop system. The unmeasured states and their derivatives have been efficiently reconstructed through the sliding mode observer design. Theoretical results have been supported by numerical simulations that validated the efficiency of the proposed controller design.

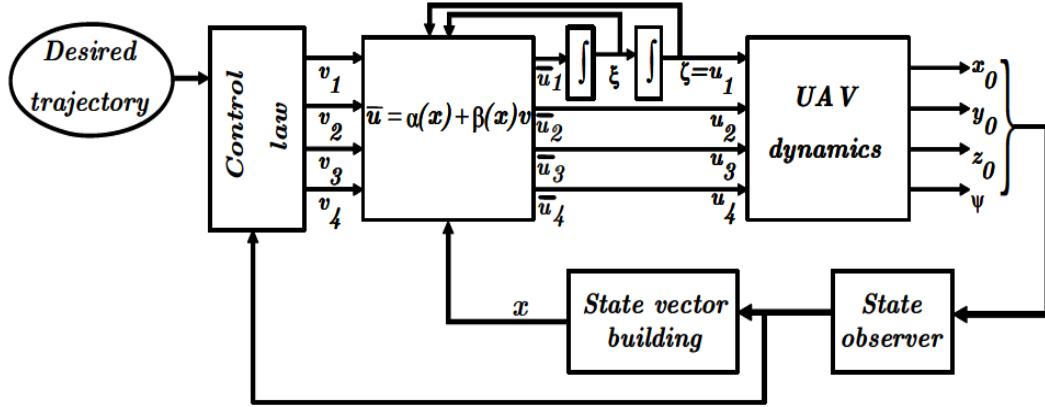


Figure 2.11: The Overall Closed-Loop System

In 2015, a study [13] was conducted by Mapopa Chipofya et. al. . This paper presents a solution to stability and trajectory tracking of a quadcopter system using a model predictive controller designed using a type of orthonormal functions referred to as Laguerre functions. A linear model of the quadcopter is derived and used. To test the overall performance of the controller they compared it with a linear quadratic regulator and a more traditional linear state space MPC. Simulations for trajectory tracking and stability are carried out in MATLAB. This paper has introduced a solution to stability and trajectory tracking of quadcopter systems using a model predictive controller designed by using a special type of orthonormal functions referred to as Laguerre functions. A quadratic cost function similar to that used in linear quadratic regulator (LQR) has been used. In stability analysis Linear Model Predictive Control (LMPC) has been compared to the optimal DLQR system whereas in trajectory tracking LMPC has been compared to a popular linear statespace MPC desgin approach in which plant constraints are modeled using linear equalities and inequalities. The results from the simulations point out that the controller performs very properly and is considered feasible. With this perceived feasibility LMPC gives the added advantage that it can manage systems where rapid sampling and more complicated process dynamics are required. By choosing the suitable gains NMPC reduces the number of parameters required for precise prediction when using the traditional (MPC) approach. This is a huge advantage in that, with the reduced number of parameters, online implementation may be feasible where the traditional MPC would have failed.

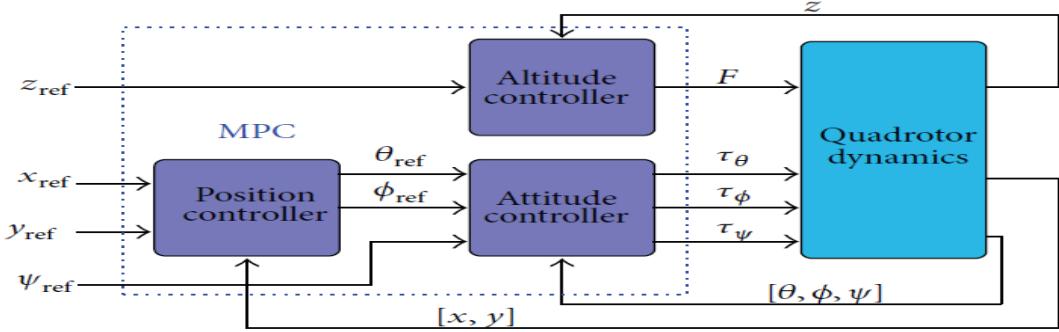


Figure 2.12: Block Diagram of The Controllers Structure

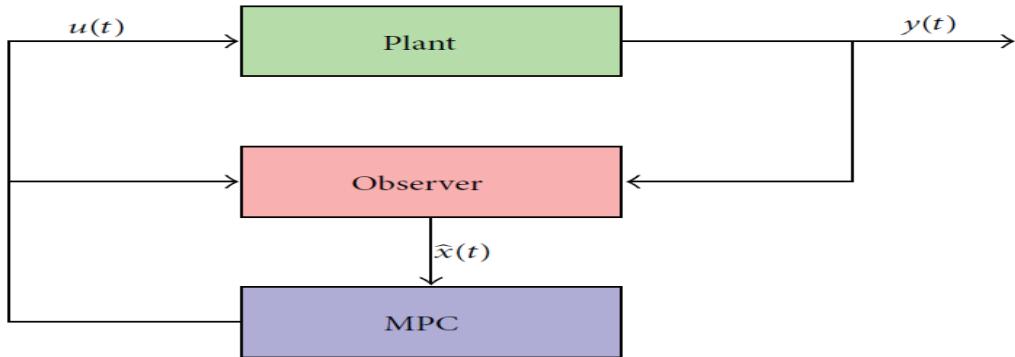


Figure 2.13: Block Diagram of The Proposed MPC

In 2016, another study [14] conducted by A. Chovancova et. al. . The purpose of this article is to design and validate various control techniques for a quadrotor using a quaternion representation of the attitude. All attitude controllers use a quaternion error to compute control signals that are calculated from an actual quaternion and a desired quaternion obtained from a function controller. Attitude and position control laws are computed using a PD, LQR and backstepping control technique. All combinations of controllers will be validated by simulation. a noise is added, apply an actuator restriction and use a different sampling period for position and attitude feedback signals to get the simulation closer to real conditions. Moreover, external disturbances were applied into the simulation; therefore a disturbance observer alongside with a position estimator will be designed to enhance the overall performance of the presented controllers. The performance of all combinations of controllers was evaluated using various quality indicators, such as the integral of absolute errors and total thrust, settling times and additionally maximum overshoots when external disturbance was applied. Two scenarios have been proposed to verify all combinations of attitude and position controllers. The first one was following the desired trajectory, where the movement along all axes and additionally the rotation round the z axis was ordered. The occurrence of an external disturbance was introduced in the second scenario.

In order to set the simulation closer to reality some adjustments were made, namely the consideration of actuator restrictions, the different sampling for feedback signals and the addition of noise to all feedback signals. The quality indicators were chosen in order to be able to compare the performance of all controllers. Some of the controllers show very similar behaviour.

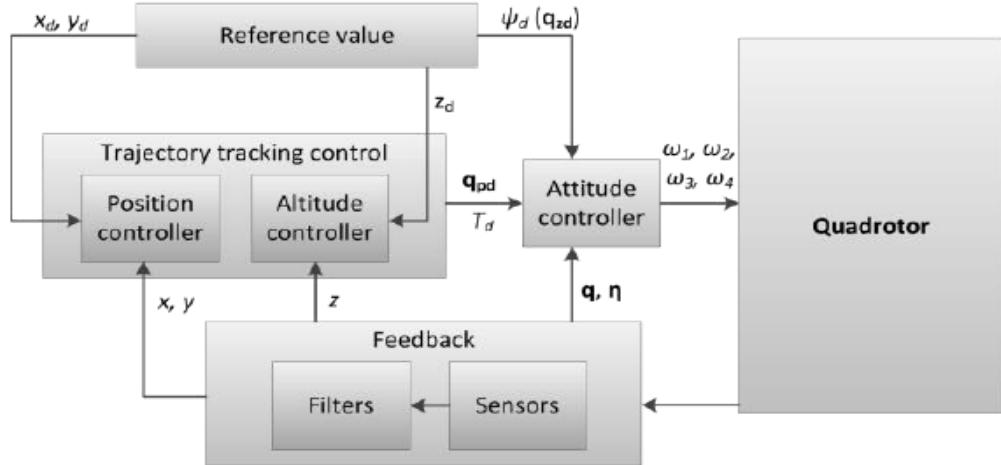


Figure 2.14: Block Diagram of The Proposed Control Structure

## 2.1 Thesis Proposition

after reviewing the literature it was concluded that the problem of quadcopters control was handled by researchers in many different ways.

This thesis presents a Model Predictive Control for quadcopters based on the literature and on the following points :

- Easy tuning.
- Control a great variety of processes.
- The multivariable case can be easily dealt with.
- Disturbance rejection. (can handle unmodeled parameters)
- Disturbance rejection. (can handle unmodeled parameters)

# **Chapter 3**

## **Methodology**

### **3.1 System Modelling**

#### **3.1.1 Basic Concepts**

Quadcopter is a four rotor rotary wing UAV, which has a rigid compact shape without a tail. Quadcopter shares the same dynamics with different rotary wing UAVs (hexacopter, octocopter, etc.), however it has the lowest power consumption. a majority of these characteristics made quadcopter this project research platform choice.

Quadcopter has two coordinate frames. A right hand inertial frame (Earth frame) denoted by  $\{ A \}$  with unit vectors along its axes denoted by  $\{ a_1, a_2, a_3 \}$ . The position of the center of mass of the quadcopter is defined relative to the inertial frame. A right hand body frame is used to define the quadcopter orientation. The body frame is denoted by  $\{ B \}$  with unit vectors along of its axes by  $\{ b_1, b_2, b_3 \}$ . Quadcopter has different two configurations: X and PLUS configuration. The body frame axis orientation varies between both configurations. In PLUS configuration, X axis lies along arm of motor one and Y axis lies along arm of motor two and Z axis is directed upward. Both motor one and two spin in opposite direction and are placed at an equivalent distance  $d$  from the center of mass of quadcopter.

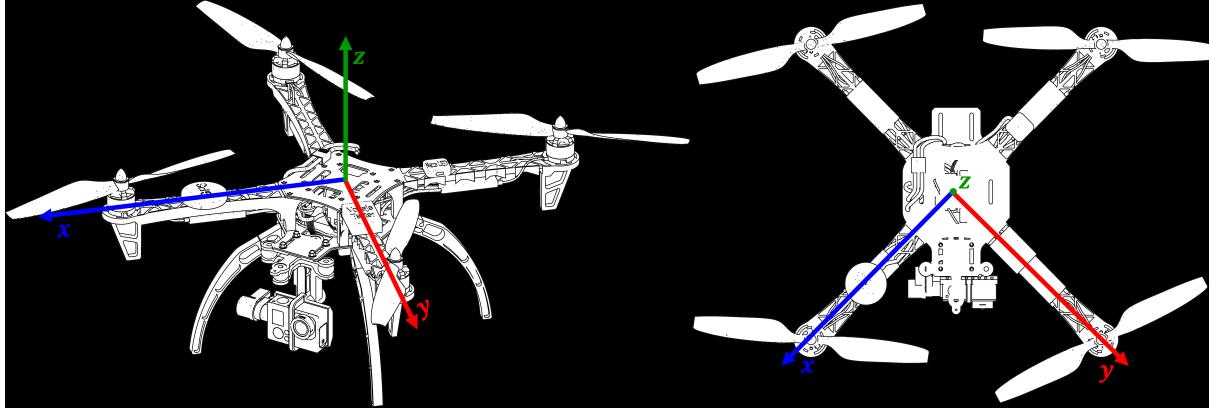


Figure 3.1: PLUS Quadcopter

In X configuration X-Y plane is rotated 45 degrees in the positive yaw direction. X axis lies symmetrically between motor one and two, Y axis lies symmetrically between motor two and three and Z axis is directed upward.

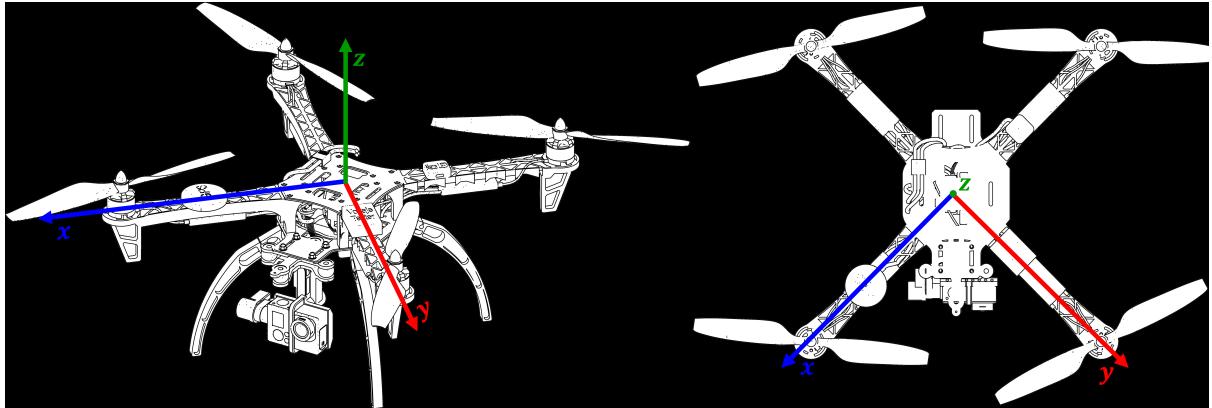


Figure 3.2: X Quadcopter

Global frame origin is defined as the starting point on the ground, from where the quadcopter takes off. Quadcopter position is described by the position vector between body frame origin and global frame origin. Quadcopter has six degrees of freedom: three rotary motions (Roll, Pitch and Yaw) and three translation motions in X, Y and Z axes. Roll ( $\phi$ ) is rotation about body frame X axis, Pitch ( $\theta$ ) is rotation about body frame Y axis and Yaw ( $\psi$ ) is rotation about body frame Z axis.

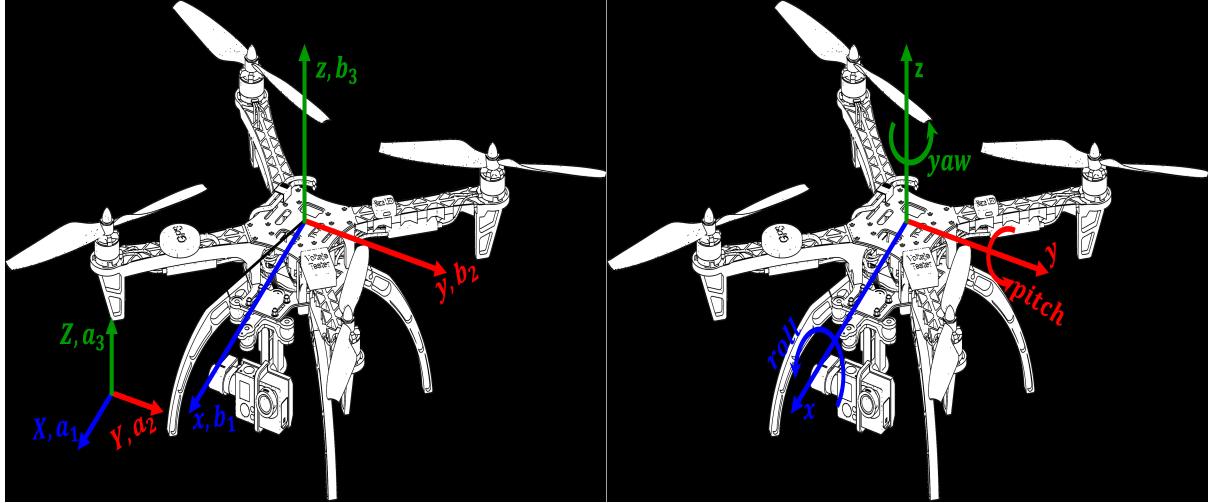


Figure 3.3: Inertial Frame and Body Frame Quadcopter

Each motor has attached to it a fixed pitch propeller blade and all the propellers' axes of rotation are parallel. By spinning these four propellers at different speeds the thrust force produced by the four rotors changes and all the common directional movements of a standard helicopter are attainable; hover, forward/backward movement, left/right movement, and yaw (turn rate) movement. If all motors of the quadcopter rotate in same direction, the heading angle of the quadcopter will be uncontrollable as nonzero drag moment is obtained. so two motors will spin in the same direction and the other two will spin in opposite direction, for example Propellers 1 and 3 rotate (clockwise) and create drag moment in counter clockwise direction while propellers 2 and 4 rotate in (counter clockwise) and create drag force in clockwise direction leading to balancing the total system torque. Moreover, this configuration of opposite directions removes the need for a tail rotor in the conventional helicopter.

Taking into consideration that this system is an under-actuated system; due to the fact that only 4 inputs(motors) control the movement of the aircraft in space while it is free to move in six degrees of freedom(three axes of translation and an angle of rotation about each translational axis), so only a maximum of four desired set-points of degrees of freedom can be achieved at one time. However, thanks to its structure, it is quite easy to choose the four best controllable variables and to decouple them to make the controller easier. The four actuators directly impact z-axis translation and rotation about each of the three principal axes, therefore, the four basic movements will allow the quadcopter to reach a desired height(altitude) and angle(attitude).

### 3.1.1.1 Throttle (U1)

This input is mainly responsible for lifting the whole aircraft upwards in the positive direction of the body-fixed z-axis (Figure 3.4a). This is simply done by increasing the speeds of the 4

propellers equally by  $\Delta_A$  hence, increasing the thrust force generated by each rotor leading to the increase of the net force acting upwards. For sustaining hovering condition, this input should be exactly equal to the weight of the aircraft. Increasing and decreasing the value of the throttle around the weight of the quadcopter will cause it to ascend and descend respectively.

### **3.1.1.2 Roll Input (U2)**

The roll input adjusts the speeds of the rotors to rotate the quadcopter around its x-axis. This rotation is desired to move the whole vehicle in the direction of positive or negative y -axis; because during the inclination of the aircraft, there is a component of the thrust force in the direction of y-axis. The rotation can be achieved by creating a balanced difference between the speeds of the right and left propellers (Figure 3.4b) while keeping the speeds of the other two propellers constant. This will increase the thrust force one side while decreasing it on the opposite one which will lead to torque generation due to the coupling effect. The direction of rotation is defined by the right- hand thumb rule around the x-axis; so a positive roll input would increase angular speed of the left propeller by  $\Delta_A$  and decrease the speed of the right propeller by  $\Delta_B$  and vice-versa for the negative input, where  $\Delta_A = \Delta_B$  for small speed changes.

### **3.1.1.3 Pitch Input (U3)**

Similar to the rolling input U2, this input rotates the quadcopter around its y-axis. The angular displacement resulting from such rotation enables the aircraft to travel along the positive or negative direction parallel to the quad-rotors x-axis. The direction of change of the angular rotation is also determined by the sign of the pitch input. In the same manner, pitching occurs when a difference appears between the speeds of the front and rear propellers (Figure 3.4c). This difference produces a couple on the body of the quadcopter generated from the change in thrust from each motor. As a result, a pitching torque is generated. The direction of rotation is found by applying the right hand thumb rule on the body-fixed y-axis. Moreover, changes of speeds in the front and rear motors should be equal (i.e.  $\Delta_A = \Delta_B$  ).

### **3.1.1.4 Yaw Input (U4)**

Finally, this last input is responsible for the rotation of the aircraft around its z-axis; changing the rate of change of yaw angle (i.e. changing the heading of the quadcopter). Similar to the previous inputs, the sign of this input determines the direction of change of the angular velocity. However, the concept of operation of this input is rather different than that of the previous inputs. The theory of operation of this rotation depends completely on the drag torque generated by the rotation of each of the four propellers. As mentioned earlier, the rotation of each propeller induces a drag torque opposite to the direction of rotation. Moreover, the torque

generated from each two opposite rotors is opposite in direction with the torque generated from the other two rotors, and since the torque is proportional to square the speeds of the propellers the heading of the aircraft could be easily changed by changing the speeds of each two opposite rotors simultaneously with the same rate and same direction. For example, in order to rotate the quadcopter around the z-axis in the positive direction (according to the right hand thumb rule), it is required to increase the speeds of the rotors rotating in the clockwise direction by  $\Delta_A$  while decreasing the speeds of the other two motors by  $\Delta_B$  (Figure 3.4d). The assumption made in the previous inputs is still valid here where  $\Delta_A = \Delta_B$ .

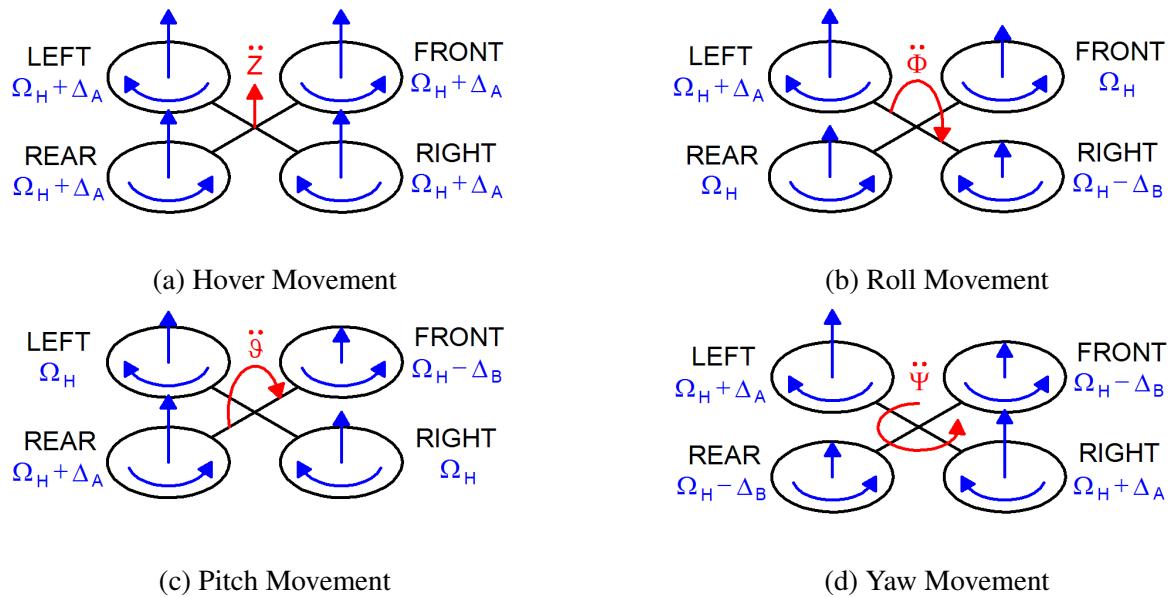


Figure 3.4: Quadcopter Movements

### 3.1.2 Newton-Euler Method

This section provides the specific model information of the quadrotor architecture starting from the generic 6 DOF rigid-body equation derived with the Newton-Euler formalism with the following assumptions:

- The structure is rigid and symmetrical.
- The center of gravity of the quadrotor coincides with the body fixed frame origin.
- The propellers are rigid.
- Thrust and drag are proportional to the square of propellers speed.
- The Gyroscopic Moment is neglected due to the linearization.

### 3.1.3 Kinematics

In order to discuss the modeling of the quadrotor, we first need to define the coordinate frames that will be used. Figure 3.5 shows the Earth reference frame with  $x$ ,  $y$  and  $z$  axes and the body frame with  $x_b$ ,  $y_b$  and  $z_b$  axes.

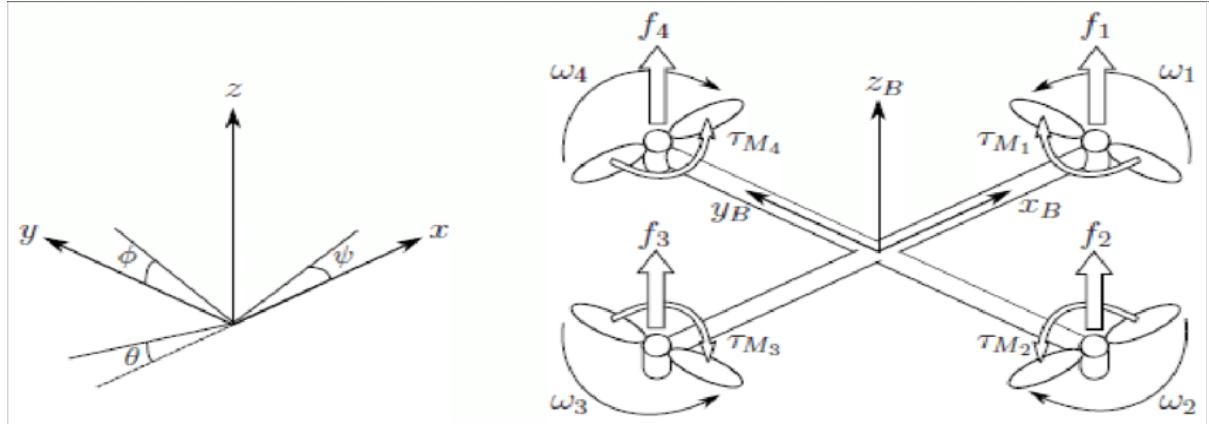


Figure 3.5: Quadcopter Reference Frames

The distance between the Earth frame and the body frame describes the absolute position of the center of mass of the quadcopter  $r = [x \ y \ z]^T$ . The rotation matrix  $R$  from the body frame to the inertial frame describes the orientation of the quadcopter. The orientation of the quadcopter is described using roll, pitch and yaw angles ( $\phi, \theta$  and  $\psi$ ) representing rotations about the X, Y and Z-axes.

The rotation matrix  $R$  will be used in formulating the dynamics model of the quadcopter, its significance is due to the fact that some states are measured in the body frame (e.g. the thrust forces produced by the propellers) while some others are measured in the inertial frame (e.g. the gravitational forces and the quadcopter's position). Thus, to have a relation between both types of states, a transformation from one frame to the other is needed.

Considering a right-hand oriented coordinate system, the three single rotations are described separately by:

- $R(x, \phi)$ , rotation around  $x - axis$
- $R(y, \theta)$ , rotation around  $y - axis$
- $R(z, \psi)$ , rotation around  $z - axis$

They are represented by:

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.1)$$

$$R(y, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.2)$$

$$R(z, \psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

The complete rotation matrix is the product of the previous three successive rotations:

$$R(\phi, \theta, \psi) = R(x, \phi)R(y, \theta)R(z, \psi) \quad (3.4)$$

which results in:

$$R = \begin{bmatrix} \cos(\psi)\cos(\theta) & \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi) \\ \sin(\psi)\cos(\theta) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) - \sin(\phi)\cos(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \quad (3.5)$$

To acquire information about the angular velocity of the quadcopter, typically an on-board Inertial Measurement Unit (IMU) is used which will in turn give the velocity in the body coordinate frame  $(p, q, r)$ . To relate the Euler rates  $\dot{\eta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$  that are measured in the inertial frame and angular body rates  $\omega = [p \ q \ r]^T$ , a transformation is needed in order to transform from  $\omega$  to  $\dot{\eta}$  as follows:

$$\omega = R_r \dot{\eta} \quad (3.6)$$

where

$$R_r = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (3.7)$$

Around the hover position, small angle assumption is made where  $\cos(\phi) \equiv 1$ ,  $\cos(\theta) \equiv 1$  and  $\sin(\phi) = \sin(\theta) = 0$ . Thus  $R_r$  can be simplified to an identity matrix I.

### 3.1.4 Dynamics Model

The motion of the quadcopter can be divided into two subsystems; rotational subsystem (roll, pitch and yaw) and translational subsystem (altitude and x and y position). The rotational subsystem is fully actuated while the translational subsystem is under-actuated.

#### 3.1.4.1 Rotational Equations of Motion

The rotational equations of motion are derived in the body frame using the Newton- Euler method with the following general formalism,

$$J\dot{\omega} + \omega \times J\omega + M_G = M_B \quad (3.8)$$

Where:

- $J$  Quadcopter's diagonal inertia Matrix.
- $\omega$  Angular body rates.
- $M_G$  Gyroscopic moments due to rotor's inertia :  $M_G = \omega \times [0 \ 0 \ J_r \omega_r]$
- $M_B$  Moments acting on the quadcopter in the body

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \ 0 \ J_r \omega_r]^T = M_B \quad (3.9)$$

Where:

- $J_r$  rotors' inertia.
- $\omega_r$  rotor's relative speed  $\omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4$ .

The inertia matrix for the quadcopter is a diagonal matrix. Where  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  are the area moments of inertia about the principle axes in the body frame.

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.10)$$

Moments acting on the quadcopter ( $M_B$ ), there is a generated force called the aerodynamic force or the lift force and there is a generated moment called the aerodynamic moment due to the rotation. the following equations show the aerodynamic force and the aerodynamic moment for the  $i^{th}$  rotor.

$$F_i = \frac{1}{2} \rho A C_T r^2 \omega_i^2 \quad (3.11)$$

$$M_i = \frac{1}{2} \rho A C_D r^2 \omega_i^2 \quad (3.12)$$

Where:

$\rho$	Air density.
$A$	Blade area.
$C_T, C_D$	aerodynamic coefficients
$r$	radius of the blade
$\omega_i$	angular velocity of rotor $i$

the aerodynamic forces and moments depend on the geometry of the propeller and the air density. Since for the case of quadcopter, the maximum altitude is usually limited, thus the air density can be considered constant, thus the equations can be simplified to:

$$F_i = K_f \omega_i^2 \quad (3.13)$$

$$M_i = K_M \omega_i^2 \quad (3.14)$$

Where  $K_f$  and  $K_M$  are the aerodynamic force and moment constants respectively and  $\omega_i$  is the angular velocity of rotor the  $i^{th}$ . the aerodynamic force and moment constants can be determined experimentally for each propeller type.

The total moment about the x-axis can be expressed as:

$$M_x = l K_f (\omega_4^2 - \omega_2^2) \quad (3.15)$$

The total moment about the y-axis can be expressed as:

$$M_y = l K_f (\omega_3^2 - \omega_1^2) \quad (3.16)$$

The total moment about the z-axis can be expressed as:

$$M_z = K_m (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (3.17)$$

Where  $l$  is the distance between the axis of rotation of each rotor to the origin of the body reference

frame. Also  $M_B$  can be formulated as:

$$M_B = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} U2 \\ U3 \\ U4 \end{bmatrix} = \begin{bmatrix} lK_f(\omega_4^2 - \omega_2^2) \\ lK_f(\omega_3^2 - \omega_1^2) \\ K_M(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (3.18)$$

Therefore, by using the assumption mentioned above while formulating  $R_r$  that it can be simplified to an identity matrix  $I$ , we can substitute in 3.9 :

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r\omega_r \end{bmatrix} = \begin{bmatrix} U2 \\ U3 \\ U4 \end{bmatrix} \quad (3.19)$$

Rewriting the previous equations, leads to:

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta}\dot{\psi}I_{zz} - \dot{\phi}\dot{\psi}I_{yy} \\ \dot{\psi}\dot{\phi}I_{xx} - \dot{\phi}\dot{\phi}I_{zz} \\ \dot{\theta}\dot{\phi}I_{yy} - \dot{\theta}\dot{\phi}I_{xx} \end{bmatrix} + \begin{bmatrix} \dot{\theta}J_r\omega_r \\ -\dot{\phi}J_r\omega_r \\ 0 \end{bmatrix} = \begin{bmatrix} U2 \\ U3 \\ U4 \end{bmatrix} \quad (3.20)$$

Rewriting the last equation to have the angular accelerations in terms of the other variables:

$$\ddot{\phi} = \frac{1}{I_{xx}}U2 - \frac{J_r}{I_{xx}}\dot{\theta}\omega_r + \frac{I_{yy}}{I_{xx}}\dot{\psi}\dot{\theta} - \frac{I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} \quad (3.21)$$

$$\ddot{\theta} = \frac{1}{I_{yy}}U3 - \frac{J_r}{I_{yy}}\dot{\phi}\omega_r + \frac{I_{zz}}{I_{yy}}\dot{\phi}\dot{\psi} - \frac{I_{xx}}{I_{yy}}\dot{\psi}\dot{\phi} \quad (3.22)$$

$$\ddot{\psi} = \frac{1}{I_{zz}}U4 + \frac{I_{xx}}{I_{zz}}\dot{\theta}\dot{\phi} - \frac{I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} \quad (3.23)$$

By neglecting the gyroscopic effects ( $\omega_r$ ) the equations can be simplified into:

$$\ddot{\phi} = \frac{1}{I_{xx}}U2 + \frac{I_{yy} - I_{zz}}{I_{xx}}\dot{\psi}\dot{\theta} \quad (3.24)$$

$$\ddot{\theta} = \frac{1}{I_{yy}}U3 + \frac{I_{zz} - I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} \quad (3.25)$$

$$\ddot{\psi} = \frac{1}{I_{zz}}U4 + \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi} \quad (3.26)$$

### 3.1.4.2 Translational Equations of Motion

Using Newton's second law the translational equation of motion for the quadcopter is formulated and they are derived in the Earth inertial frame.

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RF_B \quad (3.27)$$

Where:

$r = [x \ y \ z]^T$  Quadcopter's distance from the inertial frame.

$m$  Quadcopter's mass.

$g$  gravitational acceleration  $g = 9.81 / : m/s^2$

$F_B$  non-gravitational forces acting on the quadcopter in the body frame.

By taking a look at the non-gravitational forces acting on the quadcopter at the horizontal position where it's the only force acting on the quadcopter,  $F_B$  can be expressed as:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ K_f(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ U1 \end{bmatrix} \quad (3.28)$$

The first two rows of the force vector are zeros as there is no forces in the X and Y directions, the last row is simply an addition of the thrust forces produced by the four propellers.  $F_B$  is multiplied by the rotation matrix R to transform the thrust forces of the rotors from the body frame to the inertial frame, so that the equation can be applied in any orientation of the quadcopter.

By substituting 3.5 and 3.28 in 3.27 :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - s\phi c\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ U1 \end{bmatrix} \quad (3.29)$$

where  $c(.)$  represents  $\cos(.)$  and  $s(.)$  represents  $\sin(.)$ , by simplifying 3.29, we can obtain:

$$\ddot{x} = \frac{U1}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \quad (3.30)$$

$$\ddot{y} = \frac{U1}{m} (\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi) \quad (3.31)$$

$$\ddot{z} = -g + \frac{U1}{m} (\cos \phi \cos \theta) \quad (3.32)$$

### 3.1.5 Aerodynamic Effects

The aerodynamic effects acting on the quadcopter in the previous dynamics formulation were neglected. but in fact to have an accurate and realistic model to be used in simulations such that it's closer to the real life model, the aerodynamic effects should be taken into consideration. There are two types of aerodynamic effects, drag forces and drag moments.

#### 3.1.5.1 Drag Forces

A force acts on the body of the quadcopter resisting the motion due to the friction of the moving quadcopter body in with air. As the velocity of the quadcopter increases, the drag forces also increases. The drag forces can be approximated and expressed as :

$$F_a = K_t \dot{r} \quad (3.33)$$

where  $K_t$  is a constant matrix called the aerodynamic translation coefficient matrix and  $\dot{r}$  is the time derivative of the position vector  $r$  ( $\dot{r} = [\dot{x} \dot{y} \dot{z}]$ ). This indicates that there is an extra force acting on the quadcopter body, the translational equation of motion 3.27 should be rewritten to be :

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RF_B - F_a \quad (3.34)$$

#### 3.1.5.2 Drag Moments

The same as the drag force, due to the air friction, there is a drag moment  $M_a$  acting on the quadcopter body which can be approximated by:

$$M_a = K_r \dot{\eta} \quad (3.35)$$

where  $K_r$  is a constant matrix called the aerodynamic rotation coefficient matrix and  $\dot{\eta}$  is the Euler rates. Accordingly, the rotational equation of motion expressed by 3.9 can be rewritten to:

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \ 0 \ J_r \omega_r]^T = M_B - M_a \quad (3.36)$$

by neglecting the gyroscopic moments for simplifications as before :

$$J\dot{\omega} + \omega \times J\omega = M_B - M_a \quad (3.37)$$

To sum up the dynamic model of the quadcopter, it can be expressed in the following equations :

$$\ddot{\phi} = \frac{1}{I_{xx}}U2 + \frac{I_{yy} - I_{zz}}{I_{xx}}\dot{\psi}\dot{\theta} - \frac{K_{rx}}{I_{xx}}\dot{\phi} \quad (3.38)$$

$$\ddot{\theta} = \frac{1}{I_{yy}}U3 + \frac{I_{zz} - I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} - \frac{K_{ry}}{I_{yy}}\dot{\theta} \quad (3.39)$$

$$\ddot{\psi} = \frac{1}{I_{zz}}U4 + \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi} - \frac{K_{rz}}{I_{zz}}\dot{\psi} \quad (3.40)$$

$$\ddot{x} = \frac{U1}{m}(\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) - \frac{K_{tx}}{m}\dot{x} \quad (3.41)$$

$$\ddot{y} = \frac{U1}{m}(\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi) - \frac{K_{ty}}{m}\dot{y} \quad (3.42)$$

$$\ddot{z} = -g + \frac{U1}{m}(\cos \phi \cos \theta) - \frac{K_{tz}}{m}\dot{z} \quad (3.43)$$

### 3.1.6 State-Space Model

Formulating the acquired mathematical model for the quadcopter into state-space model will help make the control problem easier to tackle.

Defining the state vector of the quadcopter to be,

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^T \quad (3.44)$$

which is mapped to the degrees of freedom of the quadcopter as follows:

$$X = \begin{bmatrix} z & \dot{z} & \psi & \dot{\psi} & x & \dot{x} & \phi & \dot{\phi} & y & \dot{y} & \theta & \dot{\theta} \end{bmatrix}^T \quad (3.45)$$

A control input vector, U, consisting of four inputs; U1 through U4 is defined as,

$$U = \begin{bmatrix} U1 & U2 & U3 & U4 \end{bmatrix}^T \quad (3.46)$$

Where

$$U1 = K_f(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (3.47)$$

$$U_2 = lK_f(\omega_4^2 - \omega_2^2) \quad (3.48)$$

$$U_3 = lK_f(\omega_3^2 - \omega_1^2) \quad (3.49)$$

$$U_4 = K_m(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \quad (3.50)$$

Using the equations of the rotational angular acceleration. the complete mathematical model of the quadrotor can be written in a state space representation as follows,

$$\dot{x}_1 = \dot{z} = x_2 \quad (3.51)$$

$$\dot{x}_2 = \ddot{z} = -g + \frac{U_1}{m}(\cos x_7 \cos x_{11}) - \frac{K_{tz}}{m}x_2 \quad (3.52)$$

$$\dot{x}_3 = \dot{\psi} = x_4 \quad (3.53)$$

$$\dot{x}_4 = \ddot{\psi} = \frac{1}{I_{zz}}U_4 + \frac{I_{xx} - I_{yy}}{I_{zz}}x_8x_{12} - \frac{K_{rz}}{I_{zz}}x_4 \quad (3.54)$$

$$\dot{x}_5 = \dot{x} = x_6 \quad (3.55)$$

$$\dot{x}_6 = \ddot{x} = \frac{U_1}{m}(\sin x_7 \sin x_3 + \cos x_7 \cos x_3 \sin x_{11}) - \frac{K_{tx}}{m}x_6 \quad (3.56)$$

$$\dot{x}_7 = \dot{\phi} = x_8 \quad (3.57)$$

$$\dot{x}_8 = \ddot{\phi} = \frac{1}{I_{xx}}U_2 + \frac{I_{yy} - I_{zz}}{I_{xx}}x_4x_{12} - \frac{K_{rx}}{I_{xx}}x_8 \quad (3.58)$$

$$\dot{x}_9 = \dot{y} = x_{10} \quad (3.59)$$

$$\dot{x}_{10} = \ddot{y} = \frac{U_1}{m}(\cos x_7 \sin x_3 \sin x_{11} - \cos x_3 \sin x_7) - \frac{K_{ty}}{m}x_{10} \quad (3.60)$$

$$\dot{x}_{11} = \dot{\theta} = x_{12} \quad (3.61)$$

$$\dot{x}_{12} = \ddot{\theta} = \frac{1}{I_{yy}}U_3 + \frac{I_{zz} - I_{xx}}{I_{yy}}x_8x_4 - \frac{K_{ry}}{I_{yy}}x_{12} \quad (3.62)$$

$$\dot{x} = f(X, U) = \begin{bmatrix} x_2 \\ -g + \frac{U_1}{m}(\cos x_7 \cos x_{11}) - \frac{K_{tz}}{m}x_2 \\ x_4 \\ \frac{1}{I_{zz}}U_4 + \frac{I_{xx} - I_{yy}}{I_{zz}}x_8x_{12} - \frac{K_{rz}}{I_{zz}}x_4 \\ x_6 \\ \frac{U_1}{m}(\sin x_7 \sin x_3 + \cos x_7 \cos x_3 \sin x_{11}) - \frac{K_{tx}}{m}x_6 \\ x_8 \\ \frac{1}{I_{xx}}U_2 + \frac{I_{yy} - I_{zz}}{I_{xx}}x_4x_{12} - \frac{K_{rx}}{I_{xx}}x_8 \\ x_{10} \\ \frac{U_1}{m}(\cos x_7 \sin x_3 \sin x_{11} - \cos x_3 \sin x_7) - \frac{K_{ty}}{m}x_{10} \\ x_{12} \\ \frac{1}{I_{yy}}U_3 + \frac{I_{zz} - I_{xx}}{I_{yy}}x_8x_4 - \frac{K_{ry}}{I_{yy}}x_{12} \end{bmatrix} \quad (3.63)$$

### 3.1.7 Linearized Model

Around the hovering point small angle assumption has been made which results in  $\cos(.) = 1$ ,  $\sin(.) = 0$ .

It is cited that the dynamics of quadcopters are quite nonlinear and strongly coupled, so the linearization is carried out around an equilibrium point to simplify the mathematical model and decouple its dynamics primarily based on the preceding assumptions:

$$\begin{cases} \ddot{\phi} = \frac{U_2}{I_{xx}} - \frac{K_{rx}}{I_{xx}}\dot{\phi} \\ \ddot{\theta} = \frac{U_3}{I_{yy}} - \frac{K_{ry}}{I_{yy}}\dot{\theta} \\ \ddot{\psi} = \frac{U_4}{I_{zz}} - \frac{K_{rz}}{I_{zz}}\dot{\psi} \\ \ddot{x} = g\theta - \frac{K_{tx}}{m}\dot{x} \\ \ddot{y} = g\phi - \frac{K_{ty}}{m}\dot{y} \\ \ddot{z} = -g + \frac{U_1}{m} - \frac{K_{tz}}{m}\dot{z} \end{cases} \quad (3.64)$$

The states are defined in Equation (3.44) and (3.45). Therefore, the linear state-space model of the linear dynamics model can be expressed as:

$$\dot{x} = Ax + Bu \quad (3.65)$$

Where:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-K_{tz}}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-K_{rz}}{I_{zz}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{tx}}{m} & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{rx}}{I_{xx}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & \frac{-K_{ty}}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-K_{ry}}{I_{yy}} \end{bmatrix} \quad (3.66)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \end{bmatrix} \quad (3.67)$$

## 3.2 Controller Design

### 3.2.1 Introduction

The general design objective of model predictive control is to compute a trajectory of a future manipulated variable  $u$  to optimize the future behavior of the plant output  $y$ . The optimization is carried out within a confined time window through giving plant information at the beginning of the time window. To help apprehend the basic ideas that have been used in the design of predictive control, we study a common planning activity of our day-to-day work. The day begins at 9 o'clock in the morning. We are, as a team, going to complete the tasks of design and implementation of a model predictive control system for a liquid vessel. The rule of the game is that we continually plan our activities for the next eight hours work, however, we solely implement the plan for the first hour. This planning activity is repeated for each and every hour until the tasks are completed.

Given the amount of background work that we have carried out for 9 o'clock, we plan ahead for the next eight hours. Assume that the work tasks are divided into modelling, design, simulation and implementation. Completing these tasks will be a function of various factors, such as how much effort we will put in, how properly we will work as a team and whether or not we will get some additional help from others. These are the manipulated variables in the planning problem. Also, we have our limitations, such as our ability to apprehend the design problem, and whether or not we have precise skills of computer hardware and software engineering. These are the hard and soft constraints in the planning. The background information we have already received is paramount for this planning work.

After everything is considered, we decide the design tasks for the next eight hours as functions of the manipulated variables. Then we calculate hour-by-hour what we need to do in order to complete the tasks. In this calculation, primarily based on the background information, we will take our limitations into consideration as the constraints, and find the best way to attain the goal. The end result of this planning gives us our projected activities from 9 o'clock to 5 o'clock. Then we start working through implementing the activities for the first hour of our plan. At 10 o'clock, we check how much we have truly accomplished for the first hour. This information is used for the planning of our next phase of activities. Maybe we have done much less than we planned due to the fact we could not get the correct model or due to the fact one of the key members went for an emergency meeting. Nevertheless, at 10 o'clock, we make an assessment on what we have achieved, and use this updated information for planning our activities for the next eight hours. Our objective might also remain the same or may change. The length of time for the planning remains the same (8 hours). We repeat the same planning process as it was once at 9 o'clock, which then gives us the new projected activities for the next eight hours. We implement the first hour of activities at 10 o'clock. Again at 11 o'clock, we assess what we have achieved once more and use the updated information for the plan of work for the next eight hours. The planning and implementation process is repeated each and every hour until the original objective is achieved.

The planning activity described here involves the principle of Model Predictive Control (MPC). In this example, it is described by the terms that are to be used frequently in the following: the moving horizon window, prediction horizon, receding horizon control, and control objective. They are introduced as

below.

- Moving horizon window: the time-dependent window from an arbitrary time  $t_i$  to  $t_i + T_p$ . The length of the window  $T_p$  remains constant. In this example, the planning activity is performed within an 8-hour window, thus  $T_p = 8$ , with the measurement taken every hour. However,  $t_i$ , which defines the beginning of the optimization window, increases on an hourly basis, starting with  $t_i = 9$ .
- Prediction horizon: dictates how far we wish the future to be predicted for. This parameter equals the length of the moving horizon window,  $T_p$ .
- Receding horizon control: although the optimal trajectory of future control signal is completely described within the moving horizon window, the actual control input to the plant only takes the first sample of the control signal, while neglecting the rest of the trajectory.
- In the planning process, we need the information at time  $t_i$  in order to predict the future. This information is denoted as  $x(t_i)$  which is a vector containing many relevant factors, and is either directly measured (estimated).
- A given model that will describe the dynamics of the system is paramount in predictive control. A good dynamic model will give a consistent and accurate prediction of the future.
- In order to make the best decision, a criterion is needed to reflect the objective. The objective is related to an error function based on the difference between the desired and the actual responses. This objective function is often called the cost function  $J$ , and the optimal control action is found by minimizing this cost function within the optimization window.

in order to make the controller design clear we should investigate the block diagram.

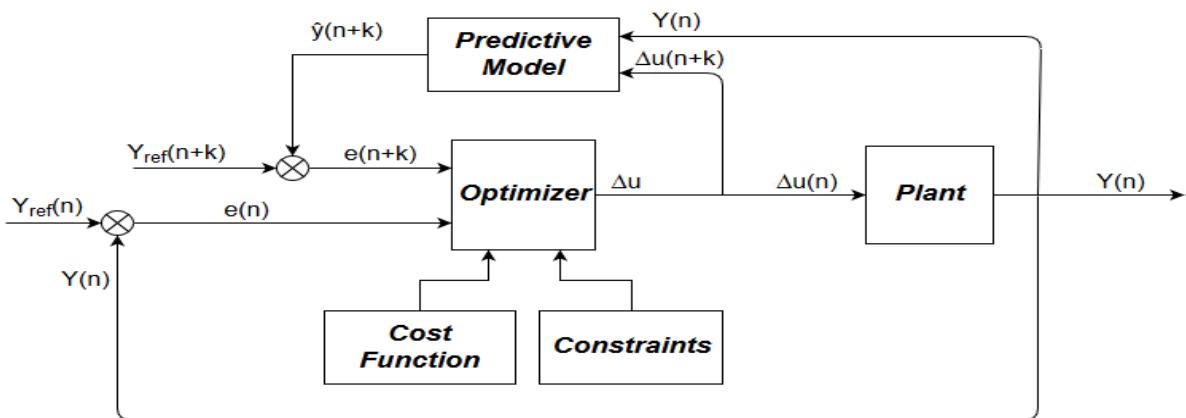


Figure 3.6: Model Predictive Control Block Diagram

### 3.2.2 Predictive Model

Assuming the plant has  $m$  inputs,  $q$  outputs,  $n_1$  states. we also assume that the number of outputs is less than or equal to the number of inputs ( i.e.,  $q \leq m$  ). if the number of outputs is greater than the number of inputs, we cannot hope to control each of the measured outputs independently with zero steady-state errors which is our case in which we have 12 states and 4 control inputs and in other cases we can have 6 states and 4 control inputs.

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (3.68)$$

$$y(k) = C_m x_m(k) \quad (3.69)$$

Note that from equation (3.68) the following difference is also valid:

$$x_m(k) = A_m x_m(k-1) + B_m u(k-1) \quad (3.70)$$

By defining  $\Delta x_m(k) = x_m(k) - x_m(k-1)$  and  $\Delta u(k) = u(k) - u(k-1)$ , then subtracting equation (3.70) from (3.68) which leads to:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (3.71)$$

In order to link the output  $y(k)$  to the state variable  $\Delta x_m(k)$ , we conclude that:

$$\Delta y(k+1) = C_m A_m \Delta x_m(k+1) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) \quad (3.72)$$

where  $\Delta y(k+1) = y(k+1) - y(k)$ . Choosing a new state variable  $x(k) = [\Delta x_m(k)^T \quad y(k)^T]^T$ , we have:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{q \times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (3.73)$$

$$y(k) = [o_m \quad I_{q \times q}] \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

Where in equation (3.73),  $A_m$ ,  $B_m$  and  $C_m$  have dimension  $n_1 \times n_1$ ,  $n_1 \times m$  and  $q \times n_1$ , respectively.  $I_{q \times q}$  is the identity matrix with dimensions  $q \times q$ , which is the number of outputs; and  $o_m$  is a  $q \times n_1$  zero matrix.

for simplicity, equation (3.73) can be expressed as

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3.74}$$

Where A, B and C are matrices that corresponds to the forms given in equation (3.73). In the following, the dimensionality of the augmented state-space terms are  $(n_1 + q) \times (n_1 + q)$  for A matrix,  $(n_1 + q) \times m$  for B matrix and  $q \times (n_1 + q)$  for C matrix

Assuming that at the sampling instant  $k_i$ ,  $k_i > 0$ , the state variable vector  $x(k_i)$  is available through measurement, the state  $x(k_i)$  provides the current plant information. The more general situation where the state is not directly measured in which we need to estimate the states so an observer is required. The future control trajectory is denoted by

$$\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + N_c - 1)$$

where  $N_c$  is referred to as the control horizon dictating the number of parameters used to capture the future control trajectory. With given information  $x(k_i)$ , the future state variables are predicted for  $N_p$  number of samples, where  $N_p$  is referred to as the prediction horizon.  $N_p$  is also the length of the optimization window. We denote the future state variables as

$$x(k_i + 1|k_i), x(k_i + 2|k_i), \dots, x(k_i + m|k_i), \dots, x(k_i + N_p|k_i)$$

where  $x(k_i + m|k_i)$  is the predicted state variable at  $k_i + m$  given the current plant information  $x(k_i)$ . The control horizon  $N_c$  is chosen to be less than or equal to the prediction horizon  $N_p$ .

Based on the state-space model (A, B, C), the future state variables are calculated sequentially using the set of future control parameters:

$$\begin{aligned} x(k_i + 1|k_i) &= Ax(k_i) + B\Delta u(k_i) \\ x(k_i + 2|k_i) &= Ax(k_i + 1|k_i) + B\Delta u(k_i + 1) \\ &= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \\ &\vdots \\ x(k_i + N_p|k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i + 1) + \dots \\ &\quad + A^{N_p-N_c}B\Delta u(k_i + N_c - 1) \end{aligned} \tag{3.75}$$

the predicted output variables can be concluded from the predicted state variables, by substitution

$$\begin{aligned}
 y(k_i + 1|k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
 y(k_i + 2|k_i) &= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\
 y(k_i + 3|k_i) &= CA^3x(k_i) + CA^2B\Delta u(k_i) + CAB\Delta u(k_i + 1) + CB\Delta u(k_i + 2) \\
 &\vdots \\
 y(k_i + N_p|k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + CA^{N_p-2}B\Delta u(k_i + 1) + \dots \\
 &\quad + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1)
 \end{aligned} \tag{3.76}$$

the previous equations show that all the predicted variables are expressed in form of current state  $x(k_i)$  and the future control signal  $\Delta u(k_i + j)$ , where  $j = 0, 1, \dots, N_c - 1$

Defining vectors:

$$\begin{aligned}
 Y &= [y(k_i + 1|k_i)^T \quad y(k_i + 2|k_i)^T \quad y(k_i + 3|k_i)^T \quad \dots \quad y(k_i + N_p|k_i)^T]^T \\
 \Delta U &= [\Delta u(k_i)^T \quad \Delta u(k_i + 1)^T \quad \Delta u(k_i + 2)^T \quad \dots \quad \Delta u(k_i + N_c - 1)^T]^T
 \end{aligned} \tag{3.77}$$

From the previous forms, we can conclude a compact matrix form expressed as:

$$Y = Fx(k_i) + \Phi\Delta U \tag{3.78}$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \tag{3.79}$$

### 3.2.3 Optimizer and Cost Function

Upon formulation of the mathematical model, the subsequent step in the design of a predictive control system is to calculate the predicted plant output with the future control signal as the adjustable variables. This prediction is described within an optimization window. This section will study in detail the optimization carried out within this window. Here, we anticipate that the current time is  $k_i$  and the length of

the optimization window is  $N_p$  as the number of samples.

For a given set-point signal  $r(k_i)$  at sample time  $k_i$ , within a prediction horizon the objective of the predictive control system is to bring the predicted output as close as feasible to the set-point signal, where we assume that the set-point signal remains constant in the optimization window. This objective is then translated into a design to find the best control parameter vector  $\Delta U$  such that an error function between the set-point and the predicted output is minimized.

Assuming that the vector that contains the set-point information is

$$R_s^T = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^{q N_p} r(k_i) \quad (3.80)$$

The cost function  $J$  that reflects the control objective is defined as

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (3.81)$$

where the first term represents the objective of minimizing the errors between the predicted output and the set-point signal while the second term represents the consideration given to the size of  $\Delta U$  when the objective function  $J$  is made to be as small as possible.  $\bar{R}$  is a diagonal matrix expressed as  $\bar{R} = r_w I_{N_c \times N_c}$  ( $r_w \leq 0$ ) where  $r_w$  is used as a tuning parameter for the desired closed-loop performance. in case  $r_w = 0$ , the cost function 3.81 is expressed as the situation where we would not want to pay attention to how large the  $\Delta U$  might be and our goal would be solely to make the error  $(R_s - Y)^T (R_s - Y)$  as small as possible. on the other hand if  $r_w$  has a large value, the cost function is expressed as the situation where we would carefully consider how large the  $\Delta U$  might be and cautiously reduce the error  $(R_s - Y)^T (R_s - Y)$ .

To find the optimal  $\Delta U$  that will minimize  $J$ , by using equation (3.78),  $J$  is expressed as

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (3.82)$$

Proceeding with getting the first derivative of the cost function  $J$  :

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k_i)) + 2(\Phi^T \Phi + \bar{R}) \Delta U \quad (3.83)$$

$$\frac{\partial J}{\partial \Delta U} = 0$$

from which we find the optimal solution for the control signal as

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)) \quad (3.84)$$

considering the case in which  $(\Phi^T \Phi + R)^{-1}$  exists. the matrix  $(\Phi^T \Phi + R)^{-1}$  is called the Hessian matrix in the optimization literature. Note  $R_s$  is a data vector that contains the set-point information expressed as

$$R_s = \begin{bmatrix} & & & q N_p \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}^T r(k_i) = \bar{R}_s r(k_i) \quad (3.85)$$

where

$$\bar{R}_s = \begin{bmatrix} & & & q N_p \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}^T \quad (3.86)$$

The optimal solution of the control signal is connected to the set-point signal  $r(k_i)$  and the state variable  $x(k_i)$  through the following equation

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)) \quad (3.87)$$

Expanding the previous equation to get

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T Fx(k_i)) \quad (3.88)$$

Where matrix  $\Phi^T \Phi$  has dimension  $mN_c \times mN_c$ , matrix  $\Phi^T F$  has dimension  $mN_c \times n$  and matrix  $\Phi^T \bar{R}_s$  has the same value as the last  $q$  columns of  $\Phi^T F$ . the weight matrix  $\bar{R}$  is a block matrix with  $m$  blocks and has its dimension equal to the dimension of  $\Phi^T \Phi$ . the set point signal is  $r(k_i) = [r_1(k_i) \ r_2(k_i) \ \cdots \ r_q(k_i)]^T$  as the  $q$  set point signals to the Multi-output system.

### 3.2.4 Receding Horizon Control

With the receding horizon control principle, we only implement the first sample of this sequence, i.e  $\Delta u(k_i)$ , while ignoring the rest of the sequence, although the optimal parameter vector  $\Delta U$  contains the controls  $\Delta u(k_i), \Delta u(k_i + 1), \Delta u(k_i + 2), \dots, \Delta u(k_i + N_c - 1)$ . When the next sample period arrives, the more recent measurement is taken to form the state vector  $x(k_i + 1)$  for calculation of the new sequence of control signal. This procedure is repeated in real time to give the receding horizon control law.

Recalling equation (3.88) and applying the receding horizon control principle, only the first  $m$  ele-

ments in  $\Delta U$  are taken to form the incremental optimal control

$$\begin{aligned}\Delta u(k_i) &= \begin{bmatrix} I_m & o_m & \cdots & o_m \end{bmatrix} (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i)) \\ &= K_y r(k_i) - K_{mpc} x(k_i)\end{aligned}\quad (3.89)$$

Where the matrix  $\begin{bmatrix} I_m & o_m & \cdots & o_m \end{bmatrix}$  has dimension  $N_c$ . Also the matrix  $I_m$  and  $o_m$  are, respectively, the identity and zero matrix with dimension  $m \times m$ .

# Chapter 4

## Results

Simulation on the quadcopter system is performed with different and various scenarios in order to verify the effectiveness of the proposed approach. The effectiveness of the proposed approach is tested using MATLAB/Simulink especially MATLAB MPC Designer Toolbox with sampling time of 0.07 seconds.

### 4.1 Open Loop Response without the aerodynamic drag

An open loop response experiments are conducted on the dynamic model of the quadcopter to test the different types of maneuvers that can be done by the quadcopter. The following maneuvers are tested:

The parameters of the quadcopter are :

Table 4.1: Parameters Of The Quadcopter

Mass of the quadcopter	m	1.4 kg
Gravitational acceleration	g	9.81 m/s <sup>2</sup>
Moment of Inertia in X	$I_{xx}$	0.08 kg.m <sup>2</sup>
Moment of Inertia in Y	$I_{yy}$	0.08 kg.m <sup>2</sup>
Moment of Inertia in Z	$I_{zz}$	0.16 kg.m <sup>2</sup>
Length of the arm	$l$	$223.5 \times 10^{-3} m$
Coefficient of Drag Forces	$K_t$	[0.1 0.1 0.1]
Coefficient of Drag Moments	$K_r$	[0.1 0.1 0.1]
Aerodynamic Force Constant	$b$	$3.13 \times 10^{-5}$
Aerodynamic Force Constant	$d$	$7.5 \times 10^{-7}$

- Thrust : where  $U_1$  is constant and not equal to zero while  $U_2$ ,  $U_3$  and  $U_4$  are zeros. (  $U_1 = 15N$  and  $U_2 = U_3 = U_4 = 0N$  )

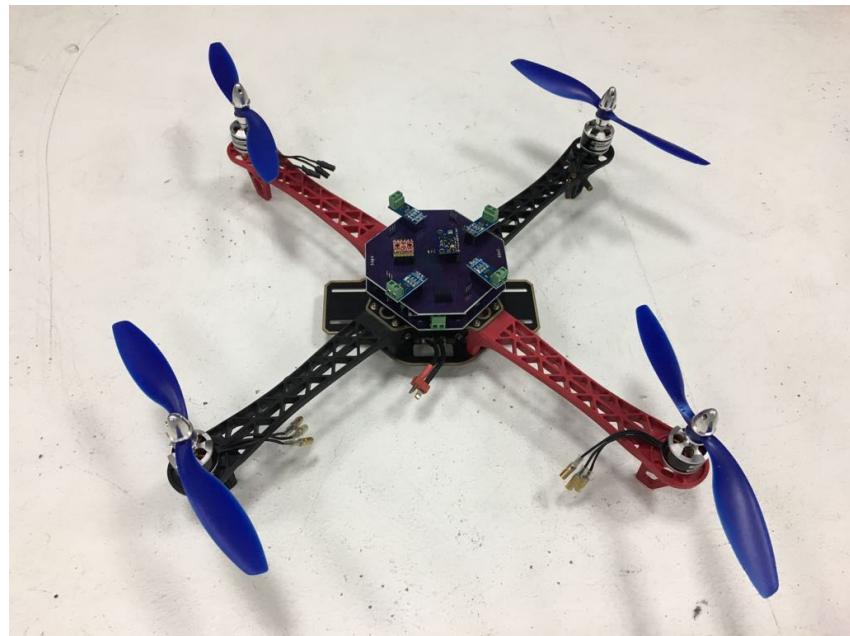
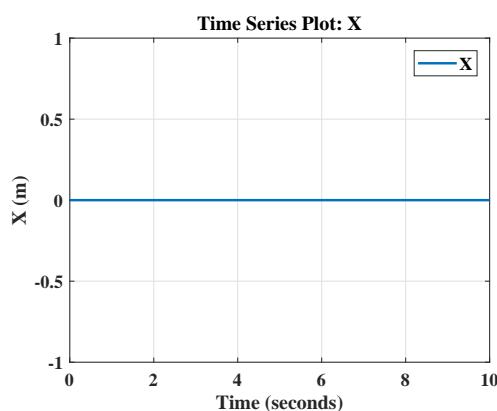
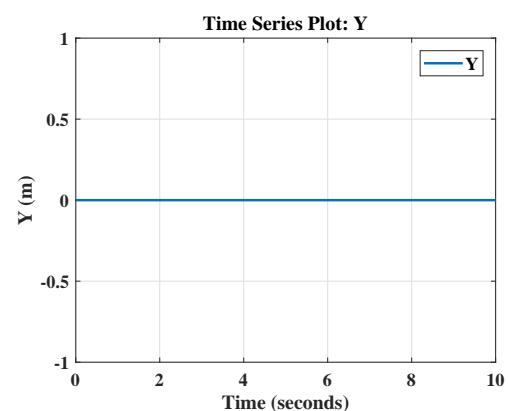


Figure 4.1: Constructed Quadcopter



(a) X plot with time



(b) Y plot with time

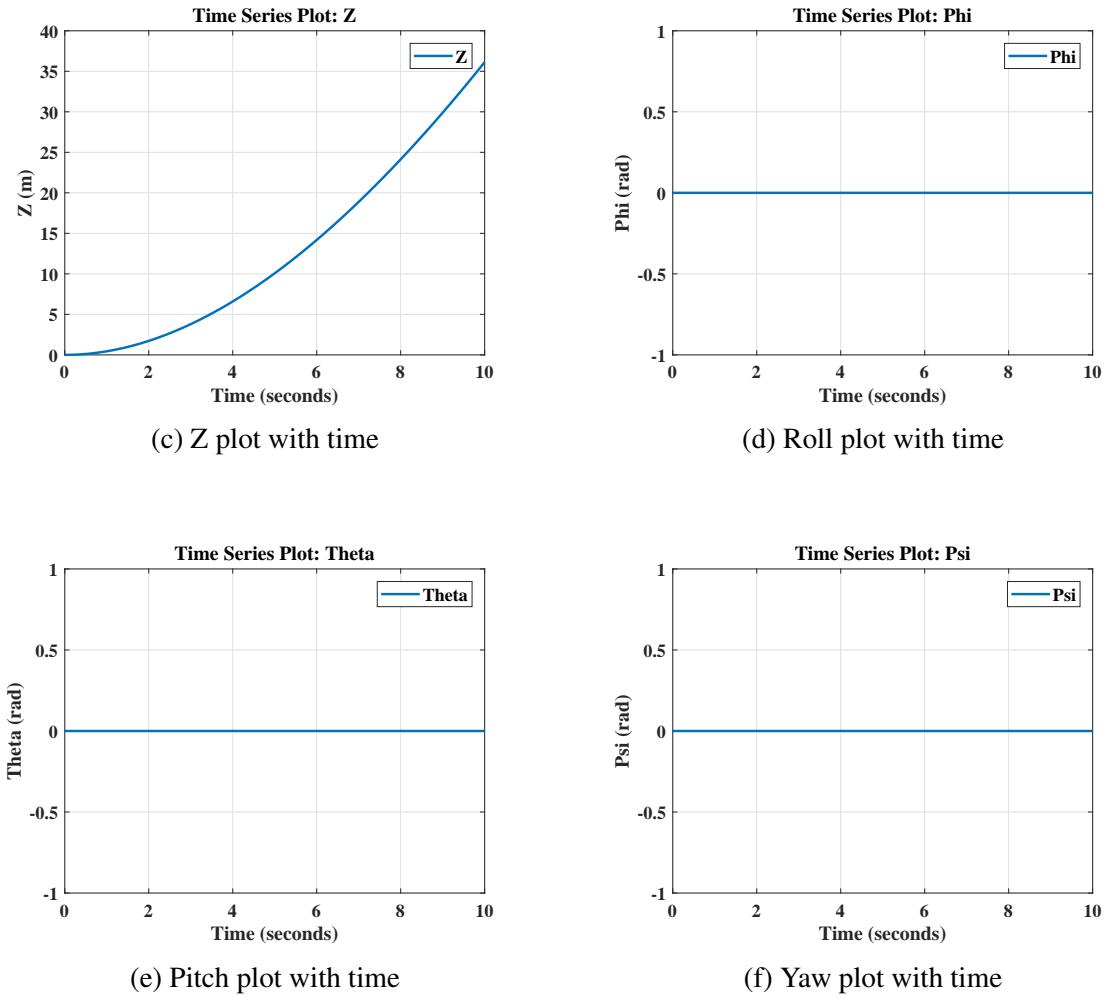
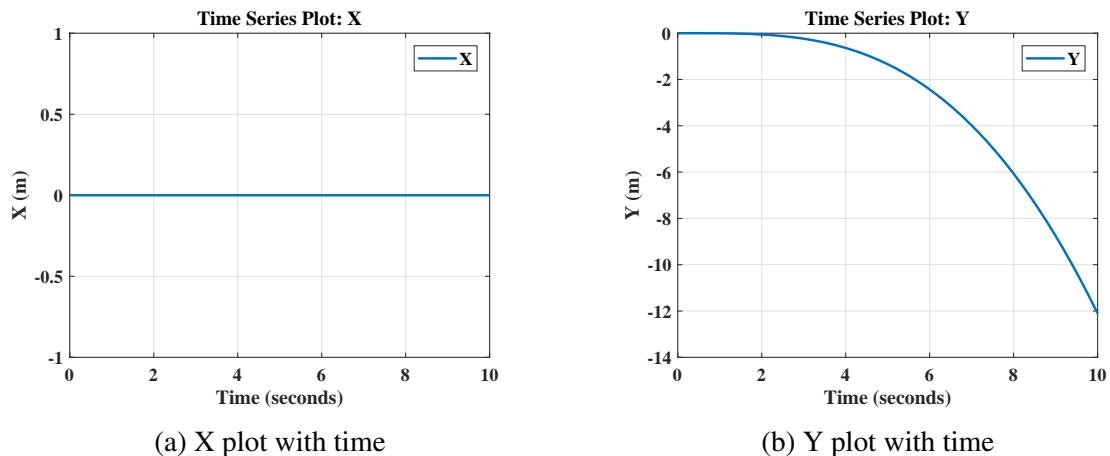


Figure 4.2: Open Loop Response for Thrust

- Thrust and Roll : where  $U_1$  and  $U_2$  are constants and not equal to zero while  $U_3$  and  $U_4$  are zeros. ( $U_1 = 15N$ ,  $U_2 = 0.001N$  and  $U_3 = U_4 = 0N$ )



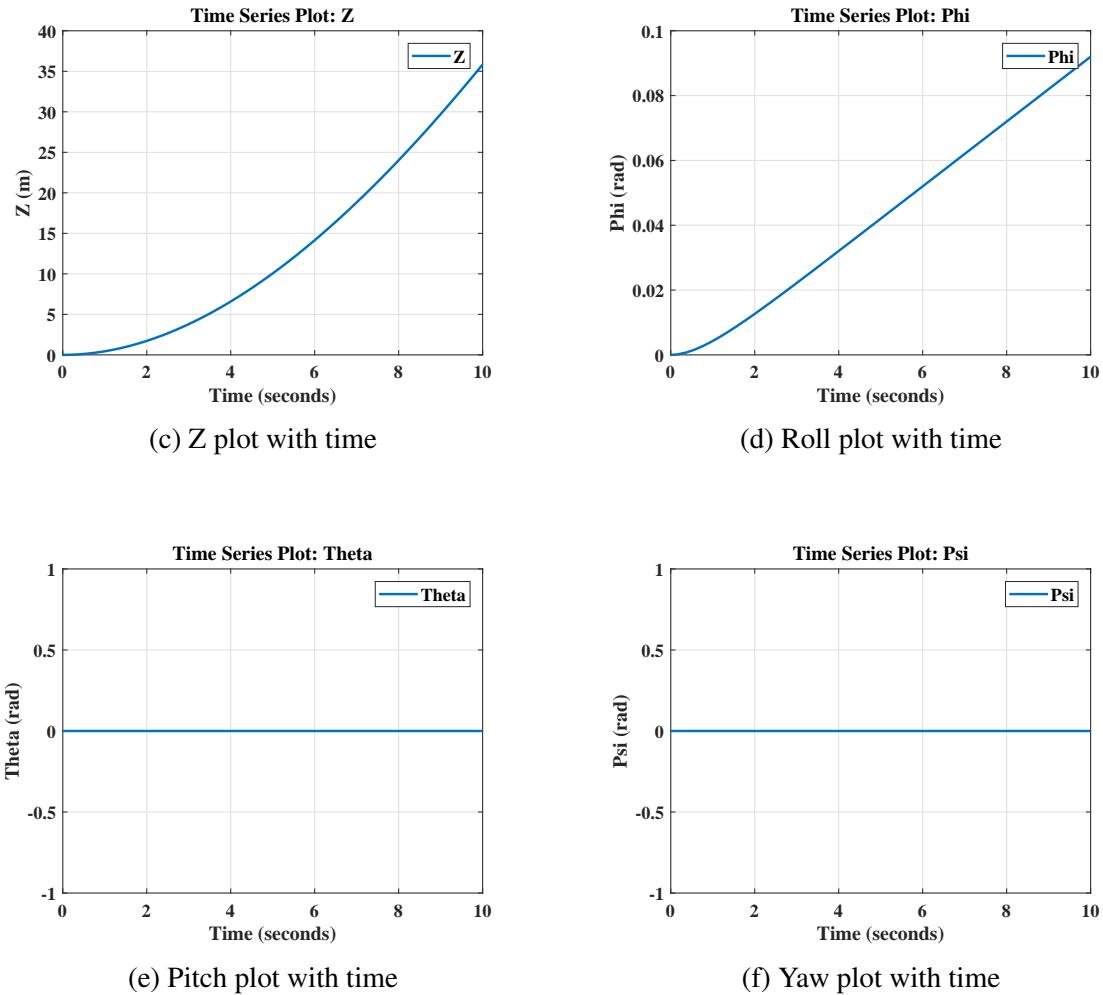
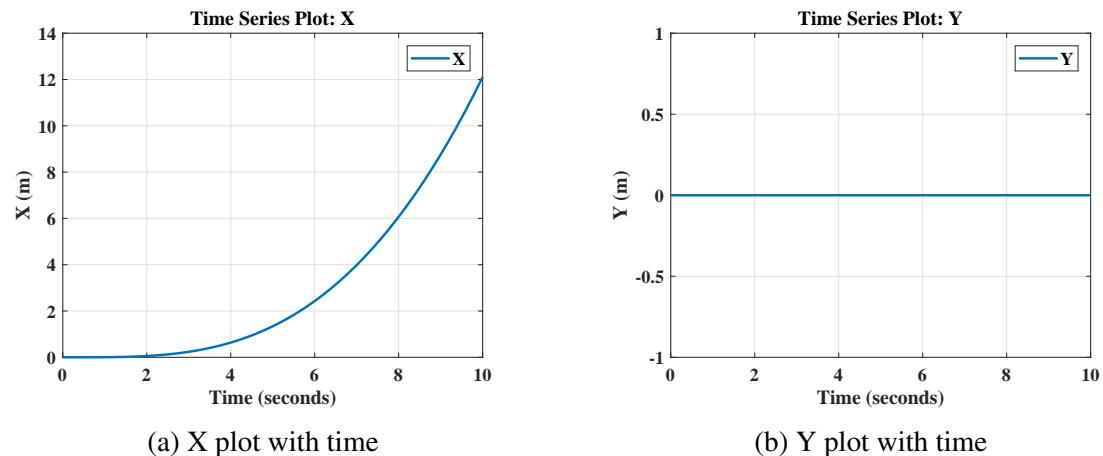


Figure 4.3: Open Loop Response for Thrust and Roll

- Thrust and Pitch : where  $U_1$  and  $U_3$  are constants and not equal to zero while  $U_2$  and  $U_4$  are zeros. ( $U_1 = 15N$ ,  $U_3 = 0.001N$  and  $U_2 = U_4 = 0N$ )



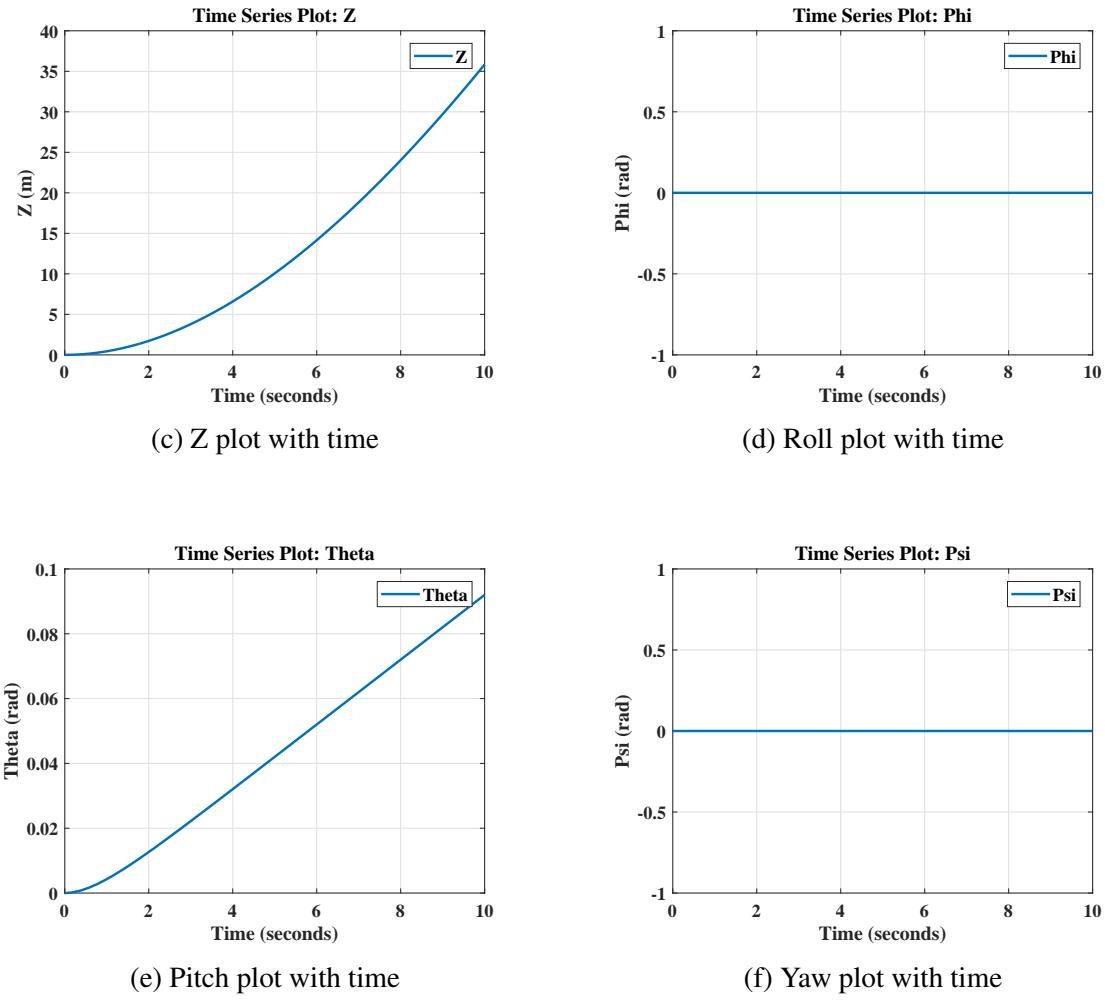
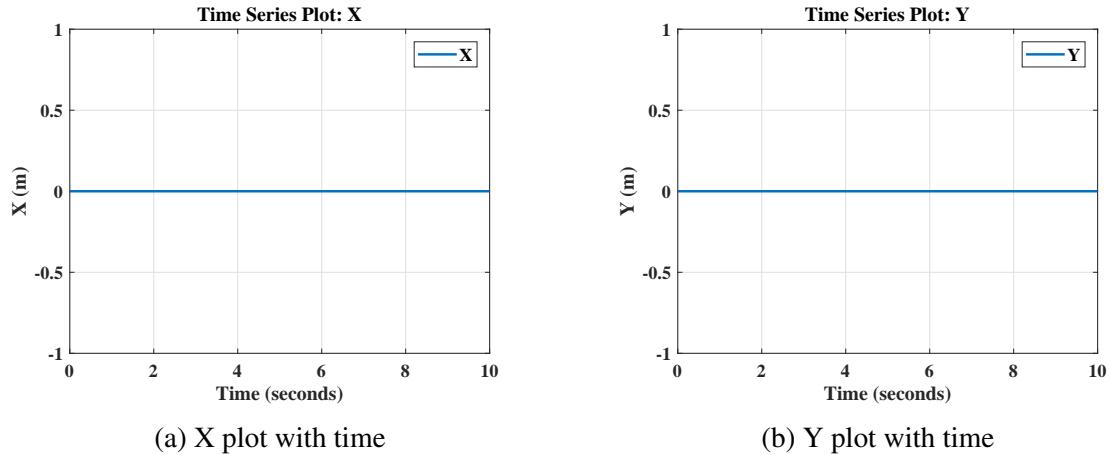


Figure 4.4: Open Loop Response for Thrust and Pitch

- Thrust and Yaw : where  $U_1$  and  $U_4$  are constants and not equal to zero while  $U_2$  and  $U_3$  are zeros. ( $U_1 = 15N$ ,  $U_4 = 0.001N$  and  $U_2 = U_3 = 0N$ )



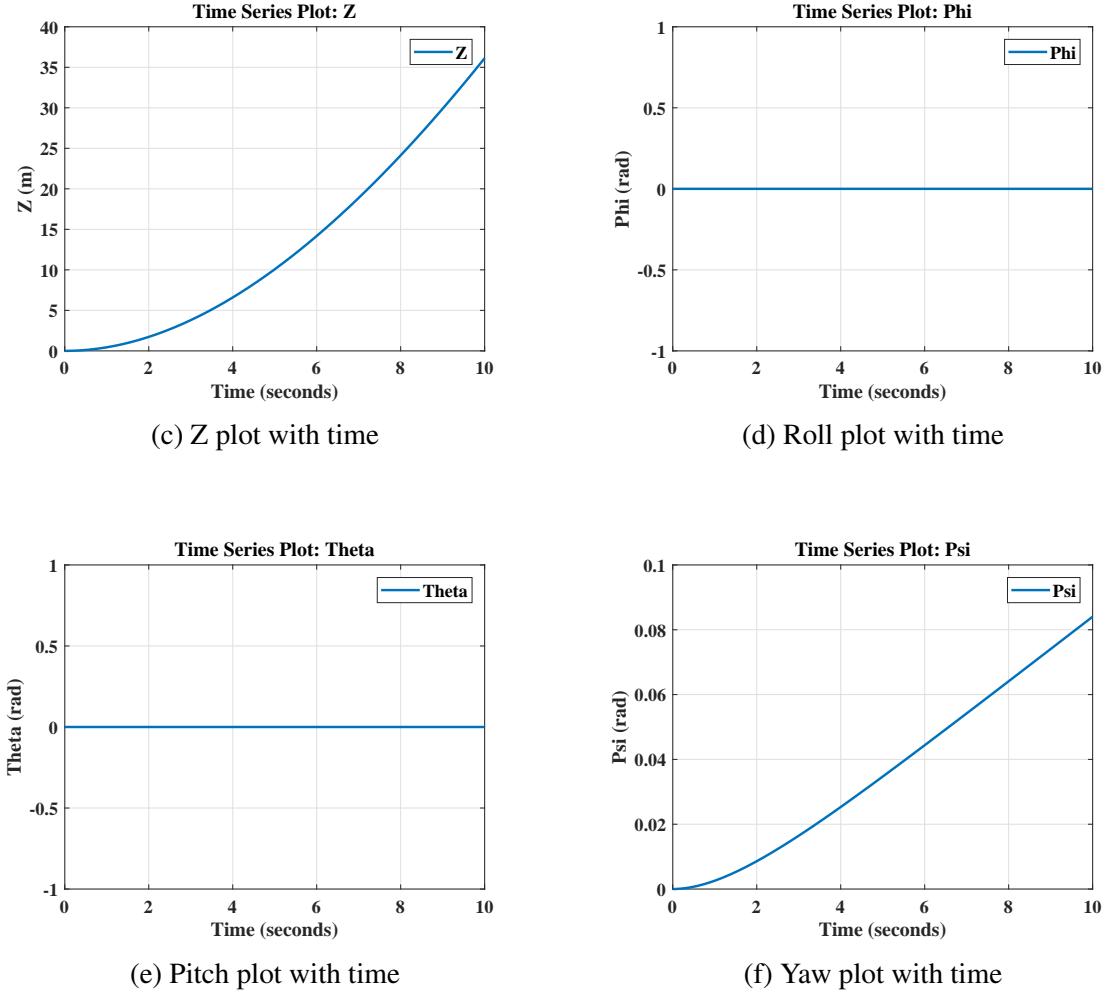


Figure 4.5: Open Loop Response for Thrust and Yaw

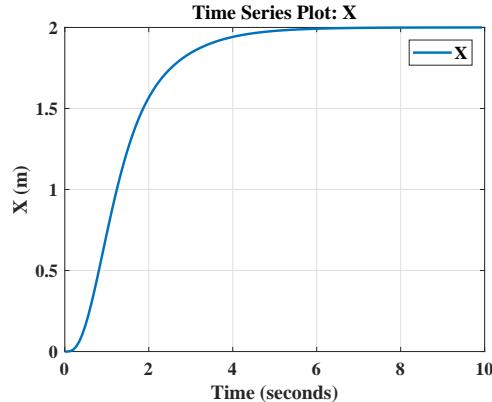
## 4.2 Model Predictive Control for point tracking

The results shown in this section is validated using MATLAB/Simulink MPC designer toolbox.

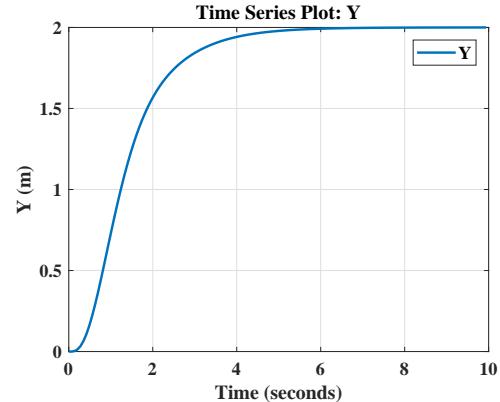
- The sampling time used in the experiments is 0.07 seconds.
- the weights of the outputs are ones.
- the prediction horizon is 100.
- the control horizon is 20.

### 4.2.1 Point Tracking With Unconstrained Control Inputs

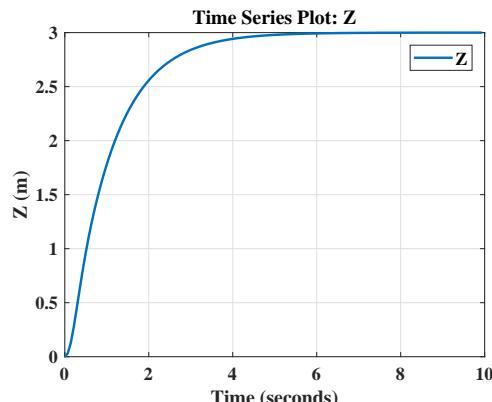
the results for the MPC test to reach desired point target  $r = \begin{bmatrix} 2 & 2 & 3 \end{bmatrix}^T$  and Yaw ( $\psi$ ) = 0.785 rad (45 degrees)



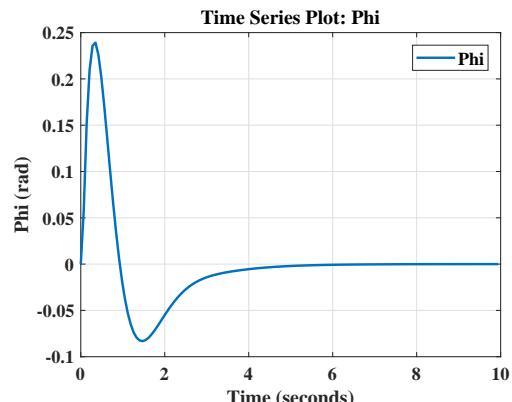
(a) X plot with time



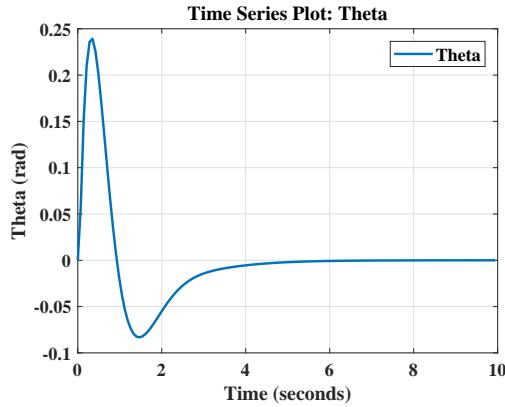
(b) Y plot with time



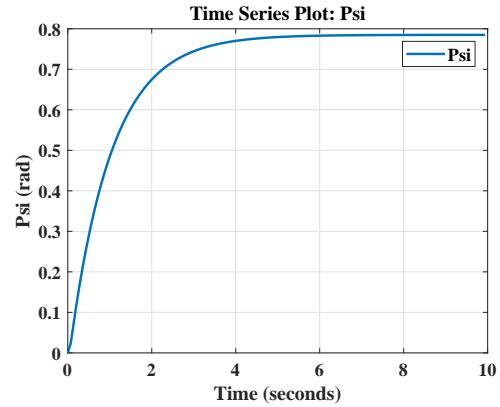
(c) Z plot with time



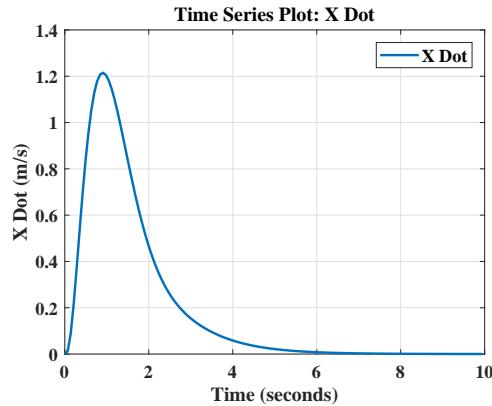
(d) Roll plot with time



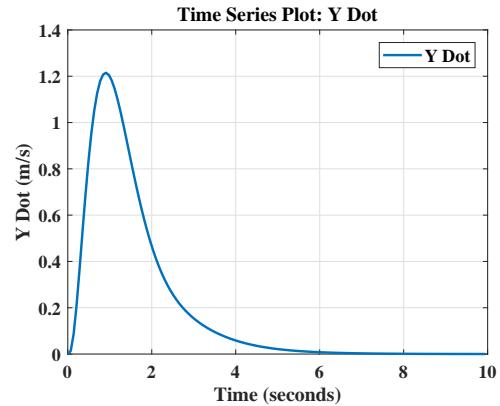
(e) Pitch plot with time



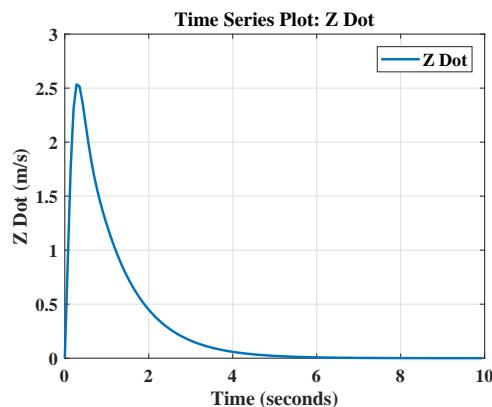
(f) Yaw plot with time



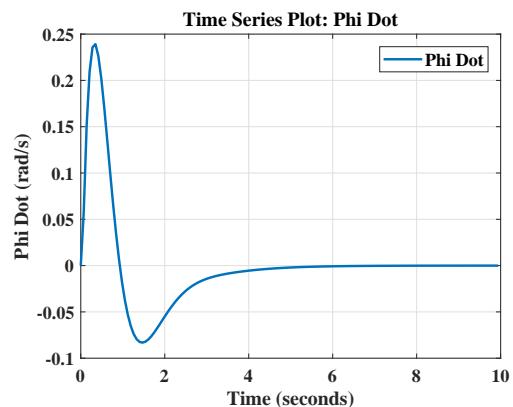
(g) X Dot plot with time



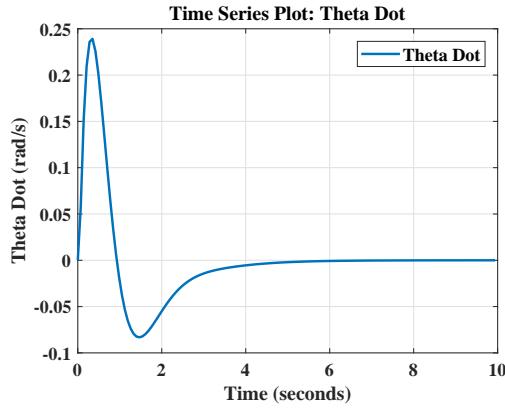
(h) Y Dot plot with time



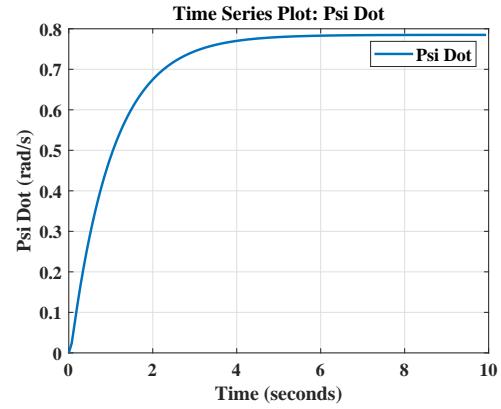
(i) Z Dot plot with time



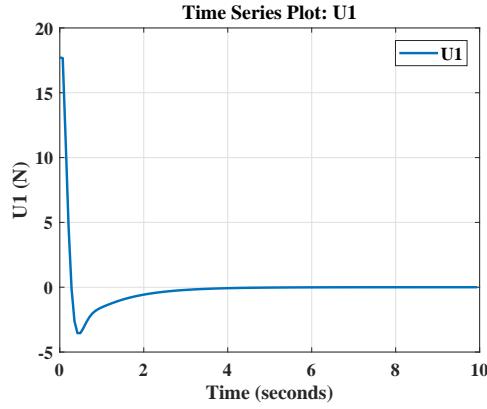
(j) Roll Dot plot with time



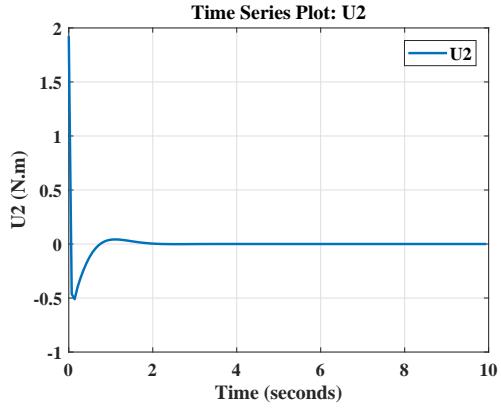
(k) Pitch Dot plot with time



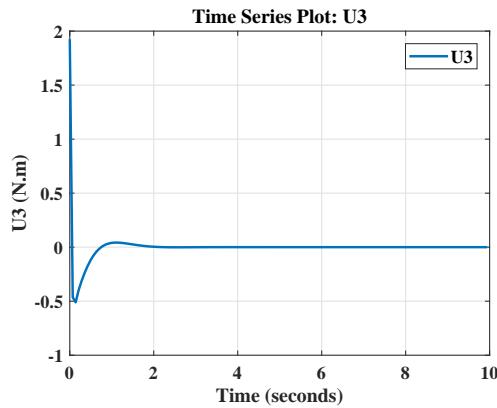
(l) Yaw plot with time



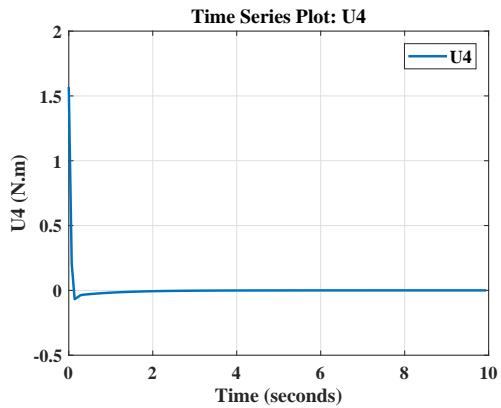
(m) U1 plot with time



(n) U2 plot with time



(o) U3 plot with time



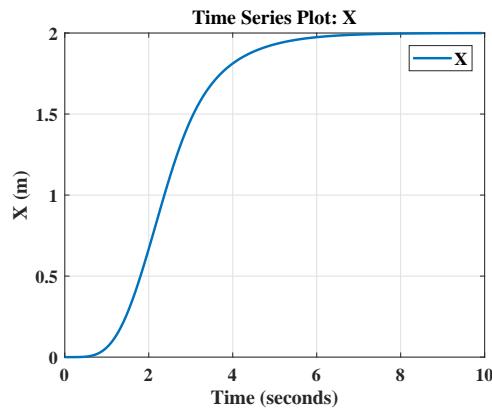
(p) U4 with time

Figure 4.6: Closed Loop Response for Point Tracking With Unconstrained Control Inputs

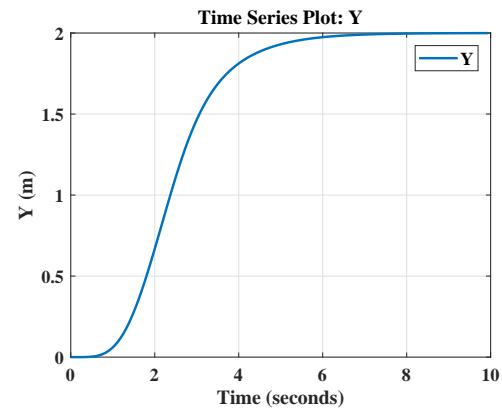
### 4.2.2 Point Tracking With Constrained Control Inputs

the results for the MPC test to reach desired point target  $r = [2 \ 2 \ 3]^T$  and Yaw ( $\psi$ ) = 0.785 rad (45 degrees) while having constraints on the rate of change of control inputs where

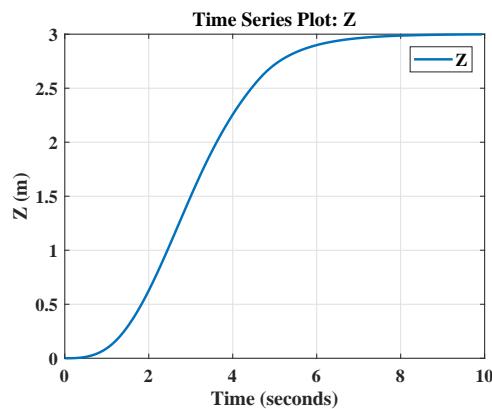
- $-0.05 \leq \dot{U}1 \leq 0.05$
- $-0.005 \leq \dot{U}2 \leq 0.005$
- $-0.005 \leq \dot{U}3 \leq 0.005$
- $-0.005 \leq \dot{U}4 \leq 0.005$



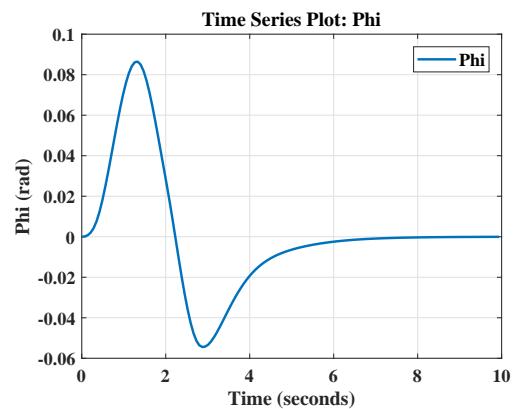
(a) X plot with time



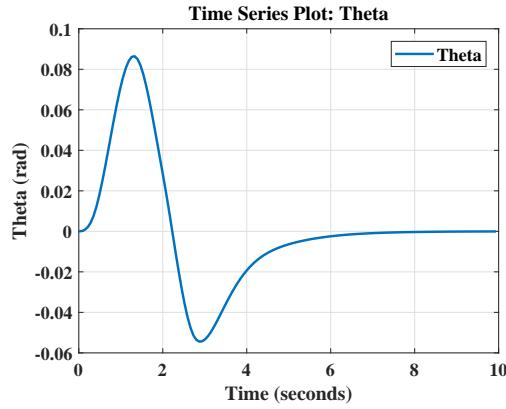
(b) Y plot with time



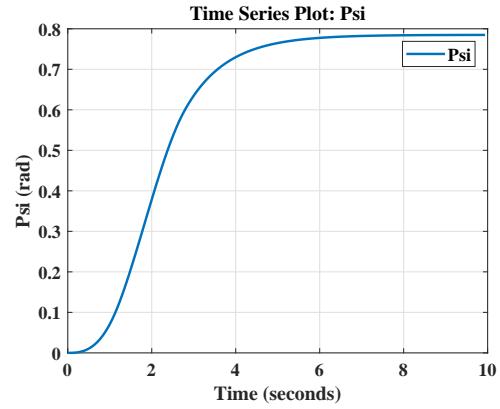
(c) Z plot with time



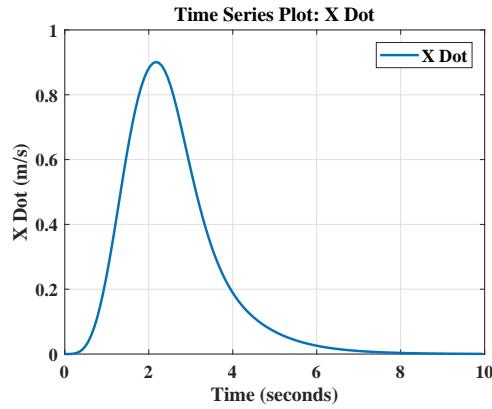
(d) Roll plot with time



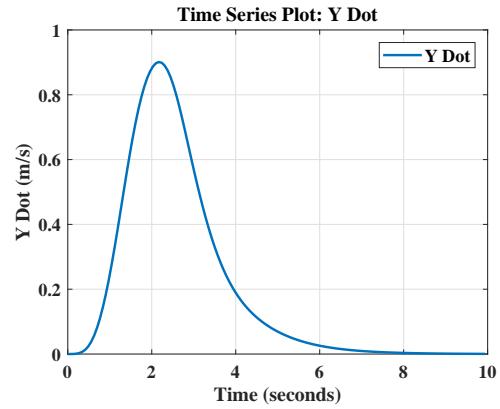
(e) Pitch plot with time



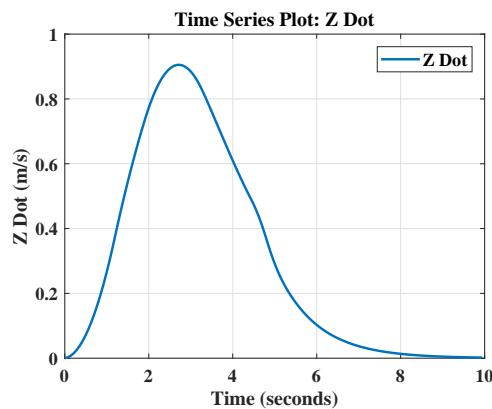
(f) Yaw plot with time



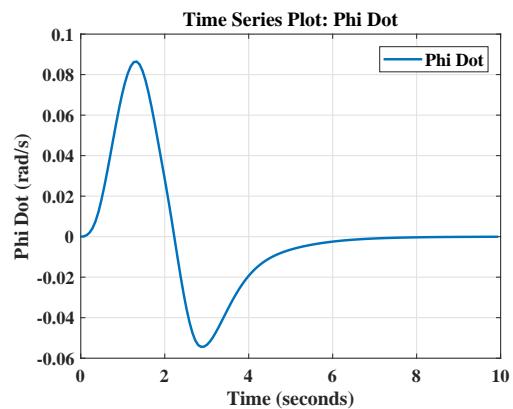
(g) X Dot plot with time



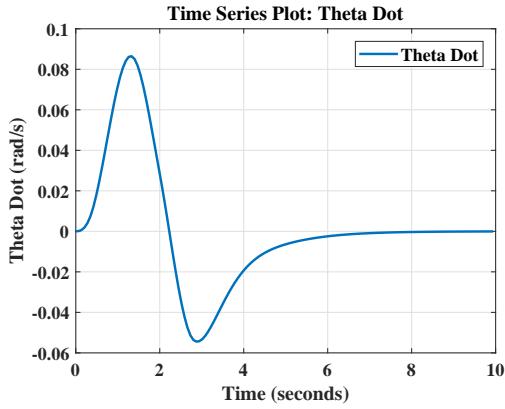
(h) Y Dot plot with time



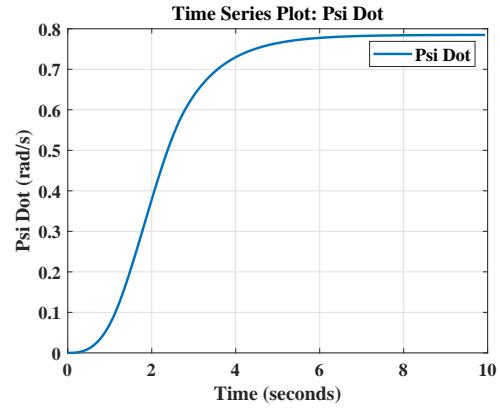
(i) Z Dot plot with time



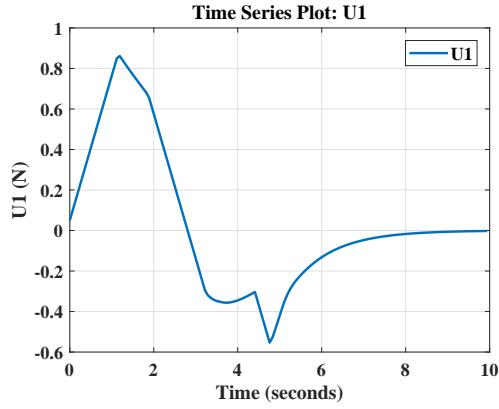
(j) Roll Dot plot with time



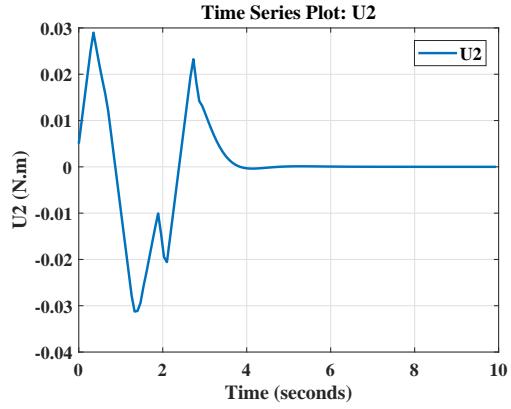
(k) Pitch Dot plot with time



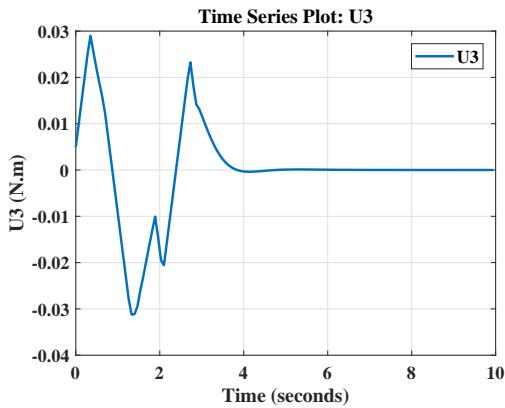
(l) Yaw plot with time



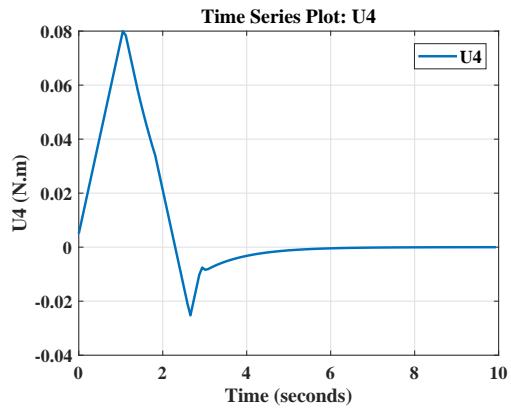
(m) U1 plot with time



(n) U2 plot with time



(o) U3 plot with time



(p) U4 with time

Figure 4.7: Closed Loop Response for Point Tracking With Constrained Control Inputs

## 4.3 Model Predictive Control For Trajectory Tracking

the results presented in this section shows the effectiveness of the model predictive control for tracking a helix, complex helix, square and star trajectories.

### 4.3.1 Model Predictive Control for helix tracking

the results for the MPC test to reach desired point target  $r = [\sin(0.1t) \quad \cos(0.1t) \quad 0.012t]^T$  while having constraints on the rate of change of control inputs presented in (4.2.2)

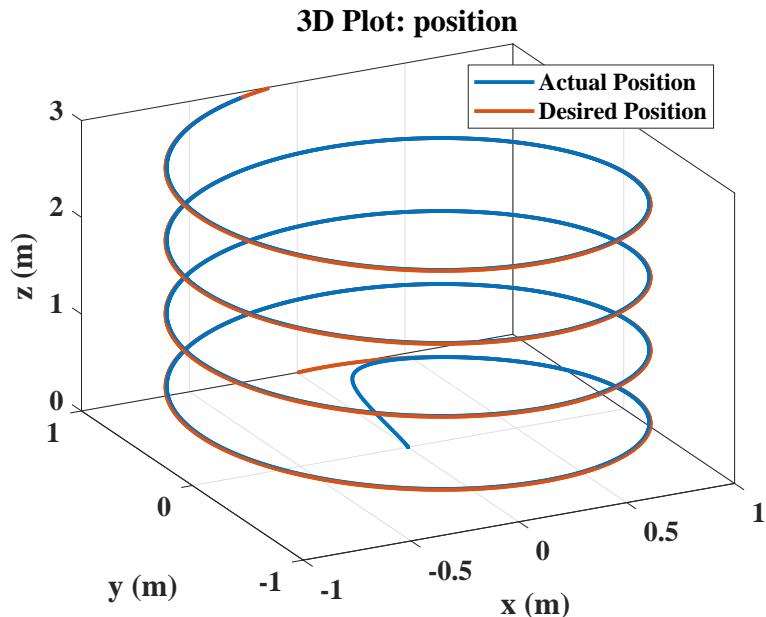
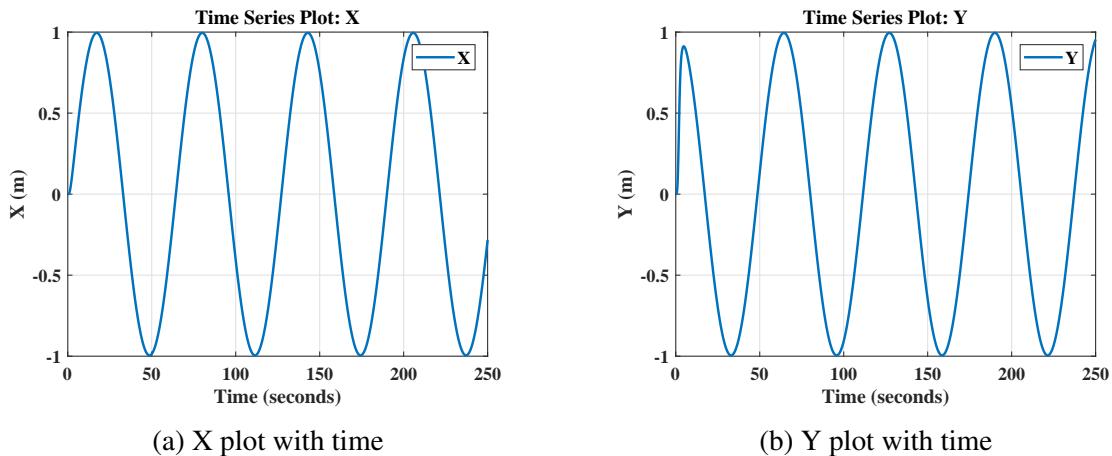
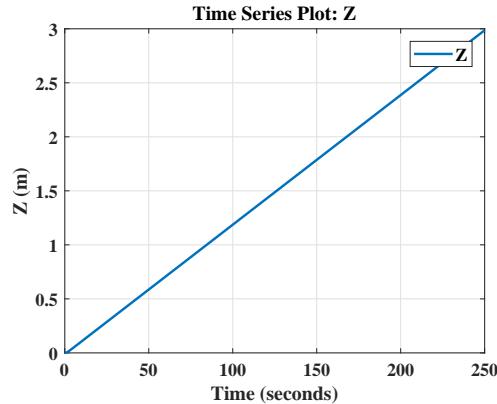
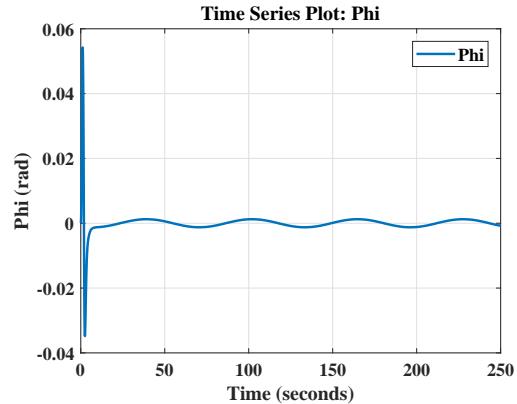


Figure 4.8: Quadcopter Tracking a Helix Trajectory

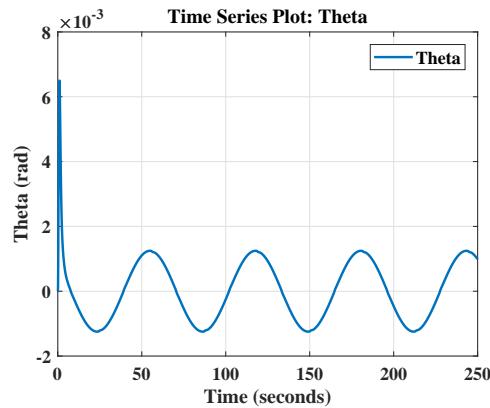




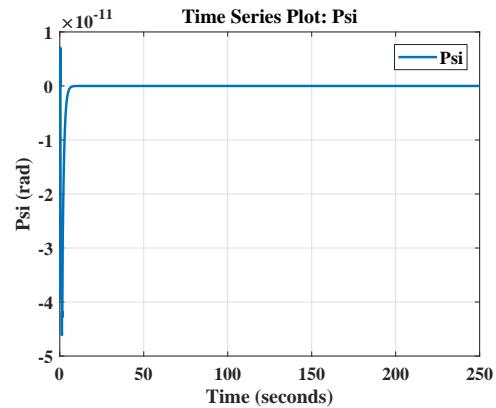
(c) Z plot with time



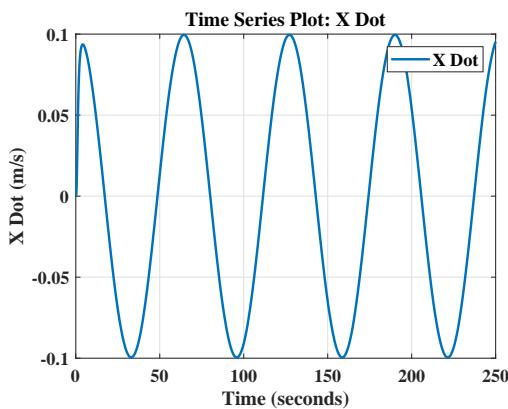
(d) Roll plot with time



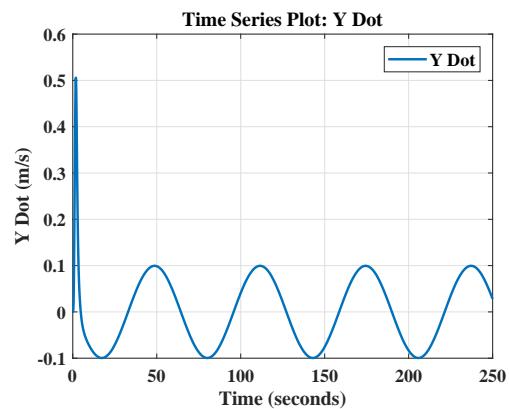
(e) Pitch plot with time



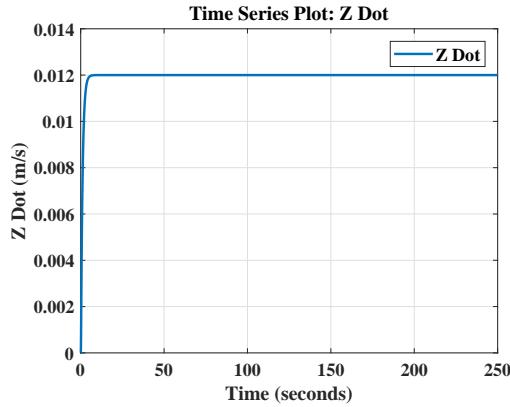
(f) Yaw plot with time



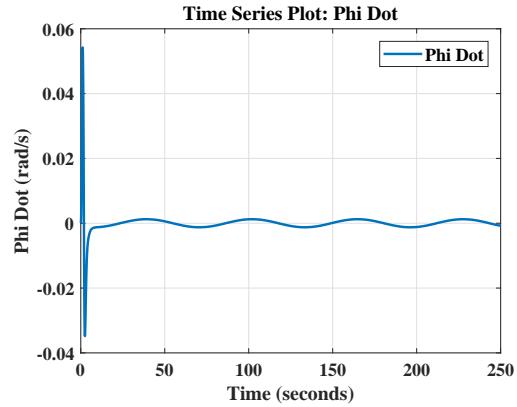
(g) X Dot plot with time



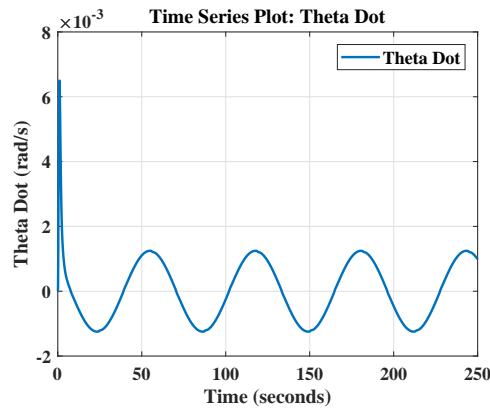
(h) Y Dot plot with time



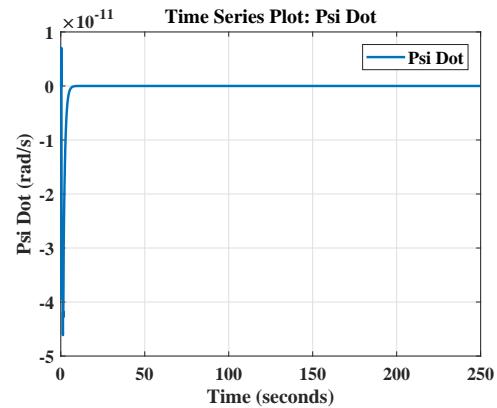
(i) Z Dot plot with time



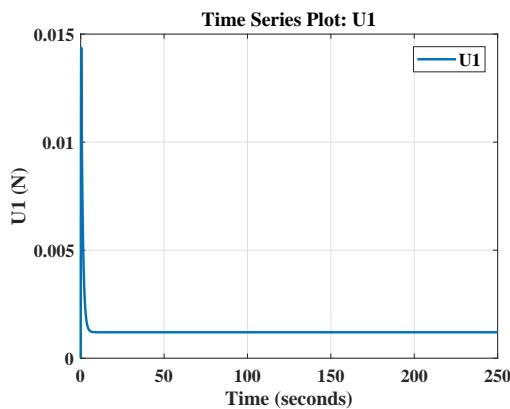
(j) Roll Dot plot with time



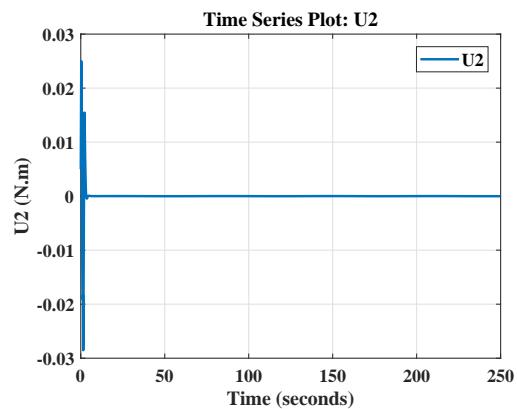
(k) Pitch Dot plot with time



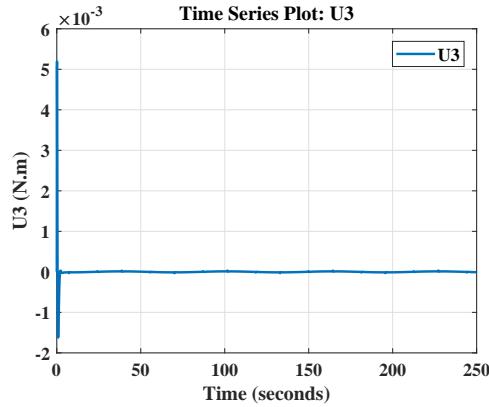
(l) Yaw plot with time



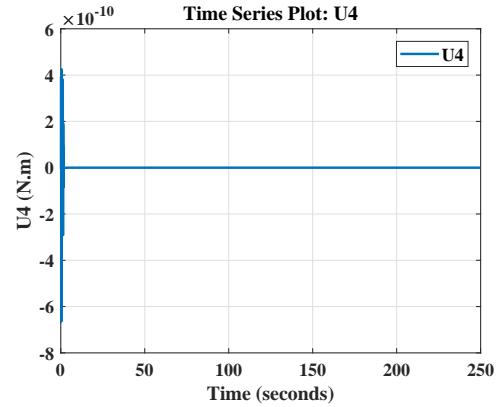
(m) U1 plot with time



(n) U2 plot with time



(o) U3 plot with time



(p) U4 with time

Figure 4.9: Closed Loop Response for Helix Trajectory Tracking

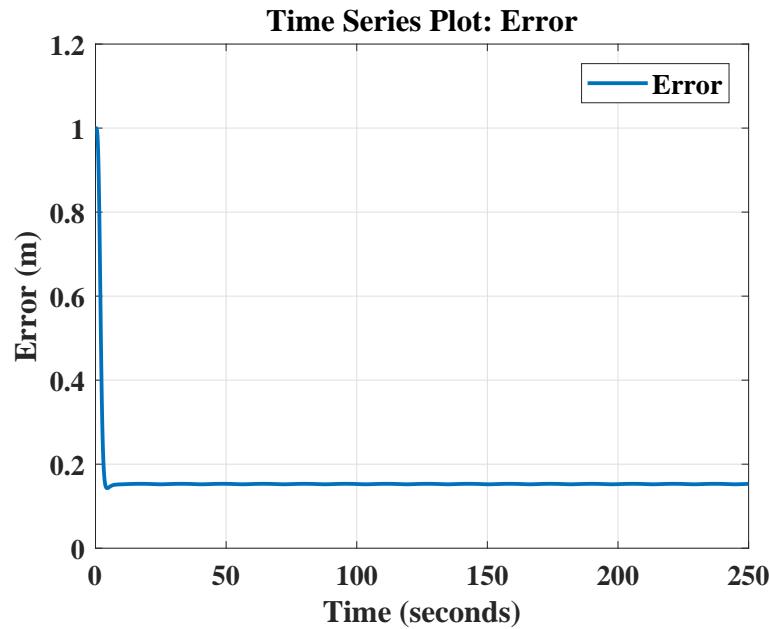


Figure 4.10: Error Between The Desired And Actual Trajectories

### 4.3.2 Model Predictive Control For Complex Helix Tracking

the results for the MPC test to reach desired point target  $r = \begin{bmatrix} \cos(0.05t) - \cos(0.05 * t)^3 & \sin(0.05t) - \sin(0.05t)^3 & 0. \end{bmatrix}$  while having constraints on the rate of change of control inputs presented in (4.2.2)

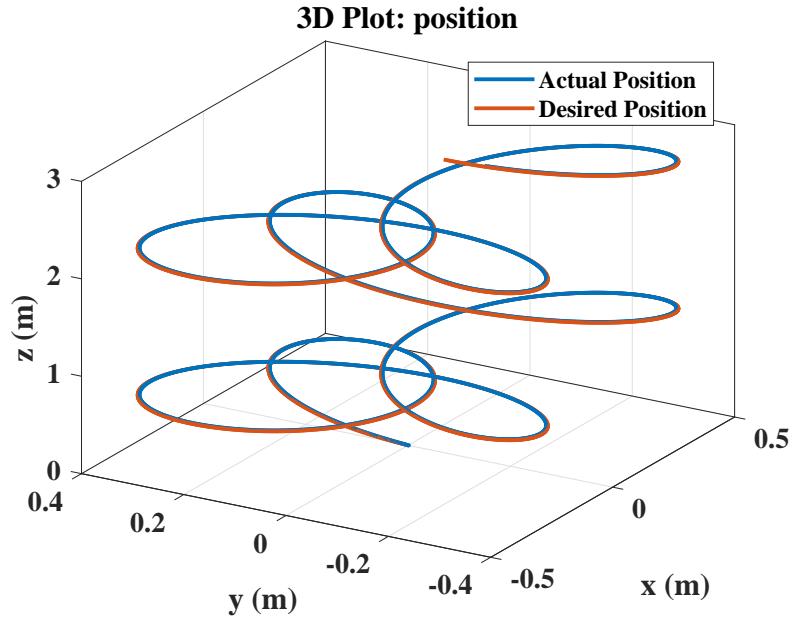
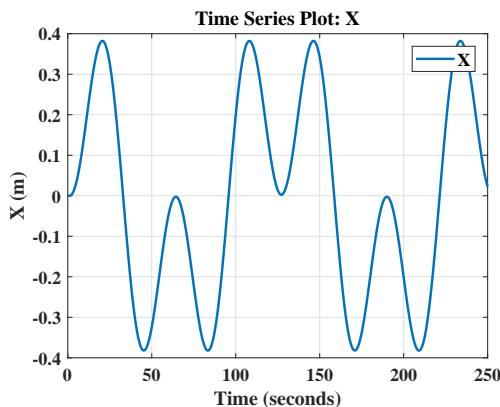
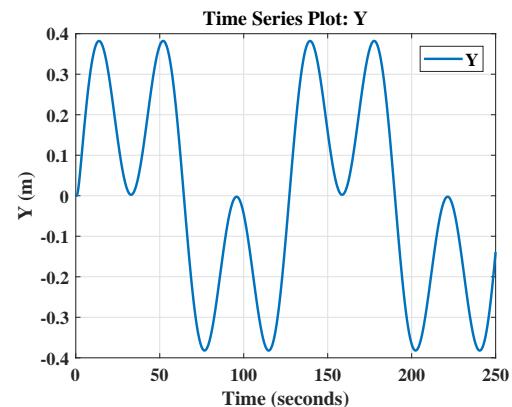


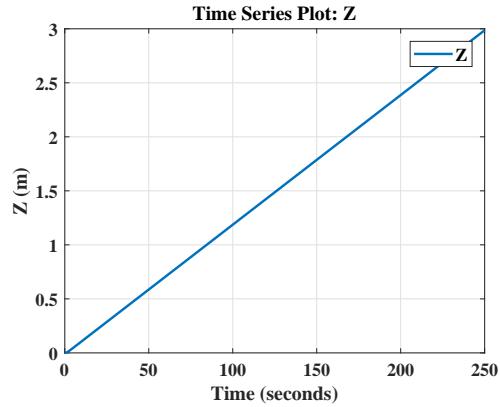
Figure 4.11: Quadcopter Tracking a Complex Helix Trajectory



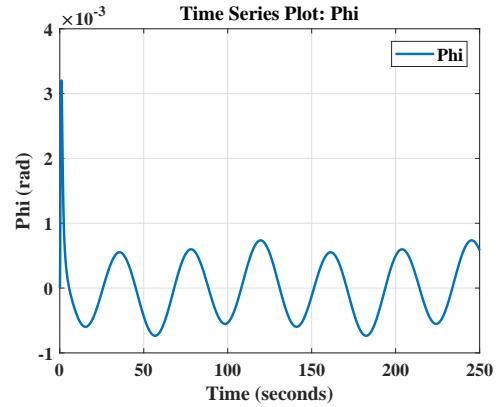
(a) X plot with time



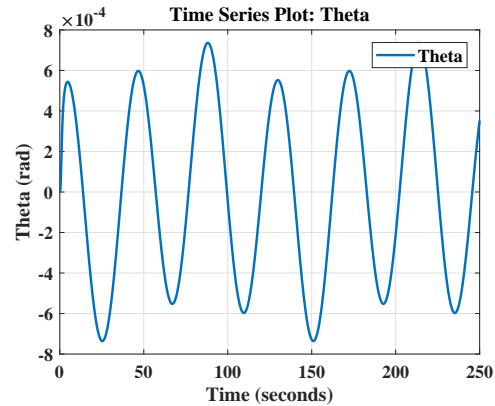
(b) Y plot with time



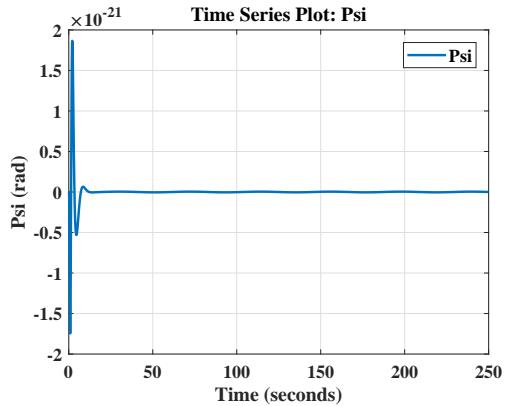
(c) Z plot with time



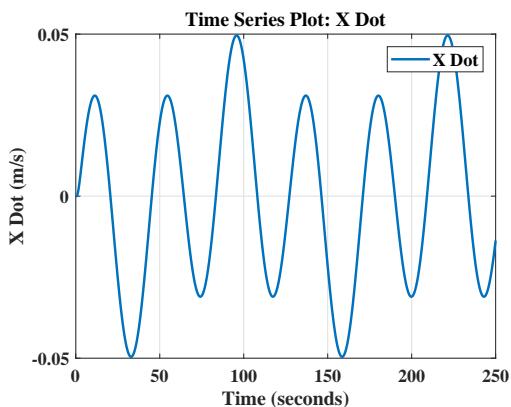
(d) Roll plot with time



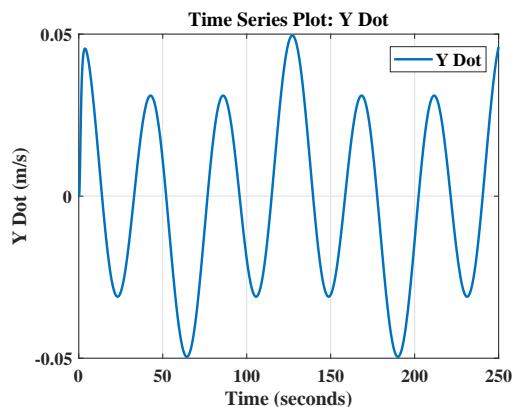
(e) Pitch plot with time



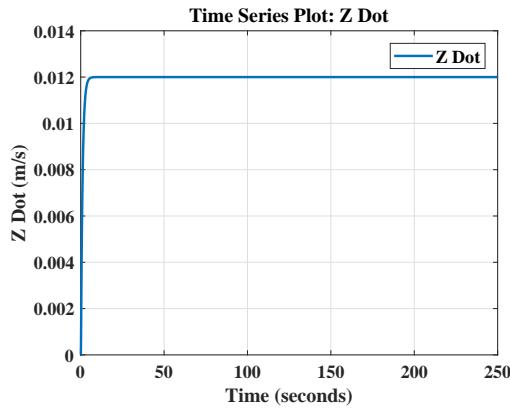
(f) Yaw plot with time



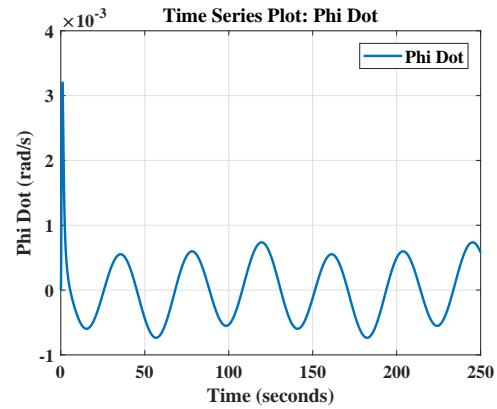
(g) X Dot plot with time



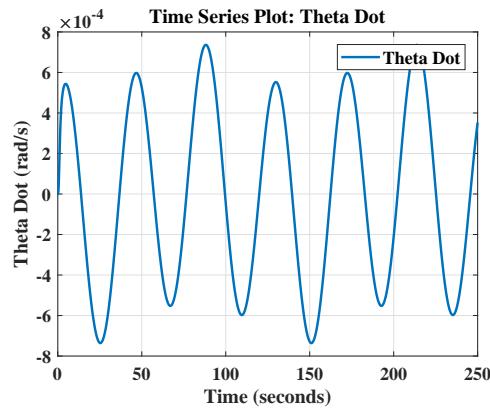
(h) Y Dot plot with time



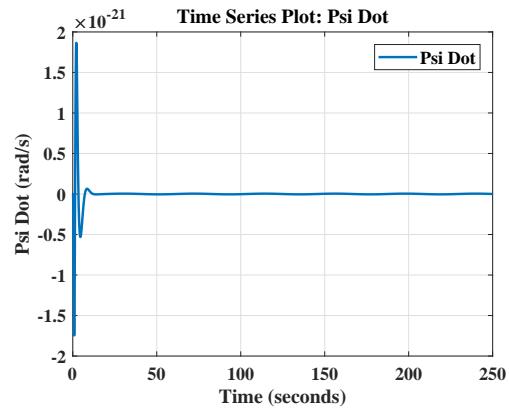
(i) Z Dot plot with time



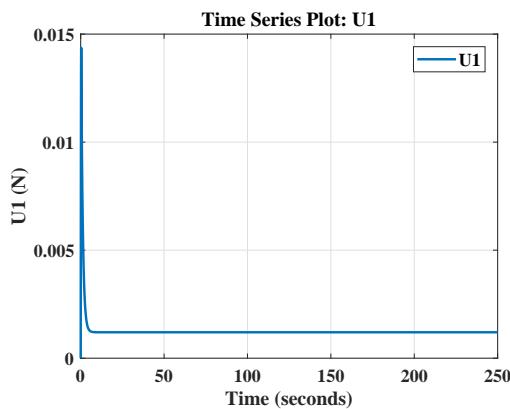
(j) Roll Dot plot with time



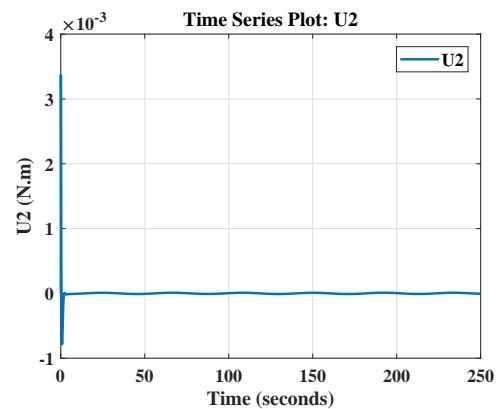
(k) Pitch Dot plot with time



(l) Yaw plot with time



(m) U1 plot with time



(n) U2 plot with time

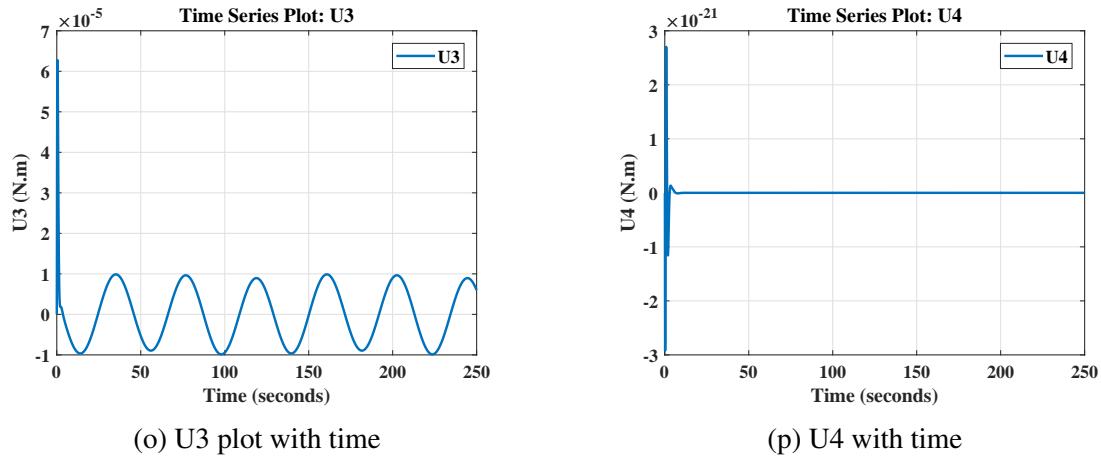


Figure 4.12: Closed Loop Response for Complex Helix Trajectory Tracking

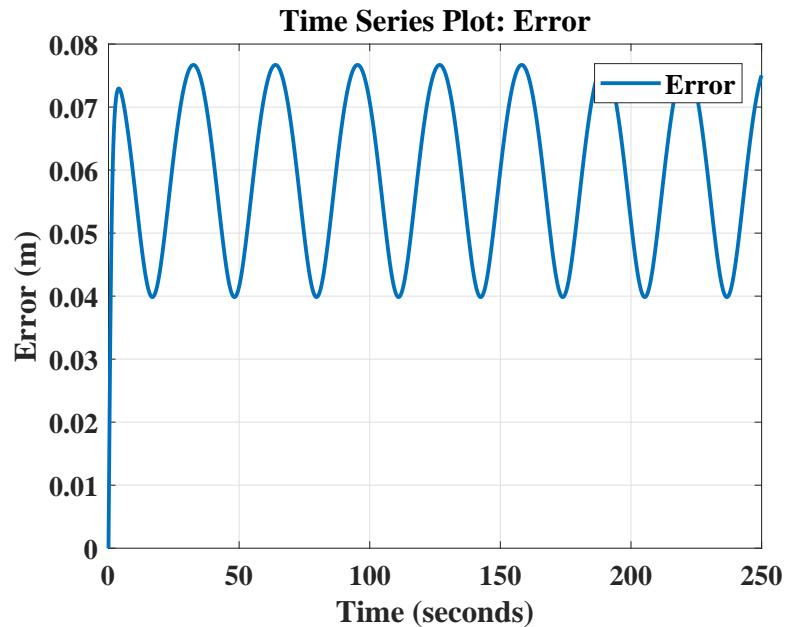


Figure 4.13: Error Between The Desired And Actual Trajectories

### 4.3.3 Model Predictive Control For Square Trajectory

the results for the MPC test to follow a square trajectory while having constraints on the rate of change of control inputs presented in (4.2.2), in order to validate the response of the controller dealing with aggressive turns.

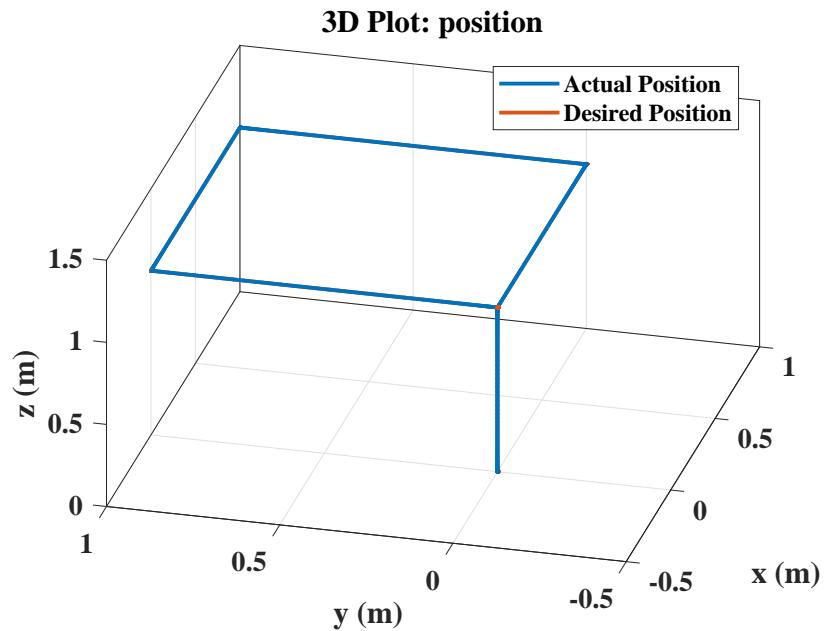
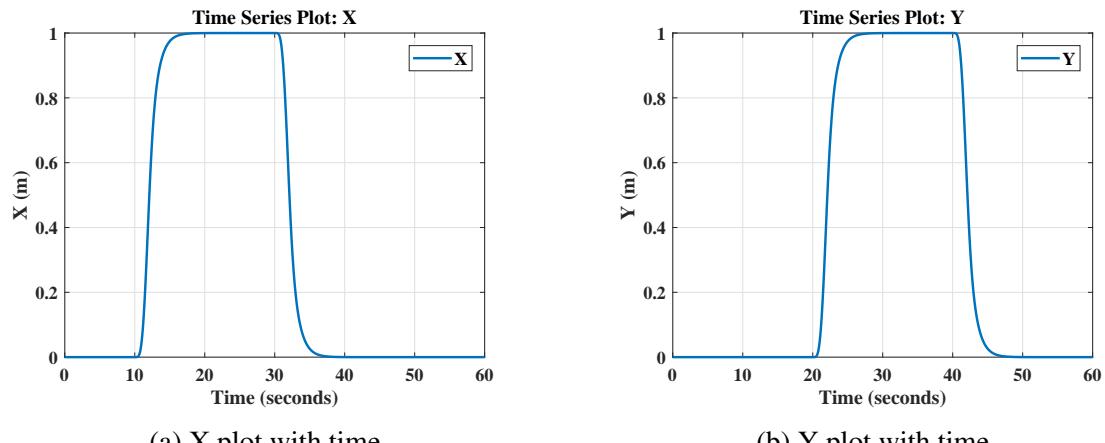
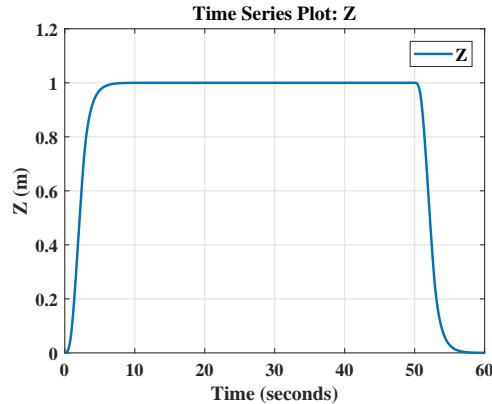
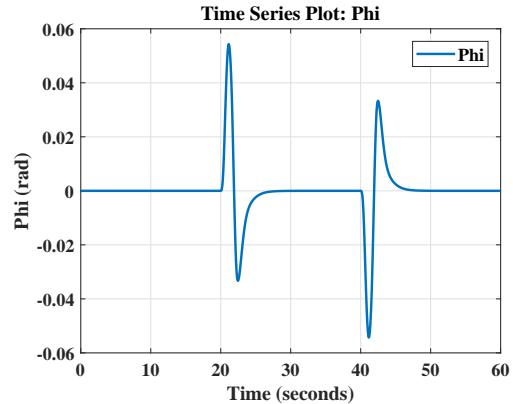


Figure 4.14: Quadcopter Tracking a Square Trajectory

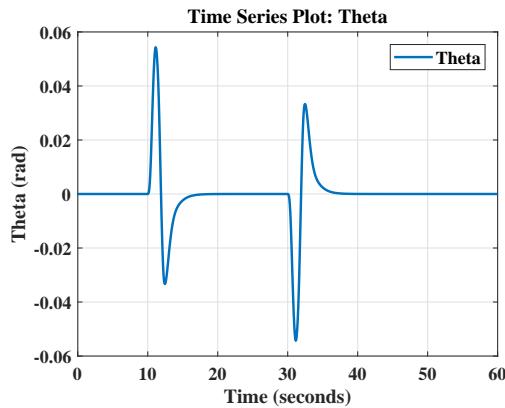




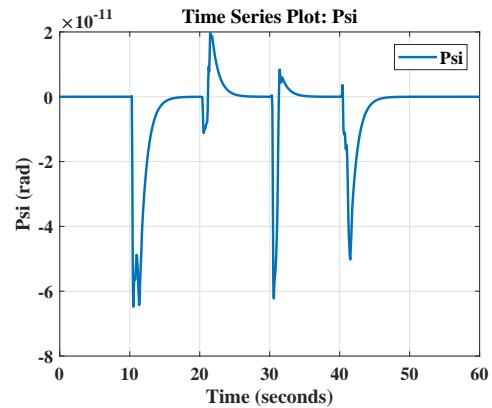
(c) Z plot with time



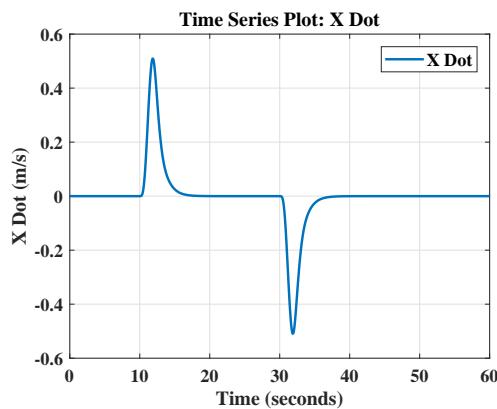
(d) Roll plot with time



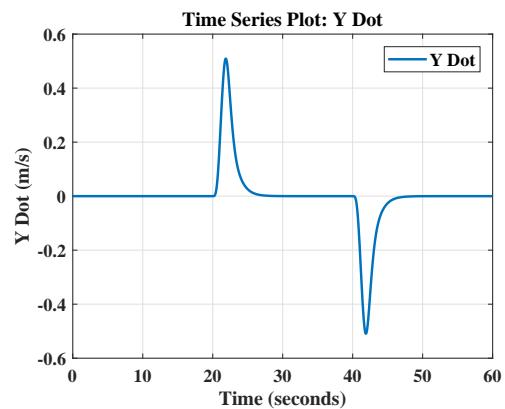
(e) Pitch plot with time



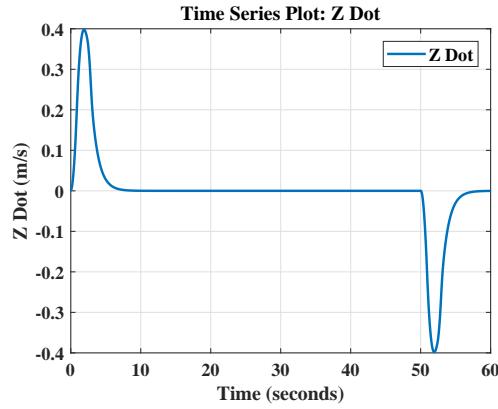
(f) Yaw plot with time



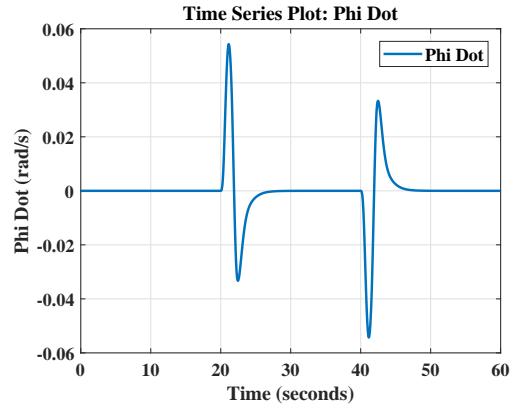
(g) X Dot plot with time



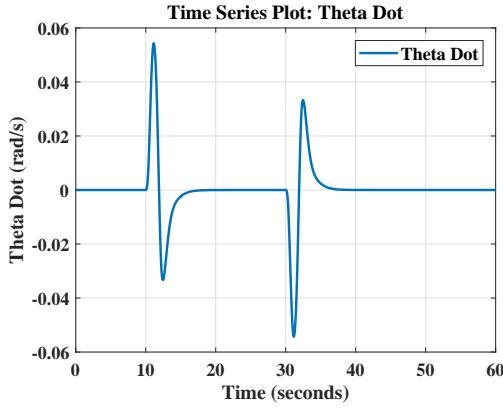
(h) Y Dot plot with time



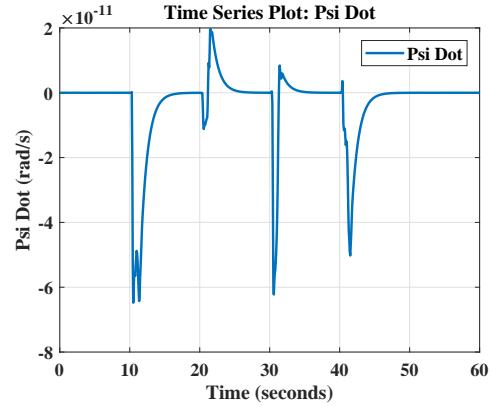
(i) Z Dot plot with time



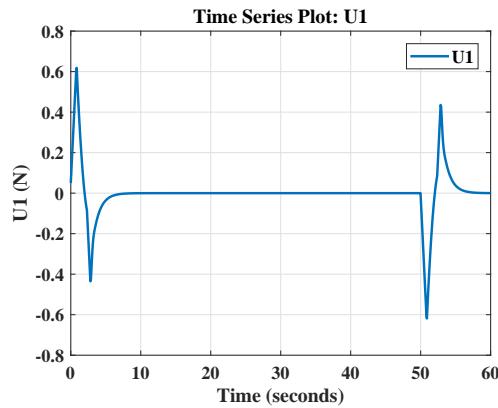
(j) Roll Dot plot with time



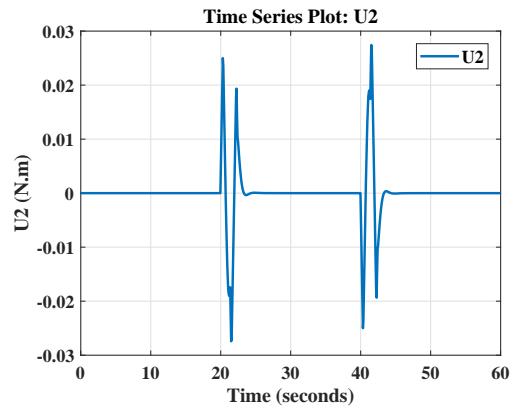
(k) Pitch Dot plot with time



(l) Yaw plot with time



(m) U1 plot with time



(n) U2 plot with time

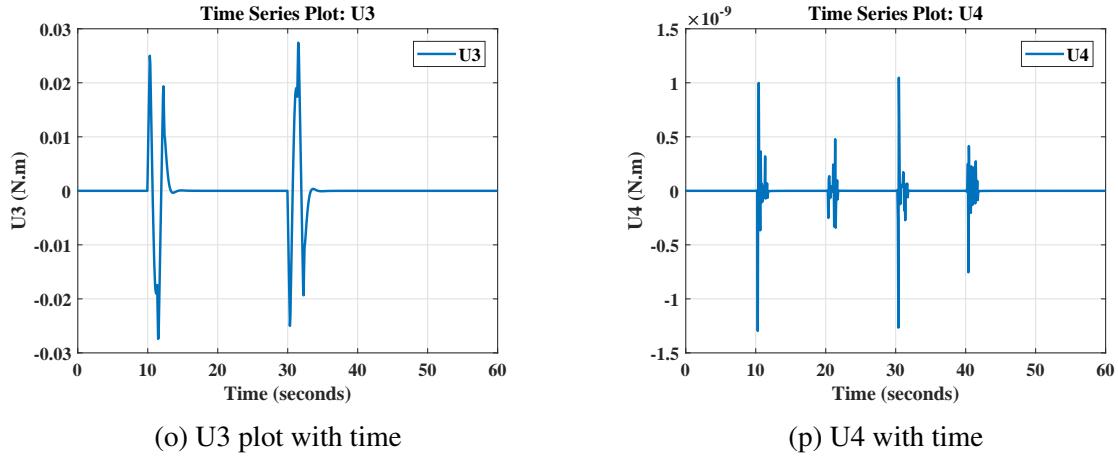


Figure 4.15: Closed Loop Response for Square Trajectory Tracking

Root-Mean-Square Error (RMSE) and Mean Absolute Error (MAE) are approaches to evaluate the accuracy of the data by comparison. The performance of the controller is evaluated using those two approaches. The tables show the comparison between reference trajectory and achieved trajectory values with respect to time for helix and complex helix trajectory.

Table 4.2: RMSE For Helix And Complex Helix

Root Mean Square Error(m)			
	X (m)	Y (m)	Z (m)
Helix	0.1066	0.1358	0.0127
Complex Helix	0.0422	0.0414	0.0127

Table 4.3: MAE For Helix And Complex Helix

Mean Absolute Error(m)			
	X (m)	Y (m)	Z (m)
Helix	0.0958	0.1053	0.0127
Complex Helix	0.0370	0.0364	0.0127

# **Chapter 5**

## **Conclusion and Future Work**

### **5.1 Conclusion**

The explosion of the market for handheld devices a few years ago accelerated incredibly the development of technologies, not only useful for mobile phones or digital cameras but also determinant for micro aerial vehicles. This project was born in this context, on the conviction that the development of miniature flying robots requires the simultaneous consideration of the system level optimization along with control design.

A lot of research is taking place in the field of quadcopter development and multiple entities are involved in modelling these unmanned aerial vehicles, designing their control laws. The mathematical model of a quadcopter was developed in details including its aerodynamic effects. A Model Predictive Control was developed. A complete simulation was then implemented on MATLAB/Simulink relying on the derived mathematical model of the quadcopter to evaluate the controller. the controller was successful in stabilizing the quadcopter and moving the three positions ( $x$ ,  $y$ ,  $z$ ) and the yaw angle to their desired values. Moreover, it provided some robustness against parametric uncertainties and external disturbance. Additionally, the controller was tested for trajectory tracking where helix, complex helix, square and star trajectories were used to validate its effectiveness.

### **5.2 Future Work**

The tasks that should be considered in the future work would be to implement the Nonlinear Model Predictive Controller. Also testing the Linear Model Predictive Control proposed in this thesis on an environment simulator called 'Gazebo' and then testing it on the real quadcopter. Changing the type of the optimizer which in return changes in the response. Extending the theory of MPC to be applied on a real life application.

# Bibliography

- [1] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.
- [2] S Norouzi Ghazbi, Y Aghli, M Alimohammadi, and AA Akbari. Quadrotors unmanned aerial vehicles: A review. *International Journal on Smart Sensing & Intelligent Systems*, 9(1), 2016.
- [3] Pengcheng Wang, Zhihong Man, Zhenwei Cao, Jinchuan Zheng, and Yong Zhao. Dynamics modelling and linear control of quadcopter. In *Advanced Mechatronic Systems (ICAMechS), 2016 International Conference on*, pages 498–503. IEEE, 2016.
- [4] Samir Bouabdallah and Roland Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2247–2252. IEEE, 2005.
- [5] Amr Nagaty, Sajad Saeedi, Carl Thibault, Mae Seto, and Howard Li. Control and navigation framework for quadrotor helicopters. *Journal of intelligent & robotic systems*, 70(1-4):1–12, 2013.
- [6] S Faiz Ahmed, K Kushsairy, M Izhar A Bakar, D Hazry, and M Kamran Joyo. Attitude stabilization of quad-rotor (uav) system using fuzzy pid controller (an experimental test). In *Computing Technology and Information Management (ICCTIM), 2015 Second International Conference on*, pages 99–104. IEEE, 2015.
- [7] Ramy Rashad, Ahmed Aboudonia, and Ayman El-Badawy. Backstepping trajectory tracking control of a quadrotor with disturbance rejection. In *Information, Communication and Automation Technologies (ICAT), 2015 XXV International Conference on*, pages 1–7. IEEE, 2015.
- [8] Mohd Ariffanan Mohd Basri, Abdul Rashid Husain, and Kumeresan A Danapalasingam. Backstepping controller with intelligent parameters selection for stabilization of quadrotor helicopter. *Journal of Engineering Science and Technology Review*, 7(2):66–74, 2014.
- [9] M Islam, M Okasha, and MM Idres. Dynamics and control of quadcopter using linear model predictive control approach, 2017.
- [10] Maidul Islam, Mohamed Okasha, and Moumen Mohammad Idres. Trajectory tracking in quadrotor platform by using pd controller and lqr control approach, 2017.

- [11] Ye Wang, Andres Ramirez-Jaime, Feng Xu, and Vicenç Puig. Nonlinear model predictive control with constraint satisfactions for a quadcopter, 2017.
- [12] A Benallegue, A Mokhtari, and L Fridman. Feedback linearization and high order sliding mode observer for a quadrotor uav. In *Variable Structure Systems, 2006. VSS'06. International Workshop on*, pages 365–372. IEEE, 2006.
- [13] Mapopa Chipofya, Deok Jin Lee, and Kil To Chong. Trajectory tracking and stabilization of a quadrotor using model predictive control of laguerre functions. In *Abstract and Applied Analysis*, volume 2015. Hindawi, 2015.
- [14] Anezka Chovancová, Tomas Fico, Peter Hubinský, and F Duchoň. Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator. *Robotics and Autonomous Systems*, 79:87–98, 2016.