6-1-2018

Dynamic modeling and robust nonlinear control of unmanned quadrotor vehicle

Amr Elhennawy

The American University in Cairo

School of Sciences and Engineering

Robotics, Control and Smart Systems

Dynamic Modeling and Robust Nonlinear Control of Unmanned Quadrotor Vehicle

By

Amr M.Elhennawy

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics, Control and Smart Systems

Under Supervision of
Dr. Maki Habib
Professor, Mechanical Engineering Department

May 14, 2018

Dedication

In the name of Allah. Praise is to you as befits the glory of your face and the greatness of your might.

I would like to dedicate this thesis to my family, especially my father, mother and wife. I would like to express my gratitude to my wife, whose support has been the backbone of my development. Likewise, special gratefulness to my parents who have always supported and encouraged me.

Finally, I would like to dedicate this work to my son, Youssuf, who have made me stronger, better and more fulfilled than I could have ever imagined. I hope you always follow the *Ayah*.

قُلۡ إِنَّ صَلَاتِي وَنُسُكِي وَمَحۡيَايَ وَمَمَاتِي
لِلَّهِ رَبِّ ٱلۡعَٰلَمِينَ ١٦٢

Aknowledgement

I would first like to thank my thesis advisor Prof. Maki Habib of the School of Science and Engineering at the American University in Cairo. Dr. Maki has always been supportive and providing the necessary tools and facilities without which this thesis would have never come to existence. The door to Prof. Maki office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently steered me in the right direction whenever he thought I needed it.

I would also like to thank my wife and my family for their uttermost support, understanding and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

I am grateful also to my managers at Nesma Airlines for their unconditional backing of my masters work, and their acceptance of my absence on several occasions for thesis work.

I must acknowledge as well the many friends, colleagues, students, teachers, and other librarians who assisted, advised, and supported my research and writing efforts over the years. Especially, I need to express my gratitude and deep appreciation to Khaled Abou Elsoud and Hashim Rezk whose help, support and encouragement saved me a lot of time and eased many difficulties. Many thanks also go to my old friends and mentors Ahmed Abo Elmakarem and Haithem E.Taha, whose lifetime experiences have always been role models for me to follow and their sincere guidance and direction have helped me go through hardships.

Abstract

[0.25 cm] It is not easy to control a quadrotor due to its highly nonlinear dynamics, variable coupling and model uncertainties. The underactuation property of the quadrotor also poses another degree of complexity to the model due to the limited availability of control techniques that can be applied to underactuated systems.
This thesis presents the development of mathematical modeling, control techniques, simulation and real-time testing on a developed quadrotor as an unmanned aerial vehicle.

Modeling of the dynamic system of a quadrotor including the motor dynamics is carried out using Newton-Euler mechanics and state space representation is obtained. Using this model a second-order Sliding Mode Control (SMC) is developed as a nonlinear robust control technique.
For the SMC development, quadrotor system is divided into two subsystems, one represents the fully actuated degrees of freedom and the other one represents the underactuated degrees of freedom. The aim of the proposed flight controller is to achieve asymptotic position and attitude tracking of the two subsystems by driving the tracking errors to zero to achieve the required tracking performance. Tackling of chattering problem associated with SMC is introduced.

Using the developed mathematical model and the developed two control techniques as linear and nonlinear approaches: the Proportional plus Derivative (PD), and SMC, simulation testing is conducted with and without the presence of external disturbances representing weight variation. Multiple simulations testing are performed to ensure the adequacy of the proposed control techniques using MATLAB and Simulink. Detailed discussion on the results of each control technique and comparison are presented with elaborate consideration of the robustness against weight variation. The simulation results demonstrate the ability of the SMC to drive the vehicle to stability and achieve the desired performance characteristics. . Finally, hardware design of a quadrotor has been developed and implemented with considerations on the hardware challenges are presented. Results of real-time fight tests using the two developed control techniques are presented and compared with that of the simulation results and it shows reliable performance of the nonlinear robust SMC controller. Flight tests results came consistent with the simulation results in terms of tracking performance, robustness and actuators efforts. Hardships in the implementation are mentioned and recommendations and future work are proposed. .

Nomenclature

Ω	Gyroscopic velocity
ω	Propeller rotational speed
ϕ	Roll angle
ψ	Yaw angle
θ	Pitch angle
d	Propeller's drag coefficient
g	Gravitational acceleration
I_y	Moment of inertia around y-axis
I_z	Moment of inertia around z-axis
$I_{x,y,z}$	Moment of inertia around x, y, z-axes
J_r	Propeller's moment of inertia
K_t	Propeller thrust constant
l	Distance from propeller's shaft and the vehicle's CG
m	Vehicle mass
U_i	Control inputs, i = 1, 2, 3, 4
x, y, z	Vehicle's coordinates in the Euclidean space

Contents

Dedication ii
Acknowledgment iii
Abstract iv
Nomenclature v
Contents viii

List of Figures **ix**

List of Tables **xii**

1 Introduction **2**
1.1 Historical Background 2
1.2 Underactuated Systems 3
1.3 Operational Principle 5
1.4 Statement of Purpose 6
1.5 Aims and Objectives 7
1.6 Thesis Structure 7

2 Literature Review and Research Methodology **10**
2.1 Literature Review 10
2.2 Achievments and Research Challenges in the Field 14
2.3 Research Statement 15

3 Mathematical Modeling **18**
3.1 Assumptions and Constraints 18
3.2 Kinematic Description 19
3.3 Rotation Matrix 20
3.4 Aerodynamic Loads 21
3.5 Forces and Moments 23
3.5.1 Forces 23
3.5.2 Moments 24
3.6 State Space Model 26

4 Sliding Mode Controller **29**
4.1 Literature Review 30
4.2 Nonholonomic Constraints 31
4.3 Sliding Mode Controller 32
4.3.1 Fully actuated subsystem controller 33
4.3.2 Underactuated subsystem controller 34
4.4 Chattering Alleviation 35
4.4.1 Saturation Function 36
4.4.2 Discontinuous to Continuous Control Law 37
4.5 Stability Analysis 37
4.6 Input Inversion 38

5 Simulation Results **40**
5.1 PD with no Disturbance 40
5.1.1 Slow-rate Input 42
5.1.2 Fast-rate Input 43
5.2 PD with Disturbance 44
5.2.1 Slow-rate Input 47
5.2.2 Fast-rate Input 48
5.3 Conclusion on PD Simulation 50
5.4 SMC with No Disturbance 50
5.5 SMC with Chattering Alleviation 52
5.6 SMC with Chattering Alleviation in the Presence of Disturbance 57
5.7 Conclusion 60

6 Quadrotor Hardware Development **62**
6.1 Design Requirements 62
6.2 Frame 63
6.3 Motors 65
6.4 Speed Controller 66
6.5 Propellers 68
6.6 Battery 70
6.7 Flight Controller 72
6.8 GPS/Compass 74
6.9 Radio Control 76
6.10 Telemetry 79
6.11 Vehicle Assembly 79

7 Experimental Results **83**
7.1 QGroundControl 83
7.2 Flight Data Parsing 84

7.3 PID Results 85
7.4 SMC Results 88
7.4.1 Pixhawk Support Package 88
7.4.2 SMC Experimental Results 90
7.5 Comments and Conclusion 94

8 Conclusions and Future Work 98
8.1 Conclusions 98
8.2 Future Work 99

Bibliography 101
References 106

Appendices 106

A MATLAB Codes 107
A.1 MATLAB Snippets 107
A.1.1 Saturation Function 107
A.1.2 Vehicle Equations 107
A.1.3 Input Inversion 108
A.1.4 Controller 109

B Hardware Setup 112
B.1 Ground Planner 112
B.2 ESC Calibration 112

C Motor Characterization 115
C.1 Introduction 115
C.2 Literature Background 117
C.3 Pixhawk Target Block Library 118

List of Figures

1.1 Quadcopter Hovering [10] 5
1.2 Roll and Pitch motion of a quadcopter [11] 6
1.3 Generation of yaw for a quadrotor [12] 6

2.1 Research approach layout 16

3.1 Schematic of quadrotor and Euler angles 20
3.2 Rotor disc according to Rankine-Froude momentum theory [37] 22
3.3 Connection of rotational and translational subsystems 27

4.1 SMC flight controller architecture 33
4.2 Typical Logistic function 36

5.1 Quadrotor PD control architecture 41
5.2 PD control attitude and altitude tracking - slow-rate input . . 42
5.3 x and y coordinates for PD control - slow-rate input 43
5.4 PD control actions - slow-rate input 44
5.5 PD control attitude and altitude tracking - fast-rate input . . 45
5.6 x and y coordinates for PD control - fast-rate input 45
5.7 PD control actions - fast-rate input 46
5.8 Disturbance Variation with Time 46
5.9 PD control attitude and altitude tracking with disturbance-slow-rate input 47
5.10 PD control actions - slow-rate inputs with disturbance 48
5.11 PD control attitude and altitude tracking with disturbance-fast-rate input 49
5.12 x and y coordinates for PD control - fast-rate input with disturbance 49
5.13 PD control actions - fast-rate input with disturbance 50
5.14 3D track of the quadrotor - slow-rate input - PD control . . . 51
5.15 3D track of the quadrotor - fast-rate input - PD control 51

5.16 SMC control attitude and altitude tracking - no disturbance . 53
5.17 Control action of sliding mode control - no disturbance 53
5.18 SMC control 3D track - no disturbance 54
5.19 SMC control attitude and altitude tracking - chattering alleviation 55
5.20 Control Action of Sliding Mode with Chattering Alleviation . 56
5.21 3D track for SMC with chattering alleviation 56
5.22 SMC control attitude and altitude tracking - chattering alleviation with disturbance 57
5.23 Control actions of SMC with Disturbance 58
5.24 3D track of SMC control - with disturbance 59
5.25 Tracking in the Presence of Disturbance - 2nd Attempt 59
5.26 3D Tracking in the Presence of Disturbance - 2nd Attempt . . 60

6.1 Quanum Venture FPV Quadcopter [46] 64
6.2 TURNIGY Multistar BLDC 2213 935KV 66
6.3 AFRO 20A ESC 68
6.4 T-Style 10x5.5 carbon fiber propellers 69
6.5 Turnigy 2200mAh 3S 35C 71
6.6 Pixhawk Mini Flight Controller 73
6.7 Pixhawk Mini Flight Controller with GPS/Compass 75
6.8 Conventional modes of an RC transmitter [51] 77
6.9 FlySky FS-i6 Radio Control 78
6.10 3DR 915MHz Telemetry 80
6.11 Quadrotor Assembly 80

7.1 PID test flight 85
7.2 Measured roll angle against RC roll input 86
7.3 Measured pitch angle against RC pitch input 86
7.4 Measured yaw angle against RC yaw input 87
7.5 Quadrotor relative position as measured by the GPS - PID Control 87
7.6 3D trajectory of the quadrotor - PID Control 88
7.7 Power discharge during flight - PID control 88
7.8 CPU and Ram usage during flight - PID control 89
7.9 First SMC test flight 91
7.10 Measured roll angle against RC input - SMC Results 91
7.11 Measured pitch angle against RC input - SMC Results 92
7.12 Measured yaw angle against RC input - SMC Results 93
7.13 Quadrotor relative position as measured by the GPS - SMC Results 93

7.14 3D trajectory of the quadrotor - SMC Results 94
7.15 Power discharge during flight - SMC control 94
7.16 CPU and Ram usage during flight - SMC control 95
7.17 3D Trajectory of PID control against 3D trajectory of SMC . 96

C.1 BLDC motor input and output schematic 116
C.2 Schematic diagram of the control system 117
C.3 Component losses as a percentage of total power [54] 118

List of Tables

5.1 Simulation Gains for PD controller 41
5.2 Simulation Gains for SMC 52
5.3 Simulation Gains for SMC with Chattering Alleviation 54

6.1 Specifications of the Quanum Venture FPV Quadcopter 64
6.2 Quadrotor Characteristics 65
6.3 Motor Specifications 66
6.4 ESC Specifications 67
6.5 Key features of the Propeller 70
6.6 Specifications of the Turnigy 2200mAh 3S 35C Battery [49] . . 71
6.7 Key features of the 3DR Pixhawk Mini autopilot 73
6.8 Key features of the 3DR GPS/Compass module 75
6.9 Key features of the FlySky FS-i6 RC 78
6.10 Technical specifications of the FlySky FS-i6 RC 79

C.1 comparison between BLDC motors and Brushed DC motors . 116

Chapter 1

Introduction

Vertical take-off and landing (VTOL) flying machines have evolved greatly in the past century and have attracted great attention from researchers in different disciplines due to their capabilities and missions they are capable of doing compared to fixed-wing Unmanned Aerial Vehicle (UAV).
Many reasons attracted researchers and engineers to VTOL planes, among them are:

1. Minimum space is needed for takeoff and landing
2. Increased portability compared to fixed wing aircraft
3. More suitable for aerial photography and videography, especially small VTOL planes

Rotor craft have drawn the main focus of development and research in the past century, including their many variations. Some of which have discontinued, some proven inefficient and some were not completed due to technological limitations at the time of development [1].

1.1 Historical Background

In the course of history, many VTOL machines were also introduced with different propulsion techniques, such as:

1. Tiltrotor which is a propeller rotor that tilts its rotors vertically for VTOL and then tilts them forwards for horizontal fixed-wing flight.
2. Jetlift which uses thrust vectoring to redirect the thrust of the jet engine to vertically take off and then moves it back to horizontal flight.

With the advance of propulsion and control, researchers started the development of mini VTOL machines; over the past decade quadrotor has been the major area of development. Different configurations, control technique, modeling methodologies and structural designs were addressed in the literature [2].

With the advent of Micro-Electro Mechanical Systems (MEMS), and the profound understanding of aerodynamics and control system even smaller flying machines have become possible and Micro Air Vehicles (MAV) came into presence [3] and researchers started studying the bio-inspired flying mechanisms.

From technical point of view, VTOL UAVs are classified as underactuated mechanical systems. Underactuation in its simplest form occurs when the system has a lower number of actuators than degrees of freedom and this poses more limitations on its control system and adds complexity the modeling of the system.

1.2 Underactuated Systems

Underactuated systems is an umbrella term that includes many real life mechanical systems such as aircraft, helicopters, vertical takeoff and landing aircraft, mobile robots, inverted pendulum, walking robots and flexible systems [2].

Underactuation describes the property of a mechanical system that cannot be commanded to follow arbitrary trajectories in generalized coordinates.

Spong defined underactuated systems as systems that have fewer control inputs than degrees of freedom and arise in applications, such as space and undersea robots, mobile robots, flexible robots, walking, brachiating, and gymnastic robots [4].

Olfati defined underactuated systems as mechanical control systems with fewer controls than the number of configuration variables. Examples of underactuated systems include flexible-link robots, mobile robots, walking robots, robots on mobile platforms, cars, locomotive systems, snake-type and swimming robots, acrobatic robots, aircraft, spacecraft, helicopters, satellites, surface vessels, and underwater vehicles [5].

Being an underactuated system poses difficulties on controlling the system, as the underactuation of a system makes arbitrary trajectory tracking impossible. This phenomenon also comes up with difficulties such as mechanical coupling and nonlinearity.

Underactuated systems cannot use feedback linearization or passivity-based control, rather partial feedback linearization or different control techniques can be utilized instead [4] as will be illustrated.

Moreover, inherent undesirable properties such as high relative degree (more than 2) and nonminimum phase behavior are also manifested in underactuated systems [4].

Because of the nonlinear coupling between the directly actuated degrees of freedom and the underactuated degrees of freedom it is impossible to use smooth feedback to stabilize the systems around equilibrium states even locally except after the existence of very restrictive necessary conditions [6].

According to Newton, the dynamics of mechanical systems are second order (F = ma). Their state is given by a vector of positions q, and a vector of velocities $\dot{q}$ and time t. The general form for a second-order controllable dynamical system is:

$$\ddot{q} = f(q, \dot{q}, u, t) \tag{1.1}$$

Where u is the control input. However, many of the mechanical systems can be expressed in control affine forms which have been studied extensively in nonlinear control theory [7, 8]. Control affine systems are linear in the actions but nonlinear with respect to the state. A control affine system has inputs which appear linearly where the nonlinearity with respect to the state is automatically implied [9]. The general form of a control affine system is

$$\dot{x} = f(x) + \sum g_i(x)u \tag{1.2}$$

However, underactuated systems represented in control affine form has a slightly constrained form

$$\ddot{q} = f_1(q, \dot{q}, t) + f_2(q, \dot{q}, t)u \tag{1.3}$$

A fully actuated system has a sufficient condition

$$rank[f_2(q, \dot{q}, t)] = dim(q) \tag{1.4}$$

While underactuated system satisfies the inequality

$$rank[f_2(q, \dot{q}, t)] < dim(q) \tag{1.5}$$

Great attention has been given to studying underactuated systems modeling and control techniques due to its rising importance and vast applications such as mobile robots and flexible manipulators [4]. Xu and Özgüner described underactuated systems by saying “Underactuated systems have very important applications such as free-flying space robots, underwater robots, surface vessels, manipulators with structural flexibility, etc. They are used for reducing weight, cost or energy consumption, while still maintaining an adequate degree of dexterity without reducing the reachable configuration space. Some other advantages of underactuated systems include less damage while hitting an object, and tolerance for failure of actuators” [6].

1.3 Operational Principle

A quadcopter is a multirotor helicopter that is lifted and propelled by four rotors. Unlike most helicopters, quadcopters use two sets of identical fixed pitched propellers; two clockwise (CW) and two counter-clockwise (CCW). This configuration causes the torque from each motor to cancel by the corresponding motor rotating the opposite direction. This uses the variation of the rotational speed to control lift and torque. Control of vehicle motion is achieved by altering the rotation rate of one or more rotor discs, this changes its torque load and thrust/lift characteristics. This controls the quadcopter in roll, pitch and yaw motions. Figure 1.2 shows how pitch and roll motions are generated for a typical quadcopter. Pitch and roll movements are generated through the increase of the rotational speed of one of the rotors while maintaining the others at the same speed, this generates a moment around the respective axis.

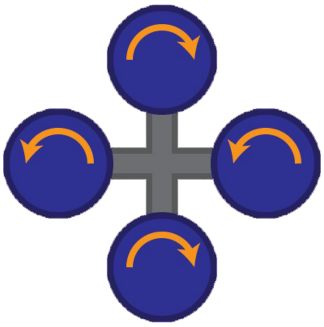

Figure 1.1: Quadcopter Hovering [10]

Applying a difference of rotational velocity between the opposing rotors forcing the vehicle to tilt towards the slowest propeller.

The horizontal motion is achieved by making the vehicle roll or pitch firstly,so the direction of the thrust vector is changed and then produce a forward component. Coupling between roll/pitch and horizontal motion is due to the underactuation property of the system. Increasing the thrust generated by one rotor as shown in figure 1.2 provokes roll or pitch moment depending on the selection of the rotor. And due to the fact that quadcopters system has coupled dynamics, roll or pitch moments are accompanied with motion in x or y directions.

Quadrotor yaw is adjusted by applying more thrust to a pair of rotors rotating in the same direction than the other two as depicted in figure 1.3.

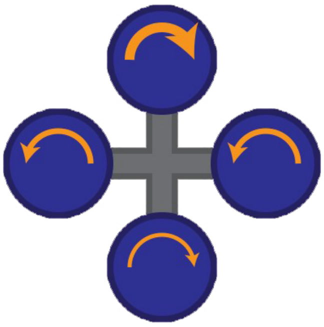

Figure 1.2: Roll and Pitch motion of a quadcopter [11]

Therefore, when yaw moment is required, the flight controller will slow down opposite pairs of motors relative to the other pair. This means the angular momentum of the two pairs of props will no longer be in balance and the craft rotates. We can make the multirotor rotate in either direction by slowing down different pairs of motors.

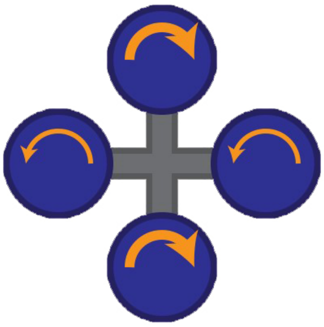

Figure 1.3: Generation of yaw for a quadrotor [12]

1.4 Statement of Purpose

The main purpose of this thesis is to study and analyze the characteristics of a quad rotorcraft, develop a nonlinear robust controller and build up a model to verify the simulation results.

Problems associated with modeling are presented along with the main assumptions, complexity of nonlinear controller are introduced and dealt with and hardware design requirements are shown.

1.5 Aims and Objectives

In the past few years, quadcopters have gained currency and been used widely in different applications. The literature reveals the wide spectrum of applications and the necessity of a maneuverable vehicle. This highlights the need for a robust and reliable controller. Consequently, this research aims to narrow the gap between theoretical and experimental implementation. By the end of this document, mathematical model should be derived, proposed controller should be developed and actual implementation on hardware should be carried out.

To achieve this aim, Newton-Euler formalism shall be used in deriving the mathematical model of the nonlinear underactuated system of the quadcopter. Then a nonlinear robust controller shall be developed using Sliding Mode Control (SMC). Problems associated with SMC will be tackled to achieve the required performance. Various simulations will be carried out afterwards to ensure the developed controller capabilities of achieving stability and robustness. Finally, implementation of the model on real quadcopter will be presented along with the used components and design requirements.

1.6 Thesis Structure

This thesis is structured as follows, chapter 2 shows a thorough review of the literature review along with the research methodology, chapter 3 presents the modeling of the quadcopter dynamics based on Newton-Euler formalism. In the modeling chapter, a detailed derivation of forces and moments equations is elaborated and assumptions are clearly stated.

In chapter 4 the proposed Sliding Mode Control (SMC) is presented, establishment of the sliding surfaces is clarified and derivation of the nonholonomic constraints on the states of the quadcopter is explained. Tackling the SMC chattering problem is presented and equations derived.

In chapter 5 the simulation results of the proposed controller are shown. The original controller's results and after applying the chattering alleviation technique. Disturbance is later fed into the model to account for outdoor phenomenon of wind in the z-direction, simulation is carried out and results

are displayed.
Chapter 6 presents the real hardware used in the development of the vehicle. Flight controller model and specifications shall be clearly stated along with other components and peripherals such as battery, motors, telemetry, Radio Control (RC) and Electronic Speed Controllers (ESC).
Chapter 7 presents the experimental results of the flight tests using both PID control and SMC. Comparison of results is conducted and conclusion is provided.
Finally, in chapter 8 the final conclusion of the thesis work is presented with potential future work.

Chapter 2

Literature Review and Research Methodology

2.1 Literature Review

Recently, quadrotors have attracted the interest of scientists and researchers because of a wide area of applications and its many advantages. Quadrotor VTOL machines with hovering capability, high maneuverability and agility. If compared with standard helicopters, quadrotors have several advantages such as small size, efficiency, and safety. These advantages made quadrotors eligible for applications like military services, surveillance, rescue, research area, remote inspection, and photography [13].

However, controlling the quadrotor into doing the assigned mission is not an easy task, due to the inherent difficulties associated with the dynamic model of the system such as high nonlinearity, coupled dynamics and multi variable mechanism. This is due the fact that the quadrotor has six degrees of freedom (three translational and three rotational) while it only has four input which are the thrust generated by each one of its propellers.

Another challenge that faces the development of quadrotor control system is the level of uncertainty associated with the physical parameters.

Several control techniques have been adopted in the literature to tackle such problems and difficulties. Linear control methods such as PID (proportional, integral and derivative control) are applied such as in [14], but normally this linear controller is only guaranteed to be stable in certain region and restricted domain of flight. The simulation and experimental testing is carried out that showed good performance in the linearization range with delicate performance when subjected to disturbance. 20% overshoot occurred in pitch angle. In [15] gain tuning is added to the PID controller using gradient method. PID tuning showed better performance when compared to constant gain PID controllers. This better performance is exhibited in lower values of overshoot but still has restricted region of guaranteed stability.

Another PID tuning is mentioned in [16] using fuzzy logic as a tuning engine. Fuzzy logic tuning significantly enhanced controller performance but was not able to handle problem such as overshoot and smooth transition of controlled values. Simulation results showed that the controller manipulated gain values to attenuate the disturbances which improved the performance with no experimental setup shown.

Likewise, in [17] a combination of PID and a fuzzy control algorithm is proposed to control a quadrotor. Experimental flight test are carried out with the developed controller which showed the adaptability of the quadrotor for exterior, environmental changes.

To expand the range of operation and domain of controllable flight, nonlinear control technique should be used. Different nonlinear control techniques are found in the literature such as backstepping, feedback linearization, sliding mode control, nonlinear modification to the PID control and different combinations of two or more techniques.

In [13] nonlinear controller using a backstepping-like feedback linearization method to control and stabilize the quadrotor is utilized and showed good performance in different maneuvers in outdoor testing without abrupt disturbances. In [18], Bouabdallah and Siegwart (2007) proposed a combination of PID controller and integral backstepping, where backstepping was used to control the quadrotor and PID was used to control motor speed.

In [19], Benallegue and Madani (2006) also used backstepping but divided the controller into three nested subsystem, underactuated subsystem, fully actuated system and propeller system. A complete derivation of the backstepping control equation is shown. Results of this techniques showed great tracking performance of position and yaw angle with stabilized roll and pitch angles.

In [20] the same technique as in [19] is followed with only the propeller subsystem is put in control affine form and state feedback to the rotor equation is presented. This small modification enhanced the stability of roll and pitch angles.

Feedback linearization is also used for designing a controller of a quadrotor. In [21], Feedback linearization coupled with a PD controller for the translational subsystem and backstepping-based PID nonlinear controller for the rotational subsystem. The proposed model showed lower overshoot when compared with conventional PID. Integral term of the backstepping-based PID controller showed good steady-state performance.

Dividing the whole system into two nested subsystems is used in [22]. The two nested subsystems have one faster than the other. The faster system which represents position control is controlled using feedback linearization while the slower one represents velocity control and is controlled by nonlinear proportional control. This technique simplified the dynamics of the system by using feedback linearization while taking into account the nonlinear nature of the system. However, results showed some overshoots especially in the transition phase of the variables.

Although nonlinear control techniques are proved to be more efficient and covering wider range of application, external disturbances still poses imminent threat to the stability of the system. As the nonlinear control techniques are applied assuming well-known and well-defined mathematical model and precise anticipation of the loads.

However, this is not the real case especially in outdoor applications. Hence, robust and adaptive control techniques should be applied to account for any external disturbances or parametric uncertainties.

The literature has many robust and adaptive control techniques used for the control of a quadcopter. Xu and Özgüner [6] proposed a Sliding Mode Control (SMC) for a class of underactuated systems that can be transformed into a cascaded form, and applied this technique on quadrotors as an example of this class of underactuated systems. The paper arrived at a general formula for sliding mode controllers of underactuated systems represented in cascade form, with controller ability to recompense any external disturbance within a priori known range.

Islam et al. used the same methodology developed by Xu and Özgüner and added an adaptation engine to recalculate the uncertainties range online and adjust controller gains accordingly [23].

In [24], a nonlinear robust control (SMC-like) with adaptation law to calculate the bounds of uncertainty online. The control scheme is intended to take large disturbances into account through the use of online adaptation law and application of SMC.

The model is divided into two subsystems in [25], fully actuated system and underactuated system, and SMC is applied to both subsystems. The underactuated subsystem controller is linearized and its coefficients obtained by Hurwitz criterion. General formula for obtaining the coefficients of the linearized sliding manifold is derived and presented as a function of Euler angles. Results showed good degree of robustness around operating points.

Sliding mode control is compared to backstepping in [26]. Backstepping showed good degree of resilience even in the presence of small perturbations, while the SMC is greatly dependent on the values of sliding manifold gains.

Feedback linearization with simplified dynamics and small angle approximation is compared to SMC with input augmentation in [27]. The input augmentation of the SMC aimed at overcoming the under actuation property of the system and minimizing controller gains. The mathematical model was simplified by neglecting Coriolis effects of the rotating propellers. Feedback linearization showed great sensitivity to noise and disturbances when compared to SMC which was able to handle perturbation due to uncertainties, unmodeled dynamics or unexpected external stimulations.

Adaptive SMC is also used in the literature to accommodate actuator failure as in [28] where two control modules with virtual control and control allocation. A control virtual module is used to redistribute the control effort among the sound actuators with online adaptive scheme. The aim of the developed controller was to maintain the closed-loop stability of the quadrotor under multiple actuator failures. Stability check is conducted in Lyapunov sense and confirmed through experimental results based on modified multirotor. The results showed the strength and robustness of the SMC for multirotor control.

The use of robust observers is also present in the literature. In [29], backstepping control is used with sliding mode observer which is used to estimate velocity and external disturbance. Simulations of results showed that the real values approximate estimated values of disturbances which the controller could overcome and track a pre-defined trajectory with small errors. Robustness of observer-controller is verified through the introduction of external disturbances.

In [30], a SMC is incorporated along with sliding mode observer. Simulation results showed robustness, fault tolerance and ability to deal with different types of severe disturbances. The controller in this case was able to maintain the stability of flight under extreme conditions such as collision and after one rotor has lost 50% of its propelling power. In the presence of wind gusts and other sources of disturbance, the controller could compensate for them and force the quadrotor to follow the input trajectory although the effect of gusts and disturbance is clearly seen.

The extensive use of SMC for the control of quadrotor highlighted the need to tackle a problem associated with high control effort that comes with applying SMC and is called chattering. Karami et al. [31] developed a fractional-order dynamic sliding mode control (FDSMC) for a class of nonlinear systems is presented where an integrator is used before the input signal of the plant, in order to remove the chattering. The proposed control scheme resulted in a controller with higher degree than original controller without an integrator. This posed difficulties to the modeling and required a complete and full understanding of system dynamics. This problem was tackled by adding a nonlinear observer to anticipate the unknown dynamics of the system. Chattering free SMC resulted from the proposed method and stability is checked in Lyapunov sense.

In [32], and due to the complexity of the mathematical model, the authors opted to use neural networks to predict and model the dynamics of the system. The model used depended on online learning scheme, so that it could always account for external disturbances and noise that occur while flying. Neural network observer is also developed to estimate translational and angular velocities. An output feedback control law is developed in which only the position and the attitude of the UAV are considered measurable. Stability of the model is proved in Lyapunov sense and illustrated by simulation results.

Fuzzy logic control is also introduced in quadrotor control as in [33] where inputs were the desired values of the height, roll, pitch and yaw. The outputs are the power of each of the four rotors. Mamdani inference and centroid method for defuzzification are used. Extra aggregator is added to implement the coupling of inputs to the four rotors. Results are fairly acceptable without the hustle needed to develop complex control systems and with simple dynamic modeling. However, the robustness of the system is not proven and noise attenuation is not dealt with.

Another trend in mathematical modeling of quadrotors is the use of quaternions instead of traditional Euler angles. This is used when extreme maneuvers are expected with angles exceeding the range [-90, 90] degrees to avoid the common gimbal lock the represents a singularity in the model and cause disorder to the system.

Liu et al. [34] derived the quadrotor mathematical model using quaternions and applied a simple PID-like controller with a robust compensator to attenuate noise and disturbance. The use of quaternion greatly complicated the mathematical model and does not offer any direct advantage in the normal operating conditions as long as maneuvers are bounded by the normal operating range of [-90, 90] degrees.

2.2 Achievments and Research Challenges in the Field

Rotorcraft development poses difficulties in the design, modeling, control and implementation. This is mainly due to the inherent nature of rotorcraft's dynamics. Quadrotors -as a type of rotorcraft- is not an exception. Its dynamics are highly nonlinear and coupled, it has an underactuated property and it suffers from vibrations, parametric uncertainties and difficulties in control.

As shown in section 2.1 the literature is very rich with different control techniques for quadrotors. Spanning from linear PID control to more advanced, mathematically exhaustive nonlinear and robust control techniques. The literature already discussed stability of the quadrotor, control of a quadrotor to achieve trajectory tracking and safe landing of a quadrotor in case of a failure occurs to one of the propellers.

The literature is also full of intelligent systems and neural network-based controllers of quadrotors. Most of them do not need exact modeling as they are dependent on online learning scheme to predict the dynamics of the system. However, these controllers need huge computational resources to adapt the heavy online computations.

The literature showed the limited capabilities of the linear control techniques to deal with a nonlinear underactuated system like the quadrotor. Although, gain tuning techniques sometimes accompany the PID control and enhance its performance, they did not increase the robustness against disturbances and rendered the quadrotor susceptible to unexpected inputs.

The nonlinear control techniques enhanced the tracking performance of the quadrotor when compared to linear control. Nonlinear control techniques are generally more mathematically exhaustive and need more computational power, however, robust nonlinear techniques proved themselves as powerful techniques with the underactuated systems.

The main challenges of the quadrotor system are the system nonlinearity, underactuation, mechanical coupling and the effect of external disturbances.

2.3 Research Statement

To tackle the problems associated with controlling the quadrotor, a Newton-Euler modeling approach is used to formulate the dynamic equations of motion of the quadcopter that is a nonlinear and underactuated system while laying down certain list of assumptions and constraints.

Sliding mode controller (SMC) is developed as a nonlinear robust technique with the aim to fulfil the control requirements of the unmanned quadrotor vehicle system. The SMC is associated with the chattering problem that affects its performance and it is required to be alleviated using the boundary layer solution to relax the control effort.

Quadrotor is developed covering the required technical specifications and characteristics and it is used as a testbed.

In order to show the performance of the SMC, both simulation and real time testing are conducted to assess the strengths and weaknessess of the controller and compared with the results obtained from a linear PD controller.

Figure 2.1 shows the layout of the research approach adopted in this thesis with the underlying subtasks.

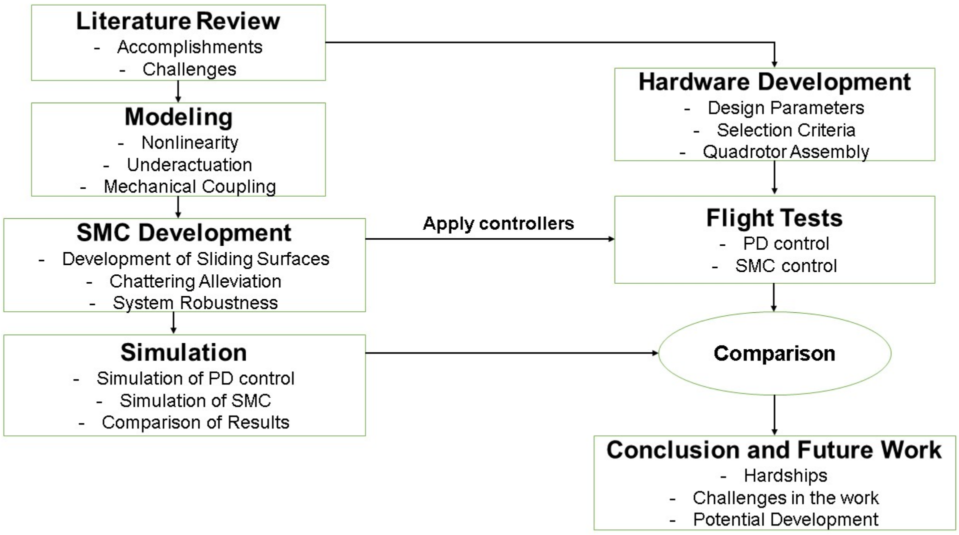

Figure 2.1: Research approach layout

Chapter 3

Mathematical Modeling

As mentioned before, quadcopter is an underactuated and nonlinear system. This poses difficulties in modeling the system which also adds complexities to the interdisciplinary nature of the vehicle.

Proper modeling of a quadcopter requires describing aerodynamic forces coming from the propellers, anticipating the response of the Brushless DC (BLDC) motor rotating the propeller and profound understanding of the dynamic behavior of the whole combined electro-mechanical system.

3.1 Assumptions and Constraints

Certain assumptions are presented to simplify complex model and are used through this thesis.

1. The structure is supposed to be rigid
2. The Centre of gravity and body fixed frame origin are supposed to coincide
3. Counter clockwise is the positive direction of rotation
4. Quadrotor is supposed to be symmetric about two of the three Cartesian planes (Products of inertia = 0)
5. Propellers are supposed to be rigid and blade flapping is neglected
6. Propellers' weight is negligible and propellers are identical with similar aerodynamic constants
7. Blades have constant pitch angle with no swash plates
8. Euler angles are assumed to be bounded as follows:

  - Roll: $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$
  - Pitch: $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$
  - Yaw: $-\pi \leq \psi \leq \pi$

  This means that the vehicle cannot undergo an inverted flight.

9. Weight force is applied to the center of gravity and directed along the negative z-axis relative to the Earth frame

10. Flow near the propellers is assumed to be low subsonic, incompressible and quasi-steady

11. Flow across propeller disc is assumed to be 1D

12. Vortex wakes and blade tip vortices of the propeller are neglected in the model

13. Ideal flow assumptions across the rotor blade is assumed (inviscid one-dimensional flow)

3.2 Kinematic Description

Let $q = (x, y, z, \phi, \theta, \psi) \in \mathbb{R}^6$ be the generalized coordinates where $\zeta = (x, y, z) \in \mathbb{R}^3$ is the absolute position of the mass of the quadrotor relative to a fixed inertial frame and $\eta = (\phi, \theta, \psi) \in \mathbb{R}^3$ denotes Euler angles, which are the orientation of the quadrotor.
ϕ is the roll angle around the x-axis, θ is the pitch angle around the y-axis and ψ is the yaw angle around the z-axis. Schematic of a quadrotor and the forces affecting its normal flight regime is shown in figure 3.1. According to the assumptions stated above, the main forces that will be affecting the quadcopter will be:

- Aerodynamic forces; Thrust, drag and propeller rotation (Propeller flapping is neglected)
- Inertial counter torques
- Gravity forces
- Gyroscopic forces; Change in orientation
- Friction
- Ground effect

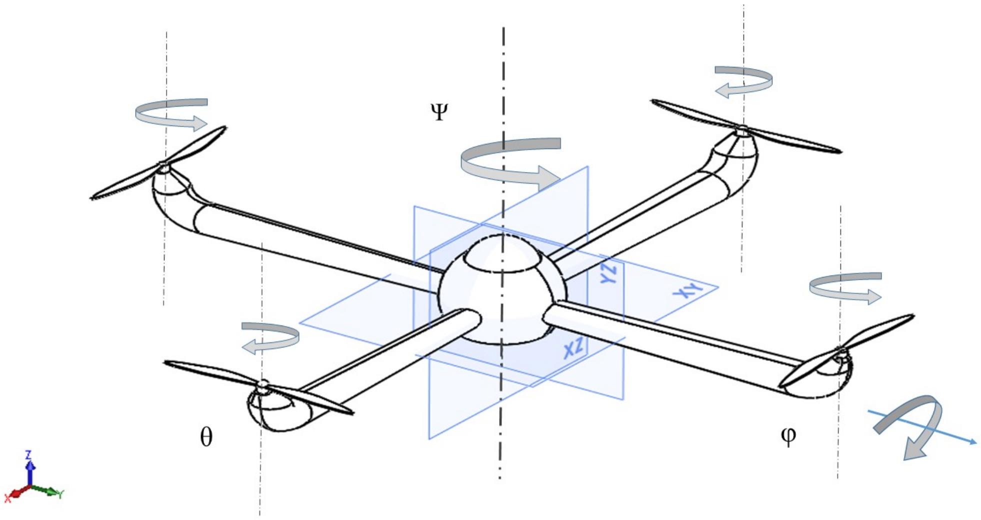

Figure 3.1: Schematic of quadrotor and Euler angles

3.3 Rotation Matrix

The rotation of a rigid body in space can be parameterized using several methods like Euler angles which are a mathematical representation of three successive rotations about different possible axes (numerous conventions) which are often confused in literature.
The three single rotations are described in the following order:

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

These rotations are represented by the following matrices:

$$
\begin{aligned}
R(x,\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
R(y,\theta) &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \\
R(z,\psi) &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.1}
$$

This makes the whole rotation matrix

$$
\begin{aligned}
&R(\phi,\theta,\psi) = R(x,\phi)R(y,\theta)R(z,\psi) \\
&= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}
\end{aligned}
\tag{3.2}
$$

This represents the transformation matrix from quadrotor frame to inertial reference frame.

3.4 Aerodynamic Loads

Aerodynamic loads are obtained by a combination of Blade Element Theory (BET) and Rankine-Froude momentum theory as mentioned in [35].
The momentum theory approach allows the derivation of a first-order prediction of the rotor thrust and power by assuming that the rotor is consisted of an infinite number of blades and may therefore be considered as an "infinitely thin actuator disk" [35, 36].
Rotor is also assumed to be uniformly accelerating the air through the disk with no loss at the tips [37]. Figure 3.2 shows schematic of momentum theory. The blade element theory assumes that each blade section acts as a quasi-2D airfoil which generates aerodynamic forces and moments [35]. Tip vortices and other losses are implied in the empirical formula.
According to [35, 36, 37], Aerodynamic thrust in hovering is governed by the following equation.

$$T = C_T \rho A R^2 \omega^2 \tag{3.3}$$

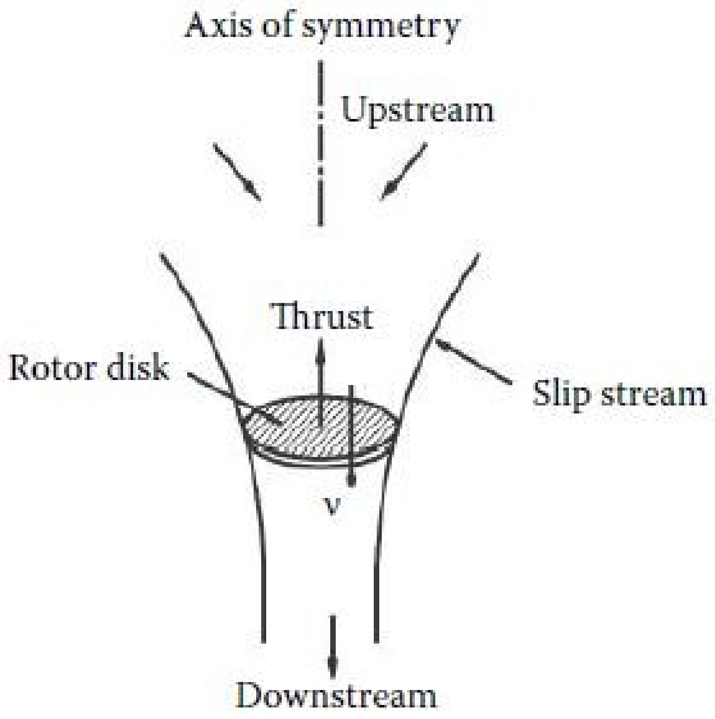

Figure 3.2: Rotor disc according to Rankine-Froude momentum theory [37]

Where A is the rotor disc area, C_T is thrust coefficient, R rotor diameter and ω is the rotational speed of the rotor. The thrust factor depends on rotor blade pitch and geometric characteristics.
Some correction factors are mentioned in the literature to account for induced velocity at the rotor tip, viscosity effects, wake swirls and some other non-linearities in the model. Rich clarified the derivation based on blade element theory and arrived at a simplified model of the thrust that can take account of these nonlinearities, and the final equation of the rotor thrust would take the form [38]

$$T = 2\lambda_o \rho \pi r^4 \omega^2 \tag{3.4}$$

Where λ_o is the inflow ratio (the ratio of the induced velocity to the rotor tip speed) and it is a constant parameter of the rotor that is available in its datasheet.
The constant value of the propeller thrust could be defined

$$K_T = 2\lambda_o \rho \pi r^4 \tag{3.5}$$

And the total thrust of the propeller would be

$$T = K_T\omega^2 \tag{3.6}$$

A simplified equation for K_T is mentioned in [37] which deals only with basic regimes of hovering flight and neglecting all kinds of nonlinearities and induced flows.

$$K_T = C_T\rho Ar^2 \tag{3.7}$$

According to [5] the power loss due to rotor drag is very hard to predict because it is a much smaller force than thrust and is thus sensitive to small changes in pressure. Also, typical flight regimes of a quadcopter are low subsonic speed where drag force does not comprise an effective parameter in the model. However, in [13] a simplified model for drag was adopted that addresses the basic level of aerodynamic problem of a propeller in which the thrust moment causes a drag moment on the rotor. The drag factor d is then defined by the equation

$$d = C_p\rho Ar^3 \tag{3.8}$$

Where C_P is the power coefficient of the propeller. The thrust and power coefficient are determined by static thrust tests which are given by manufacturer's or propeller database site. Elaboration on this is presented in section 6.5.

3.5 Forces and Moments

3.5.1 Forces

The forces affecting the quadrotor in the body axes are

$$F_B = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} T_i \end{bmatrix} \tag{3.9}$$

Where $\sum_{i=1}^{4} T_i$ is the total thrust of the four propellers. Transforming from body axes to earth axes using the transformation matrix gives us

$$F = R(\phi,\theta,\psi) = \begin{bmatrix} (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)\sum_{i=1}^{4} T_i \\ (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\sum_{i=1}^{4} T_i \\ (\cos\theta\cos\phi)\sum_{i=1}^{4} T_i \end{bmatrix} \tag{3.10}$$

– Forces in z-direction:

Weight: $w = mg$
Thrust: $T = (\cos\theta\cos\phi)\sum_{i=1}^{4} T_i$

- Forces in the y-direction:
  Actuator action: $(\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\sum_{i=1}^{4} T_i$
- Forces in the x-direction:
  Actuator action: $(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)\sum_{i=1}^{4} T_i$

And again for low subsonic speeds as in the case of the quadrotor drag force in both x and y directions are negligible compared to weight and thrust force. Thus, The translational equations of motions hence are

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)\frac{F_B}{m} \\ (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\frac{F_B}{m} \\ (\cos\theta\cos\phi)\frac{F_B}{m} - g \end{bmatrix} \tag{3.11}$$

3.5.2 Moments

1. **Actuation Torques**
   The actuation torques around x and y axes of the body frame are described by the equations

$$\tau_{xB} = (T_4 - T_2)l = K_T l(\omega_4^2 - \omega_2^2) \tag{3.12}$$

$$\tau_{yB} = (T_3 - T_1)l = K_T l(\omega_3^2 - \omega_1^2) \tag{3.13}$$

   where l is the distance from propeller's shaft and the CG of the vehicle. Realization of these equations is obvious due to the fact that any change in thrust of any motor will result in a torque around the perspective direction.
   The actuation torque around the z axis of the body frame is much less intuitive, as each rotor generates some torque about the body z axis. This torque is required to keep the propeller spinning and providing thrust; it creates the instantaneous angular acceleration and overcomes the frictional drag forces.
   According to the momentum theory discussed in [35, 36, 37, 38], drag force is a function of drag coefficient which depends on air density, blade configuration and dimensions and it is normally provided in the propeller data sheet. Yaw torque is the resultant of the blade drag force and the gyroscopic effect of the propeller around the z-axis. This could be represented in the form

$$\tau_{zB} = \tau_D + J_r\dot{\omega} \tag{3.14}$$

Where J_r is the propeller's rotor moment of inertia and $\dot{\omega}$ is the angular acceleration of the propeller. In normal flight conditions and hovering, the angular acceleration of the propeller is almost zero, so the gyroscopic term of the equation can be neglected without much influence on the system.
The drag torque according to [13] can be calculated from the equation

$$\tau_{zB} = \sum_{i=4}^{4} d\omega_i = d(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{3.15}$$

The constant d always satisfies $d > 0$ and it depends on air density, the radius and the shape of the blade as derived in [39].

2. **Gyroscopic Torques**
Due to the gyroscopic effect of the propellers, moments are generated along x and y axes. These moments are governed by the equation

$$\grave{\tau}_x = -J_r\dot{\theta}\omega \tag{3.16}$$

$$\grave{\tau}_y = -J_r\dot{\phi}\omega \tag{3.17}$$

No gyroscopic torque is generated about the z axis because the rotation of the propeller assumed to be planar in the direction of the z axis.
Derivation of gyroscopic moment equations and interpretation of the negative or positive signs is illustrated in Dynamics textbooks such as [40].

3. **Conservation of Angular Momentum**
According to the conservation of angular momentum law and clarified in [40] chapter 21 the moments around a certain axis of the Cartesian system is governed by equation

$$\sum M_x = I_x\dot{\omega}_x - (I_y - I_z)\omega_y\omega_z \tag{3.18}$$

So, the roll, pitch and yaw torques of the quadrotor which are generated by a difference between motors' thrust are governed by the following set of equations.

   - Roll:

$$M_\phi = I_x\ddot{\phi} - (I_y - I_z)\dot{\psi}\dot{\theta} \tag{3.19}$$

   - Pitch:

$$M_\theta = I_y\ddot{\theta} - (I_z - I_x)\dot{\phi}\dot{\psi} \tag{3.20}$$

   - Pitch:

$$M_\psi = I_y\ddot{\psi} - (I_x - I_y)\dot{\phi}\dot{\theta} \tag{3.21}$$

Moments' Equations

Applying the conservation of angular momentum, and summing up the moments in equations 3.12, 3.13, 3.15, 3.16, 3.17, 3.19, 3.20 and 3.21 we get the equations of the moments

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{(I_y - I_z)\dot{\psi}\dot{\theta}}{I_x} - \frac{J_r}{I_x}\dot{\theta}(\omega_2 + \omega_4 - \omega_3 - \omega_1) \\ \frac{(I_z - I_x)\dot{\psi}\dot{\phi}}{I_y} + \frac{J_r}{I_y}\dot{\phi}(\omega_2 + \omega_4 - \omega_3 - \omega_1) \\ \frac{(I_x - I_y)\dot{\theta}\dot{\phi}}{I_z} \end{bmatrix} + \begin{bmatrix} \frac{K_T l(\omega_4^2 - \omega_2^2)}{I_x} \\ \frac{K_T l(\omega_3^2 - \omega_1^2)}{I_y} \\ \frac{d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)}{I_z} \end{bmatrix} \tag{3.22}$$

The term $\omega_2 + \omega_4 - \omega_3 - \omega_1$ is the overall residual rotor angular velocity and will be denoted Ω, that is

$$\Omega = \omega_2 + \omega_4 - \omega_3 - \omega_1 \tag{3.23}$$

3.6 State Space Model

From equations 3.11 and 3.22 the six degrees of freedom of the quadcopter are fully described with second order equations. From these equations we can derive the state space model which is comprised of 12 equations.
The states of the system would be

$$\begin{array}{rcl} x_1 = x & | & x_2 = \dot{x_1} = \dot{x} \\ x_3 = y & | & x_4 = \dot{x_3} = \dot{y} \\ x_5 = z & | & x_6 = \dot{x_5} = \dot{z} \\ x_7 = \phi & | & x_8 = \dot{x_7} = \dot{\phi} \\ x_9 = \theta & | & x_{10} = \dot{x_9} = \dot{\theta} \\ x_{11} = \psi & | & x_{12} = \dot{x_{10}} = \dot{\psi} \end{array}$$

And the inputs to the system are

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_T(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ K_T(\omega_4^2 - \omega_2^2) \\ K_T(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \end{bmatrix} \tag{3.24}$$

Manipulating equations 3.11, 3.22 and 3.24 gives us the state space model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \\ \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} x_2 \\ (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)\frac{U_1}{m} \\ x_4 \\ (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\frac{U_1}{m} \\ x_6 \\ (\cos\theta\cos\phi)\frac{U_1}{m} - g \\ x_8 \\ \frac{1}{I_x}[(I_y - I_z)\dot{\psi}\dot{\theta} - J_r\dot{\theta}(\omega_2 + \omega_4 - \omega_3 - \omega_1) + U_2 l] \\ x_{10} \\ \frac{1}{I_y}[(I_z - I_x)\dot{\psi}\dot{\phi} + J_r\dot{\phi}(\omega_2 + \omega_4 - \omega_3 - \omega_1) + U_3 l] \\ x_{12} \\ \frac{1}{I_z}[(I_x - I_y)\dot{\theta}\dot{\phi} + U_4] \end{bmatrix} \tag{3.25}$$

It is noteworthy that the system described by the state space equation consists of two subsystems, the rotational and the translational subsystems. While the rotational subsystem does not depend on the translational subsystem, the opposite is not true and the dependency of the translational subsystem on the rotational one is obvious from the governing equations. This could ideally allow the thought of two consecutive subsystems to ease the development of a controller as shown in figure 3.3.

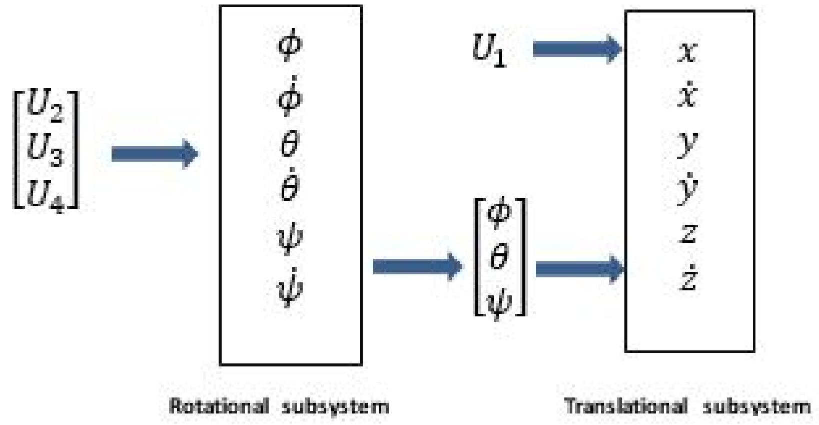

Figure 3.3: Connection of rotational and translational subsystems

Chapter 4

Sliding Mode Controller

Variable structure control (VSC) is a form of discontinuous nonlinear control. The method alters the dynamics of a nonlinear system by application of a high-frequency switching control. The state-feedback control law is not a continuous function of time; it switches from one smooth condition to another. So the structure of the control law varies based on the position of the state trajectory.

The main mode of VSC operation is sliding mode control (SMC). Sliding Mode Control, or SMC, is a nonlinear control method that alters the dynamics of a nonlinear system by application of a discontinuous control signal that forces the system to ”slide” along a cross-section of the system’s normal behavior. The state-feedback control law is not a continuous function of time. Instead, it can switch from one continuous structure to another based on the current position in the state space. Hence, sliding mode control is a variable structure control method. SMC has established itself as an effective robust control technique as robustness is inherent in this control scheme [41]. The strengths of SMC include [42]:

- Low sensitivity to plant parameter uncertainty
- Greatly reduced-order modeling of plant dynamics
- Finite-time convergence (due to discontinuous control law)
- Solution does not depend on plant parameters nor disturbance, and this is called invariance property
- Its ability to globally stabilize the system

The weaknesses of SMC include:

- Chattering due to implementation imperfections, fast-switching of controller and discretization chatter due to the fixed sampling rate [42].

Chattering is normally caused by imperfections of switching devices such as delays, dead zones or hysteresis. It also can appear as a results of small time constants of sensors and actuators neglected in the model [42].
Chattering is problematic because it can wear down the actuator and it can excite unmodeled dynamics in the system, possibly compromising performance and even stability. One can mitigate the problem of chatter by relaxing the requirement that trajectories remain within the set.
The sliding mode dynamics depend on the switching surface equation and do not depend on control. This eases the design procedure and makes it consist of two stages, first, select an equation for the manifold to achieve certain desired dynamics. Then, discontinuous control should be found so that the state would reach the manifold [42].
The design procedure of a SMC consists of two stages. First, selection of the equation of the manifold that will achieve the desired dynamics. Then, the discontinuous control should be found such that the state would reach the manifold where the sliding mode exists.

4.1 Literature Review

SMC has been applied extensively in controlling quadcopter vehicles. Alone or combined with different control techniques, SMC has been proved efficient in stabilizing the system and attenuating disturbances.
In [6], rigorous mathematics is used to develop SMC that can stabilize all Degrees of Freedom (DOFs) including those which are indirectly actuated through nonlinear coupling for a class of underactuated systems that can be put in a form featured as cascaded form with bounded external disturbances. Controller is developed, assumptions are stated and several experiments are carried out which included one applied on a quadcopter.
The quadcopter system is divided into two subsystems, fully actuated and underactuated subsystem. Sliding surfaces are developed for both subsystems based on a PID sliding surface and several simulations are performed to ensure the stability of every state. In [24], position and attitude tracking of a quadcopter is considered using two separate second order sliding mode controllers. As in [6]the model here is also divided into two subsystems, one accounting for fully actuated subsystem comprising altitude and yaw motions and the other subsystem represents the underactuated coupled dynamics in roll/pitch and position.

For the fully actuated subsystem, a sliding manifold is defined by combining the position and velocity tracking errors of one state variable (either altitude or yaw). For the underactuated subsystem, the sliding manifold is constructed via a linear combination of position and velocity tracking errors of two state variables.
Selection of sliding manifold coefficients is carried out through Hurwitz stability analysis and stability of the selected sliding surface is guaranteed through Lyapunov analysis.
Several simulations are performed and showed that the position and velocity tracking errors of all the system state variables go to zero, which means that all state variables converge eventually to their sliding surface. Controller also showed great amount of robustness against disturbances.
In [43], a second order sliding mode control (SOSM) with a focus on altitude control is developed. The control law is derived baed on Lyapunov stability theory and sliding surface is developed in resemblance with the traditional PID dynamics. Different comparison with results from the literature are shown which demonstrated that the proposed PID-based SOSM control improved the performance of quadrotor with better altitude tracking and faster convergence.
Another adaptive SOSM controller with a nonlinear sliding surface was proposed in [41]. The nonlinear sliding surface consists of a gain matrix having a variable damping ratio to better control transient settling time of the system without subjecting the system to excessive overshoot.
Merheb et al. combined sliding mode controller with sliding mode observer to develop a fault tolerant controller of a quadcopter. The observer aimed at online estimating the amount of fault injected into one of the quadcopter motors and inject this information into the reconfigurable sliding controller. SMC proved adequate degree of robustness despite fault [44].

4.2 Nonholonomic Constraints

The nonlinear coupling between states x & ϕ and states y & θ is very clear from the state space equation 3.25. The control action basically generates torque around x and/or y axis giving rotational angles θ and/or ϕ that in turn and depending on the value of *U1* moves the vehicle in x and y directions.
The nonholonomic constraints of the quadcopter arises from the fact that derivatives of the position are constrained, i.e. we cannot achieve arbitrary trajectory tracking.
From equation 3.11, multiply $\ddot{x}$ by $\cos\psi$ and $\ddot{y}$ by $\sin\psi$

$$\ddot{x}\cos\psi = (\cos^2\psi\sin\theta\cos\phi + \sin\psi\cos\psi\sin\phi)\frac{U_1}{m} \tag{4.1}$$

$$\ddot{y}\sin\psi = (\sin^2\psi\sin\theta\cos\phi - \sin\psi\cos\psi\sin\phi)\frac{U_1}{m} \tag{4.2}$$

Adding both equations yields

$$\ddot{x}\cos\psi + \ddot{y}\sin\psi = (\cos\phi\sin\theta)\frac{U_1}{m} \tag{4.3}$$

And from the $\ddot{z}$ equation we have

$$\ddot{z} + g = (\cos\phi\cos\theta)\frac{U_1}{m} \tag{4.4}$$

Dividing both equations we get

$$\tan\theta = \frac{\ddot{x}\cos\psi + \ddot{y}\sin\psi}{\ddot{z} + g} \tag{4.5}$$

Likewise, the relationship between ϕ and translational states is

$$\sin\phi = \frac{\ddot{y}\cos\psi - \ddot{x}\sin\psi}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}} \tag{4.6}$$

Position tracking of the quadcopter is achieved by calculating the rotational angles ϕ and θ from the desired position and altitude x_d, y_d and z_d.

4.3 Sliding Mode Controller

The aim of the proposed flight controller is to achieve asymptotic position and attitude tracking of the quadcopter based on Sliding Mode Control Approach. This is obtained through driving the tracking errors to zero to achieve the required tracking performance.

Based on the dynamical model in 3.6, the control system is divided in to multiple subsystems (fully actuated subsystem composed of $\ddot{z}$ and $\ddot{\psi}$, underactuated subsystem made up of $\ddot{x}$, $\ddot{y}$, $\ddot{\phi}$ and $\ddot{\theta}$).

A second order Sliding Mode Controller (SMC) is constructed using a linear combination of position and velocity tracking errors of related state variables. Then the tracking errors are driven to zero. For the underactuated subsystem, nonholonomic constraints are established between the coupled states to generate the required control action that allows the tracking of position (x,y). A schematic diagram of the proposed controller is shown in figure 4.1.

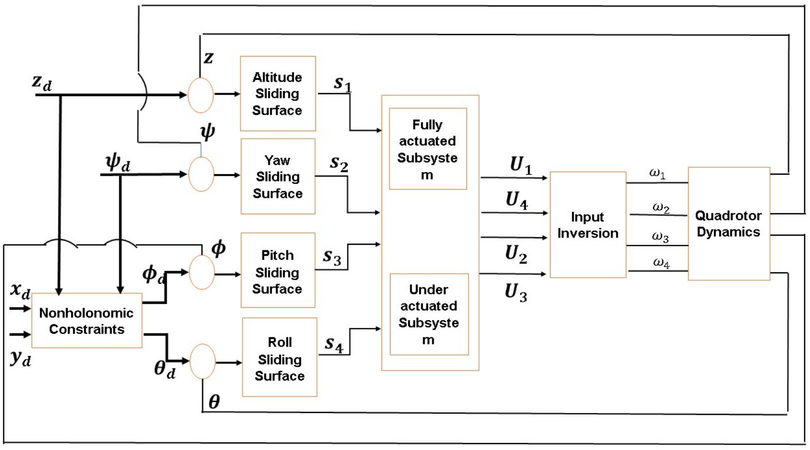

Figure 4.1: SMC flight controller architecture

4.3.1 Fully actuated subsystem controller

The objective of this controller is to ensure that subsystem state variables $[z, \psi]$ converge to the desired variables $[z_d, \psi_d]$.
The sliding manifolds for these two DOFs are:

$$s_1 = c_z(z_d - z) + (\dot{z}_d - \dot{z}) \tag{4.7}$$
$$s_2 = c_\psi(\psi_d - \psi) + (\dot{\psi}_d - \dot{\psi}) \tag{4.8}$$

where the sliding manifold coefficients $c_z, c_\psi > 0$. One of the most important aims of SMC is to keep the system trajectories on the sliding surface once they arrive to it, or in other words, SMC aims at making the sliding surface an invariant set.
To enforce the sliding mode with the desired dynamics[1], the slope of the control surfaces should follow the equation

$$\dot{s}_i = -\epsilon_i sgn(s_i) - \eta_i s_i \tag{4.9}$$

Where η_i and ϵ_i are the sliding surface exponential approach coefficients.
Differentiating equations 4.7 and 4.8 and equating the result with corresponding equation 4.9, using the fact that the dynamics while in sliding mode are governed by [45]

$$\dot{s} = 0 \tag{4.10}$$

[1] As clarified in [42], time derivative of s_i should be a discontinuous state function to enforce sliding mode on the surface. For more information, refer to [42] chapters 3 & 4

gives us the following equations

$$\dot{s_1} = -\epsilon_1 sgn(s_1) - \eta_1 s_1 = c_z(\dot{z}_d - \dot{z}) + (\ddot{z}_d - \ddot{z}) \tag{4.11}$$

$$\dot{s_2} = -\epsilon_2 sgn(s_2) - \eta_2 s_2 = c_\psi(\dot{\psi}_d - \dot{\psi}) + (\ddot{\psi}_d - \ddot{\psi}) \tag{4.12}$$

Substituting for $\ddot{z}$ and $\ddot{\psi}$ from equations 3.11 and 3.22 gives us the corresponding control laws.
Equations 4.11 and 3.11 give us the altitude control law

$$U_1 = m\frac{\epsilon_1 sgn(s_1) + \eta_1 s_1 + c_z(\dot{z}_d - \dot{z}) + \ddot{z}_d + g}{\cos\theta\cos\phi} \tag{4.13}$$

And equations 4.12 and 3.22 give us the yaw control law that would achieve $\dot{s} = 0$

$$U_4 = I_z[\epsilon_2 sgn(s_2) + \eta_2 s_2 + c_\psi(\dot{\psi}_d - \dot{\psi}) + \ddot{\psi}_d] - (I_x - I_y)\dot{\theta}\dot{\phi} \tag{4.14}$$

Equations 4.13 and 4.14 represent the control laws for the fully actuated subsystem.

4.3.2 Underactuated subsystem controller

The flight controller developed for the underactuated subsystem differs significantly from the one developed for the fully actuated subsystem. Underactuation indicates the coupling of some degrees of freedom which makes a single control input governs more than one output, and this poses difficulty to the sliding surface development.
The aim of this controller is also to ensure that subsystem state variables pairs $[x, \theta]$ and $[y, \phi]$ converge to the desired variables $[x_d, \theta_d]$ and $[y_d, \phi_d]$.
The sliding manifold for the pair $[x, \theta]$ is defined as

$$s_3 = c_1(\dot{\theta}_d - \dot{\theta}) + c_2(\theta_d - \theta) \tag{4.15}$$

The sliding manifold for the pair $[y, \phi]$ is defined as

$$s_4 = c_3(\dot{\phi}_d - \dot{\phi}) + c_4(\phi_d - \phi) \tag{4.16}$$

The time derivative of the two sliding manifolds is

$$\dot{s_3} = c_1(\ddot{\theta}_d - \ddot{\theta}) + c_2(\dot{\theta}_d - \dot{\theta}) \tag{4.17}$$

$$\dot{s_4} = c_3(\ddot{\phi}_d - \ddot{\phi}) + c_4(\dot{\phi}_d - \dot{\phi}) \tag{4.18}$$

By making the slope of the sliding surfaces equal to

$$\dot{s_i} = -\epsilon_i sgn(s_i) - \eta_i s_i \tag{4.19}$$

And equating equations 4.17 and 4.19 and equations 4.18 and 4.19, and substituting for $\ddot{\theta}$ and $\ddot{\phi}$ from equation 3.22 we get the control inputs for the underactuated subsystem

$$U_2 = \frac{I_x}{c_3 l}[c_3\ddot{\phi}_d + c_4(\dot{\phi}_d - \dot{\phi}) + \epsilon_4 sgn(s_4) + \eta_4 s_4] - A_2\dot{\psi}\dot{\theta} - J_r\dot{\theta}\Omega] \tag{4.20}$$

$$U_3 = \frac{I_y}{c_1 l}[c_1\ddot{\theta}_d + c_2(\dot{\theta}_d - \dot{\theta}) + \epsilon_3 sgn(s_3) + \eta_3 s_3] - A_3\dot{\psi}\dot{\phi} + J_r\dot{\phi}\Omega] \tag{4.21}$$

Where

$$A_2 = \frac{1}{l}[(I_y - I_z)$$

$$A_3 = \frac{1}{l}[(I_z - I_x)$$

4.4 Chattering Alleviation

The controller proposed and developed in 4.3 is proven to be stable as will be shown in the next section and achieves adequate performance as will be shown in the simulation results in chapter 5. However, this comes at a cost of chattering of the control action, and this chattering increases in the presence of disturbance and with increasing system gains.

Chattering is attributed to the fast switching of sliding mode controllers and is worsened by the digital implementation in microcontrollers with fixed sampling rates due to discretization chatter [42].

Utkin et al. described chattering as ”the phenomenon of finite-frequency, finite-amplitude oscillations appearing in many sliding mode implementations. These oscillations are caused by the high-frequency switching of a sliding mode controller exciting unmodeled dynamics in the closed loop.” [42]

Different techniques were proposed in the literature to address the chattering problem, Slotine and Li proposed the ”boundary layer solution” in which ”the control law in equation 4.9 is replaced by a saturation function that approximates the sign term in a boundary layer of the manifold $s(t) = 0$" [45]. The use of the boundary layer approach is proved useful in different texts in the literature. Although it decreases the robustness of the system, it also decreases the chattering of the input actuators which allows the increase of the controller gains, bigger controller gains result in better robustness.

4.4.1 Saturation Function

Different types of saturation functions $sat(s)$ have been proposed in the literature. The main characteristic of the $sat(s)$ is that in the vicinity of the origin, i.e. the boundary layer, $sat(s)$ is continuous. However, away from the boundary layer it exhibits the same behavior of the $sign(s)$.
In this thesis, a common type of sigmoid curves is used as the saturation function. The logistic curve is an S-shaped function that has the form

$$f(s) = \frac{L}{1 + e^{-k(s-s_o)}} \tag{4.22}$$

Where, s_o is x-value of the sigmoid's midpoint, k is the steepness of the curve and L is the maximum value of the curve. Typical logistic function is shown in the figure 4.2.

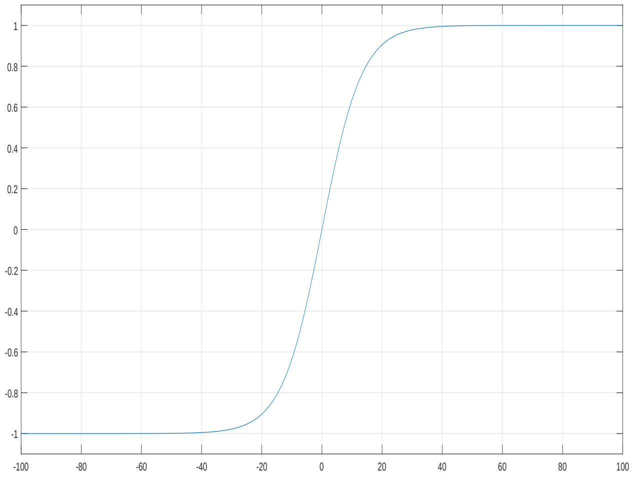

Figure 4.2: Typical Logistic function

4.4.2 Discontinuous to Continuous Control Law

The boundary layer approach in SMC changes the control law from discontinuous to continuous. Discontinuity of the proposed control law is the main source of chattering.
Changing the *sign* function with the *saturation* function in equations 4.13, 4.14, 4.20 and 4.21 gives us the following equations.

$$U_1 = m\frac{\epsilon_1 sat(s_1) + \eta_1 s_1 + c_z(\dot{z}_d - \dot{z}) + \ddot{z}_d + g}{\cos\theta\cos\phi} \tag{4.23}$$

$$U_4 = I_z[\epsilon_2 sat(s_2) + \eta_2 s_2 + c_\psi(\dot{\psi}_d - \dot{\psi}) + \ddot{\psi}_d] - (I_x - I_y)\dot{\theta}\dot{\phi} \tag{4.24}$$

$$U_2 = \frac{I_x}{c_3 l}[c_3\ddot{\phi}_d + c_4(\dot{\phi}_d - \dot{\phi}) + \epsilon_4 sat(s_4) + \eta_4 s_4] - A_2\dot{\psi}\dot{\theta} - J_r\dot{\theta}\Omega] \tag{4.25}$$

$$U_3 = \frac{I_y}{c_1 l}[c_1\ddot{\theta}_d + c_2(\dot{\theta}_d - \dot{\theta}) + \epsilon_3 sat(s_3) + \eta_3 s_3] - A_3\dot{\psi}\dot{\phi} + J_r\dot{\phi}\Omega] \tag{4.26}$$

4.5 Stability Analysis

Theorem: the flight controller described by equations 4.13, 4.14, 4.20 and 4.21 renders the nonlinear quad rotor system described by equation 3.25 stable in Lyapunov sense.
Proof: The Lyapunov candidate function for each control actions is described by the strictly positive function

$$V_i = \frac{1}{2}s_i^2 \qquad (i = 1, 2, 3, 4) \tag{4.27}$$

The derivative of equation 4.27 is

$$\dot{V}_i = s_i\dot{s}_i \tag{4.28}$$

Invoking equation 4.9 into equation 4.28 we get

$$\dot{V}_i = -\epsilon_i|s_i| - \eta_i s_i^2 \leq 0 \tag{4.29}$$

Which is negative semi-definite for all s_i, therefore, the control action drives the origin of each subsystem to stability.
Thus under the control law U_i (i = 1,2,3,4), all system state trajectories can reach and stay on the corresponding sliding surface.

4.6 Input Inversion

The control equations developed in section 4.3 and the chattering alleviation technique developed in section 4.4 outputs the forces that should be generated from each motor to achieve the required characteristics.
These forces should be transformed to their corresponding value of the required rotational speed from each motor. In equation 3.24 we have the forces as functions of motors' RPM.
After obtaining the input forces, these should be transformed to the required motor rotational speed so the flight controller sends to each motor.
Transformation of equation 3.24 to get U's as functions of $\omega' s$ is mandatory for feeding the flight controller with the correct input to the motors.

Inversion of equation 3.24 yields equation 4.30.

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} \sqrt{\omega_2^2 + 0.5(\frac{U_2 - U_3}{K_t} + \frac{U_4}{d})} \\ \sqrt{0.25(\frac{U_1 - 2U_2)}{K_t} - \frac{U_4}{d})}; \\ \sqrt{\omega_1^2 + \frac{U_3}{K_t}} \\ \sqrt{\omega_2^2 + \frac{U_2}{K_t}} \end{bmatrix} \tag{4.30}$$

Chapter 5

Simulation Results

Different simulations are carried out on the proposed control techniques to ensure the developed controller drives the system along the desired manifold and achieves required performance.

A quick representation of the results associated with the implementation of the Proportional plus Derivative (PD) control are also presented along with the control laws themselves. The PD control results are used to compare with the results obtained from the PD controller that comes pre-installed on the flight controller as illustrated in section 7.3.

Simulations are carried out using Simulink®. Values of mass and moments of inertia of the quadrotor used in the simulations are the real values of the assembled vehicle as illustrated in section 6.11.

Comparisons and conclusion are shown at the end of the chapter.

5.1 PD with no Disturbance

A PD controller is developed with the following input equations

$$U_1 = K_{p_1}(z_d - z) + k_{d_1}(\dot{z}_d - \dot{z}) \tag{5.1}$$

$$U_2 = K_{p_2}(\phi_d - \phi) + k_{d_2}(\dot{\phi}_d - \dot{\phi}) \tag{5.2}$$

$$U_3 = K_{p_3}(\theta_d - \theta) + k_{d_3}(\dot{\theta}_d - \dot{\theta}) \tag{5.3}$$

$$U_4 = K_{p_4}(\psi_d - \psi) + k_{d_4}(\dot{\psi}_d - \dot{\psi}) \tag{5.4}$$

The above equations are used to drive the quadcopter system to stability [14]. The simulations results obtained from the PD control will be used to compare the results between linear and nonlinear control techniques. Schematic of the PD control architecture is shown in figure 5.1.

The PD controller is chosen because unlike the PID it does not result in a

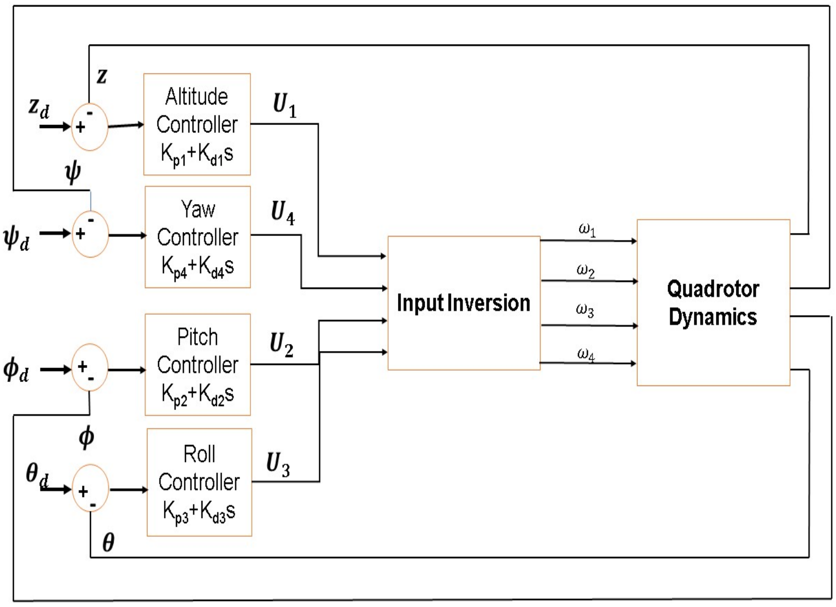
z
z_d
+
-
Altitude Controller K_p1+K_d1s
U_1
ψ
ψ_d
+
-
Yaw Controller K_p4+K_d4s
U_4
ω_1
ω_2
ω_3
ω_4
Input Inversion
Quadrotor Dynamics
φ_d
+
-
Pitch Controller K_p2+K_d2s
U_2
φ
θ_d
+
-
Roll Controller K_p3+K_d3s
U_3
θ

Figure 5.1: Quadrotor PD control architecture

higher order controller which in turn requires higher accelerations from the actuators which needs more power.

Table 5.1 shows the gains used with the PD controller [14].

To better assess the performance of a PD control, two simulations are carried out with the presence of disturbance and two without disturbance. The first simulation has mild inputs without any abrupt changes of fast dynamics, the second simulation has more aggressive inputs and fast changing dynamics to better compare with the SMC.

Table 5.1: Simulation Gains for PD controller

Gain	**Value**	**Gain**	**Value**
K_{p_1}	156.8	k_{d_1}	78.2
K_{p_2}	44.5	k_{d_2}	30.5
K_{p_3}	44.5	k_{d_3}	30.5
K_{p_4}	33.9	k_{d_4}	20.7

5.1.1 Slow-rate Input

The first simulation results are shown in figures 5.2 and 5.3. Figure 5.2 shows the mild input signals to the PD controller, the input is mildly changing with no abrupt or fast alterations. The tracking performance of the quadrotor is also shown in the same figure.

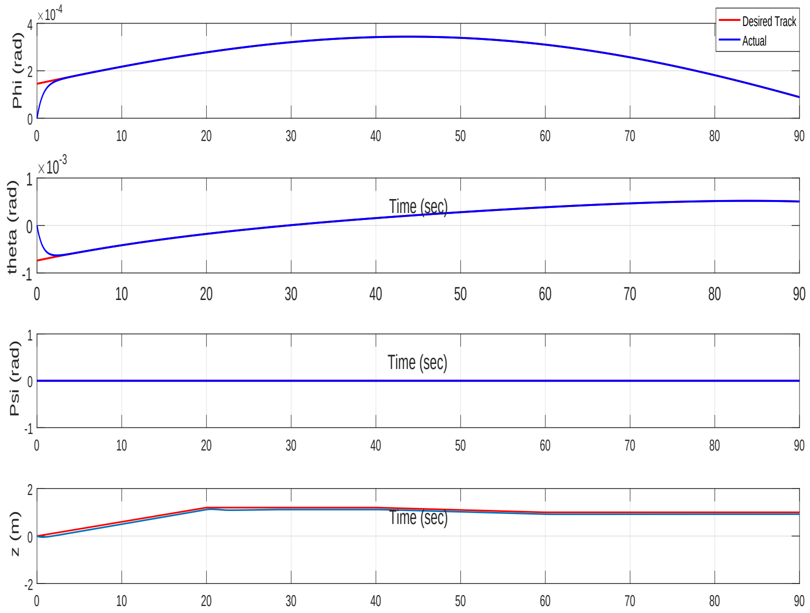

Figure 5.2: PD control attitude and altitude tracking - slow-rate input

Fig. 5.3 shows the x and y coordinates. x and y are not inputs to the quadrotor equations, but rather they are dynamically coupled with both θ and ϕ respectively. Fig. 5.2 shows the input tracking of the attitude angles and altitude. The figure shows that the PD controller was able to track both ϕ and θ angles with settling time less than 5 seconds. The altitude tracking also showed good performance despite the persistent steady state error that followed the whole trajectory.

The input action of the PD control is shown also in figure 5.4. As can be shown, the inputs of a PD controller did not undergo chattering or increased control effort due to the smooth nature of the input and the slow dynamics accompanied with the attitude trajectory, however, the input for yaw moment U_4 suffers from some oscillations around zero, and this is normal due to the coupled dynamics of the system, so any change in U_2 or U_3 is accompanied

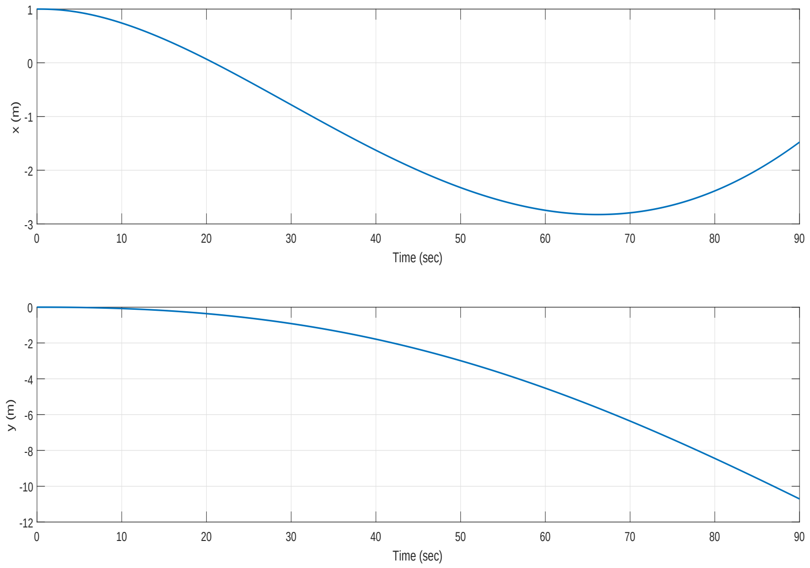

Figure 5.3: x and y coordinates for PD control - slow-rate input

with a change in U_4.
The ability of the PD control to track the input signals when the input signal is smooth and does not undergo any abrupt changes is clear.

5.1.2 Fast-rate Input

A fast-changing input is fed into the system to assess the tracking capabilities of the PD controller. The tracking results are shown in Fig. 5.5. The fast changing inputs show the limitations of the linear PD control for a nonlinear system with coupled dynamics like the quadrotor. For fast-changing input, the PD controller failed to perfectly track the input signals and could not comply with the fast dynamics.
Fig. 5.6 shows the x and y trajectories of the quadrotor.
Fig. 5.7 shows the control actions of the actuators for the input signals shown in fig. 5.5. The difference in control action between the slow dynamics and fast-changing dynamics is very clear from fig. 5.7 and fig. 5.4. The fast-changing dynamics resulted in an increased control effort for all inputs, increased chattering and higher RPM for the motors accordingly. The effect of the aggressive signals is very clear with U_2 and U_3 specifically because both input ϕ and θ have changed dramatically, while it is less for U_1 because the altitude input did not change as much.

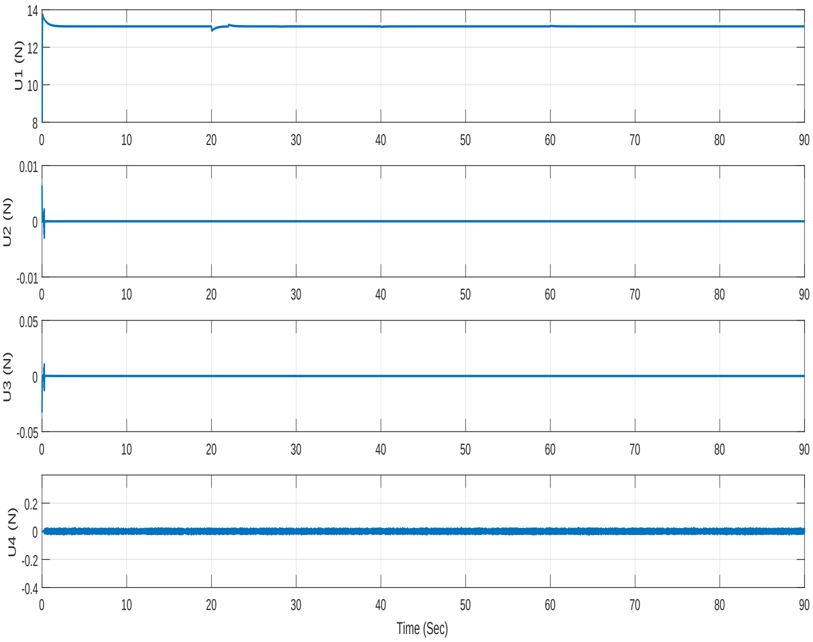

Figure 5.4: PD control actions - slow-rate input

5.2 PD with Disturbance

The PD controller is subjected to disturbance in the vertical direction (z-axis) to check the robustness of the controller and to compare its output with the results obtained from the SMC shown in section 5.6.
The quadrotor is subjected to variable disturbance in the vertical direction. The disturbance could resemble sudden weight change of wind gust that acts in the direction of z-axis.
The disturbance is fed into the state equations such that the $\ddot{x}_6$ becomes

$$\ddot{x}_6 = \ddot{z} = (\cos\theta\cos\phi)\frac{U_1}{m} - g + Dist. \tag{5.5}$$

The disturbance term changes with time as depicted in Fig. 5.8.
Controller gains are the same of table 5.1.
The same steps followed for PD control without the presence of disturbance are done in the presence of disturbance. Simulation with smooth signal is carried out then fast-changing inputs are introduced and both are in the presence of the same disturbance signal shown in fig. 5.8.

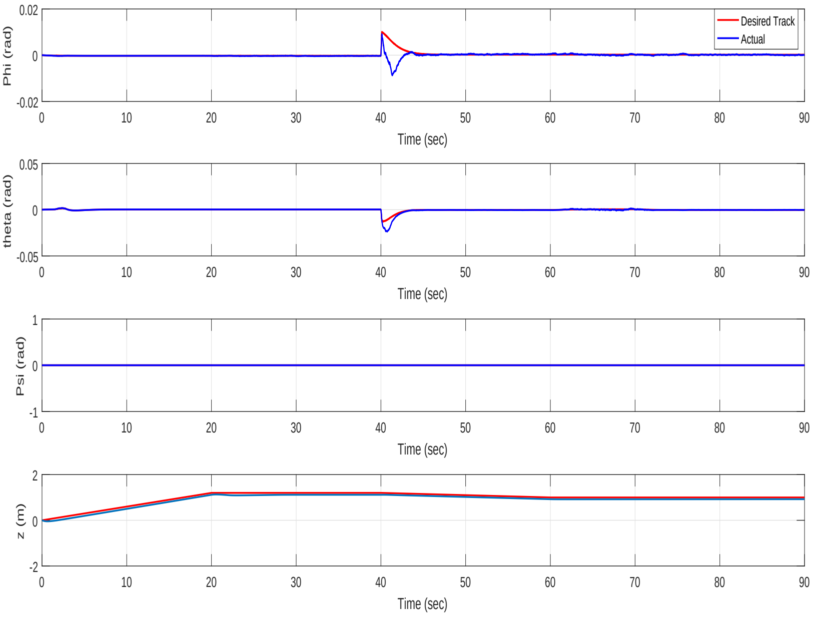

Figure 5.5: PD control attitude and altitude tracking - fast-rate input

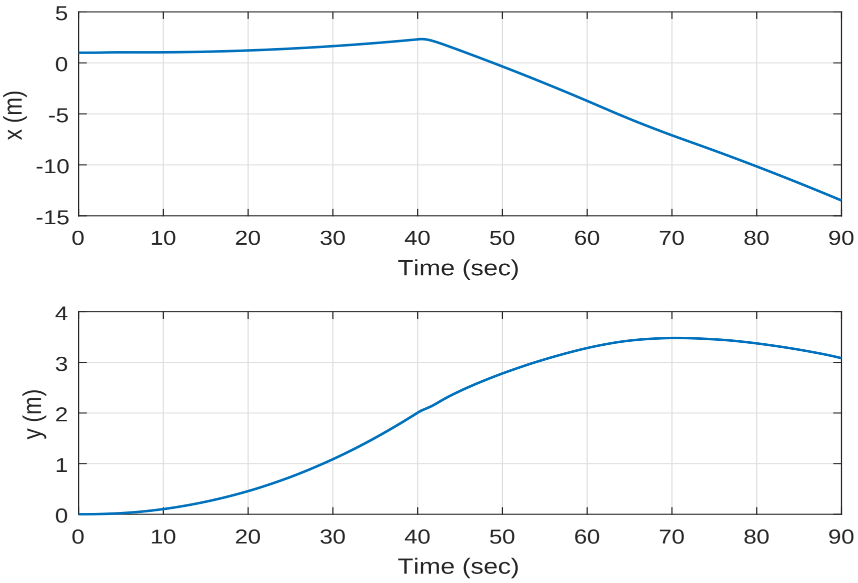

Figure 5.6: x and y coordinates for PD control - fast-rate input

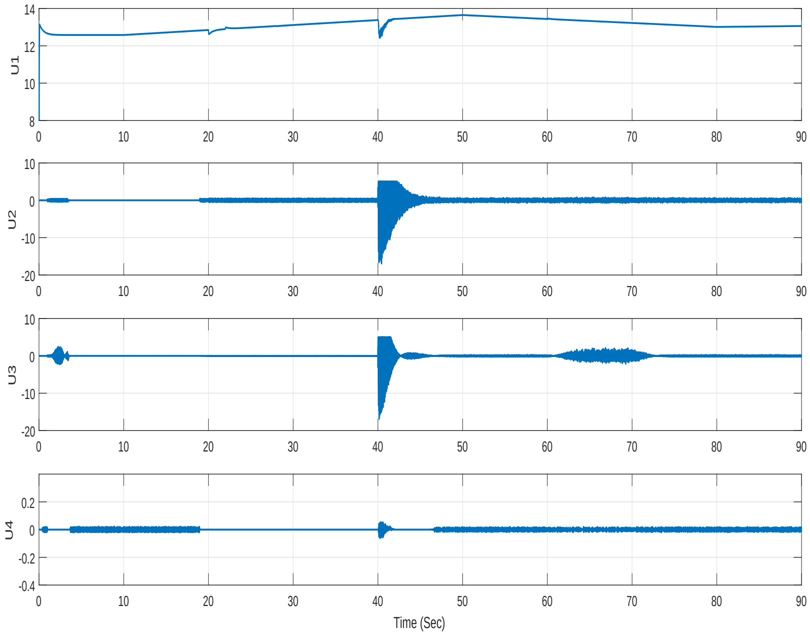

Figure 5.7: PD control actions - fast-rate input

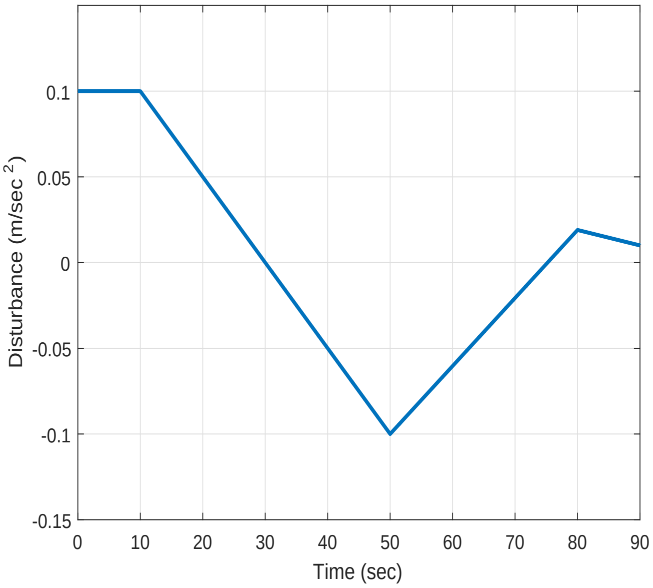

Figure 5.8: Disturbance Variation with Time

5.2.1 Slow-rate Input

The smooth input signals are sent to the quadrotor system, fig. 5.9 shows the results obtained with the existence of the disturbance signal. And it is clear that the disturbance in the z-direction increased the steady-state error of the altitude tracking while having no direct effect on the attitude angles. This results is logical from the equations of motions as there is no coupling between the altitude subsystem and attitude subsystem as illustrated in sections 4.3.1 and 4.3.2.

The results show a 10% increase in both steady state error in the altitude tracking and the corresponding input action. Fig. 5.10 shows the input action for PD control with the presence of disturbance.

It is clear that the PD control showed a degree of robustness against disturbances as long as the input dynamics are not fast. It adequately achieved input tracking with and without disturbance without significantly increasing the control effort. The PD control with smooth input also did not show any chattering or increased control effort.

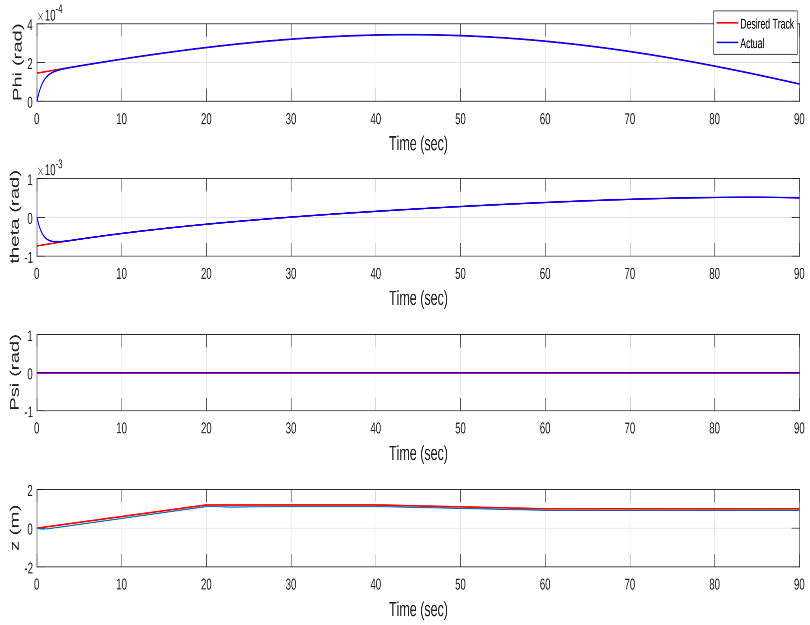

Figure 5.9: PD control attitude and altitude tracking with disturbance- slow-rate input

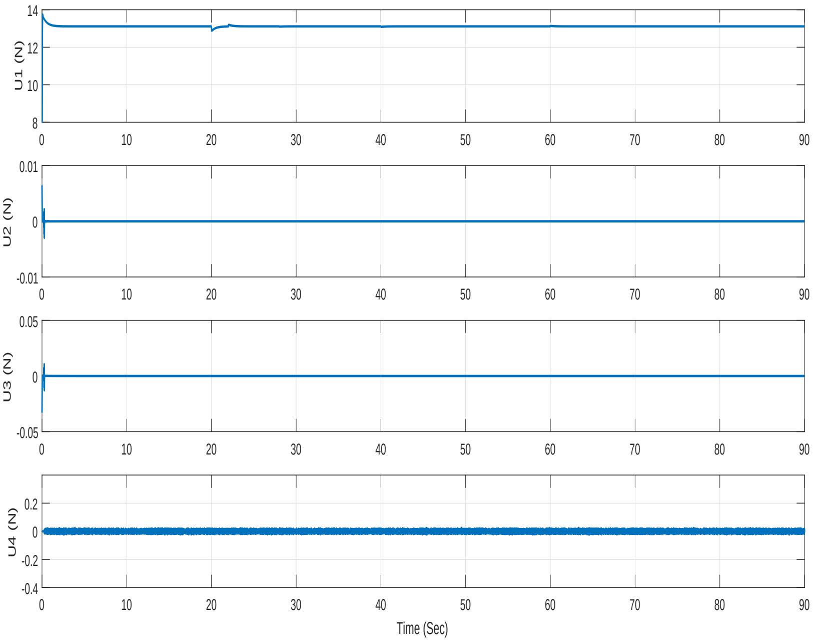

Figure 5.10: PD control actions - slow-rate inputs with disturbance

5.2.2 Fast-rate Input

The response to fast-changing inputs with the presence of disturbance is shown in this section. Fig. 5.11 shows inputs tracking. It is clear that the disturbance affected the altitude tracking performance.

Fig. 5.12 shows the x and y coordinates that resulted from the tracking of both θ and ϕ respectively.

Fig. 5.13 shows the input actions associated with the input signals and the present disturbance. The figure shows that the input actions did not significantly increase due the presence of disturbance, rather the effect of the fast-changing input was more overwhelming.

Like the case of no disturbance in section 5.1.2, the fast-changing dynamics resulted in a more aggressive input action, chattering and increased RPMs from the motors.

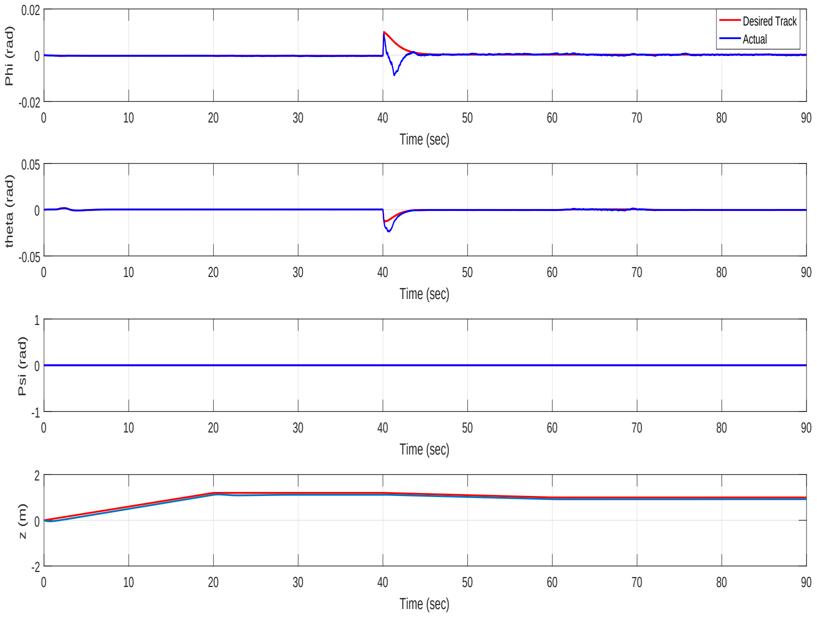

Figure 5.11: PD control attitude and altitude tracking with disturbance-fast-rate input

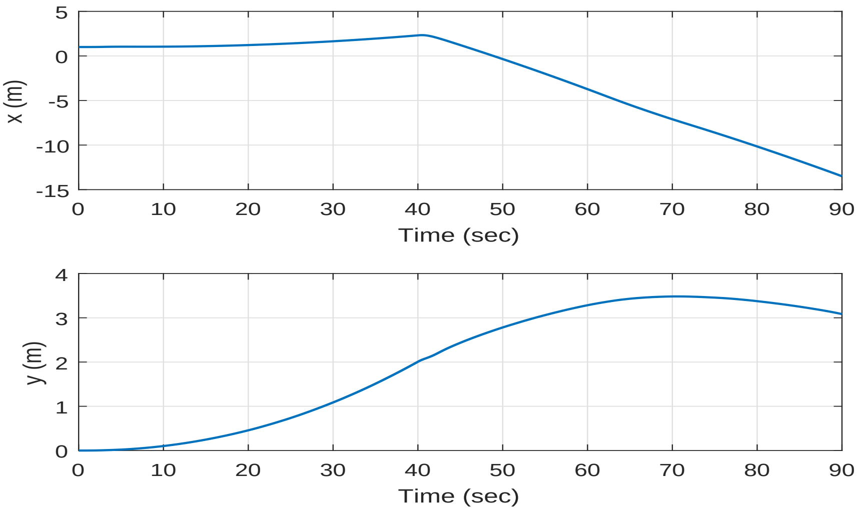

Figure 5.12: x and y coordinates for PD control - fast-rate input with disturbance

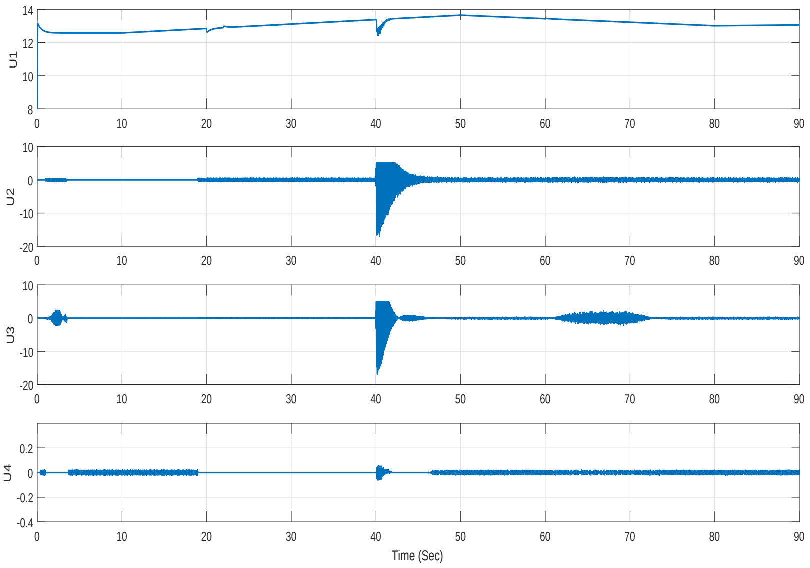

Figure 5.13: PD control actions - fast-rate input with disturbance

5.3 Conclusion on PD Simulation

It has been clear that the PD control was able to track the input signals of a slow-changing input, even in the presence of disturbance. This would help make the quadrotor follow a pre-defined trajectory without aggressive control effort. Fig.5.14 shows the 3D track of the quadrotor with PD controller. It shows that the capability of the PD control with the slow-changing signal.

However, this is not the case with a fast-changing input where the PD controller failed to achieve the required tracking performance. Also, the presence of disturbance aggravated the performance and resulted in more diversion from the input signal as shown in Fig. 5.15.

5.4 SMC with No Disturbance

In this section, simulation of the controller developed in section 4.3. The developed controller is a SMC with *signum* function. Results from the literature show chattering as one of the main drawbacks of the traditional SMC. Controller gains used in the simulations are shown in table 5.2.

Unlike the PD control simulation results shown in sections 5.1 and 5.2 consecutively, the nonholonomic constraints developed in section 4.2 are used. Nonholonomic constraints generate both attitude angles θ and ϕ from both

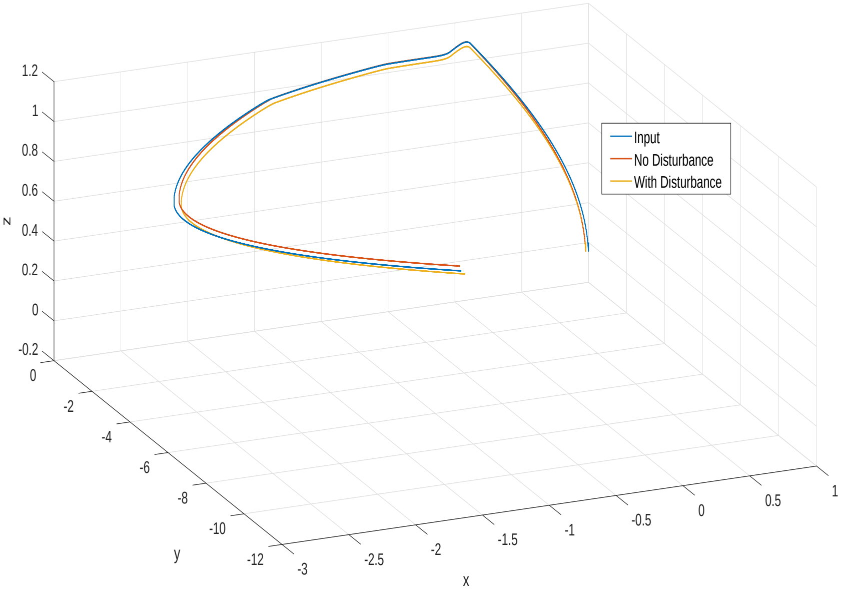

Figure 5.14: 3D track of the quadrotor - slow-rate input - PD control

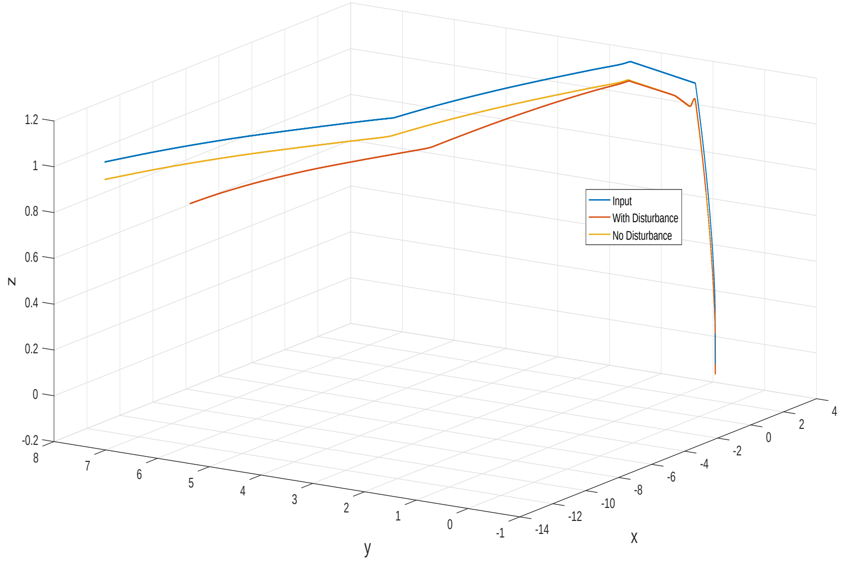

Figure 5.15: 3D track of the quadrotor - fast-rate input - PD control

x and y. The signals fed into the system are fast-changing.
Simulation results are shown in Fig. 5.16, 5.17 and 5.18. Fig. 5.16 shows

Table 5.2: Simulation Gains for SMC

Gain	**Value**	**Gain**	**Value**
ϵ_1	0.1	η_3	0.25
η_1	0.02	c_3	1.1
c_z	1.2	c_4	1.25
ϵ_2	0.6	ϵ_4	0.2
η_2	0.04	η_4	0.15
c_ψ	2	c_7	1.2
ϵ_3	0.28	c_8	1.3

the trajectory tracking under the developed control law with no present disturbance. It shows good tracking of system states with the presence of slight overshoot. Overshoot is attributed to low values of gains used in the controller. Chattering increases with increased gains, that is why low gain values where used.

The SMC was able to follow both θ and ϕ with their fast dynamics to allow the tracking of both x and y.

Fig. 5.17 shows the control action of vehicle's actuators, the presence of chattering is obvious despite the use of low values of gains.

Fig. 5.18 shows the desired 3D trajectory against the real one.

It is clear from the simulation results that the proposed control achieves the required trajectory tracking with no presence of disturbance. The enhanced performance compared to the PD control comes at a cost of chattering that limits the capability of the controller and which was not present with PD control.

5.5 SMC with Chattering Alleviation

The boundary layer method in sliding mode described in section 4.4 is used to alleviate the chattering through the developed control scheme.

Controller gains used in the simulations are shown in table 5.3.

Simulation results of the SMC with boundary layer approximation are shown in Fig. 5.19-5.21.

Fig. 5.19 shows trajectory tracking with no present disturbance. Unlike the performance shown in Fig. 5.16, tracking of states with chattering alleviation shows damped overshoot due to the higher gains used. This is attributed to the use of the *Saturation* function.

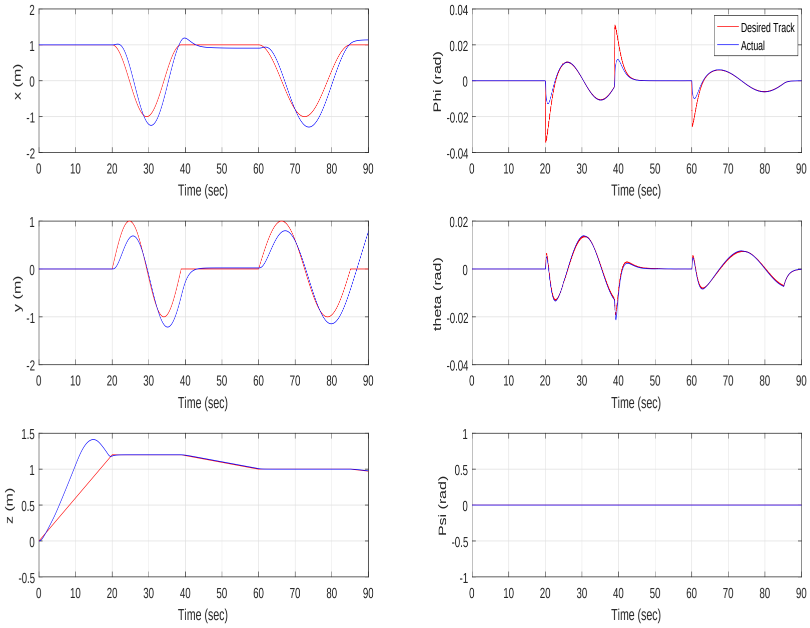

Figure 5.16: SMC control attitude and altitude tracking - no disturbance

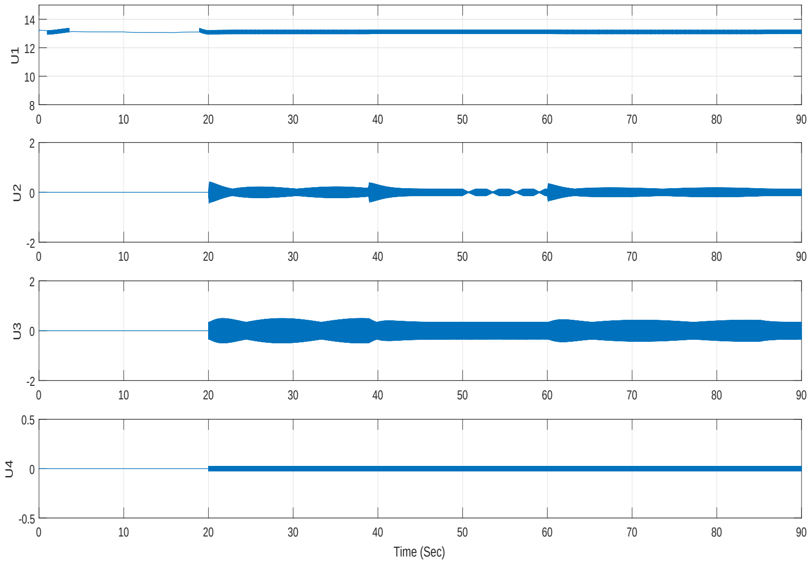

Figure 5.17: Control action of sliding mode control - no disturbance

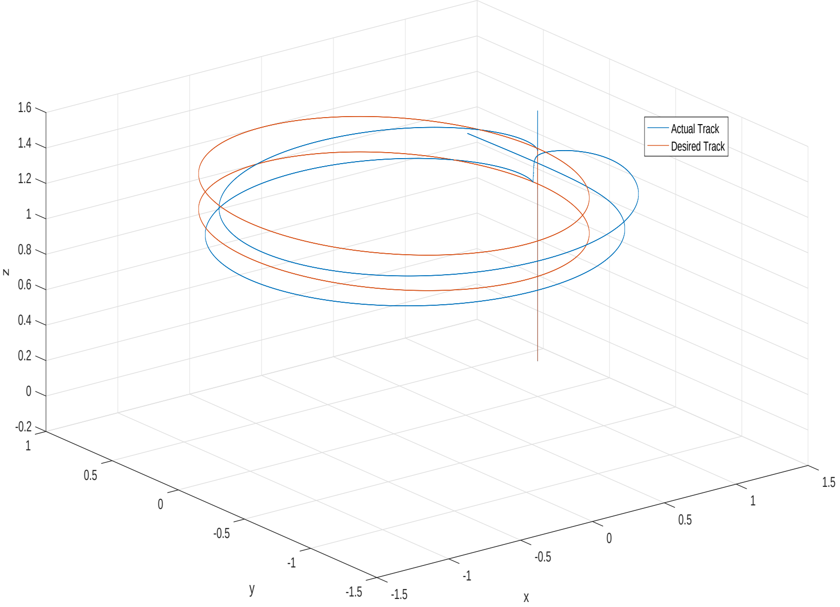

Figure 5.18: SMC control 3D track - no disturbance

Table 5.3: Simulation Gains for SMC with Chattering Alleviation

Gain	**Value**	**Gain**	**Value**
ϵ_1	2.2	η_3	0.28
η_1	0.4	c_3	1.3
c_z	3.0	c_4	1.45
ϵ_2	1.1	ϵ_4	2.4
η_2	0.04	η_4	0.22
c_ψ	2	c_7	1.35
ϵ_3	2.5	c_8	1.4

Fig. 5.20 shows the control action of the quadrotor's actuators. Unlike the actuator action shown in Fig.5.17, the presence of chattering is diminished despite the use of higher values of gains. Weak chattering is only present when the vehicle undergoes aggressive maneuvers.
Fig. 5.21 shows the desired 3D trajectory against the real one.
It is clear from the simulation results that the boundary layer technique pro-

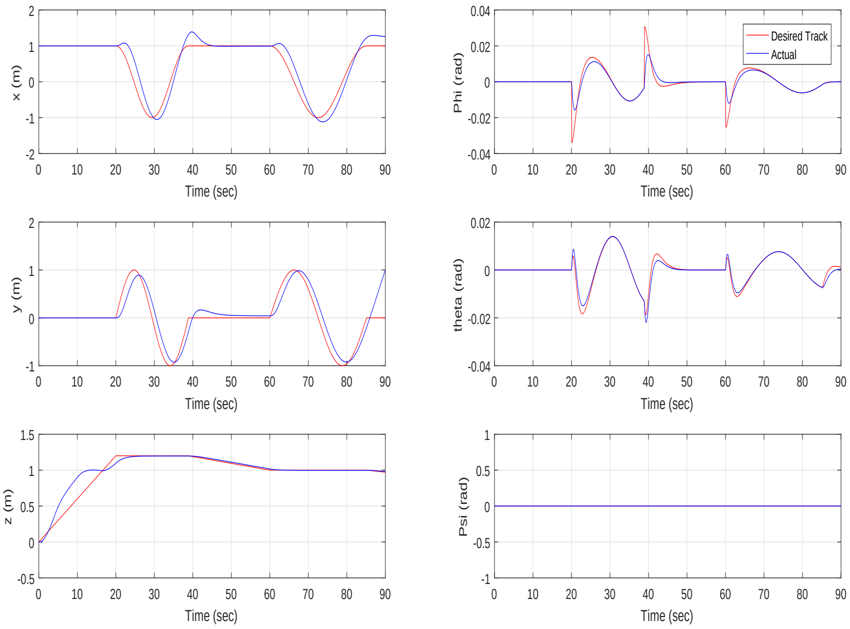

Figure 5.19: SMC control attitude and altitude tracking - chattering alleviation

posed by Slotine and Li significantly reduces chattering problem in SMC. It also allowed the use of higher gains which enhanced tracking performance and damped overshooting [45].

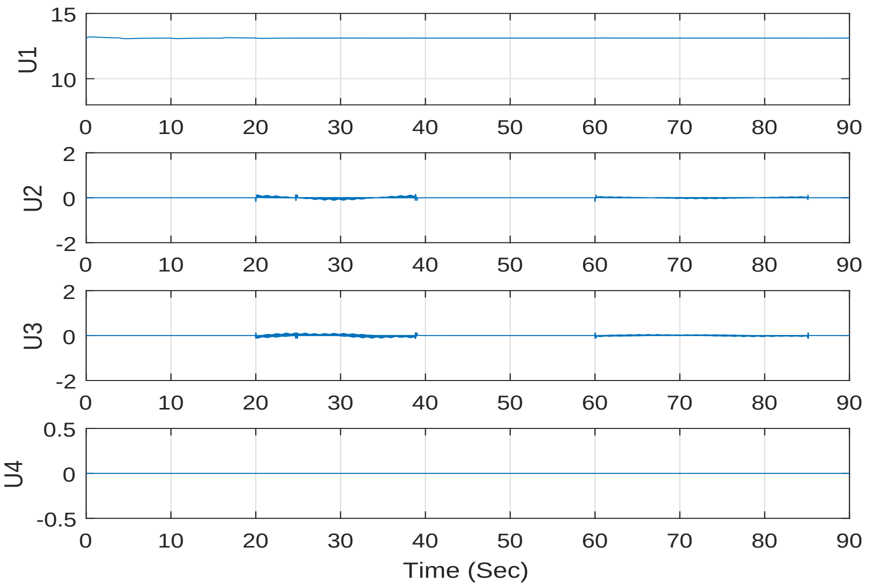

Figure 5.20: Control Action of Sliding Mode with Chattering Alleviation

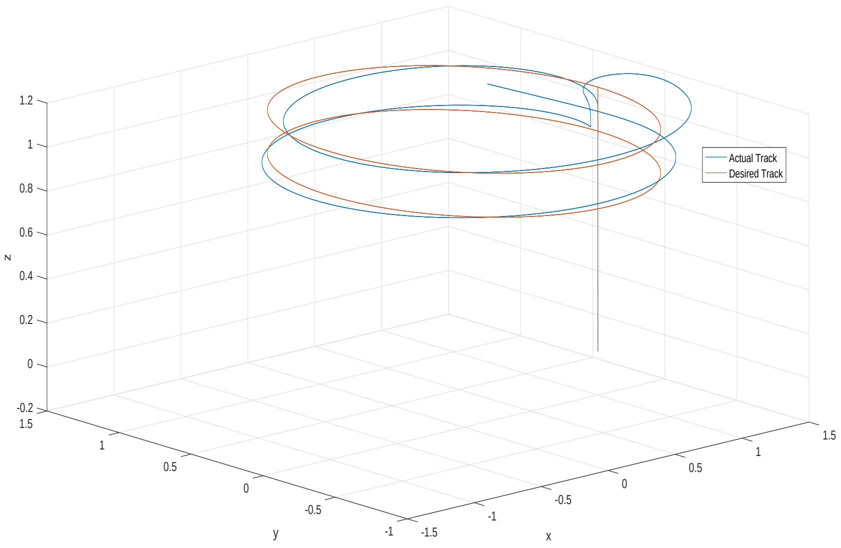

Figure 5.21: 3D track for SMC with chattering alleviation

The boundary layer approach with SMC allowed the use of higher gains which in turn damped overshoots and enhanced the tracking performance.

5.6 SMC with Chattering Alleviation in the Presence of Disturbance

The proposed SMC is subjected to disturbance in the z-direction shown in figure 5.8. This is the same disturbance subjected to the PD controller in section 5.2.

Simulations are carried out using controller gains presented in table 5.3. Results are shown in Fig.5.22-5.24.

Fig. 5.22 shows trajectory tracking in the presence of disturbance. The plot shows the robustness of the controller against the disturbance affecting the system. The controller was able to drive the trajectories along the desired track and damp overshoot.

Robustness of the controller is obvious from Fig. 5.22, in which the controller was able to get the states to follow the desired trajectory despite the presence of variable disturbance and maintain its track. Both ϕ and θ had fast dynamics and abrupt changes, however, the SMC exhibited resilience and ability to maintain the output on the same trajectory.

Actuator actions are shown in Fig. 5.23 which also proves the robustness of

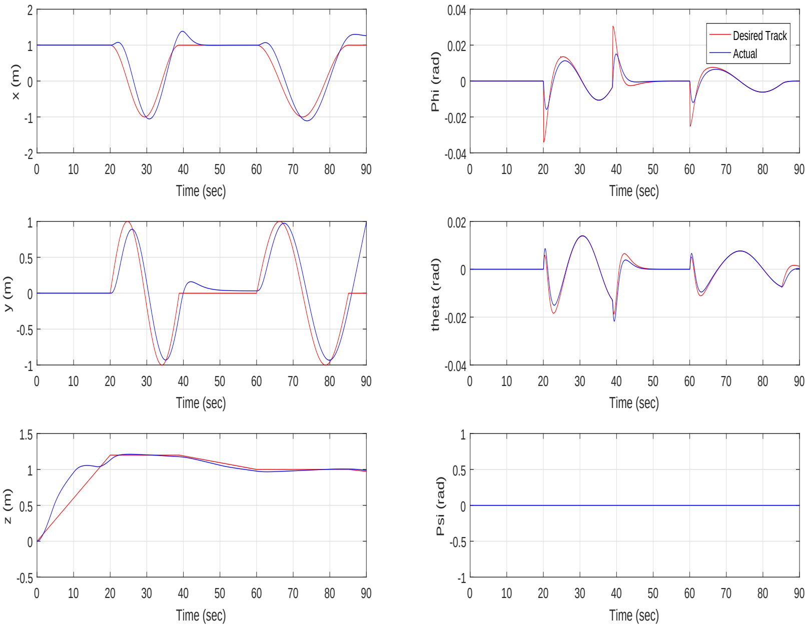

Figure 5.22: SMC control attitude and altitude tracking - chattering alleviation with disturbance

the system and the effectivity of the boundary layer approach in sliding mode control. The control actions, and despite the presence of disturbance, were not greatly affected either in magnitude of force applied or in chattering.
Fig. 10 shows 3D tracking in the presence of disturbance.

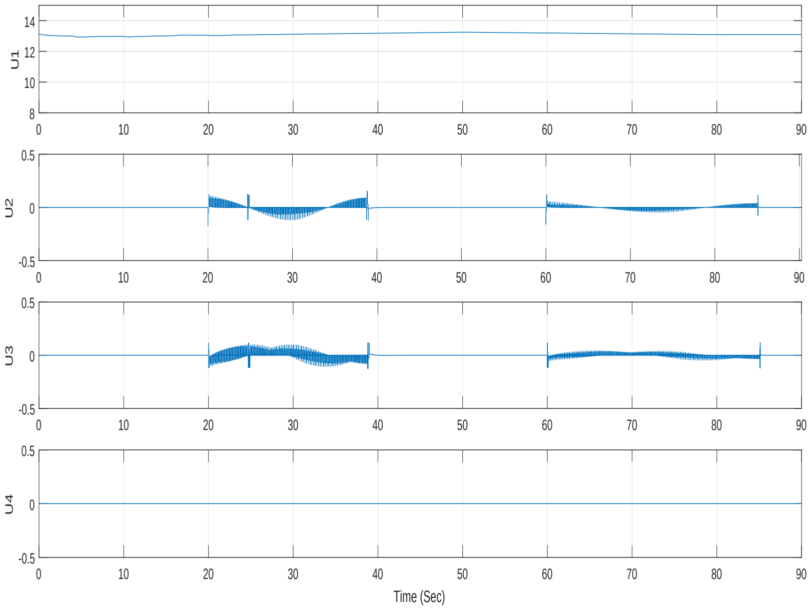

Figure 5.23: Control actions of SMC with Disturbance

Simulation results of SMC with boundary layer approach prove the adequacy and robustness of the developed controller. It was able to drive the vehicle along the desired track, damp overshoot and alleviate the effect of sudden variable disturbance.
Using the *Saturation* function instead of *Signum* function in the controller enhanced the tracking performance by allowing the use of higher gains which damped the overshoot while significantly lowering the chattering in the control action.
Fig. 5.25-5.26 show the tracking performance of a more dynamic trajectory to highlight the robustness of the controller. In this simulation, a constant change in altitude in the presence of disturbance accompanies the movement in both x and y directions.
The simulation results in the presence of disturbance show the robustness and reliability of the used chattering alleviation technique. Comparison of the results with the ordinary SMC shows the advantages of using *Saturation* function over *Signum* function.

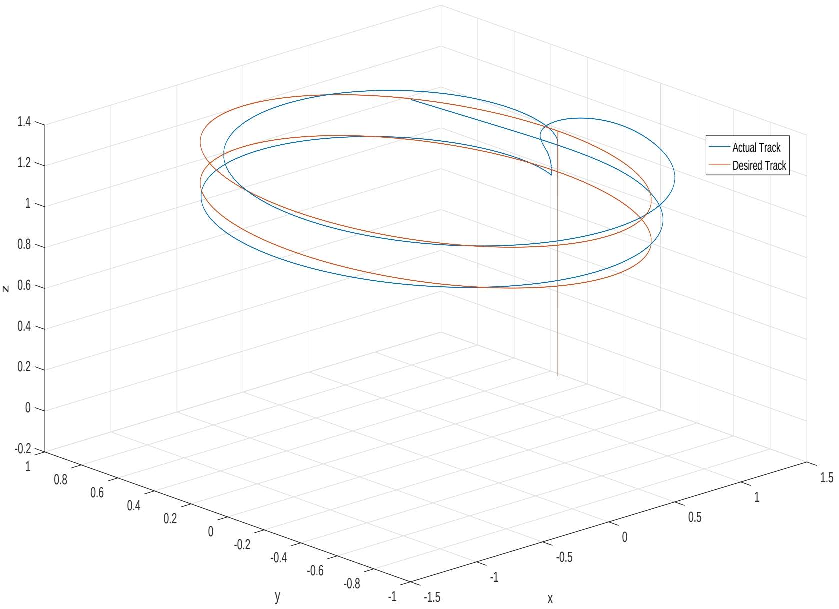

Figure 5.24: 3D track of SMC control - with disturbance

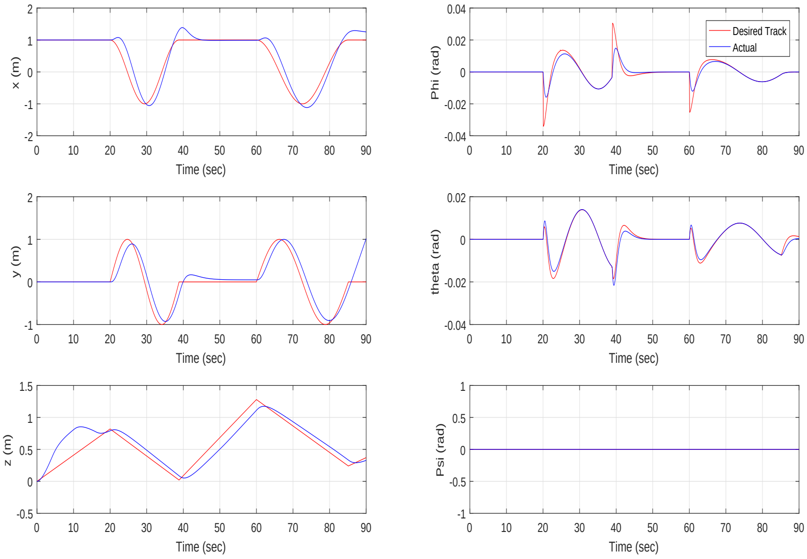

Figure 5.25: Tracking in the Presence of Disturbance - 2nd Attempt

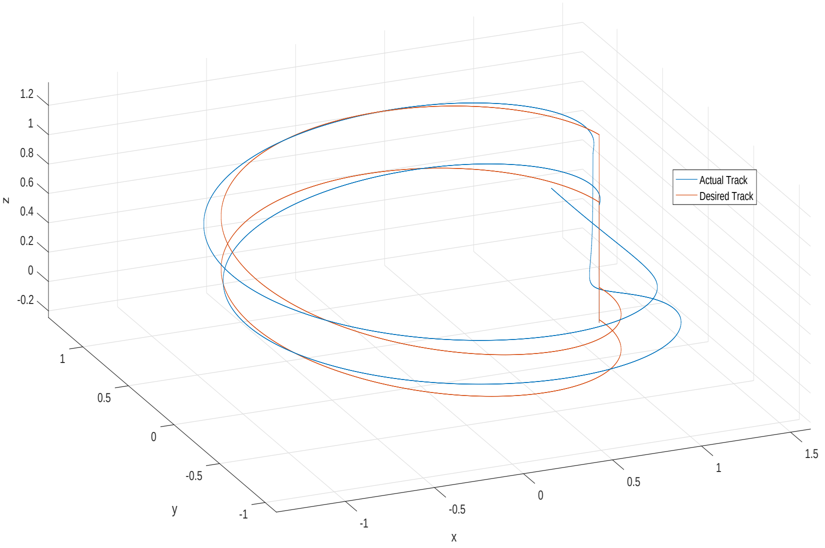

Figure 5.26: 3D Tracking in the Presence of Disturbance - 2nd Attempt

5.7 Conclusion

In this chapter, different simulations are carried out with different conditions. First, PD control results are shown with and without disturbance for both a slow changing input signal and a fast-changing input signal. The PD control showed its ability to achieve the required tracking performance with the slow-changing signals and showed good robustness against the injected disturbance. However, with fast-changing signal it did not show the same performance.

SMC simulations are carried out with the presence of chattering and with chattering alleviation technique used, both with disturbance and without.

Unlike the PD control, the signals fed to the SMC simulations are rather aggressive and fast-changing which the SMC was able to deal with the track the required trajectory.

SMC suffered from aggressive input effort and chattering, this was alleviated by using the boundary layer technique shown in section 5.5 and section 5.6, the use of chattering alleviation allowed the use of higher gains which in turn damped the overshoot and enhanced performance without increasing the control effort significantly.

Chapter 6

Quadrotor Hardware Development

In this chapter the design requirements of the quadrotor are going to be presented and the design problem will be broken down along with the anatomy of a quadrotor that will be explained, and what each component does in a quadrotor.
Various commercial off-the-shelf components are used. They are going to be presented along with their characteristics.

6.1 Design Requirements

The aim of the hardware implementation is to assemble a quadrotor vehicle from commercial components that can achieve the required performance.
The requirements for hardware implementation includes:

1- Build a quadrotor that is capable of carrying its own weight plus 500 gm payload

2- Selection of motors and propellers that can provide sufficient thrust to accommodate the required control action

3- Selection of a battery that is able to power the components of the quadrotor for a flying time not less than 20 minutes

4- Incorporation of a Radio Control (RC) that provide sufficient range (not less than 600 m)for manually controlling the vehicle

5- Provide means for reading sensors data and comparing with actual inputs to test and validate the designed control technique

To achieve the aforementioned design requirements, a comprehensive process of component selection was undertaken. The selection criteria of the used components are illustrated in the following sections. Every section shows one used component along with its specifications as follows:

- the quadrotor's frame
- the motors
- the electronic speed controller
- the propellers
- the battery
- the flight controller
- the GPS and compass
- the RC
- the telemetry

6.2 Frame

Quadcopter frame is the basic structure that carries all components, controller and RC. It shall have adequate space to accommodate all required components and sensors, it shall also allow for the installation of propellers that are able to generate the required thrust. For Field Point View (FPV) photography and future development, it is preferable to have a facility to install a gimbal and camera mounting.

The used frame is the Quanum Venture FPV Quadcopter. A commercial H-shaped quadcraft. This model comes with pre-installed mount/power distribution board, 20A Afro Electronic Speed Controllers and TURNIGY Multistar brushless DC motors.

Vehicle dimensions allow the use of up to 10 inch propellers and the ESC characteristics can accommodate big BLDC motors. This give the frame the capacity for higher thrust generation [46].

The vehicle is also equipped with universal gimbal mounting and tall skids as an option. Fig. 6.2 shows a photo of the assembled frame.

The vehicle comes with most of the setup has been already done. It only needs the installation of the flight controller, battery, RC and FPV if required.

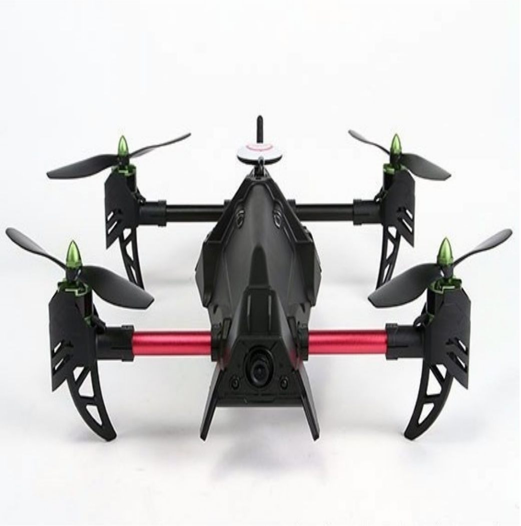

Figure 6.1: Quanum Venture FPV Quadcopter [46]

Specifications of the frame is shown in table 6.1.

Table 6.1: Specifications of the Quanum Venture FPV Quadcopter

Weight	440 grams
Max prop	8-10 inches
Power Distribution	4 Afro 20A ESC pre-installed with XT60 power lead
Wheelbase	430 mm
Motors	2213 935KV Multistar Motors
Length	495 mm
Width	170 mm

The moments of inertia of the assembled vehicle are provided, they do not include the effects of the battery, telemetry and the flight controller.Their values are shown in table 6.2.

Table 6.2: Quadrotor Characteristics

Symbol	Value	Description
m	1.06 kg	total mass of the vehicle and installed components
I_{xx}	$7.14 * 10^{-3} kg.m^2$	Moment of inertia around x-axis without battery and flight controller
I_{yy}	$7.55 * 10^{-3} kg.m^2$	Moment of inertia around y-axis without battery and flight controller
I_{zz}	$1.24 * 10^{-2} kg.m^2$	Moment of inertia around z-axis without battery and flight controller

6.3 Motors

Selection of a motor should come hand in hand with the selection of the propeller to fulfill the payload requirements. They both should come after the complete estimation of the aircraft weight to be able to determine the required thrust.

The generated thrust is a function of both lift coefficient of the propeller and the rotational speed of the motor as shown in section 3.4. Normally, a high thrust to weight ratio (more than 2) shall be used to ensure a fast-responding maneuvers.

In chapter C advantages of BLDC over brushed DC motors were presented, motor parameters were illustrated along with the characterization of the performance.

The selected motor should be able to provide adequate RPM that allows the propellers to generate the required thrust, for that a higher KV value for motor is better. The KV value represent the RPM that is generated due to 1 V of power supply. In the propeller section 6.5 propeller performance with the motors will be illustrated.

Also, the selected motors should have light weight to allow for more payload at given thrust and maximum current that is lower than the maximum current an ESC can accommodate.

The higher KV value, the bigger the motor hence the more the weight. Hence, a trade-off between size and maximum RPM shall be made.

In this work, TURNIGY MULTISTAR 2213 935KV 12 pole multi-rotor outrunners V2 are used. They have high-end magnets, high quality bearings and all are precision balanced for smooth running. The TURNIGY MULTISTAR

2213 integrates the prop adapter into the motor housing for easy propeller attachment and centering.
Figure 6.3 shows TURNIGY MULTISTART motor while its specifications are shown in table 6.3.

Elaboration on BLDC motor characterization along with a quick glimpse on the literature is provided in appendix C.

Figure 6.2: TURNIGY Multistar BLDC 2213 935KV

Table 6.3: Motor Specifications

Weight	55 grams
KV	935 RPM/V
Max Current	15.0A
No Load Current	0.4A
Max Surge Watts	200W
No. of Poles	14
Internal Resistance	0.180Ω
Motor Shaft	3mm
Motor Inertia	$6 * 10^{-5} kg.m^2$

6.4 Speed Controller

As stated in chapter C the use of Brushless DC motors entails the use of Electronic Speed Controllers (ESC) to perform electronic commutation.ESCs are very important because they control the speed of the BLDC.

A standard commercial ESC should tell the number of amps it supplies to your motor (this is the 'size' of the ESC) along with additional information such as if it has a Battery Eliminating Circuit (BEC) or not, and what batteries it supports. Normally the size and weight of an ESC is proportional to the amp rating.

Heavier propellers also draw more current, so selection of a motor and an ESC should come with meticulous selection of light weight efficient propellers.

BEC equipped ESC means that they are able to output a constant voltage, and so power some equipment like the flight controller. BEC spares the developer of the quadcopter platform the headache of providing different power sources for both the motors and other components.

Selection of ESC depends on the current capacity, firmware and commutation protocol. The maximum current drawn from the used motors is 15 amp as mentioned in section 6.3, an ESC current capacity should be larger than 15 amp to prevent ESC overheating. The higher the motor KV value, the higher its current draw.

ESC protocol manages the speed of signals between the flight controller and the ESC. The oldest ESC protocol – standard PWM, has delay up to 2ms, new generations of ESC protocols have significantly higher speeds up to $2\mu S$.

Two of the most common ESC firmware for RC are SimonK and BLHeli, they are extensively used.

The used ESC is AFRO 20A and it is shown in figure 6.3 while table 6.4 shows its specifications.

The firmware may be upgraded and/or customized to enable or disable various features. Firmware is open source and available for download using the regular PWM input cable and instructions are available in the datasheet.

Table 6.4: ESC Specifications

Type	AFRO ESC
Max. Current	20A
Voltage Range	2-4 Cell LiPo
Built-in Voltage Regulator	Yes
Weight	22.8 grams
Firmware	SimonK
Protocol	OneShot125

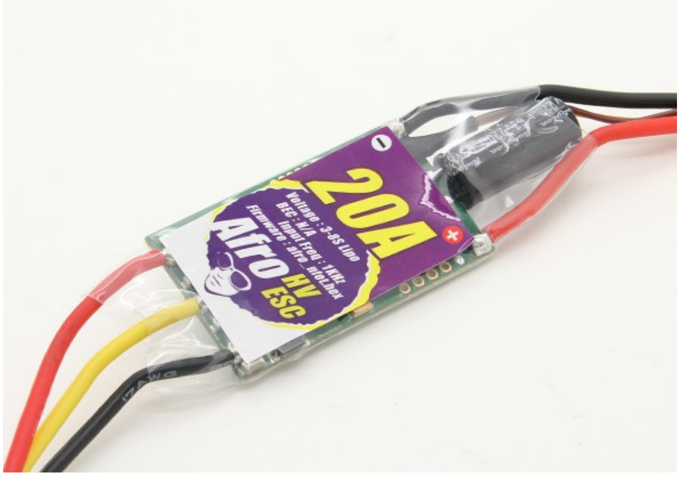

Figure 6.3: AFRO 20A ESC

6.5 Propellers

In rotorcraft, the propellers are the sole generators of aerodynamic forces. They generate lift and drag forces. its rotary motion through the air creates a difference in air pressure between the front and back surfaces of its blades. Propellers exist in different length and pitch:

- The length of a propeller is the diameter of a disc the propeller makes when it is spinning
- The pitch is the turning of the angle of the blades of a propeller to control the production or absorption of power

Larger propeller or higher pitch will increase aircraft's speed but also use more power, the larger the propeller (either increasing diameter, pitch or both) the more energy it takes to spin.
Normally, a propeller for a quadrotor is characterized by its material, weight and rigidity of its structure. Most durable propellers are made of composite plastic or carbon fiber. Composite plastic is ubiquitously used, however carbon fiber is lighter in weight and has a more rigid frame.
Lighter weight means lower drag and less moment of inertia, this enhances the performance and increases controllability.
The more rigid the propeller frame is, the better performance at high RPMs. Weak propellers may bend or deform at high RPM, this would make the propeller lose thrust and destabilize the vehicle.
Characterization of aerodynamic forces discussed in section 3.4 requires prior knowledge of aerodynamic constants of the propeller, the aerodynamics of

propellers has been extensively studied for the development of turbo prop aircraft. These data are generally useful for the design of rotor based vehicles.

In this work, a basic level of aerodynamic problem of a propeller is addressed. Complex nonlinear interactions are easily neglected due to the low-speed nature of the problem.

Data of the propeller are either published by the manufacturer or can be found in results published by laboratories like the propellers data published by the University of Illinois at Urbana-Champaign [47] and the data published by APC Propeller Performance Data [48] which are based on vortex theory, using actual propeller geometry.

The aim of propeller characterization is to obtain the thrust coefficient defined by equation 3.7 and drag coefficient defined by equation 3.8.

In this work, a T-Style 10x5.5 carbon fiber propellers are going to be used. Carbon fiber propellers have light weight with rigid structure that has more capability to resist flapping. Fig 6.5 Based on the geometry data and aerodynamics parameters extracted from [48] and are shown in table 6.5, the thrust and drag factors can be obtained from equations 3.7 and 3.8.

Figure 6.4: T-Style 10x5.5 carbon fiber propellers

$$\begin{aligned} K_T &= 1.241e^{-4} \\ d &= 6.452e^{-6} \end{aligned} \tag{6.1}$$

These value will determine the relationship between motor speed and aerodynamic load generated.

Table 6.5: Key features of the Propeller

Name	Name	Value
Radius	r	0.127 m
Thrust Coefficient	C_T	0.121
Power Coefficient	C_P	0.0495
Air density	ρ	1.255 kg/m^3
Disk Area	A	0.05067 m^2

6.6 Battery

Brushless motors are driven by batteries. The most common type of batteries used in UAV applications is the Lithium polymer batteries, more commonly known as LiPo. This could be attributed to their high energy density, high discharge rate and light weight which make them a great candidate.

LiPo batteries are made up of individual battery cells connected in series. Each cell has a nominal voltage of 3.7V. Motor RPM is directly related to battery voltage.The output RPM from the BLDC motor depends on both the motor's KV value and supplied voltage. However, the higher the voltage the bigger the battery.

Characterization of LiPo batteries does not depend solely on voltage, the capacity of battery is measured in Milli-amp hour (mAh) which indicates the amount of current the motor can draw from the battery for an hour until it is empty. For example, an 2200 mAh would take an hour to get empty if the motors are constantly drawing 2.2 amp. Normally, increasing battery capacity increases both size and weight of the battery.

With the advance of battery manufacturing technology, the 4-cell LiPo battery has become lighter in weight and providing more voltage than the 3-cell battery. That's why the 4-cell battery is becoming the dominant among quadrotor hobbyists around the world.

Normally, in most quadrotor applications 3 or 4 cell LiPo batteries are used. In our case, and as mentioned in ESC specifications in section 6.4, the maximum number of cells the ESCs can accommodate is 4 cells and this would be the limiting number.

Also, battery discharge capacity (C-rating) is another factor that can affect motor performance. The battery discharge capacity should be adequate enough to supply motors maximum current draw.

As shown in section 6.3, the used motors have maximum current draw of 15 amp per motor, this means total current draw of 60 amp. Normally motors account for the maximum source of energy consumption in a quadrotor. The battery discharge capacity should be able to cover this current draw plus an extra margin for safety.

The maximum current supply from a LiPo battery is governed by the following equation

$$A_{max} = Ah \times C_{rating}$$

Where Ah is the current capacity.

The battery used in building this model is Turnigy 2200mAh 3S 35C shown in Fig 6.6.

This battery is equipped with 3 cells with 35C rating at constant level with 45C burst discharge. Complete specifications are shown in table 6.6. The battery will be installed in the mid-section of the quadrotor, so its center

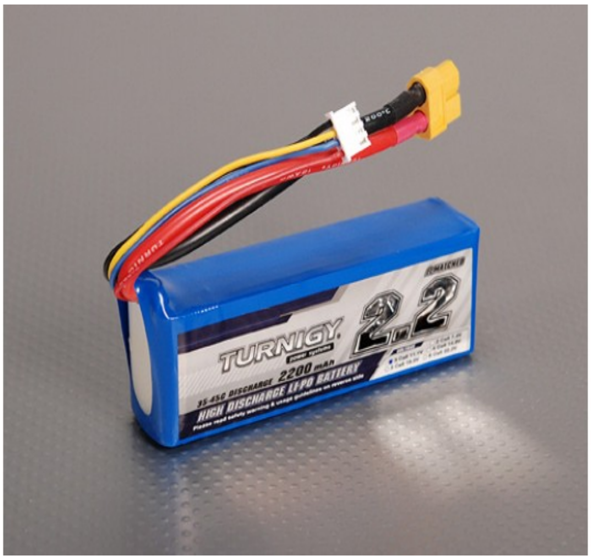

Figure 6.5: Turnigy 2200mAh 3S 35C

Table 6.6: Specifications of the Turnigy 2200mAh 3S 35C Battery [49]

Minimum Capacity	2200mAh
Configuration	3S1P / 11.1v / 3Cell
Constant Discharge	35C
Peak Discharge (10sec)	45C
Weight	199 g
Dimensions (w x h x d)	105 x 33 x 24 mm

of mass coincides with the center of mass of the vehicle. This will ease the process of moments of inertia calculations.

The battery will be treated as a solid cuboid of width w, height h and depth d.

The moments of inertia are:

$$I_{xx} = \frac{1}{12}m(h^2 + d^2) = 1.9238e - 04kg.m^2$$
$$I_{yy} = \frac{1}{12}m(w^2 + d^2) = 2.7611e - 05kg.m^2$$
$$I_{zz} = \frac{1}{12}m(w^2 + h^2) = 2.0089e - 04kg.m^2$$

These values are going to be used for the calculation of the total moments of inertia for the vehicle assembly as in 6.11.

6.7 Flight Controller

A flight controller (FC) is a microcontroller that is used to direct the RPM of each motor in response to input. It processes the data received through different sensors and calculates the corresponding output that achieves certain performance.

Flight controllers are configurable and programmable, allowing for adjustments based on varying multi-rotor configurations. The majority of flight controllers also employ sensors to supplement their calculations. These range from simple gyroscopes for orientation to barometers for automatically holding altitudes. GPS can also be used for auto-pilot or fail-safe purposes.

The flight controller used to conduct the experiment is **3DR Pixhawk Mini autopilot** which is a next-generation evolution of the traditional Pixhawk flight controller. Pixhawk Mini is based on the PX4 open-hardware project. Key features of the 3DR Pixhawk Mini autopilot are shown in table 6.7.

It also features an advanced processor and sensor technology from ST Microelectronics® and a NuttX real-time operating system for flexible and reliable controllability of the vehicle. The new model comes with improved sensors, including both primary and secondary (Inertial Measurement Unit) IMU (MPU9250 and ICM20608, respectively), lead to much better vibration handling and increased reliability [50].

The Pixhawk mini autopilot has an advanced 32 bit ARM Cortex $^{®}$ M4 Processor with 8 PWM/servo outputs. It allows Multiple connectivity options for additional peripherals (UART, I2C, CAN) as shown in figure 6.7. The Pixhawk mini flight controller is based on state-of-the-art Micro Air Vehicle Link (MAVLink) protocol for communication. The new generation Pixhawk Mini flight controller comes with rich interface that includes:

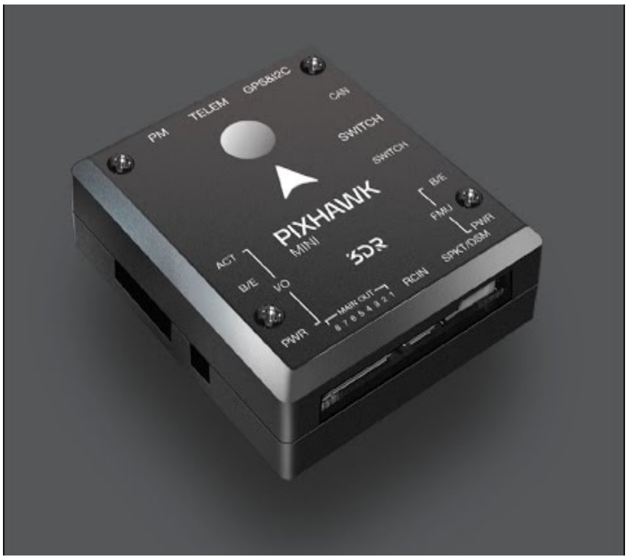

Figure 6.6: Pixhawk Mini Flight Controller

Table 6.7: Key features of the 3DR Pixhawk Mini autopilot

Main Processor	STM32F427 Rev 3
IO Processor	STM32F103
Accel/Gyro/Mag	MPU9250
Accel/Gyro	ICM20608
Barometer	MS5611
Power Output	4.1 5.5V
Flash Memory	2 MB
Max Input Voltage	45V
Weight	15.8g
Dimensions	38x43x12mm

- 1 x UART Serial Port (for GPS)
- Spektrum DSM/DSM2/DSM-X® Satellite Compatible RC input
- Futaba S BUS® Compatible RC input
- PPM Sum Signal RC Input
- I2C (for digital sensors)
- CAN (for digital motor control with compatible controllers)

- ADC (for analog sensors)
- Micro USB port

In addition, Pixhawk Mini comes in a kit that includes many accessories to facilitate the connection and use of the board. These accessories include:

- Safety Switch and buzzer connector
- Standard Power Module (up to 6s battery) with 6 pin connector for an APM power module to provide power to the flight controller along with voltage and current measurements from your flight battery
- 8-channel servo output board
- A bundle of cables

The moments of inertia of the flight controller will be calculated just like the battery in section 6.6 assuming the flight controller is a solid cuboid. The values of the moments of inertia are:

$$I_{xx} = 2.6573e - 06kg.m^2$$
$$I_{yy} = 2.1173e - 06kg.m^2$$
$$I_{zz} = 4.3907e - 06kg.m^2$$

These values will be used to account for the total moments of inertia for the assembled quadrotor.

6.8 GPS/Compass

The GPS module provides the geolocation to the flight controller, it is specifically useful for providing coordinates of the vehicle that the flight controller can use.

The used model also comes with an integrated GPS module based on the uBlox M8N with 167 dBm navigation sensitivity, advanced jamming and spoofing detection and various security and integrity protection, it also comes with 35 mm high gain patch antenna with large ground plane for reliable reception and an integrated magnetometer.

The GPS modules has a state-of-the-art concurrent reception of up to three Global Navigation Satellite System (GNSS) that provides autonomous geospatial positioning with global coverage. The GNSS system provides GPS, GLONASS, Galileo, Beidou and other regional systems with advantage to having access to multiple satellites is accuracy, redundancy and availability at all times [50].

The module also comes with 3-axis digital compass based on Honeywell HMC5883L with continuous self diagnostics and failure detection.

Along with the GPS and compass, the module comes with a high precision barometric altimeter with altitude resolution up to 10 cm.

Features of the GPS/Compass module are shown in table 6.8. The moment of inertia of the GPS module is calculated just like the flight controller and

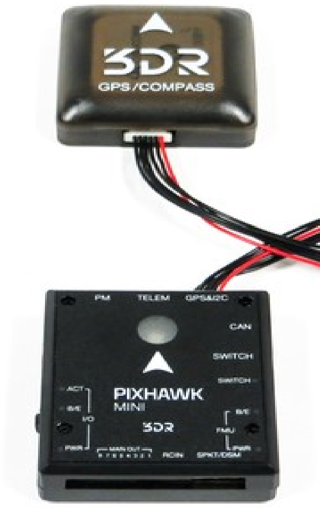

Figure 6.7: Pixhawk Mini Flight Controller with GPS/Compass

Table 6.8: Key features of the 3DR GPS/Compass module

Receiver	uBlox M8N with high gain antenna
Compass	STMicroelectronics LIS3MDL with thermal compensation
Barometer	MS5611
Magnetometer	HMC5983
Navigation Sensitivity	167 dbm
Update Rate	Up to 15 Hz
Supply Voltage	2.7 - 3.6 V
Weight	22.4g
Dimensions	37x37x12mm

battery, it is assumed that the GPS is a solid cuboid. The values of the moments of inertia are

$$I_{xx} = I_{yy} = 2.8524e - 07kg.m^2$$
$$I_{zz} = 5.1618e - 07kg.m^2$$

6.9 Radio Control

A Radio Control (RC) is a device that allows the control of the aircraft wirelessly through the wireless transmitter (TX). The signal/commands are then received by a radio receiver (RX) which is connected to a flight controller.
Normally, an RC is being selected based on the number of channels, modes, frequency technology and type of receiver.
The number of channels in an RC determines the number of individual actions by which the aircraft can be controlled. For example, a throttle control needs one channel and so does roll, pitch and yaw control.
This makes the minimum number required to control a quadrotor is a 4-channel RC. However, normal control of a quadrotor requires more channels to allow for extra functions and control of the vehicle and keep space for changing mode or trigger a certain function of the multirotor (i.e. emergency landing, fail safe mode, etc).
The modes of the transmitter are basically the configuration of the control stick. There are two common modes of any transmitter shown in Fig.6.9 . These modes can be set up and changed through the flight controller. The frequency technology used in hobbyists RCs is 2.4GHz. The 2.4GHz systems are the new technology in RC for hobbyists. They are equipped with frequency hopping technology which is highly resistant against jamming and picking up signals from other devices.
There are other frequencies that have been used among RC communities like 1.3ghz, 900mhz or 433MHz but they are less common.
Finally, the selection criteria of an RC should come to the type of receiver. Receiver protocols an make a big difference. There are many receiver protocols that are being used:

- **PWM – Pulse Width Modulation**: the old fashioned receivers, they use one servo wire for each channel. They are the cheapest option, however and due to the messy wiring they require they have been replaced recently.

- **PPM – Pulse Position Modulation**: The advantage of PPM is that only one signal wire is needed for several channels, instead of a number of individual wires. Technically, they send multiple PWM signals down

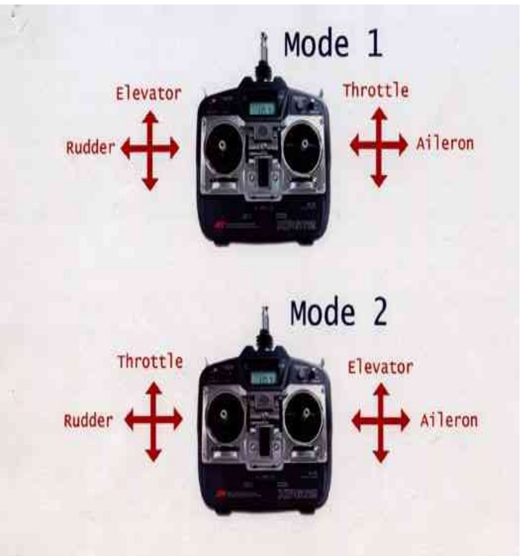

Figure 6.8: Conventional modes of an RC transmitter [51]

a single wire in succession. The PPM signal is just like the PWM signal, both are analog signals.

- **PCM – Pulse Code Modulation**: which is similar to PPM in its data types, however, PCM sends a digital signal while PPM is analog.
- **SBUS - Serial BUS**: it is an inverted UART communication signal and a type of serial communication protocols, used by Futaba and FrSky RCs. It supports up to 18 channels using only one signal cable.
- **DSM2/DSMx**: an interference-resistant transmitting protocol. It always "hops" between frequencies to maintain the best connection to your radio. It has two advantages, first, the signal is spread out over a wider frequency band and each transmitter/receiver pair uses it's own coding scheme to scramble the signal.

In this work, FlySky FS-i6 radio control is used. It is a 6-channel, 2.4GHz transmitter with Automatic Frequency Hopping Digital System (AFHDS). This transmission system guarantees long range and jamming free control.
FlySky FS-i6 radio system works in frequency range of 2.405 to 2.475GHz. This band is divided into 142 independent channels which used 160 different hopping algorithms [52].
The radio system uses a high gain antenna to cover the whole frequency band with a sensitive receiver to guarantee a jamming free signals. Fig. 6.9 shows the complete controller kit.

Transmitter specifications of the FlySky FS-i6 radio control are shown in table 6.9.

Figure 6.9: FlySky FS-i6 Radio Control

Receiver specifications are shown in table 6.10. The receiver and transmitter come already bound at production time which eliminates the need to manually bind them. This model of radio controllers is compatible with the Pixhawk Mini flight controller as stated in section 6.7. However, it needs

Table 6.9: Key features of the FlySky FS-i6 RC

No. of Channels	6
Channel Resolution	1024
RF Frequency	2.405 - 2.475GHz
Bandwidth	500KHz
Power Supply	6V (1.5AAx4)
Low Voltage Warning	HMC5983
Antenna Lengh	26mm x 2 (dual antenna)
RF Power	Less than 20 dBm
Size	174x89x190 mm
Weight	392g
Output	PPM
Low Voltage Warning	Below 4.2V
Certifications	CE0678 & FCC

Table 6.10: Technical specifications of the FlySky FS-i6 RC

RF Receiver Sensitivity	-105 dBm
Power	4-6.5 V
Antenna Length	26mm
Weight	6.4g
Size	40.4x21.1x7.35 mm

a PPM encoder to encode up to 8 PWM signals into one PPM signal the Pixhawk can read.

6.10 Telemetry

Radio telemetry allow your ground station computer to communicate with your vehicle (aircraft, quadrotor, ground robot) wirelessly, providing easy way for viewing real time -flight data, changing missions on the fly.

Telemetry provides an automated communications means by which measurements data are collected at remote points and transmitted to receiving equipment for monitoring. Telemetry can be used to send or receive data for distant devices.

For this work, 3DR 915MHz telemetry is used. It has a small size , light weight and have a communication range of more than 1 km. It is fully compatible with Pixhawk flight controller and APM Ardupilot. The system provides a full-duplex link with open source firmware. Interface to the module is via standard 5V-tolerant TTL serial / FTDI USB serial. Fig. 6.10 shows the air and ground telemetry modules along with cables and antennas used.

The 3DR telemetry has a free open source firmware that can be upgraded and Frequency Hopping Spread Spectrum (FHSS) for spoofing protection. It comes with Micro-USB port USB link on the ground module and its maximum power consumption is 100 mW.

6.11 Vehicle Assembly

Vehicle assembly is done using the aforementioned components. Fig.6.11 shows the assembled quadrotor used in the flight tests in this thesis.

The total weight of the quadrotor is 1.05 kg and this can be calculated right

Figure 6.10: 3DR 915MHz Telemetry

away from the addition of every single component's mass as done in section 6.2.

The moment of inertia of the quadrotor is also as important as its mass,

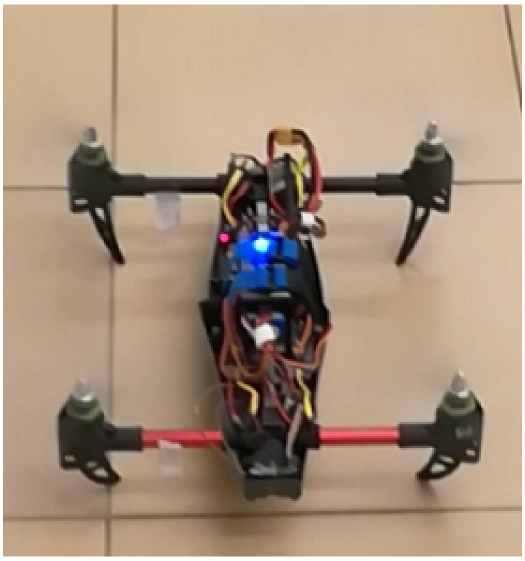

Figure 6.11: Quadrotor Assembly

this was very clear in the quadrotor's equations of motion equation 3.25. The moments of inertia of the frame and the components that come pre-installed on it are shown in table 6.1, to account for the items that are added, their moments of inertia shall be added to the value of the frame.

The total moments of inertia of the quadrotor are

$$I_{xx} = 0.0073 kg.m^2$$
$$I_{yy} = 0.0076 kg.m^2$$
$$I_{zz} = 0.0126 kg.m^2$$

The values of mass and moments of inertia obtained in this section are the values used in the simulation as illustrated in chapter 5.

The distance l can be calculated directly from the dimensions of the frame mentioned in table 6.1. For the Quanum Venture FPV Quadcopter

$$l = 261.7mm$$

Chapter 7

Experimental Results

The Pixhawk flight controller used with the vehicle frame has the ability to record vehicle data. Dataflash logs are stored on the flight controller's onboard dataflash memory and can be downloaded after a flight either through a standard USB connection or through Telemetry wireless download.

The sensors built in inside the flight controller measures different variables and RC inputs. The measurements stored are attitude angles (pitch, roll, yaw) both the RC input signal and the quadrotor measurements, the quadrotor relative position through the GPS module and power consumption.

By default, a new dataflash log file is created when the vehicle is initially armed (i.e. data logging starts when the vehicle is armed and ready for RC input), and logging is performed only while the vehicle is armed.

Two runs were carried out, one with a PID controller that comes pre-installed on the flight board and the other after the download of the SMC control onto the board. The flight controller comes with a pre-installed and customizable PID control.

The flight controller reads four signals from the RC which are the roll, pitch, yaw and altitude only and it measures different outputs from the controller like actual attitude of the vehicle, power consumption and rates of change of attitudes. The GPS module reads the relative position (relative to the takeoff point) and the current date of the flight.

7.1 QGroundControl

QGroundControl is the application used to set up and configure the **Pixhawk** autopilot and enable the download of the latest PX4 firmware used with this board. It can also be used to configure any MAVLINK-enabled drone.

It provides easy and straightforward usage, while still delivering high end feature support for experienced users. Through QGroundControl one can set up/configure PX4 Pro-powered vehicles, plan missions for autonomous flights and retrieve data log files.
QGroundControl is a cross-platform application that also has a mobile applications that runs on both Android and iOS.

7.2 Flight Data Parsing

The latest firmware of **Pixhawk** flight controller is the PX4 firmware which uses a unique file format for data logging. PX4 uses a binary file with *.ULOG extension for logging system data. The ULOG file contains quadrotor parameters and message types that are logged.The ULOG format uses Little Endian for all binary types.
Downloading of the PX4 firmware is done through the QGroundControl tool using the MAVLINK protocol.
Parsing of the binary ULOG file is rather complicated because of it is a new binary file format and not many parsers are available by **Pixhawk**.
However, some tools are available that enable the drawing of most of the quadrotor parameters in the ULOG file, the most common drawing tools are:

- Flight Review: a web-based tool that parses the ULOG file and generate charts for measured values against input values.

- Flight Plot: a cross-platform, Java-based desktop application that does the same job as the Flight Review website.

Despite their general ease of use of both drawing tools, their interface is not intuitive and there is no documentation of their output.
Along with the drawing tools, a Python-based parser is available open-source that transforms the binary ULOG file into readable excel sheets. The script is called **pyulog**. One good aspect of pyulog is that it converts the ULOG file to Comma-Separated Values (CSV) excel files which in turn can be read into MATLAB and drawn. However, not all data that appear using Flight Review or Flight Plot are available in the excel files, that is why some results shown below will be displayed using MATLAB charts and others are obtained directly from Flight Review.

7.3 PID Results

The flight controller comes with a pre-installed and customizable PID control. Customization can be done through the ground control station (QGround-Control tool). The results of the run shows the response and behavior of the PID controller.

The run with the PID controller took place in an open area inside the American University in Cairo (AUC), where the average wind speed was 2.8 m/sec [53]. Fig. 7.1 shows the flight test using the PID control.

Figures 7.2, 7.3 and 7.4 show the measured roll, pitch and yaw angles consecutively versus the input signal from the RC. Fig. 7.2 shows the tracking of roll angle against the RC input, overshoots are apparent at the beginning and the controller took 12 seconds to settle down to the input track. Throughout the input trajectory, the controller exhibited a consistent steady state error.

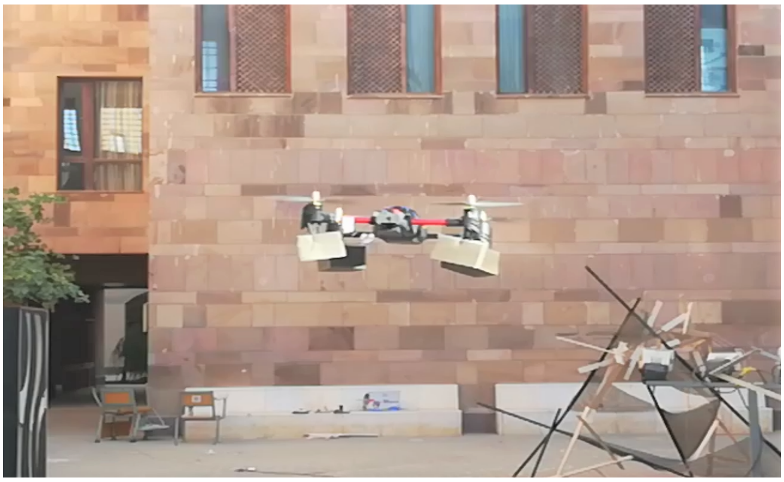

Figure 7.1: PID test flight

Fig. 7.3 shows the tracking of pitch angle against the RC pitch input, it pretty much resembles the roll angle tracking performance. It starts with an overshoot that took the first 10 seconds to damp out. The signal tracking suffered also from a persistent steady state error.

Figure 7.4 shows the measured psi angle against the RC input. The figure shows that the vehicle underwent a slight rotation despite no input from the RC is recorded.

The slight difference is a direct result of the linearization of the controller where the coupling between attitude angles is eliminated. Also, the linearization of the system equations of motion is another reason where the coupling between vehicle dynamics is removed and this affects controller performance.

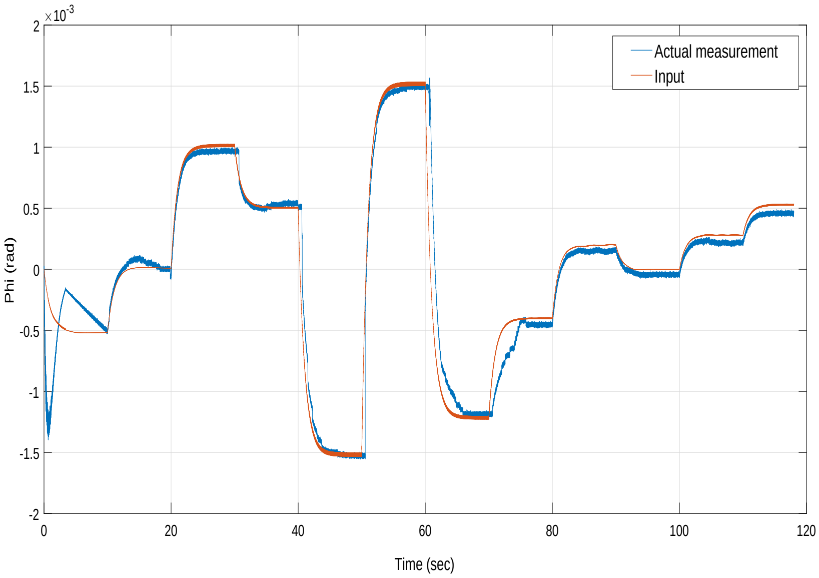

Figure 7.2: Measured roll angle against RC roll input

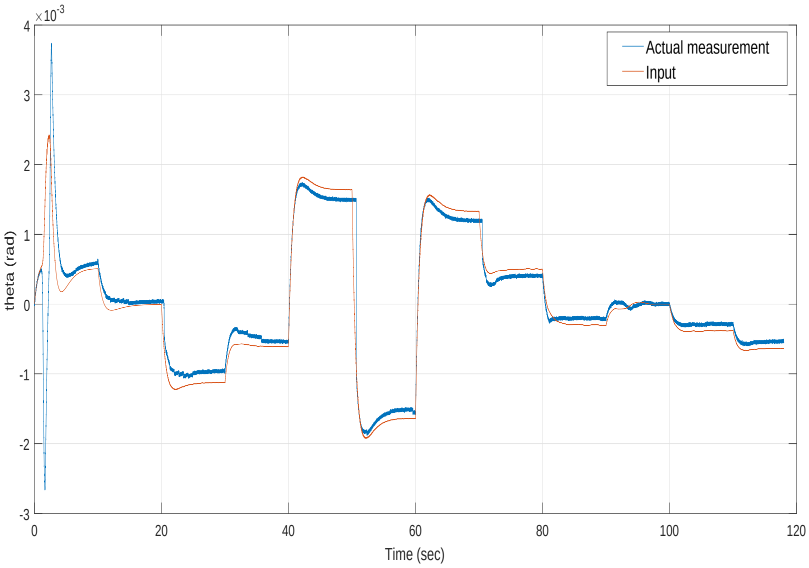

Figure 7.3: Measured pitch angle against RC pitch input

Figure 7.5 shows the location of the vehicle relative to the starting point as measured by the GPS module.

Fig. 7.6 shows the 3D trajectory of the quadrotor with the PID control.

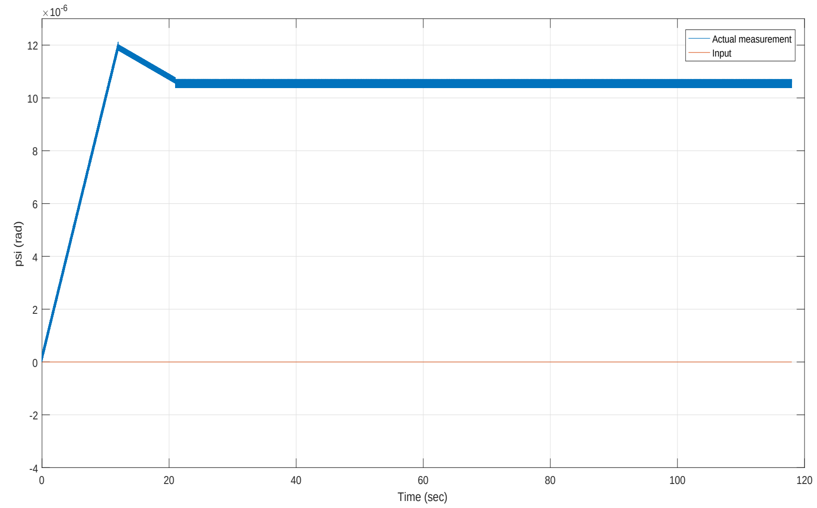

Figure 7.4: Measured yaw angle against RC yaw input

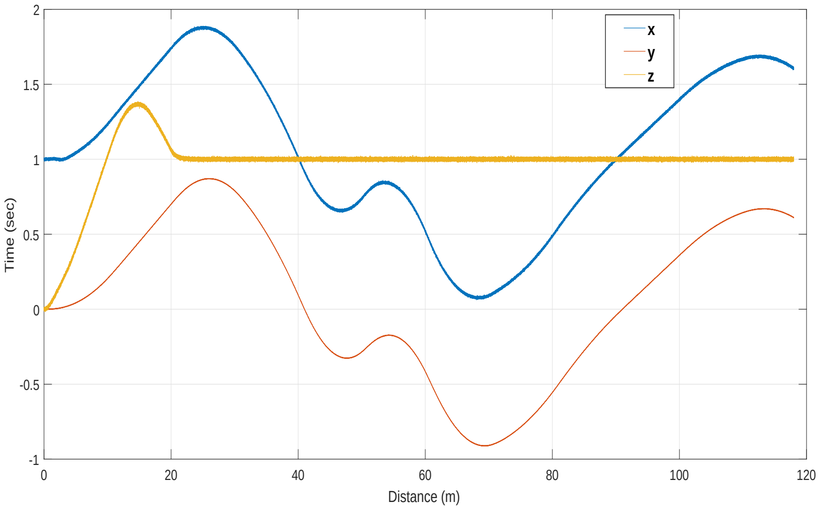

Figure 7.5: Quadrotor relative position as measured by the GPS - PID Control

Figures 7.7 and 7.8 shows power discharge, CPU loads and RAM usage during flight. Figures are obtained directly from Flight Review web-based tool. They are indicative of how much energy consumed during the maneuvers and the processing power the controller needs.

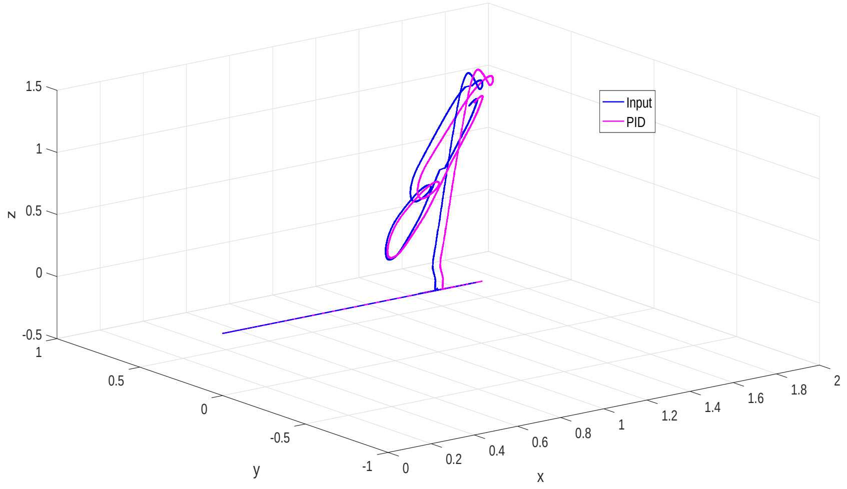

Figure 7.6: 3D trajectory of the quadrotor - PID Control

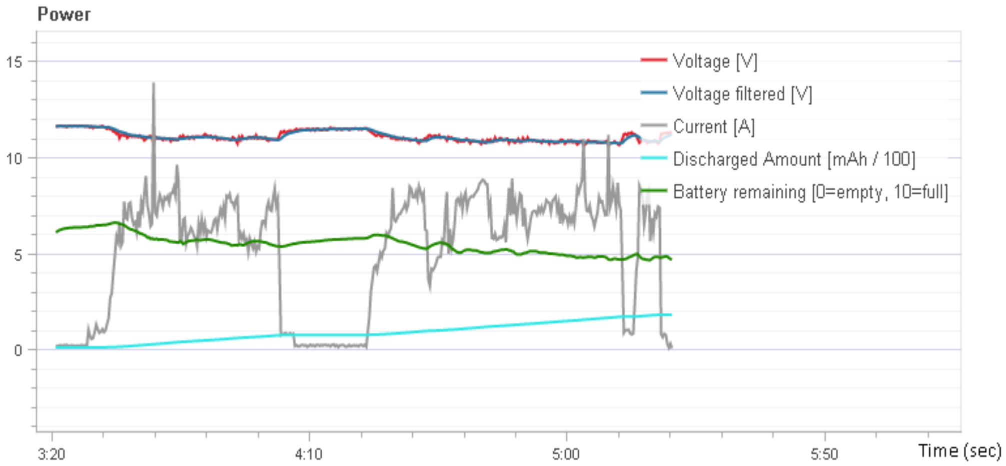

Figure 7.7: Power discharge during flight - PID control

7.4 SMC Results

7.4.1 Pixhawk Support Package

The download of the SMC onto the **Pixhawk** flight controller is done through MATLAB by using Pixhawk Pilot Support Package (PSP) version 2.1 issued on February 2017.
Using Pixhawk support for Embedded Coder, once can generate ANSI/ISO

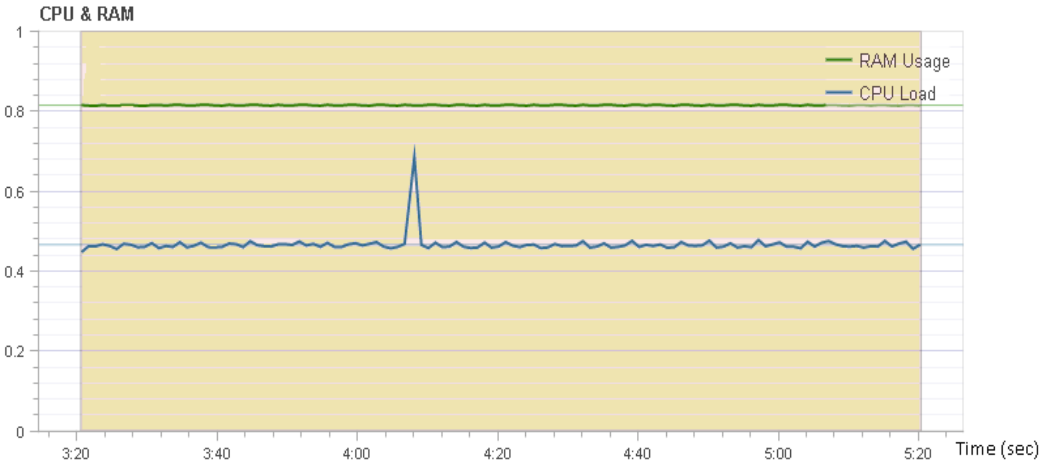

Figure 7.8: CPU and Ram usage during flight - PID control

C from Simulink models specifically tailored for the Pixhawk FMU (flight management unit) using the Pixhawk Toolchain.
The Pixhawk Pilot Support Package (PSP) feature allows users to use Simulink models to generate code targeted for the Pixhawk FMUv2 (Flight Management Unit). The PSP provides the ability to incorporate the Pixhawk Toolchain for complete firmware build and download to the px4fmu Version 2 unit. It does not provide exact function behavior blocks for other services running on the Pixhawk like the Attitude Estimation using Extended Kalman Filter (EKF). That is why the results obtained using the Support Package are noisier than the results obtained using the original **Pixhawk** firmware.
The Pixhawk Simulink blocks allows users to access sensor data and other calculations available to be used in their Simulink model at runtime. Generated code can then be compiled and executed on the Pixhawk platform controlling a multi-rotor airframe.
The Pixhawk Simulink blocks allows users to access sensor data and other calculations available to be used in their Simulink model at runtime. Generated code can then be compiled and executed on the Pixhawk platform controlling a multi-rotor airframe.
The Pixhawk Pilot Support Package is based off a forked version of the official Pixhawk Firmware available on the online web repository **GitHub**.
The use of the PSP also requires the installation of the **Pixhawk** toolchain to allow software update and pre-compiling. The setting of the sampling time inside the PSP is set to 1E-04 second.

7.4.2 SMC Experimental Results

Due to the fact that the PSP does not support advanced filtering as of the original firmware provided by **Pixhawk**, the results obtained from the actual implementation of the controller on the board is noisier than the measurements obtained in 7.3.
However, due to the accurate modeling in the SMC and taking into consideration the actual nonlinear terms within the equations of motion, the performance of the vehicle is greatly enhanced and the tracking capability is improved.
The flight test with the SMC controller took place in an open area inside the American University in Cairo on March 17th 2018, where the average wind speed was 4.4 m/sec [53].
The flight test included basis maneuvers and attitude change (roll, pitch and altitude inputs only) with no yaw input to assess the maneuverability of the vehicle with the SMC.
Due to some hardships with the GPS and unreliable signals received to the flight controller, trajectory tracking is carried out by embedding a signal inside the code downloaded to the board, while the RC fifth channel acts as a switch that would change between the embedded signal and the RC input itself. This would make the quadrotor follow the same tracks used for the PID control in section 7.3.

SMC Flight Test

Fig. 7.9 shows the results of the test flight using the SMC.
In this run, the input to the quadrotor is the same as the one used in the test flight using PID control in section 7.3. This shall be used to compare the performance of each controller.
The roll angle obtained from the SMC is shown in figure 7.10 shows enhancement compared to the results using the PID control in 7.2. The slight overshoot that took place at the start of the maneuver is diminished and persistent difference between input and actual measurement is reduced. Also, the quadrotor showed quicker response to input signals. However, and as expected, the measurement is slight noisier due the the lack of implementation of better filters. Using SMC, the vehicle was able to achieve better tracking of the roll input signal.

The results for the pitch angle is also improved as shown in figure 7.11, the tracking performance is enhanced and the overshoot that happened at the beginning is completely damped and better tracking performance is achieved.

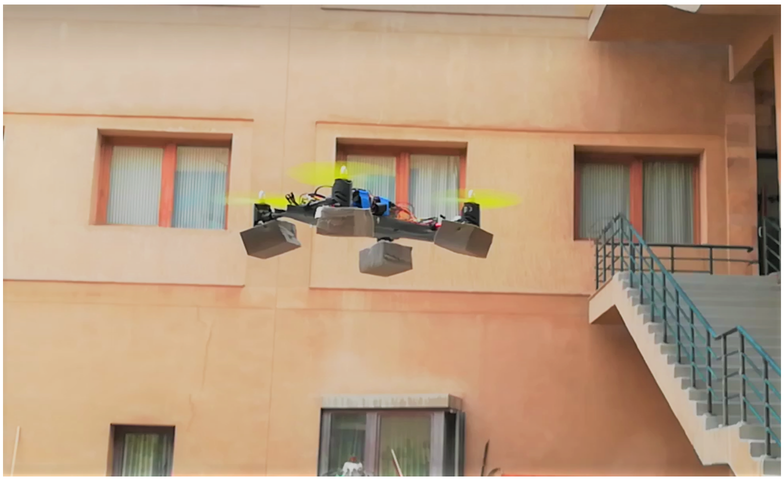

Figure 7.9: First SMC test flight

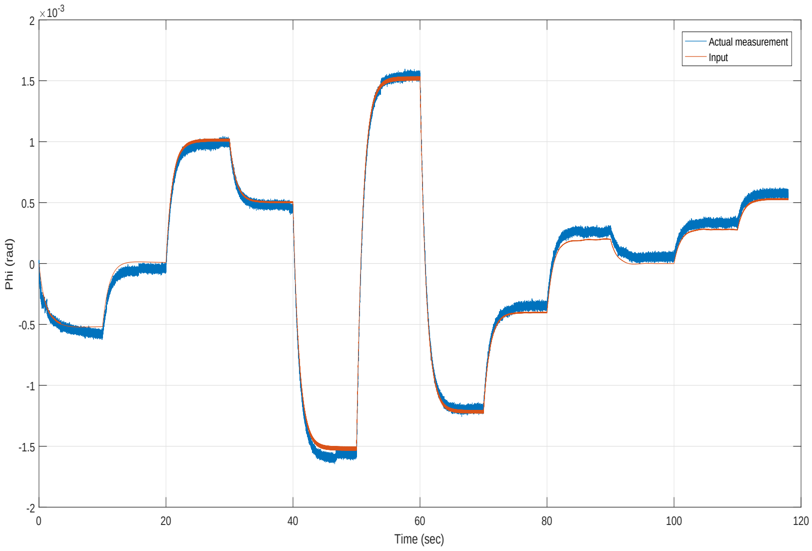

Figure 7.10: Measured roll angle against RC input - SMC Results

A slight steady state error is noticed at the beginning which the controller eventually managed to damp.

Yaw performance is also enhanced, although in this run no yaw input was present, the yaw suffered from uncontrolled rotation with the PID control, however, with the SMC the yaw angle tracking performance is also improved and the slight rotation accompanied with the PID control is eliminated as shown in figure 7.12, small oscillations however took place around the z axis

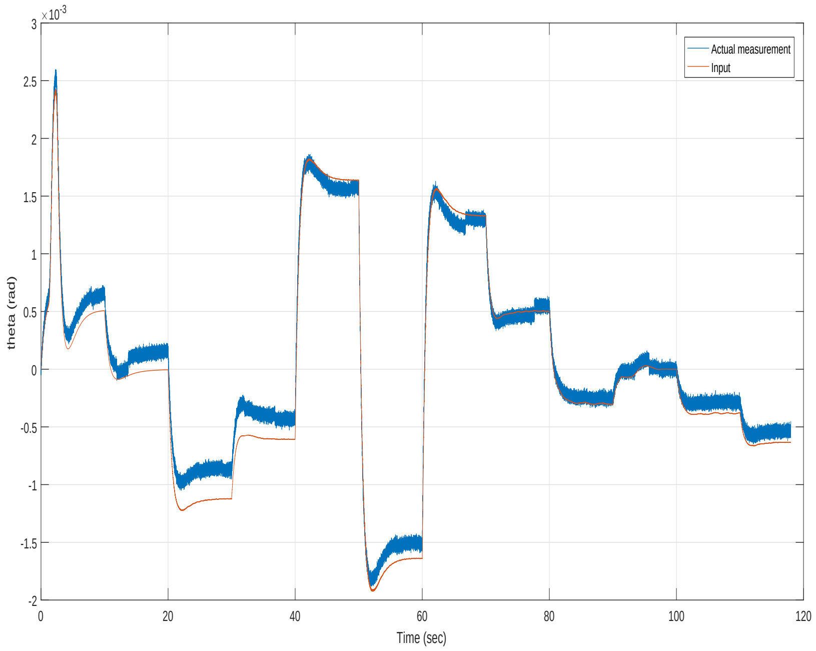

Figure 7.11: Measured pitch angle against RC input - SMC Results

due to the mechanical coupling between attitude angles. The same noise problem is also present with the yaw angle due to the absence of the adequate filters.

Fig. 7.13 shows the location of the vehicle relative to the starting point as measured by the GPS module with SMC.

Figure 7.14 shows the 3D trajectory of the quadrotor with the SMC control. It is clear that the trajectory tracking of the SMC is greatly enhanced from the PID controller shown in figure 7.6 even in the presence of a stronger wind.

Figures 7.15 and 7.16 show power discharge, CPU and RAM usage during the flying test using the SMC. It is clear that the power discharge is slightly decreased from the PD control. This confirms the results obtained from the simulations results where the PD control effort for a fast changing input signals was greater than the control effort of the SMC.

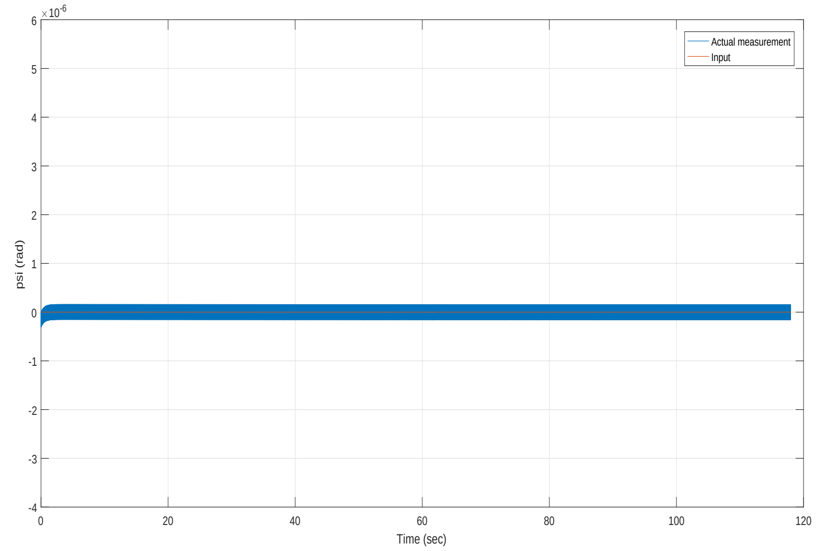

Figure 7.12: Measured yaw angle against RC input - SMC Results

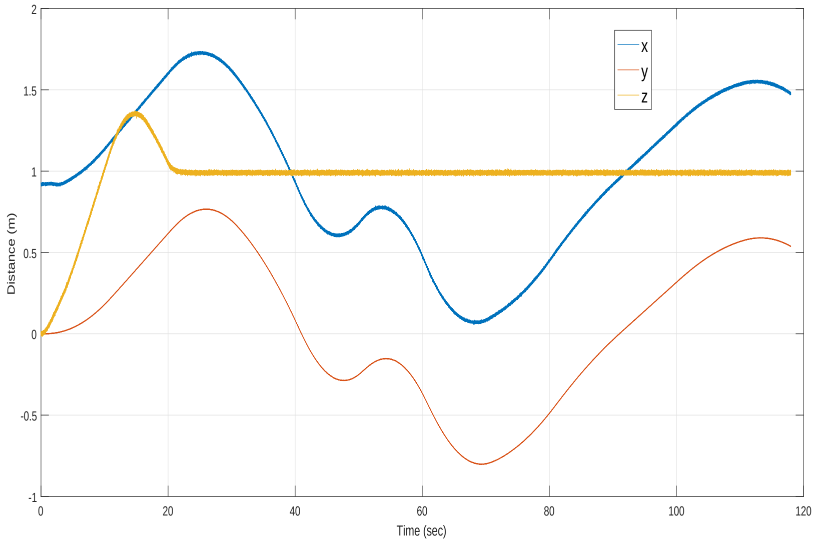

Figure 7.13: Quadrotor relative position as measured by the GPS - SMC Results

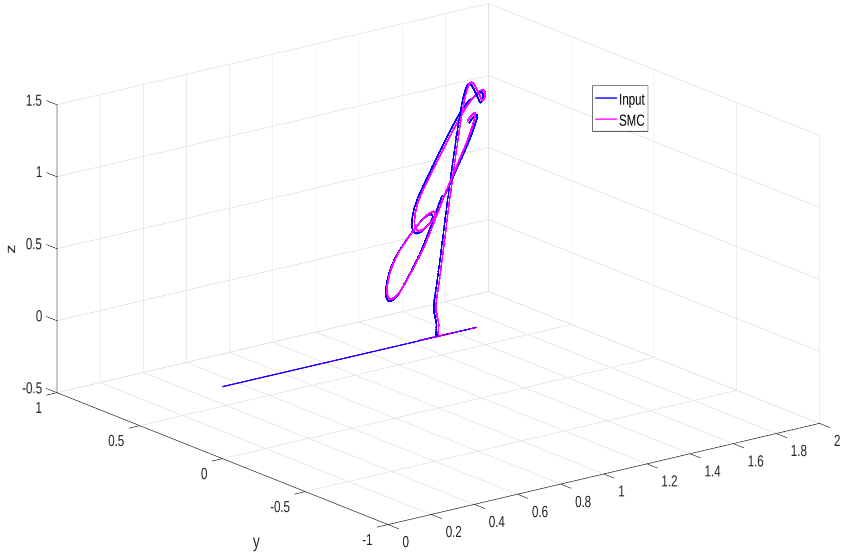

Figure 7.14: 3D trajectory of the quadrotor - SMC Results

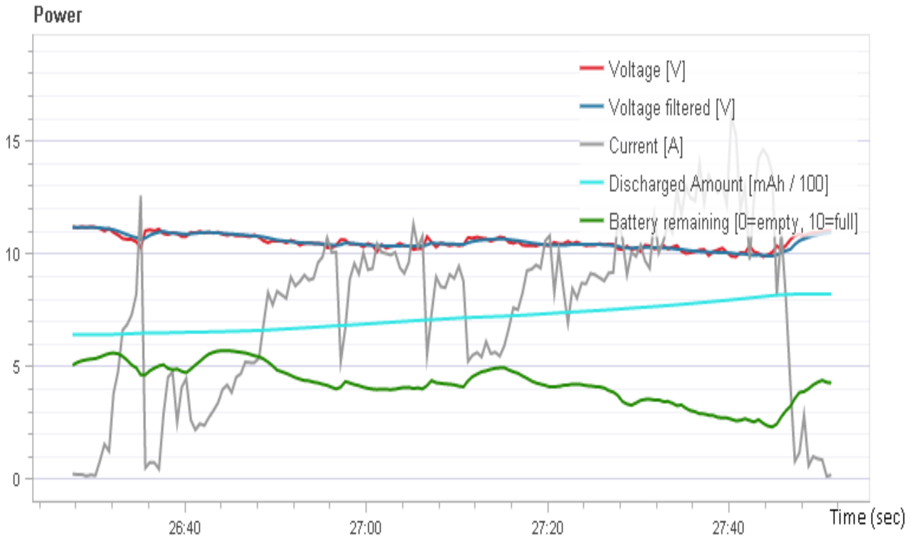

Figure 7.15: Power discharge during flight - SMC control

7.5 Comments and Conclusion

The results presented in this chapter shows the experimental outcomes of a quadrotor with **Pixhawk** FMU, once with PID controller and another with SMC. It was clear that the SMC enhanced the performance of the quadro-

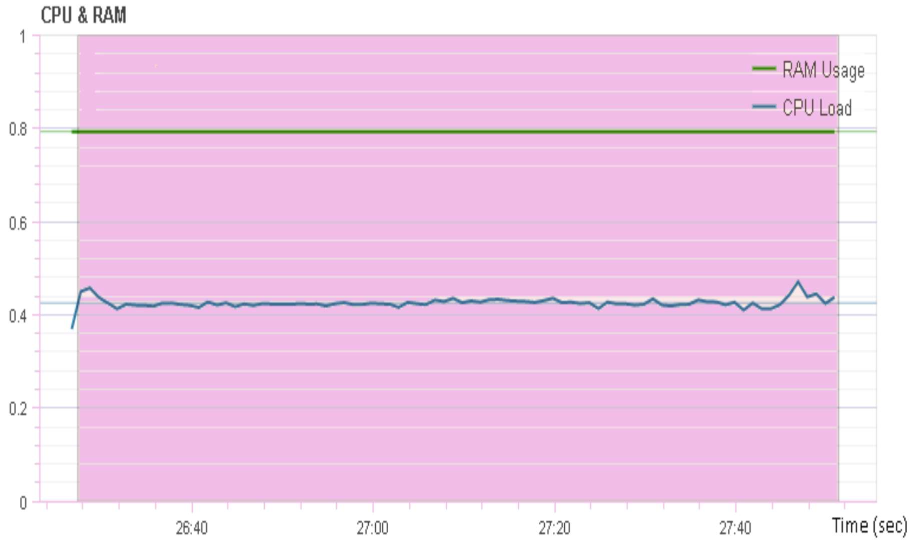

Figure 7.16: CPU and Ram usage during flight - SMC control

tor to RC inputs. Both pitch and roll angles tracking are enhanced with overshoot that occurred with PID controller damped and steady state error diminished.

Also, a slight rotation around z axis took place with PID controller as a results of the coupling between attitude angles, this slight rotation is eliminated with the SMC and the controller was able to maintain the yaw angle as desired with no unexpected rotations.

The download of the SMC onto the FMU removed some enhanced filteration techniques that came preinstalled with the Pixhawk firmware, this resulted in slightly noisier signals.

Experimental results showed the robustness of the SMC with boundary layer chattering alleviation technique against normal day flight test. SMC tracking performance of the quadrotor enhanced from the PD performance even in the presence of higher wind speed when SMC flight test was run.

In harmony with the simulations results, the control effort for an aggressive input is higher for PID control than the SMC, this is substantiated with experimental results as shown in figures 7.8 and 7.16 where the both the CPU and RAM loading are less for the SMC.

The enhanced tracking of vehicle attitude improved the position of the vehicle, 3D trajectories are changed due to the better tracking of the SMC against the PID as shown in figure 7.13.

The conclusion of the experimental tests is that the SMC was able to provide better performance for the nonlinear system of the quadrotor in terms

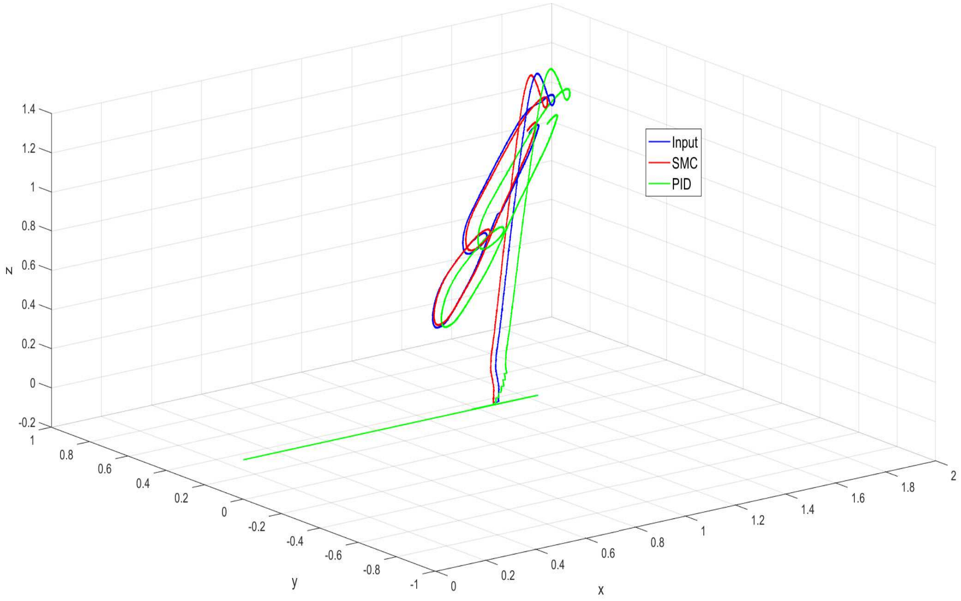

Figure 7.17: 3D Trajectory of PID control against 3D trajectory of SMC

of tracking of inputs and settling time. The chattering alleviation technique used with the SMC enhanced the control action and made the controller robust.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The work of this thesis shows that the required objectives are accomplished to good extent. The quadrotor equations of motion have been developed and discussions on different assumptions have been provided. The development of different types of controllers has been done, this includes PD and sliding mode controller. The limitations of each were taken into account, simulations of the two controllers are carried out with and without disturbance. PD control simulations with both smooth input and fast-changing input. The simulations revealed the limited capability of the PD control to deal with aggressive input signals especially in the presence of disturbance.
SMC simulations showed that the nonlinear technique was able to track the input signal even when accompanied by fast dynamics and abrupt changes. The SMC control effort however was very high with significant chattering. This was tackled by chattering alleviation technique from the literature. Chattering alleviation for SMC allowed the use of higher gains which in turn damped the overshoot and allowed for better performance. SMC also dealt significantly better than the PD in the presence of disturbance. Nonholonomic constraints that enabled the trajectory tracking of the underactuated system.
Motor characterization for hardware implementation is presented with the techniques used in literature. A built-in BLDC motor characterization inside the PSP is used to model the dynamics of the motors.
Quadrotor hardware development and design requirements components' specifications are presented and along with the implementation of the proposed controller with chattering alleviation on a **Pixhawk** flight controller using Pixhawk Support Package (PSP).

Exprimental results of the sliding mode controller along with the PID controller that comes preinstalled on the board are presented that showed the difference of performance between the linear and nonlinear robust controller. Different hardware implementation problems that arose during the actual assembly are presented also with the solutions undertaken, then the results obtained from different sensors on the board are shown.
Experimental results were consistent with the simulations results, that the SMC was able to better track the input trajectory than the PID even when the average wind speed was higher. Control effort of the SMC also was less than the PID as was clear from both the simulation and experimental outputs.
The main focus of this thesis was on the development of a robust nonlinear control for the underactuated quadrotor vehicle, this result was clear in the simulation even in the presence of disturbance, regulated disturbance has been fed into the model and results obtained that showed the reliability of the controller, however, in the hardware implementation and due to limitations imposed by the available components and ease of access to replacements.
Finally, some code snippets and comments on the hardware implementation are included in the appendices at the end of the thesis document for future reference.

8.2 Future Work

Continuation of the work done in this thesis could include:

- Although the SMC showed good robustness and reliability of the system, the chattering alleviation technique made the quadrotor more susceptible to unexpected disturbances. Combination of SMC with another nonlinear robust control and comparison of results is a potential task to do.

- Use of new chattering alleviation techniques such as the one mentioned in [31].

- Use of a powerful flight board with both better computational power and storage capacity would potentially enhance the performance.

- The flight test area was encircled by buildings. The GPS measurements are greatly affected indoors and in areas encircled by buildings. In different occasions the GPS module did not show reliability in measurements. Better selection of the flight test location in open areas away from buildings re-establishment of the results is another potential work.

- Develop a filteration technique that can be downloaded along with the SMC onto the board

- The simulation results did not test the robustness of the developed controller against different disturbances. Future work could include disturbances for both ϕ, θ and ψ.

Bibliography

[1] Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. *Introduction and Historical Background*, pages 3–11. Modelling and Control of Mini-Flying Machines. Springer London, 01/01 2005.

[2] Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. *The Quad-rotor Rotorcraft*, pages 39–59. Modelling and Control of Mini-Flying Machines. Springer London, 01/01 2005.

[3] Thomas Mueller. On the birth of micro air vehicles. *International Journal of Micro Air Vehicles*, 1(1):1–12, 2009.

[4] MarkW Spong. *Underactuated mechanical systems*, volume 230, pages 135–150. Springer Berlin Heidelberg, 01/01 1998.

[5] Reza Olfati-Saber. *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*. PhD thesis, Massachusetts Institute of Technology, February 2001.

[6] Rong Xu and Ümit Özgüner. Sliding mode control of a class of underactuated systems. *Automatica*, 44(1):233–241, 1 2008.

[7] A. Isidori. *Nonlinear Control Systems*. Springer, London, 1995.

[8] S. Sastry. *Nonlinear Systems*. Springer-Verlag, New York, 1999.

[9] Control affine systems. http://planning.cs.uiuc.edu/node825.html, August 2017.

[10] Quadrotor hovering. https://en.wikipedia.org/wiki/File:Quadrotorhover.svg, August 2015.

[11] Quadrotor roll and pitch. https://en.wikipedia.org/wiki/File:Quadrotorpitch.svg, August 2015.

[12] Quadrotor yaw. https://en.wikipedia.org/wiki/File:Quadrotoryaw.svg, August 2015.

[13] Young Cheol Choi and Hyo Sung Ahn. Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. *Mechatronics, IEEE/ASME Transactions on*, 20(3):1179–1192, 2015.

[14] Jun Li and Yuntang Li. Dynamic analysis and PID control for a quadrotor. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 573–578, 2011.

[15] J. Zhu, E. Liu, E. Guo, and C. Xu. A gradient optimization based PID tuning approach on quadrotor. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, page 1588, May 2015.

[16] M. R. Mehranpour, O. Emamgholi, A. M. Shahri, and M. Farrokhi. A new fuzzy adaptive control for a quadrotor flying robot. In *Fuzzy Systems (IFSC), 2013 13th Iranian Conference on*, pages 1–5, 2013.

[17] Tao Yong, Xie Guang, Chen Youdong, Xiong Hegen, Liu Hui, Zheng Jiaqi, and Gao Jinpeng. A PID and fuzzy logic based method for quadrotor aircraft control motion. *Journal of Intelligent & Fuzzy Systems*, 31(6):2975 – 2983, 2016.

[18] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 153–158, 2007.

[19] T. Madani and A. Benallegue. Control of a quadrotor mini-helicopter via full state backstepping technique. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1515–1520, 2006.

[20] T. Madani and A. Benallegue. Backstepping control for a quadrotor helicopter. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3255–3260, 2006.

[21] A. A. Mian and W. Daobo. Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter. *Chinese Journal of Aeronautics*, 21(3):261–268, 6 2008.

[22] Holger Voos. Nonlinear control of a quadrotor micro-uav using feedback-linearization. In *Proceedings of the 2009 IEEE International Conference on Mechatronics*, 2009.

[23] S. Islam, P. X. Liu, and A. El Saddik. Nonlinear adaptive control for quadrotor flying vehicle. *Nonlinear Dynamics*, 78(1):117–133, 10/01 2014. J2: Nonlinear Dyn.

[24] En-Hui Zheng, Jing-Jing Xiong, and Ji-Liang Luo. Second order sliding mode control for a quadrotor uav. *ISA transactions*, 53(4):1350–1356, 7 2014.

[25] S. Bouabdallah and R. 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, 2005. ID: 1.

[26] Daewon Lee, H. Jin Kim, and Shankar Sastry. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of Control, Automation and Systems*, 7(3):419–428, 06/01 2009. J2: Int. J. Control Autom. Syst.

[27] Hao Liu, Yongqiang Bai, Geng Lu, Zongying Shi, and Yisheng Zhong. Robust tracking control of a quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, 75(3-4):595–608, 09/01 2014. J2: J Intell Robot Syst.

[28] B. Wang and Y. Zhang. An adaptive fault-tolerant sliding mode control allocation scheme for multirotor helicopter subject to simultaneous actuator faults. *IEEE Transactions on Industrial Electronics*, 65(5):4227–4236, May 2018.

[29] T. Madani and A. Benallegue. Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles. In *American Control Conference, 2007. ACC '07*, pages 5887–5892, 2007. ID: 1.

[30] Lénaïck Besnard, Yuri B. Shtessel, and Brian Landrum. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *Journal of the Franklin Institute*, 349(2):658–684, 3 2012.

[31] Ali Karami-Mollaee, Hamed Tirandaz, and Oscar Barambones. On dynamic sliding mode control of nonlinear fractional-order systems using sliding observer. *Nonlinear Dynamics*, 92(3):1379–1393, May 2018.

[32] T. Dierks and S. Jagannathan. Output feedback control of a quadrotor uav using neural networks. *Neural Networks, IEEE Transactions on*, 21(1):50–66, 2010. ID: 1.

[33] M. Santos, V. Lopez, and F. Morata. Intelligent fuzzy controller of a quadrotor. In *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, pages 141–146, 2010.

[34] Hao Liu, Xiafu Wang, and Yisheng Zhong. Quaternion-based robust attitude control for uncertain robotic quadrotors. *Industrial Informatics, IEEE Transactions on*, 11(2):406–415, 2015. ID: 1.

[35] John M. Seddon and Simon Newman. *Basic helicopter aerodynamics.* Wiley, Chichester, West Sussex, UK ; Hoboken, N.J., 3 edition, 2011. [electronic resource] / John Seddon, Simon Newman.; Includes bibliographical references and index.; Mode of access: World wide Web.

[36] J. Gordon Leishman. *Principles of helicopter aerodynamics*, volume 18. Cambridge University Press, New York, NY, 2 edition, 2006. J. Gordon Leishman.; Includes bibliographical references and index.

[37] C. Venkatesan. *Fundamentals of helicopter dynamics.* CRC Press, Taylor & Francis Group, Boca Raton, 2015; 2015. C. Venkatesan.; xx, 318 pages : illustrations ; 24 cm; text; unmediated; volume; Includes bibliographical references (pages 305-309) and index.

[38] A. T. Conlisk. Modern helicopter rotor aerodynamics. *Progress in Aerospace Sciences*, 37(5):419–476, 7 2001.

[39] Carrillo Luis Rodolfo Garca, Dzul Lapez Alejandro Enrique, Rogelio Lozano, and Claude PAgard. *Modeling the Quad-Rotor Mini-Rotorcraft*, pages 23–34. Springer London, 01/01 2013.

[40] Hibbeler R.C. *Engineering Mechanics, Dynamics.* Pearson Prentice Hall, New Jersey, 12th edition, 2010.

[41] Sanjoy Mondal and Chitralekha Mahanta. A fast converging robust controller using adaptive second order sliding mode. *ISA transactions*, 51(6):713–721, 11 2012.

[42] V. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electro-Mechanical Systems.* CRC Press, second edition, 2009.

[43] Sudhir Nadda and A. Swarup. Improved quadrotor altitude control design using second-order sliding mode. *Journal of Aerospace Engineering*, 30(6):04017065, 2017.

[44] A. R. Merheb, H. Noura, and F. Bateman. Active fault tolerant control of quadrotor uav using sliding mode control. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 156–166, May 2014.

[45] J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ, 1991.

[46] Quanum venture fpv deluxe quad copter set. https://hobbyking.com/enus/quanum-venture-fpv-deluxe-quad-copter-set-w-dji-fatshark-afro-parts-pnf.html, April 2018.

[47] UIUC propeller data site. http://m-selig.ae.illinois.edu/props/propDB.html, August 2017.

[48] APC propeller performance data, August 2017.

[49] Turnigy 2200mah 3s 25c lipo pack. https://hobbyking.com/turnigy-2200mah-3s-25c-lipo-pack.html, April 2018.

[50] Pixhawk mini. https://docs.px4.io/en/flightcontroller/pixhawkmini.html, May 2018.

[51] How to choose radio transmitter & receiver for racing drones and quadcopter. https://oscarliang.com/choose-rc-transmitter-quadcopter/, May 2018.

[52] FlySky. *Digital Proportional Radio Control System - FlySky FS-i6 Instructions Manual*.

[53] Weather underground. https://www.wunderground.com/history/, April 2018.

[54] Michael J. Stepaniak, Frank Van Graas, and Maarten Uijt De Haag. Design of an electric propulsion system for a quadrotor unmanned aerial vehicle. *Journal of Aircraft*, 46(3):1050–1058, 05/01; 2016/01 2009. doi: 10.2514/1.38409; 16.

[55] D. L. Gabriel, J. Meyer, and F. Du Plessis. Brushless dc motor characterisation and selection for a fixed wing uav. In *AFRICON, 2011*, pages 1–6, 2011. ID: 1.

Appendices

Appendix A

MATLAB Codes

A.1 MATLAB Snippets

In this section, lists of the critical MATLAB code snippets will be presented. A repository that contains all files used for the generation of the code can be downloaded from GitHub link Thesis Repository.

A.1.1 Saturation Function

```
function y = sat(s)
        k = 0.15;
        y = (2./(1+exp(-k*s))) - 1.0;
end
```

A.1.2 Vehicle Equations

The dynamic equations of the vehicle is expressed in MATLAB level-2 S-function in which the Ordinary Differential Equations (ODEs) that govern the dynamics.

MATLAB S-functions (System functions) provide a powerful mechanism for extending the capabilities of the Simulink® environment. They allow the creation of user-defined time dependent block where ODEs can be solved.

A Level-2 MATLAB S-function consists of a setup routine to configure the basic properties of the S-function, and a number of callback methods that the Simulink engine invokes at appropriate times during the simulation using the "block" argument.

```
function Derivatives(block)
```

```
        x1_dot = xdot;
        x2_dot = (cos(psi)*sin(theta)*cos(phi) +
                        sin(psi)*sin(phi))*(U1/m);
        x3_dot = ydot;

        x4_dot = (sin(psi)*sin(theta)*cos(phi) -
                        cos(psi)*sin(phi))*(U1/m);
        x5_dot = zdot ;
        x6_dot = (cos(theta)*cos(phi))*(U1/m) - g + DistZ;
        x7_dot = phidot;
        x8_dot =  (1/Iy)*((Iy - Iz)*psidot*thetadot -
                        Jr*thetadot*Omega_r + U2*l);
        x9_dot = thetadot;
        x10_dot = (1/Iy)*((Iz - Ix)*psidot*phidot +
                        Jr*phidot*Omega_r + U3*l);
        x11_dot = psidot;
        x12_dot = (1/Iz)*((Ix - Iy)*thetadot*phidot + U4);

end
```

A.1.3 Input Inversion

In this section, the code use to implement input inversion equation discussed in section 4.6 is presented.

```
function [Omega1, Omega2, Omega3, Omega4] = InverseInput(U1, U2, U3, U

        Kt = 1.22641e-04; d = 6.48447e-06;
        Omega2_sqr = 0.25*(((U1-2*U2)/Kt)-(U4/d));
        Omega1_sqr = Omega2_sqr + 0.5*(((U2–U3)/Kt) + (U4/d));
        Omega3_sqr = Omega1_sqr + U3/Kt;
        Omega4_sqr = Omega2_sqr + U2/Kt;
    Omega1_sqr ,Omega2_sqr , Omega3_sqr ,Omega4_sqr );

        if(Omega2_sqr < 0)
        Omega2_sqr = 0;
        end
        if(Omega1_sqr < 0)
        Omega1_sqr = 0;
        end
        if(Omega3_sqr < 0)
        Omega3_sqr = 0;
        end
        if(Omega4_sqr < 0)
        Omega4_sqr = 0;
        end

        Omega1 = Omega1_sqr;
        Omega2 = Omega2_sqr;
        Omega3 = Omega3_sqr;
        Omega4 = Omega4_sqr;

end
```

A.1.4 Controller

The sliding mode control equations are stated in this section. Here, the *Signum* function will be used, however, it can be replaced with the *Saturation* function which is presented in the code snippet in section A.1.1.

```
function [U1, U2, U3, U4] = SMC(States, Constants, Omegar, Track, XYd

        %% x:                  States
        %% Constants:          Controller Constants (To be added)
        %% Omegar:             Gyroscopic term
        %% Track:              Desired states

        %% Data Retrieval %%%%%

        x        = States(1);            % x
        xdot     = States(2);            % xdot
        y        = States(3);            % y
        ydot     = States(4);            % ydot
        z        = States(5);            % z
        zdot     = States(6);            % zdot
        phi      = States(7);            % phi
        phidot   =  States(8);           % phi_dot
        theta    = States(9);            % theta
        thetadot = States(10);           % theta_dot
        psi      = States(11);           % psi
        psidot   = States(12);           % psi_dot
```

```
xdoubledot = XYdoubledot(1);
ydoubledot = XYdoubledot(2);

%% Constants
Ix = Constants(1);
Iy = Constants(2);
Iz = Constants(3);
m = Constants(4);
l = Constants(5);
Jr = Constants(6);
g = Constants(7);
Kt = Constants(8);
d = Constants(9);

%% Desired Track
xd          = Track(1);       phid        = Track(10);
xddot       = Track(2);       phiddot     = Track(11);
xdddot      = Track(3);       phidddot    = Track(12);
yd          = Track(4);       thetad      = Track(13);
yddot       = Track(5);       thetaddot   = Track(14);
ydddot      = Track(6);       thetadddot  = Track(15);
zd          = Track(7);       psid        = Track(16);
zddot       = Track(8);       psiddot     = Track(17);
zdddot      = Track(9);       psidddot    = Track(18);

%% U1 Controller
eps1 = 0.1; eta1 = 0.02; cz = 1.2;      % sign parameters
%       eps1 = 2.2; eta1 = 0.4; cz = 3.0;     % sat parameters
s1 = cz*(zd - z) + (zddot - zdot);
U1 = (eps1*sign(s1) + eta1*s1 + cz*(zddot - zdot) +
      zdddot +g)*(m/(cos(theta)*cos(phi)));

%% U4 Controller
eps2 = 0.6; eta2 = 0.04; cpsi = 2;      % sign parameters
%       eps2 = 1.1; eta2 = 0.04; cpsi = 2;    % sat parameters
s2 = cpsi*(psid - psi) + (psiddot - psidot);
U4 = Iz*(eps2*sat(s2) + eta2*s2 + cpsi*(psiddot - psidot)
                 + psidddot) - (Ix - Iy)*thetadot*phidot;


%% U2 Controller  —— y & phi
```

```
eps4 = 0.2; eta4 = 0.15; c7 = 1.2; c8 = 1.3;
% eps4 = 2.4; eta4 = 0.22; c7 = 1.35; c8 = 1.4;
s4 =   c7*(phiddot - phidot) + c8*(phid - phi);
ldummy =   c7*phidddot + c8*(phiddot - phidot);
U2 = ((Ix/(c7*l))*(ldummy + eps4*sat(s4) + eta4*s4))-
              (1/l)*((Iy - Iz)*psidot*thetadot - Jr*Omegar*thetad

%% U3 Controller
%      eps3 = 0.28; eta3 = 0.25; c3 = 1.1; c4 = 1.25;
eps3 = 2.5; eta3 = 0.28; c3 = 1.3; c4 = 1.45;
s3 =    c3*(thetaddot - thetadot) + c4*(thetad - theta);
mdummy =    c3*thetadddot + c4*(thetaddot - thetadot);
U3 = ((Iy/(c3*l))*(mdummy + eps3*sat(s3) + eta3*s3))
                  - (1/l)*((Iz - Ix)*thetadot*phidot + Jr*Omegar
```

Appendix B

Hardware Setup

B.1 Ground Planner

Pixhawk flight controller can be configured and have its built-in sensors calibrated using a ground planner tool called QGroundControl. QGroundControl provides full flight control and mission planning for any MAVLink enabled drone. It provides configuration for PX4 Pro powered vehicles. Its primary goal is ease of use for first time and professional users. All the code is open-source source, so you can contribute and evolve it as you want.
The features of QGroundControl include:

- Flight support for vehicles running PX4
- Sensor calibration and setup
- Mission planning for autonomous flight

QGroundControl also allows the download and reading of flight log data, provide analysis for the vehicle performance and tune RC parameters.

B.2 ESC Calibration

ESC calibration will vary based on what brand of ESC in use. Afro 20A ESC illustrated in section 6.4 should be calibrated as indicated in its datasheet.
Afro 20A ESC can be manually or automatically calibrated. Automatic calibration which is also known as ”All-at-once” which means that all the four ESCs are calibrated at the same time is easier and faster.
To start automatic calibration the following steps should be followed:

1. Turn on the transmitter with the throttle stick at maximum

2. Connect the Lipo battery. The flight controller should give light and sound pattern to indicate it has entered the ESC calibration mode.
3. Disconnect and connect the Lipo batter. Keeping the throttle to it maximum position
4. Press the safety button to arm the flight controller
5. The ESCs should emit a tone, the regular number of beeps indicating your battery's cell count and then an additional two beeps to indicate that the maximum throttle has been captured.
6. Move the throttle stick down to its minimum position.
7. The ESCs should emit long sound indicating the completeness of ESC calibration

Manual ESC calibration differs greatly depending on ESC type. For manual calibration the ESC datasheet shall be consulted.

Appendix C

Motor Characterization

C.1 Introduction

The quadrotor propulsion system consists of Brushless Direct Current (BLDC) electric motors, electronic speed controller and propellers.

BLDC motors have proven payload efficiency, high power and reliable performance. They replace the field windings with permanent magnets located on the rotor and move the armature windings to the stator. In this manner, the need for mechanical commutation is eliminated reducing noise and electromagnetic interference. The elimination of mechanical commutation allowed greater rotational speeds and eased maintenance.

BLDC motors are synchronous motors which mean that both the magnetic fields generated by the stator and the rotor rotate at the same frequency. BLDC motors do not experience the “slip” that is normally seen in induction motors.

A common type of BLDC motors is the outrunner, which has a spinning outer case as part of the rotor. This type of BLDC motors can provide a greater amount of torque at low speeds without the need for mechanical gearing [54].

BLDC motors exist in different configurations, but the most common configuration is the three phase motor due to its efficiency, low torque and good precision in control.

A disadvantage of the brushless motor is the need to perform the commutation electronically in a separate unit, and this is the function of the Electronic Speed Controller (ESC).

The electronic speed controller is the circuit responsible for varying the speed and the direction of the BLDC motors. It receives a control signal from the quadcopter main controller through the form of Pulse-Width Modulation (PWM) form. The ESC converts the battery pack’s dc voltage to a three-

phase alternating signal that is synchronized to the rotation of the rotor and applied to the armature windings. The motor speed is then proportional to the root-mean square (RMS) value of the armature voltage and is set by the ESC in response to a pulse-width-modulated control signal. The electric motors convert this electric signal into torque.
Figure C.1 shows a schematic of transfer of power in a quadcopter propulsion system.

A vital part of the BLDC motors is the hall sensor which takes part in electronic commutation. Hall sensors are used to determine rotor position in order to energize stator winding in the correct sequence.
Table 1 shows a comparison between BLDC motors and conventional Direct Current (DC) brushed motors.

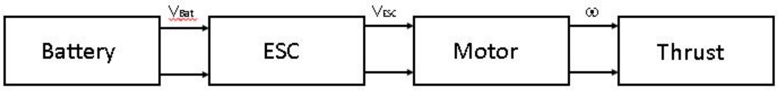

Figure C.1: BLDC motor input and output schematic

Table C.1: comparison between BLDC motors and Brushed DC motors

Feature	**BLDC Motors**	**Brushed DC Motor**
Commutation	Electronic Commutation	Brushed Commutation
Maintenance	Less due to absence of brushes	Periodic due to brush friction
Speed/Torque	Flat at rated speed	Moderately flat but frictionincreases at high speeds reducing useful torque
Speed Range	High due to absence of brushes	Limited mechanically by the presence of brushes
Control	Complex and always needs a controller	Simpler and does not need a controller at fixed speed

Accurate characterization of motors' rotational speeds is crucial for adequate controllability of the vehicle. Figure C.1 is a schematic diagram of the control system in which the importance of motors characterization is apparent.

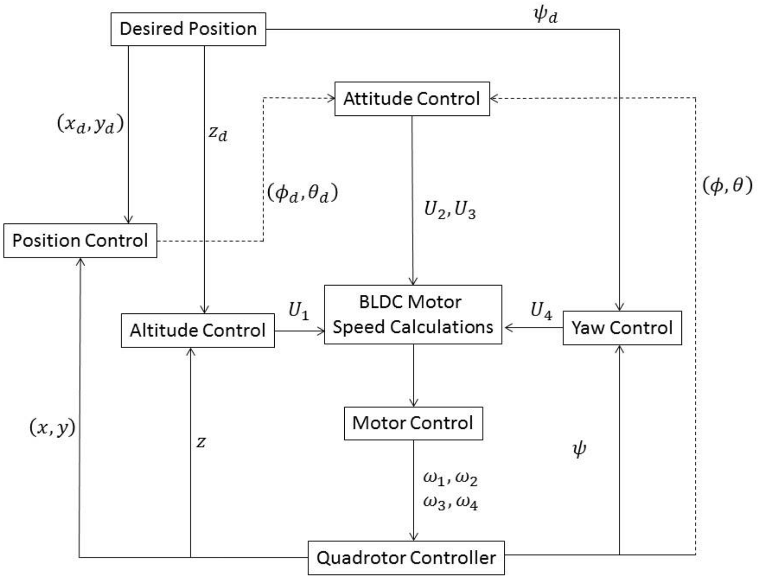

Figure C.2: Schematic diagram of the control system

C.2 Literature Background

Different approaches had been adopted in the literature to characterize the BLDC motors in rotorcraft applications. The characterization process is aimed at providing the relationship between the control input PWM and the output angular velocity ω in different battery conditions for the stable flying. In [13] a rather simple approach is used. An experimental setup is constructed to measure the angular velocity of the BLDC motor in various battery conditions assuming that the supply voltage is well regulated and the torque load is almost not changing. The rotation speed is measured 30 times at every ten PWM step and the mean value is used in actual implementation. This resulted in a link between Pulse Width Modulation (PWM) duty cycle and motor speed values. Finally, a fifth order polynomial is obtained through curve fitting of the data measured. This approach, despite its simplicity, provides a clear and direct way of obtaining the required PWM as a function of the required rotational speed using open loop control.

In [54], the authors modeled every component of the propulsion system separately. This included modeling the motor and its core losses as a function of its rotational speed, the power consumed by the electronic speed controller and the battery voltage degradation. Component losses as a percentage of total power derived from the battery are concluded as a function of the required propeller RPM as shown in figure C.2. In [55], parameterization of a BLDC

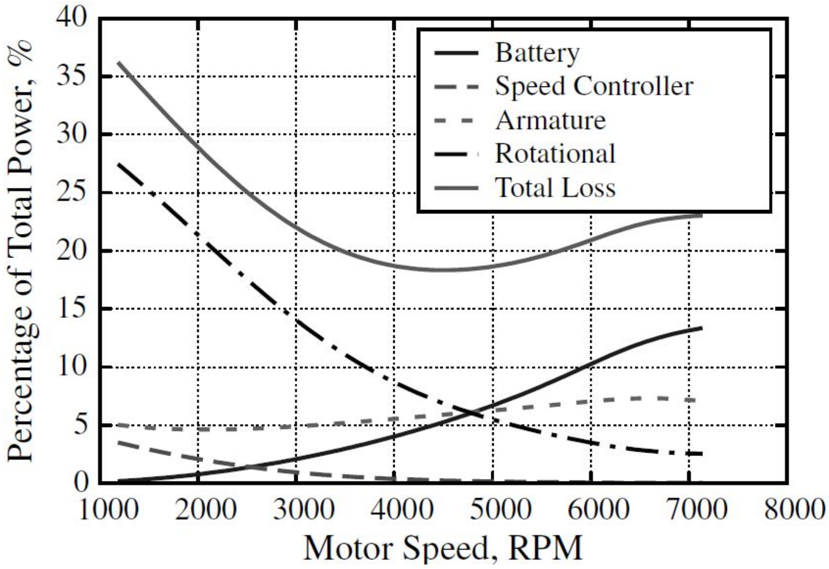

Figure C.3: Component losses as a percentage of total power [54]

motor based on a four-constant model is presented. The proposed model represents the BLDC motors in terms of four constants,$R_m, I_o, K_v and R_1.R_m$ is the combined ESC and winding resistance. I_o is the no-load current at zero battery voltage. K_v is the RPM per back-EMF voltage. The fourth constant, R_1, is used to model the Eddy current and viscous damping losses. The determination of the four constants allows the proper description of the characteristic model equation which is the relationship between efficiency and rotational speed.

C.3 Pixhawk Target Block Library

The Pixhawk Support Package allows the download of Simulink custom controller on the Pixhawk Flight Management Unit.
The Pixhawk Pilot Support Package (PSP) feature allows users to use Simulink models to generate code targeted for the Pixhawk FMUv2 (Flight Management Unit).
The PSP allows to access sensor data and other calculations available to be used in their Simulink model at runtime. Generated code can then be compiled and executed on the Pixhawk platform controlling a multi-rotor airframe.
In the PSP a ready-made motor control block is present which allows the

user to send the appropriate PWM signals out to the PX4 outputs. These are connected to the ESCs which control the motor speeds. The input to the Pixhawk PWM block comes from the control block which determines the values of the output required from each motor. More on the PSP will be illustrated in 7.4.1.

This block allows the user to send the appropriate PWM signals out to the PX4 outputs. The PWM unit value is in micro-seconds which corresponds to the pulse width and can accommodate up to 8 PWM signals.

In this work, Pixhawk PSP PWM block will be used to control the PWM input to the motors. This shall be tested and validated before experimental use.