

American University in Cairo

## AUC Knowledge Fountain

---

Theses and Dissertations

---

2-1-2018

### Quadcopter: Design, modelling, control and trajectory tracking

Khaled Abouelsoud

Follow this and additional works at: <https://fount.aucegypt.edu/etds>

---

#### Recommended Citation

##### APA Citation

Abouelsoud, K. (2018). *Quadcopter: Design, modelling, control and trajectory tracking* [Master's thesis, the American University in Cairo]. AUC Knowledge Fountain.


<https://fount.aucegypt.edu/etds/396>

##### MLA Citation

Abouelsoud, Khaled. *Quadcopter: Design, modelling, control and trajectory tracking*. 2018. American University in Cairo, Master's thesis. *AUC Knowledge Fountain*.

<https://fount.aucegypt.edu/etds/396>

This Thesis is brought to you for free and open access by AUC Knowledge Fountain. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AUC Knowledge Fountain. For more information, please contact [mark.muehlhaeusler@aucegypt.edu](mailto:mark.muehlhaeusler@aucegypt.edu).

 THE AMERICAN UNIVERSITY IN CAIRO  
SCHOOL OF SCIENCES AND ENGINEERING  
ROBOTICS, CONTROL AND SMART SYSTEMS

# **Quadcopter: Design, Modelling, Control and Trajectory Tracking**

By  
Khaled Mahmoud Abouelsoud

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 K. Habib  
Professor, Mechanical Engineering Department

August, 2017

# 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 parents. A special feeling of gratitude to my loving parents; Mahmoud and Amina Abouelsoud for there words of encouragement and for always believing in me. For my dad who supports me perpetually. To my mom who always believed that i can achieve anything i set my mind to. Thank you.

To my charming wife, Injy Diab who always backed me up and gave me endless support. Thanks for bearing me up.

Finally, i would like to dedicate this thesis to my idol who is the main reason after god for changing my career path. Who taught me that anything is possible if i set my mind to achieve it. To my exceptional brother Mohamed Abouelsoud. I shall be forever in your debt.

# **Acknowledgment**

I would like to thank my supervisor, Prof. Dr. Maki Habib, for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

I would also like to thank my fellow graduate student Mohammed Gamal for helping me out in the assembly of the hardware. Also I would like to thank Chimsom Chukwue-meka for helping me in the quadcopter flight tests.

Finally, Completing this work would have been all the more difficult were it not for the support and friendship provided by my fellow graduate student Hashem Rizk.



## Summary

A quadcopter is a type of unmanned aerial vehicles (UAV). The industry of this type of UAVs is growing exponentially in terms of new technology development and the increase of potential applications that may cover construction inspections, search and rescue, surveillance, aerial photography, monitoring, mapping, etc.

A quadcopter is a nonlinear and under-actuated system that introduces complex aerodynamics properties and create challenges which demands the development of new, reliable and effective control techniques to enhance the stability of flight control, plan and track a desired trajectory while minimizing the effect induced by the operational environment and its own sensors. Hence, many control techniques have been developed and researched. Some of such developments work well with the provision of having an accurate mathematical model of the system while other work is associated with a mathematical model that can accommodate certain level of wind disturbances and uncertainties related to measurement noise. Moreover, various linear, nonlinear and intelligent control techniques were developed and recognized in the literature. Each one of such control techniques has some aspect that excels in under certain conditions.

The focus of this thesis is to develop different control techniques that can improve flight control stability, trajectory tracking of a quadcopter and evaluate their performance to select the best suitable control technique that can realize the stated technical flight control requirements. Accordingly, three main techniques have been developed: Standard PID, Fuzzy based control technique that tune PID parameters in real time (FPID) and a Hybrid control strategy that consists of three control techniques:

- (a) FPID with state coordinates transformation
- (b) State feedback
- (c) Sliding mode

The configuration of the hybrid control strategy consists of two control loops. The inner control loop aims to control the quadcopter's attitude and altitude while the outer control loop aims to control the quadcopter's position. Two configurations were used to configure the developed control techniques of the control loops. These configurations are:

- (a) A sliding mode control is used for the outer loop while for the inner loop two control techniques are used to realize it: a Fuzzy gain scheduled PID with state coordinates transformation and a state feedback control.
- (b) Fuzzy gain scheduled PID control is used for the outer loop while for the inner loop two control techniques are used to realize it using the same formation as in (a) above.

Furthermore, in order to ensure a feasible desired trajectory before tracking it, a trajectory planning algorithm has been developed and tested successfully. Subsequently, a simulation testing environment with friendly graphical User Interface (GUI) has been developed to simulate the quadcopter mathematical model and then to use it as a test bed to validate the developed control techniques with and without the effect of wind disturbance and measurement noise.

The quadcopter with each control technique has been tested using the simulation environment under different operational conditions. The results in terms of tracking a desired trajectory shows the robustness of the first configuration of control techniques within the hybrid control strategy under the presence of wind disturbance and measurement noise compared to all the other techniques developed. Then, the second configuration of the control techniques came second in terms of results quality. The third and fourth results in the sequence shown by the fuzzy scheduled PID and the standard PID respectively.

Finally, Validating the simulation results on a real system, a quadcopter has been successfully designed, implemented and tested. The developed control techniques were tested using the implemented quadcopter and the results were demonstrated and compared with the simulation results.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Literature Review . . . . .	2
1.1.1	PID control . . . . .	2
1.1.2	Linear Quadratic Regulator (LQR) . . . . .	3
1.1.3	Sliding Mode Control (SMC) . . . . .	4
1.1.4	Feedback Linearization Control . . . . .	5
1.1.5	Integral Backstepping Control . . . . .	6
1.1.6	Intelligent controllers . . . . .	7
1.1.7	Hybrid control . . . . .	9
1.2	Challenges . . . . .	10
1.3	Objectives . . . . .	11
<b>2</b>	<b>Mathematical Model</b>	<b>13</b>
2.1	Axes and basic setup . . . . .	13
2.2	Rotation and Transformation . . . . .	15
2.3	Kinematic and Dynamic model . . . . .	17
2.3.1	Forces and linear acceleration . . . . .	18
2.3.2	Torques and angular acceleration . . . . .	21
2.4	Control Forces . . . . .	23
2.5	Linear Model . . . . .	24
2.5.1	Linearization of Rotation Matrices . . . . .	25
2.5.2	Linearization of forces . . . . .	26
2.6	State Space and Control Model . . . . .	27
2.6.1	Equations of Motion and Control Inputs . . . . .	28
2.6.2	State Space Representation . . . . .	28
<b>3</b>	<b>Linear Control (PID)</b>	<b>33</b>
3.1	PID Control . . . . .	33

3.2	Effect of Nonlinearities and Noise . . . . .	35
3.2.1	Derivative filtering for PID . . . . .	35
3.2.2	Anti integral windup . . . . .	35
3.2.3	Modified PID controller structure . . . . .	36
3.3	PID Stability Controller . . . . .	36
3.3.1	Attitude and heading control . . . . .	36
3.3.2	Position control . . . . .	39
3.3.3	Position and Attitude control tuning . . . . .	41
<b>4</b>	<b>Fuzzy Gain Scheduled PID Controller</b>	<b>42</b>
4.1	Developing of the Fuzzy Controller . . . . .	43
4.1.1	Membership functions . . . . .	44
4.1.2	Fuzzy rules . . . . .	45
4.1.3	Defuzzification . . . . .	46
4.2	Implementing the Fuzzy Controllerl on the Quadcopter . . . . .	48
<b>5</b>	<b>Hybrid Control Strategy</b>	<b>49</b>
5.1	Hybrid Control Strategy: Configuration 1 . . . . .	49
5.1.1	Inner control loop: FPID with state coordinates transformation and a state feedback control . . . . .	50
5.1.2	Outer control loop: Stability of the internal dynamics using SMC	55
5.2	Hybrid Control Strategy: Configuration 2 . . . . .	59
5.2.1	Outer control loop: Stability of the internal dynamics using FPID	60
<b>6</b>	<b>Trajectory Planning</b>	<b>61</b>
<b>7</b>	<b>Simulation and Results</b>	<b>70</b>
7.1	GUI . . . . .	70
7.2	Parameters . . . . .	71
7.3	Simulation Results:	
	Case 1 without including disturbance or noise . . . . .	72

7.3.1	Altitude and attitude stabilization . . . . .	72
7.3.2	Position tracking . . . . .	81
7.3.3	3D trajectory tracking . . . . .	84
7.4	Simulation Results:	
	Case 2 with the presence of disturbance and noise . . . . .	88
7.4.1	Altitude and attitude stabilization with noise and disturbance . . .	88
7.4.2	Position tracking with noise and disturbance . . . . .	93
7.4.3	3D Trajectory tracking with noise and disturbance . . . . .	96
<b>8</b>	<b>Hardware Development and Implementation</b>	<b>100</b>
8.1	Quadcopter Design and Requirements . . . . .	100
8.2	Hardware Modules . . . . .	103
8.2.1	Quadcopter frame structure. . . . .	103
8.2.2	Brushless Motor Module . . . . .	104
8.2.3	Sensor module . . . . .	107
8.2.4	Wireless transceiver . . . . .	109
8.2.5	Quadcopter controller . . . . .	109
8.2.6	Battery module . . . . .	112
8.3	Quadcopter's Parameters Identification . . . . .	114
8.3.1	Airframe . . . . .	114
8.3.2	Propellers . . . . .	115
8.3.3	Motors . . . . .	117
8.4	Real Time Control Implementation . . . . .	118
8.4.1	Run on target hardware . . . . .	118
8.5	Quadcopter Flight Test . . . . .	122
8.5.1	Flight test procedure . . . . .	122
8.5.2	Flight test results: Calm weather . . . . .	123
8.5.3	Flight test results: Windy weather . . . . .	128
<b>9</b>	<b>Conclusion and Future Work</b>	<b>130</b>

<b>Appendices</b>	<b>133</b>
A   Feedback Linearization . . . . .	134
<b>Bibliography</b>	<b>138</b>

# List of Figures

1.1	Quadcopter [1]	1
2.1	Axis and Coordinates system.	13
2.2	Gimbal lock phenomenon[25]	15
3.1	PID main control scheme	33
3.2	Traditional PID controller	34
3.3	Modified PID controller	36
3.4	Main controller structure	37
3.5	Roll PID control	37
3.6	Pitch PID control	38
3.7	Yaw PID control	38
3.8	Altitude PID control	39
3.9	Longitude PID control	40
3.10	Lattitude PID control	40
4.1	Fuzzy PID control	42
4.2	General fuzzy control structure	43
4.3	Inputs membership functions (e and ed )	45
4.4	Outputs membership functions (Gains)	45
4.5	Control surfaces for $K_P$ and $K_I$	47
4.6	Control surfaces for $K_D$	48
4.7	Fuzzy gain scheduled PID control	48
5.1	Configuration 1 of the hybrid control startegy scheme using FPID with state transformation, state feedback and SMC	50
5.2	Inner control loop for the hybrid control strategy	55
5.3	Configuration 2 of the hybrid startegy scheme using FPID, with state transformation, state feedback and FPID	59

6.1	Trajectory generation for Equation 6.12 . . . . .	64
6.2	Trajectory generation for Equation 6.14 . . . . .	65
6.3	Trajectory generation with a way point . . . . .	65
6.4	X trajectory planning . . . . .	67
6.5	Y trajectory planning . . . . .	68
6.6	Z trajectory planning . . . . .	68
6.7	3D trajectory generation . . . . .	69
6.8	Trajectory planning added to the system . . . . .	69
7.1	Graphical User Interface . . . . .	71
7.2	Altitude stability response using PID control . . . . .	73
7.3	Attitude stability response using PID control . . . . .	74
7.4	Altitude stability response using FPID control . . . . .	75
7.5	Attitude stability response using FPID control . . . . .	75
7.6	$x$ and $y$ position with chattering effect. . . . .	76
7.7	Roll and pitch with chattering effect. . . . .	77
7.8	$x$ and $y$ position with reduced chattering effect. . . . .	77
7.9	Roll and pitch with reduced chattering effect. . . . .	78
7.10	Altitude stability response using first hybrid control configuration . . . .	78
7.11	Attitude stability response using first hybrid control configuration . . . .	78
7.12	Altitude stability response using first hybrid control configuration with chattering . . . . .	79
7.13	Attitude stability response using first hybrid control configuration with chattering . . . . .	79
7.14	Altitude stability response using second hybrid control configuration . . .	80
7.15	Attitude stability response using second hybrid control configuration . . .	80
7.16	Altitude stability comparison without disturbances . . . . .	81
7.17	Attitude stability comparison . . . . .	81
7.18	Longitude and latitude tracking using PID control . . . . .	82
7.19	Longitude and latitude tracking using FPID control . . . . .	82



7.20	Longitude and latitude tracking using first control configuration . . . . .	83
7.21	Longitude and latitude tracking using second hybrid control configuration	83
7.22	Longitude and latitude tracking response (comparison) . . . . .	84
7.23	Spiral trajectory tracking response with each of the developed control techniques . . . . .	85
7.24	Spiral trajectory tracking response comparison between all the control techniques . . . . .	86
7.25	Random trajectory tracking with each of the developed control techniques.	87
7.26	Random trajectory tracking comparison between all the control techniques	87
7.27	Altitude stability response using PID control . . . . .	88
7.28	Attitude stability response using PID control . . . . .	89
7.29	Altitude stability response using FPID control . . . . .	89
7.30	Attitude stability response using FPID control . . . . .	90
7.31	Altitude stability response using first hybrid control configuration . . . .	90
7.32	Attitude stability response using first hybrid control configuration . . . .	91
7.33	Altitude stability response using second hybrid control configuration . . .	91
7.34	Attitude stability response using second hybrid control configuration . . .	92
7.35	Altitude stability comparison between all the control techniques . . . . .	92
7.36	Attitude stability comparison between all the control techniques . . . . .	93
7.37	Longitude and latitude tracking using PID control . . . . .	94
7.38	Longitude and latitude tracking using FPID control . . . . .	94
7.39	Longitude and latitude tracking using first hybrid control configuration . .	95
7.40	Longitude and latitude tracking using second hybrid control configuration	95
7.41	Longitude and latitude tracking comparison between all the control techniques . . . . .	96
7.42	Spiral trajectory tracking with each control technique. . . . .	97
7.43	Spiral trajectory tracking comparison between all the control techniques .	98
7.44	Random trajectory tracking with each control technique) . . . . .	99
7.45	Random trajectory tracking comparison between all the control techniques	99

8.1	The layout of the quadcopter system. . . . .	102
8.2	The developed quadcopter. . . . .	102
8.3	Frame specifications. . . . .	104
8.4	Emax BL2220 Motor. . . . .	105
8.5	Turningy ESC. . . . .	107
8.6	Quantom GPS module . . . . .	108
8.7	3DR telemetry . . . . .	108
8.8	FlySky FS-i6 RC. . . . .	109
8.9	APM 2.6. . . . .	110
8.10	APM 2.6 board layout. . . . .	110
8.11	Lithium Polymer Battery. . . . .	113
8.12	Battery monitor module. . . . .	113
8.13	Power Distribution Board. . . . .	114
8.14	Thrust Coefficient vs Propeller Speed Plot.[41] . . . . .	116
8.15	Power Coefficient vs Propeller Speed Plot. [41] . . . . .	117
8.16	Main simulink model layout for hardware implementation. . . . .	120
8.17	Hardware inputs subsystem. . . . .	120
8.18	Control subsystem. . . . .	121
8.19	Hardware outputs subsystem. . . . .	122
8.20	Quadcopter flight test. . . . .	123
8.21	Roll plot (PID) . . . . .	124
8.22	Roll plot (FPID) . . . . .	124
8.23	Roll plot (FL) . . . . .	125
8.24	Pitch plot (PID) . . . . .	125
8.25	Pitch plot (FPID) . . . . .	126
8.26	Pitch plot (FL) . . . . .	126
8.27	Yaw plot (PID) . . . . .	127
8.28	Yaw plot (FPID) . . . . .	127
8.29	Yaw plot (FL) . . . . .	127
8.30	Roll plot (comparison) . . . . .	128

8.31 Pitch plot (comparison) . . . . .	129
8.32 Yaw plot (comparison) . . . . .	129

# List of Tables

3.1	PID gains effect . . . . .	41
4.1	Linguistic variables . . . . .	44
4.2	Fuzzy rules for $K_P$ and $K_I$ . . . . .	47
4.3	Fuzzy rules for $K_D$ . . . . .	47
6.1	Desired way points . . . . .	66
7.1	Parameters . . . . .	72
7.2	Attitude and Altitude initial and desired values . . . . .	72
7.3	Gains and Filter coefficient Parameters . . . . .	73
7.4	Outputs gain range . . . . .	74
7.5	Inputs gain range . . . . .	74
7.6	Outputs gain range . . . . .	75
7.7	Inputs gain range . . . . .	76
7.8	Sliding Mode control simulation parameters . . . . .	76
7.9	Sliding Mode control refined simulation parameters . . . . .	77
8.1	Frame Specification . . . . .	103
8.2	Motors Specification . . . . .	106
8.3	Turningy ESC specification . . . . .	106
8.4	APM 2.6 Specification . . . . .	111
8.5	Identified quadcopter airframe parameters. . . . .	115
8.6	(10 x 4.7) Propeller Thrust & Power Coefficients at Different Speed.[41] .	116
8.7	Motor Parameters . . . . .	117

# Nomenclature

## List of Symbols

Symbol	Description
$A$	System matrix
$A_r$	Cross section of the propeller
$B$	Input matrix
$b$	Drag coefficient
$C$	Output matrix
$D$	Feed forward matrix
$e$	Error
$f_b$	Linear force
$f_{lift}$	Lift force
$f_w$	Disturbance forces
$g$	Gravitational acceleration
$i$	Induced inflow constant
$I_{xx}$	Moment of inertia about the x axis
$I_{yy}$	Moment of inertia about the y axis
$I_{zz}$	Moment of inertia about the z axis
$j$	Total inertia of the quadcopter
$j_r$	Rotor's inertia
$k_d$	Derivative gain
$k_f$	Thrust/Lift coefficient
$k_i$	Integral gain
$k_p$	Proportional gain
$l_a$	Rotor's lever length to the quadcopter's CG
$m$	Total mass of the quadcopter
$R$	Rotation matrix

Symbol	Description
$r$	Propeller's radius
$R_a$	Armature current
$R_w$	Angular rate transformation matrix
$S$	Sliding surface
$T$	Torque
$T_w$	Disturbance moments
$U1$	Control force
$U2$	Control force
$U3$	Control force
$U4$	Control force
$v$	Linear velocity relative to the earth frame
$V$	Lyapunov function
$V_b$	Linear velocity relative to the body frame
$x_d$	Desired x position
$y_d$	Desired y position
$z_d$	Desired z position
$\omega$	Angular velocity relative to the earth frame
$\omega$	Rotor speed
$\omega_b$	Angular velocity relative to the body frame
$\phi$	Roll angle
$\phi_d$	Desired roll angle
$\psi$	Yaw angle
$\psi_d$	Desired yaw angle
$\rho$	Density of air
$\theta$	Pitch angle
$\theta_d$	Desired pitch angle

## Acronyms

Acronym	Description
APM	Ardupilot mega
FPID	Fuzzy logic PID
GPS	Global positioning system
$H_\infty$	H infinity
IMU	Inertial measurement unit
LQ	Linear quadratic
LQR	Linear quadratic regulator
LQG	Linear quadratic gaussian
PD	Proportional derivative
PID	Proportional integral derivative
PWM	Pulse width modulation
RC	Remote controller
SMC	Sliding mode control
UAV	Unmanned aerial vehicle
VTOL	Vertical takeoff and landing

# Chapter 1

## Introduction

Quadcopter is one type of the unmanned aerial vehicles (UAV) and it's also named as Vertical Take OFF and Landing (VTOL) aircraft. It is used in many applications such as filming, construction inspections, search and rescue, surveillance, aerial photography, mapping, traffic monitoring, crop monitoring, fire detection and many other uses.[2][3]



Figure 1.1: Quadcopter [1]

A quadcopter is a small aircraft with four rotors. This gives it many advantages such as simplicity, maneuverability and agility. In addition, it can be customized easily for different applications. It consists of four rotors which control the quadcopter by adjusting the angular velocities of each rotor. In order to achieve control of the roll, pitch, yaw and lift it uses 2 sets of identical propellers, 2 clockwise and 2 counterclockwise. Thrust of each rotor is manipulated by changing the speed of each one. By doing this control of the system is achieved. [2][3][4]. The quadcopter consists of six degrees of freedom. They are the translation motion in x,y and z in addition to the rotational motion in Roll( $\phi$ ), Pitch( $\theta$ ) and Yaw( $\psi$ ). Those degrees of freedom are considered to be the outputs of the



system. On the other hand, The total thrust of the rotors and the torques produced around each axis are considered to be the inputs to the system. Consequently, the quadcopter system is considered as an under actuated system since the number of the inputs are less than the number of the outputs. Hence, direct control of each degree of freedom is not possible. Moreover, since the quadcopter is relatively small in size it is prone to be influenced by wind effects or disturbances. Furthermore, the dynamics of the quadcopter is highly nonlinear with some uncertainties which leads to an inaccurate mathematical model of the system. Hence, controlling a quadcopter is a cumbersome task considering the fact it needs to deal with non linearities of the system and disturbances.[2][3][4][5]

Many control techniques have been suggested for the control of the quadcopter. The main task each technique is trying to achieve is to stabilize the attitude of the quadcopter while following a given translational trajectory. Some control techniques are based on a linearized mathematical model for the quadcopter and applying a linear controller such as PID control. While other control techniques deals with the nonlinear model by applying a non linear controller such as backstepping and Sliding Mode control technique. A more modern approaches are the adaptive controllers such as fuzzy logic and neural networks. Some also introduced the idea of a hybrid control technique which combines between two of the techniques stated above. Each control technique suggested has its own pros and cons [2][3][4][6][7].

## **1.1 Literature Review**

### **1.1.1 PID control**

The PID controller is the most applied controller in the industry. This is due to its simplicity. It is also easy to adjust the gain parameters of the controller whose parameters are chosen by experience, trial and error and some new techniques such as fuzzy logic [3] and neural networks. However, Due to the nonlinearity and uncertainties found in the

quadcopter model the PID limits the performance of the quadcopter.

A PID controller was applied to a quadcopter for the purpose of stabilizing the attitude [4]. The model was linearized around the hover position in order to design the controller. Hence, the gyroscopic effects has not been taken into consideration in the controller design. The simulation on Simulink showed satisfactory results in the sense of altitude stability. The quadcopter attitude stabilizes itself after 3 seconds. The result was validated by applying the controller into a real system. The results were consistent with the one acquired from the simulation. The controller is efficient for the hover position. On the other hand, this is only valid for hovering (altitude control) as for the attitude control the controller was not robust for any sudden perturbations in the attitude.

Another PID controller was developed for the purpose of stabilizing the attitude and altitude [6]. The quadcopter was simulated to test the applied developed control technique. The altitude and attitude stabilized after 5 seconds while the position of  $x$  and  $y$  had a large steady state error. This is mostly due to an inaccurate model and the fact that the system is an under-actuated system with no control over the  $x$  and  $y$  positions.

The main disadvantage of the PID control is that it simplifies the dynamics which results in limiting the control and maneuverability of the quadcopter also the gains must be chosen on the basis of experience and trial and error. On the other hand, it is easy and simple to implement.

### **1.1.2 Linear Quadratic Regulator (LQR)**

Linear quadratic regulator is an optimal control technique which is main function to reduce the cost function. In other words, it minimizes the deviation from the desired values. This is carried out by adjusting the weighing factors.

An LQR controller was applied on a quadcopter and compared with a PID controller

[8]. The LQR was applied on the full dynamic model while the PID controller was applied on the simplified dynamic model. The results for both systems were close through simulation only. However, the LQR controlled system showed slightly better results and that's mainly because it deals with the full dynamic model.

A comparison was made between an LQR controller and an adaptive controller based on the Lyapunov stability. They were both tested under uncertainties which is a loss in thrust in this case through simulation only. Both showed good response and stability however they both have a small steady state error while the adaptive controller had a better response in the presence of uncertainty or loss of thrust where it returns back to the desired position smoother and faster compared to the LQR controller.

In addition, An LQR controller was applied on a quadcopter to follow a desired path in [9]. The developed controller was tested using a real quadcopter under the influence of wind and measurement noise and it showed good results in terms of path following. However, the controller response was not accurate when there multiple obstacles were introduced in the desired path.

The used LQR controllers shows an average results in the stability of the quadcopter. It also has a drawback in choosing the weighing factors or matrices which needs iterative simulations and experience by the designer.

### **1.1.3 Sliding Mode Control (SMC)**

Sliding mode control is a nonlinear control algorithm that works by applying a discontinuous control signal to the system to command it to slide along a prescribed path.

An adaptive sliding mode controller was implemented on a quadcopter for attitude stabilization and altitude tracking in[5]. The controller proved to be very robust with a very small steady state error and small settling time through simulation only

Another approach was carried out in [7]. The model was tested first without uncertainties and noise through simulation. Later white Gaussian noise and uncertainties of 35% on mass and 15% on inertia matrix were added. However, the attitude control was not tested.

A sliding mode controller was implemented by Rong Xu and Umit Ozguner in [10] in order to follow a desired position under the effect of uncertainties. The system was divided into two subsystems the first for controlling the altitude ( $z$ ) and yaw angle ( $\psi$ ), while the second subsystem for controlling the positions  $x$  and  $y$  and the roll ( $\phi$ ) and pitch ( $\theta$ ) angles. Chattering effect was reduced with continuous approximation by introducing a saturation function instead of using the sign function in the sliding mode control law. The quadcopter was tested to follow a desired position under the effect of uncertainties through simulation only. The results showed good stability and robustness against uncertainties such as wind and measurement noise

The main advantage for the SMC controller is that it is very robust against model inaccuracy, uncertainty, noise and tracking. However it does not show good results specifically in the attitude control (roll and pitch).

#### **1.1.4 Feedback Linearization Control**

Feedback linearization is carried out by transforming a nonlinear system to an equivalent linear system. In order to do that the model must be very accurate. If the model is not accurate the control would be very poor.

A feedback linearization adaptive controller was introduced in [8] to stabilize the system and act as trajectory follower controller. The controller proved to be very efficient in trajectory and path following while having poor results especially in the presence of noises. The system was tested through simulation only.

Another comparison took place between a feedback linearization controller and a sliding mode controller in [11]. The feedback linearization proved to be very sensitive against noises and uncertainties while the sliding mode controller proved to be very robust against noises and uncertainties. This is mainly because the feedback linearized system has very simplified dynamics which doesn't account for many of the model aspects. The comparison was done through simulation only.

Feedback linearization controller is very efficient in trajectory planning. However, it's not good in stabilizing the quadcopter. Thus, combining the feedback linearization with another controller such as SMC where the feedback linearization role would be trajectory planning while the SMC role would be stability would probably show promising results.

### **1.1.5 Integral Backstepping Control**

Backstepping controller is a nonlinear controller. It divides the system into several subsystems and in a recursive steps it deals with each subsystem independently to achieve stability of each subsystem.

An integral Backstepping controller was applied on a quadcopter to stabilize the system and trajectory planning in [12]. The stabilizing/tracking controller was tested with external disturbance (wind) to fly in a helical trajectory through simulation. It showed a very robust result even with the external disturbances and had a very small steady state error.

A backstepping technique was used to improve a quadcopter stability [13]. The quadcopter was modeled using quaternion transformation instead of using Euler transformation method. The quaternion modelling technique was carried out to avoid singularity and to make the quadcopter able to exceed 90 degrees rotations. Finally, Lyapunov stability theory was used to ensure that the backstepping controller achieved system stability. The attitude showed good results in terms of stability through simulation only.

The integral Backstepping controller showed promising results in trajectory planning as well as stability of the system. It also computationally inexpensive since it divide the system into simple subsystems. However it's complicated to apply.

### **1.1.6 Intelligent controllers**

#### **A Fuzzy Logic controller**

Fuzzy logic is considered as an intelligent controller. It suits very well the nature of any nonlinear systems with model imprecisions and uncertainties. Hence, it should be able to adapt well with the nature of the quadcopter dynamic model.

A fuzzy logic controller was applied on a quadcopter in [14]. Six fuzzy controllers were implemented. One for each degree of freedom. Sensor noise and wind were introduced and the system was tested using two inference engines; Mamdani and TSK. Both converged to the desired set points. However, the Mamdani converged faster with a smaller steady state error even under noise and disturbance. The testing was carried out using simulation and a real quadcopter.

Another implementation of a fuzzy logic controller was implemented in [15]. It was only tested to control the stability of the quadcopter through simulation and using a real quadcopter. However, it didn't show satisfactory results especially in the presence of disturbance. This is mainly due to hardware problems that was faced were the position controller could not be uploaded to the real quadcopter.

An intelligent controller used to control the altitude and attitude only [16]. Testing the controller was carried out through simulation only. The controller showed good results in terms of the attitude stability for the angles roll, potch and yaw. Moreover, a good tracking response for the altitude  $z$  was shown in the simulation results.

Fuzzy logic is a very promising control technique that can be used to control the quadcopter. It can adapt with the nonlinearity of the quadcopter model very well and can also adapt with external disturbances and wind. However, very few researches were done in this area.

## **B Neural Network control**

Neural network control is adopted from the brain of the human, they mimic the function of the neurons found in the brain. It is similar to the fuzzy controller in being an intelligent and adaptive controller. It also doesn't require a specific model instead it learns by trial (learning phase). So by testing and trial the controller gets more robust.

A neural network trajectory planning controller was implemented in [17]. The quadcopter was given a destination and the neural network controller managed to plan a trajectory and reach the required destination. However at the initial trials (learning phase) the result was not satisfactory but with each trial the result was more satisfactory until reaching a stable condition. The result was satisfactory. However, the controller was tested only to track a given trajectory and not focusing on stabilizing the quadcopter.

Another adaptive neural network controller was introduced in [18]. Two single hidden layers were used instead of 1 and the learning algorithm of those layers were based on Lyapunov stability. A sinusoidal disturbance was introduced and the controller were able to deal with those disturbances through simulation.

A neural network controller was developed aiming to learn the unmodeled dynamics of the quadcopter[19] . Control of all the degrees of freedom was achieved using the neural network controller. Finally, Lyapunov stability theory was used to proof the stability of the system. The system was tested using simulation only and showed good tracking results.

Neural networks is an intelligent adaptive controller which can learn the behavior of

the system with each iteration. It does not requires a dynamic model for the quadcopter and can adapt with disturbances.

### 1.1.7 Hybrid control

Hybrid control is a mix between two controllers. For instance, A PID controller could be combined with a fuzzy logic controller. The PID role would be to stabilize the quadcopter while the fuzzy logic control would be to follow a trajectory.

Another example was introduced in [20], where a Backstepping controller was combined with a neural network controller. The Backstepping controller role was to stabilize the quadcopter while following a trajectory while the neural network role was to compensate the uncertainty of the dynamical model of the quadcopter. Hence, using the neural network there was no need to derive a model for the system.

A hybrid technique using sliding mode control with fuzzy technique to tune it. In addition, a backstepping controller was applied[21]. A sliding surface was obtained using a backstepping controller, then a sliding mode controller was applied to enforce the system on the sliding surface obtained. The fuzzy controller was then used to eliminate the chattering effect. The technique was tested through simulation only and the technique proved to reduce the chattering effect. Moreover, the quadcopter was able to track a given desired position while stabilizing the attitude.

A sliding mode controller was implemented along with a partial feedback linearization technique in [22]. The sliding mode controller aimed to stabilize the internal dynamics or the position  $x$  and  $y$ . Moreover, the dynamics of the quadcopter were simplified by assuming the yaw angle ( $\psi$ ) equal zero. Testing was carried out under the effect of disturbances through simulation only. The controller proved to be robust and able to track the desired trajectories in the  $x$  and  $y$  positions.



Hybrid control theory is promising for controlling quadcopters, since it combines the best of multiple controllers to achieve trajectory tracking while stabilizing of the attitude for a quadcopter.

## 1.2 Challenges

The quadcopter system is an under actuated system having four control inputs while consisting of six degrees of freedom. This results in hindering a full control of the system. Many linear and nonlinear controllers were proposed to control the quadcopter as efficient as possible and tackle the under actuation problem such as in [14]. However the dynamic model of the quadcopter is a nonlinear model with many uncertainties. Thus, the need for a controller to deal with this nonlinearity and uncertainties.

As for the nonlinear controllers. Each has its point of strength and weakness. For instance, The full state feedback linearization provide very good stability and trajectory planning however its not robust against disturbance which is the main drawback also it needs a very accurate model. On the other hand there is the Backstepping controller which showed very good results especially under disturbance but its main drawback is its complexity.

Intelligent controllers such as fuzzy logic and neural network on the contrary don't need an accurate model. They are very adaptive to disturbances and easy to implement on the system. They are very efficient in trajectory control which is their main advantage. However, since they disregard the model this hinders full stabilization of the system.

According to the aforementioned discussion a hybrid controller would be the optimum controller in the case of a quadcopter. This is mainly because one controller could be specifically designed to stabilize the quadcopter attitude and altitude while the other controller would be designed specifically for trajectory planning. For instance, an integral Backstepping controller could be used for stability where it excels in this area while

a fuzzy logic controller act as the trajectory planning tool. Another example would be using a PID controller with automatic gaining for stability while using another method for trajectory tracking such as sliding mode control.

The research challenges in the field are highlighted as follows:

1. The nonlinearity of the quadcopter model which makes it challenging to model the system and apply a controller.
2. Designing a controller and deciding which technique should be used to control the quadcopter (Linear control, nonlinear control, intelligent control, Hybrid control).
3. The quadcopter is an under-actuated system.
4. A coupled nonlinear mathematical model.
5. The quadcopter is influenced by uncertainties such as wind disturbance and noise measurement.

### **1.3 Objectives**

The aim of this thesis is to develop different control techniques that can improve flight control stability, trajectory tracking of a quadcopter and evaluate their performance to select the best suitable control technique. Moreover, to ensure a feasible desired trajectory before tracking it; trajectory planning algorithm is proposed. In addition, a simulation environment facilitated by a graphical user interface (GUI) is presented. Finally, a physical quadcopter is designed and implemented to test the control techniques for validation of the simulation results.

Therefore, the main objectives of the thesis are:

1. Derive a mathematical model for the quadcopter system.
2. Develop multiple controllers for attitude, altitude and position control and compare their results.

3. Develop an algorithm for trajectory planning.
4. Develop a simulation environment facilitated by a GUI.
5. Test and validate the developed controllers using the simulation environment.
6. Design a quadcopter and then implement the developed controllers on it to test and validate the simulation results.

# Chapter 2

## Mathematical Model

In this chapter the mathematical model of the quadcopter is derived using Euler transformation method. The mathematical model describes how the quadcopter moves due to the forces and torques created by the four rotors. This model is used to provide a controller to stabilize the quadcopter and trajectory tracking based on the positions, velocities and accelerations given by the mathematical model through IMU or any other kind of sensors.

The outline of the quadcopter with its body frame and earth frame is shown in Figure 2.1.

### 2.1 Axes and basic setup

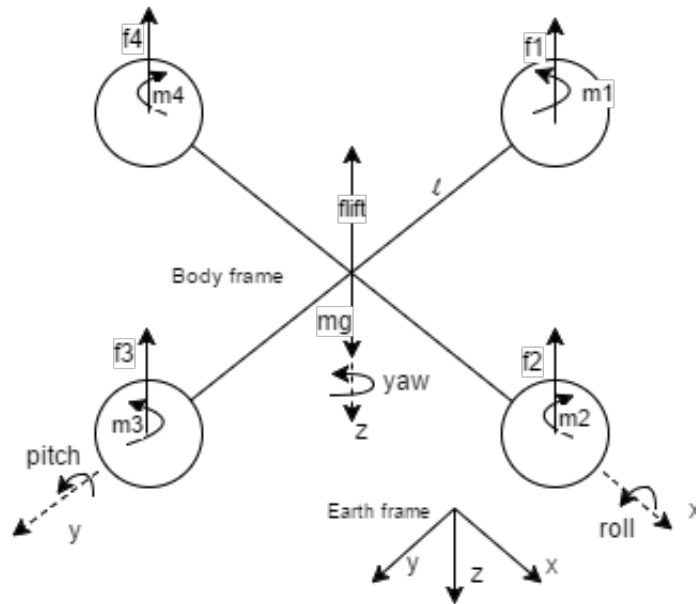


Figure 2.1: Axis and Coordinates system.

The earth frame is assigned as the inertial frame, with gravity pointing in the negative  $z$  direction. The body frame is defined by the orientation of the quadcopter, with the rotors

axes pointing in the positive  $z$  direction and the arms pointing in the  $x$  and  $y$  directions. All the propellers axis are parallel to each other. Rotors 1 and 3 rotate anticlockwise while rotors 2 and 4 rotate clockwise. This is done in order to cancel the yawing effect (Rotation around the  $z$ -axis). Two Degrees of freedom are indirectly controlled which are the longitude ( $x$ ) and latitude ( $y$ ) by adjusting the roll ( $\phi$ ) and pitch ( $\theta$ )[23][24]. Moreover the other four degrees of freedom will be directly controlled and they are:

- Hover (translation along  $z$ -axis): The four rotors must have equal thrust which is equal to the total weight of the quadcopter in order to stabilize it and hover. This is due to the lift force generated by the four rotors.
- Roll  $\phi$  (around  $x$ -axis): rotors 2 and 4 must have different speed. This is due to the torque generated by rotors 2 and 4.
- Pitch  $\theta$  (around  $y$ -axis): rotors 1 and 3 must have different speed. This is due to the torque generated by rotors 1 and 3.
- Yaw  $\psi$  (around  $z$ -axis): rotors 1-3 and rotors 2-4 must have different speed. This happens due to the fact that rotors 1-3 rotate clockwise while rotors 2-4 rotate anticlockwise which make the overall torque unbalanced resulting in the quadcopter turning on itself (around  $z$ -axis)

Thus Newton-Euler formulation is used to describe the forces that act on the quadcopter[2][6][24]. In order to represent a mathematical model for the quadcopter some Equations must be derived to eventually formulate the Equations of motion which will help in simulating the quadcopter and add a controller to the system. These required Equations are:

- The rotation matrix from the body frame to the earth frame and vice versa using Euler or quaternions. It will be used for linear velocity transformation.
- The angular rates transformation matrix to transform the body angular velocities to the earth angular velocities.

- Induced inflow force.
- Gravitational pull vector.
- Forces and torques generated by the four rotors:
  1. Thrust force. [U1]
  2. Rolling moment (Torques). [U2, U3 and U4].

## 2.2 Rotation and Transformation

In this section, a transformation from the body frame to the earth frame and vice versa using Euler angles is described. Euler angle method is widely used and clearly understood and interpreted. However, the transformation matrix which is a 3x3 matrix includes a trigonometric function which results in a nonlinearity and also has a singularity. This causes the phenomenon known as gimbal lock where the system loses one of its degrees of freedom. This is avoided by limiting the attitude angles not to reach 90 degrees.

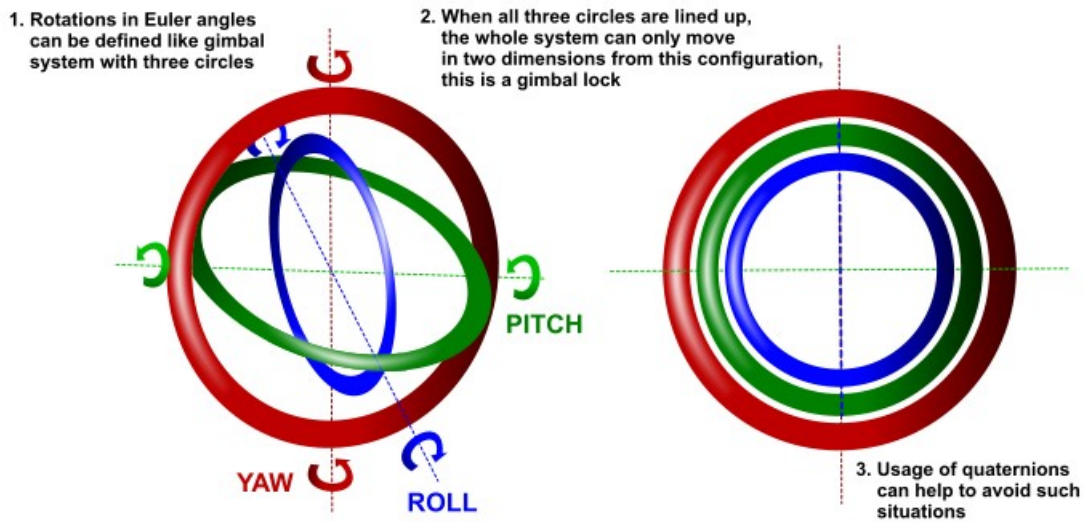


Figure 2.2: Gimbal lock phenomenon[25]

Euler angles were introduced by Leonhard Euler to describe the orientation of a rigid body. They are used to describe the orientation of a frame relative to another. Moreover,

it is used to transform a point from one frame to another frame. The Euler angles are denoted by  $\phi\theta\psi$ .

The transformation matrices representing the rotation about each principle axis is shown in Equations (2.1), (2.2) and (2.3).

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.1)$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.2)$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

The transformation matrix from the body frame to the earth frame is the product of the three rotations in the sequence zyx:

$$R(\psi, \theta, \phi) = R_z(\psi)R_y(\theta)R_x(\phi) \quad (2.4)$$

Thus the matrix describing the rotation from body frame to earth frame is shown in (2.5),

$$R(\psi, \theta, \phi) = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.5)$$

Where  $c = \cos$ ,  $s = \sin$ .

The angular rates transformation matrix from body angular rates to Euler angular rates or velocity ( $\omega$ ) can be described as [26]:

$$\omega = R_x(\phi)R_y(\theta) \begin{bmatrix} 0 \\ 0 \\ t \end{bmatrix} + R_z(\phi) \begin{bmatrix} 0 \\ s \\ 0 \end{bmatrix} + \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi) \cos(\theta) \end{bmatrix} \begin{bmatrix} e \\ s \\ t \end{bmatrix} \quad (2.6)$$

Taking the inverse gives,

$$Rw = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \quad (2.7)$$

## 2.3 Kinematic and Dynamic model

In this section the goal is to describe the quadcopter Equation of motion with respect to the earth frame using the transformation matrices (2.5) and (2.7). The quadcopter linear and angular motion relative to the earth frame is denoted as  $[x \ y \ z \ \phi \ \theta \ \psi]$  while the linear and angular velocities relative to the body frame is denoted as  $[u \ v \ w \ e \ s \ t]$ .

From Equations (2.5) and (2.7) linear and angular velocities could be written as [1][4][19]:

$$v = R.v_B \quad (2.8)$$

$$\omega = Rw.\omega_B \quad (2.9)$$

where  $v$  is the linear velocity vector relative to the earth frame  $[\dot{x} \ \dot{y} \ \dot{z}]$ ,  $v_B$  is the linear velocity vector relative to the body frame  $[u \ v \ w]$ ,  $\omega$  is angular velocity vector relative to the earth frame  $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]$  and  $\omega_B$  is the angular velocity vector relative to the body frame  $[e \ s \ t]$ .

Thus the kinematic model can be written as:

$$v = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R.v_b = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ \psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.10)$$

$$\omega = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = Rw.\omega = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} e \\ s \\ t \end{bmatrix} \quad (2.11)$$



In order to represent the dynamical model of the system the total forces acting on the quadcopter should be accounted for. This is done using Newton's law for linear forces and Euler's law for torques. Newton's law is applied on the quadcopter as shown in Equation (2.12) to calculate the linear forces acting on the quadcopter [1][4][19].

$$f_B = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = m(\dot{v}_B + \omega_B \times V_B) \quad (2.12)$$

Where  $f_B$  is the total force acting on the quadcopter and  $m$  is the mass of the quadcopter

As for the total torques (T) acting on the quadcopter are calculated using Euler's law shown in Equation (2.13) [1][4][19].

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = j.\dot{\omega}_B + \omega_B \times (j.\omega_B) \quad (2.13)$$

where  $j$  is the moment of inertia for the quadcopter:

$$j = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.14)$$

### 2.3.1 Forces and linear acceleration

Linear acceleration happens due to the net forces acting on the quadcopter. So it is essential to calculate all the linear forces affecting the quadcopter to derive eventually the linear translation. The forces derived is based on Newton second law which states that:

$$f_B = m(\dot{V}_B + \omega_B \times V_B) \quad (2.15)$$

Thus,

$$\begin{aligned} f_x &= m(\dot{u} + ew - tv) \\ f_y &= m(\dot{v} - ew + tu) \\ f_z &= m(\dot{w} + ev - tu) \end{aligned} \quad (2.16)$$

### A Lift force

The main force which is provided by the four rotors is the lift force  $f_{lift}$  that is also considered as the first control input (U1) which is responsible for the altitude (Translation along the Z-axis) [2].

$$f_{lift} = c_t \rho A_r r^2 (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) = U1$$

$$f_{lift} = k_f (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) = U1 \quad (2.17)$$

where  $c_t$  is the lift coefficient,  $\rho$  is the density of the air,  $A_r$  is the cross sectional area of the propeller's rotation,  $r$  is propeller's radius and  $\Omega$  is the rotor speed. The above parameters are lumped into one parameter that is denoted by  $k_f$ . Thus. The lift force within the body frame is

$$f_{lift-B} = \begin{bmatrix} 0 \\ 0 \\ k_f (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (2.18)$$

### B Gravitational pull

This force is pulling the quadcopter downwards relative to the earth frame.

$$f_g = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

To represent the gravitational force relative to the body frame it's stated as:

$$f_{g-B} = R^T m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

$$f_{g-B} = \begin{bmatrix} f_{gx} \\ f_{gy} \\ f_{gz} \end{bmatrix} = \begin{bmatrix} -mg \sin(\theta) \\ mg \cos(\theta) \sin(\phi) \\ mg \cos(\theta) \cos(\phi) \end{bmatrix} \quad (2.19)$$

where  $m$  is the total mass of the quadcopter, while  $g$  is the gravitational acceleration  $= 9.82 \text{ m/s}^2$

### C Induced inflow force

When the lift force is larger than the gravitational force the quadcopter do not rise with constant acceleration. This is mainly because of a phenomenon called the induced inflow. As each rotor starts rising through the air, the airflow through the propellers starts in increasing which generates a negative small force perpendicular to the rotor called the induced inflow force [6].

$$f_i = i \begin{bmatrix} 0 \\ 0 \\ V_z \end{bmatrix} \quad (2.20)$$

Where  $V_z$  is the vertical velocity along the  $z$  axis relative to the body frame and  $i$  is the induced inflow constant which is calculated through imperial tests.

### D Disturbance

Wind disturbance has high influence on the quadcopter stability. Thus, it is accounted for in the form of forces  $f_w$  and moments  $T_w$ .

### E Total forces

From the above deduced forces the total force relative to the body frame can be written as:

$$f_{Total-B} = -f_{lift-B} + f_{g-B} + f_i + f_w \quad (2.21)$$

## F Linear acceleration

The dynamic model relative to the body frame can be deduced by substituting in Equation (2.21)

$$\begin{aligned} m(\dot{u} + ew - tv) &= f_{wx} - mg \sin(\theta) \\ m(\dot{v} - ew + tu) &= f_{wy} + mg \cos(\theta) \sin(\phi) \\ m(\dot{w} + ev - tu) &= f_i + f_{wz} + mg \cos(\theta) \cos(\phi) - U1 \end{aligned} \quad (2.22)$$

Using Newtons second law it can be deduced that

$$\dot{V} = \frac{1}{m}F \quad (2.23)$$

Then Equation (2.23) is used to derive the linear acceleration with respect to the earth frame,

$$R.f_{Total-B} = f_g - R.f_{lift-B} + f_w + f_i \quad (2.24)$$

Therefore, the linear acceleration with respect to the earth frame using Euler angles can be written as,

$$\begin{aligned} \ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \cdot \frac{U1}{m} \\ \ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \cdot \frac{U1}{m} \\ \ddot{z} &= g - (\cos \phi \cos \theta) \cdot \frac{U1}{m} \end{aligned} \quad (2.25)$$

### 2.3.2 Torques and angular acceleration

Angular acceleration occurs mainly due to the torques acting on the quadcopter.

The total torques applied to the quadcopter is stated by Euler's law:[1][4][19]

$$T = j * \dot{\omega}_B + (\omega_B \times j * \omega_B) \quad (2.26)$$

Thus

$$\begin{aligned} T_x &= \dot{e}I_{xx} - stI_{yy} + stI_{zz} \\ T_y &= \dot{s}I_{yy} - etI_{xx} + etI_{zz} \end{aligned} \quad (2.27)$$

$$T_z = \dot{t}I_{zz} - esI_{xx} + esI_{yy}$$

Finally, the net moments acting on the quadcopter can be written as:

$$T = T_B + T_w - G \quad (2.28)$$

Where  $T_B$  are the applied torques by the four rotors  $[T_\phi \ T_\theta \ T_\psi]$ ,  $T_w$  are the torques generated by the wind disturbances  $[T_{\phi w} \ T_{\theta w} \ T_{\psi w}]$  and  $G$  represents the gyroscopic effect caused by the four rotors. Since the moment of inertia for each rotor is relatively very small the gyroscopic effect will be neglected for simplicity.

### A Input Torques

The main reason for torques is the difference between each rotor speed. There are three torques generated which results in three angular displacement (Roll, Pitch and Yaw).

$$T_B = I\omega^2 \quad (2.29)$$

$$T_\phi = l * kf(-\Omega_1^2 + \Omega_3^2) = U2 \quad (2.30)$$

$$T_\theta = l * kf(-\Omega_2^2 + \Omega_4^2) = U3 \quad (2.31)$$

$$T_\psi = b(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) = U4 \quad (2.32)$$

$$T = \begin{bmatrix} T_\phi \\ T_\theta \\ T_\psi \end{bmatrix}$$

Where  $b$  is the drag constant and  $l$  is the arm length.

The dynamic model for the total moments relative to the body frame can be deduced by substituting in Equation (2.28).

$$\begin{aligned} \dot{e}I_{xx} - stI_{yy} + stI_{zz} &= U2 + T_{\phi w} \\ \dot{s}I_{yy} - etI_{xx} + etI_{zz} &= U3 + T_{\theta w} \\ \dot{t}I_{zz} - esI_{xx} + esI_{yy} &= U4 + T_{\psi w} \end{aligned} \quad (2.33)$$

## B Angular acceleration

Angular acceleration relative to the body frame can be deduced from the torques in the following manner, [1][4][19]

$$T = j * \dot{\omega}_B + (\omega_B \times j * \omega_B) \quad (2.34)$$

$$\dot{\omega}_B = [j^{-1} * T] - [j^{-1} * (\omega_B \times j * \omega_B)] \quad (2.35)$$

$$\dot{\omega}_B = \begin{bmatrix} \dot{e} \\ \dot{s} \\ \dot{t} \end{bmatrix} = \begin{bmatrix} l * kf(-\Omega_1^2 + \Omega_3^2/I_{xx}) \\ l * kf(-\Omega_2^2 + \Omega_4^2/I_{yy}) \\ b(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)/I_{zz} \end{bmatrix} - [j^{-1} * (\omega_B \times j * \omega_B)] \quad (2.36)$$

To simplify Equation (2.36) the assumption  $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}] = [e \ s \ t]$  is made. This is valid only for small angle changes. Thus the angular accelerations relative to earth frame using Euler transformation could be written as:

$$\begin{aligned} \ddot{\phi} &= \frac{U2}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \\ \ddot{\theta} &= \frac{U3}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} \\ \ddot{\psi} &= \frac{U4}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \end{aligned} \quad (2.37)$$

## 2.4 Control Forces

The input controls for the quadcopter are chosen as  $U = [U1 \ U2 \ U3 \ U4]^T$  where the first control input is for the altitude control while the remaining three control inputs are for controlling the attitude (Roll, Pitch and Yaw) [2]. As shown in Equation (2.38) The basic need is to stabilize the quadcopter at the required position and follow a desired reference trajectory. This is done by calculating the required rotors angular speed using inverse dynamics for the quadcopter.

$$U1 = T = u_z(Z_d - Z)m = kf(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (2.38a)$$

$$U2 = T_\phi = u_\phi(\phi_d - \phi)I_{xx} = l * kf(-\omega_2^2 + \omega_4^2) \quad (2.38b)$$

$$U3 = T_\theta = u_\theta(\theta_d - \theta)I_{yy} = l * kf(-\omega_1^2 + \omega_3^2) \quad (2.38c)$$

$$U4 = T_\psi = u_\psi(\psi_d - \psi)I_{zz} = b(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \quad (2.38d)$$

And the inverse dynamics could be written as in Equation (2.39) [23].

$$\omega_1^2 = \frac{1}{4kf}U1 - \frac{1}{2kf.l}U3 - \frac{1}{4b}U4 \quad (2.39a)$$

$$\omega_2^2 = \frac{1}{4kf}U1 - \frac{1}{2kf.l}U3 + \frac{1}{4b}U4 \quad (2.39b)$$

$$\omega_3^2 = \frac{1}{4kf}U1 + \frac{1}{2kf.l}U3 - \frac{1}{4b}U4 \quad (2.39c)$$

$$\omega_4^2 = \frac{1}{4kf}U1 + \frac{1}{2kf.l}U3 + \frac{1}{4b}U4 \quad (2.39d)$$

## 2.5 Linear Model

In order to use a linear controller for the quadcopter model. The model derived in the previous section must be linearized. To achieve this a method for linearization must be chosen and implemented on the model of the quadcopter derived in the previous section. 1st order Taylor series linearization for multivariable is used to linearize the nonlinear elements within the Equations of motion derived for the quadcopter model.

The Equation for the linearization of a function  $f(x,y)$  at the operating point  $p(a,b)$  using Taylors series method is:

$$f(x, y) \approx f(a, b) + \frac{\partial f(x, y)}{\partial x}|_{a,b}(x - a) + \frac{\partial f(x, y)}{\partial y}|_{a,b}(y - b) \quad (2.40)$$

The quadcopter is stabilized around the hover point which will be chosen as the operating point. These operating points are expressed within the body frame.

$$\hat{V}_b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.41)$$

$$\hat{\omega} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.42)$$

As for the control inputs:

$$\hat{U} = \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.43)$$

### 2.5.1 Linearization of Rotation Matrices

To carry out the linearization using the Taylors series equation the rotation matrices are calculated using the operating points as follows:

From Equation (2.40) the trigonometric functions can be approximated at the operating point as shown in Equations (2.44),(2.45) and (2.46).

$$\sin i \approx \sin(0) + \cos(0) \cdot \hat{i} \approx \hat{i} \quad (2.44)$$

$$\cos i \approx \cos(0) + -\sin(0) \cdot \hat{i} \approx 1 \quad (2.45)$$

$$\tan i \approx \tan(0) + \frac{1}{(\cos(0))^2} \cdot \hat{i} \approx \hat{i} \quad (2.46)$$

Using the approximation derived in (2.44),(2.45) and (2.46) the linearized linear and angular transformation matrices can be deduced as,

$$R_B = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.47)$$

$$\hat{R}_B = \begin{bmatrix} 1 & \phi\theta\psi - \psi & \theta + \phi\psi \\ \psi & \phi\theta\psi + 1 & \theta\psi - \phi \\ -\theta & \phi & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} \quad (2.48)$$

Substituting (2.48) in Equation (2.10) using taylor from Equation (2.40).

$$\dot{V} = \hat{R} * V_B + R * \hat{V}_B$$



$$\hat{V} = \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} * \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$V = V_b \quad (2.49)$$

The same method for the angular transformation matrix and angular acceleration transformation is carried. Using the approximation derived in (2.44), (2.45) and (2.46) the linearized angular transformation matrix can be deduced from the rotation matrix (2.7) as,

$$Rw = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix}$$

$$\hat{R}w = \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix} \quad (2.50)$$

To get the linearized angular acceleration the linearized transformation matrix (2.50) will be substituted into Equation (2.9) using taylor's method (2.40).

$$\hat{\omega} = \hat{R}w \omega_B + Rw \hat{\omega}_B$$

$$\hat{\omega} = \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & -\phi \\ 0 & \phi & 1 \end{bmatrix} \dot{\omega}_B$$

$$\hat{\omega} = \omega_B \quad (2.51)$$

## 2.5.2 Linearization of forces

Linear force and moment are affecting the quadcopter. Both of the forces are derived in Equations (2.21) and (2.36) respectively. Each one is linearized using the operating points. The first Equation is the linear force Equation which consists of three individual parts. Thus, its possible to linearize the gravitational part individually. Equation (2.52)

shows the gravitational part in the  $F_{Total-b}$  Equation linearized using the inverse of the linearized linear transformation matrix deduced in Equation (2.48).

$$f_{Total-B} = -f_{lift-B} + f_{g-B} + f_i + f_w$$

$$\hat{R} * f_{g-B} = \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} m = \begin{bmatrix} -\theta \\ \phi \\ 1 \end{bmatrix} gm \quad (2.52)$$

now we can substitute Equation (2.52) and Rewrite the Equation of linear acceleration (2.23) in vector form as in (2.54),

$$\dot{V}_b = \frac{1}{m} \left( - \begin{bmatrix} 0 \\ 0 \\ kf(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} + \begin{bmatrix} -\theta \\ \phi \\ 1 \end{bmatrix} gm + i \begin{bmatrix} 0 \\ 0 \\ \dot{V}_z \end{bmatrix} + \begin{bmatrix} f_{wx} \\ f_{wy} \\ f_{wz} \end{bmatrix} \right) \quad (2.53)$$

$$\dot{V}_b = \dot{V} = \begin{bmatrix} -\theta g + \frac{f_{wx}}{m} \\ \phi g + \frac{f_{wy}}{m} \\ \frac{-U1}{m} + g + \frac{i\dot{V}_z}{m} + \frac{f_{wz}}{m} \end{bmatrix} \quad (2.54)$$

The second Equation is the moment or torques deduced in Equation (2.34) and (2.36). in order to linearize the angular acceleration the operating points is substituted in Equation (2.36) and rewrite it as in Equation (2.55) based on the linearization deduced in Equation (2.51)

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} e \\ s \\ t \end{bmatrix} = \begin{bmatrix} f_{wx} + U2/I_{xx} \\ f_{wy} + U3/I_{yy} \\ f_{wz} + U4/I_{zz} \end{bmatrix} (Euler) \quad (2.55)$$

The linearized equations of motions can now be represented in State Space form.

## 2.6 State Space and Control Model

In this section the linearized equations of motions derived in the previous section are converted into state space form. This will make it easier to implement the desired linear controllers.

### 2.6.1 Equations of Motion and Control Inputs

#### A Linear accelerations

$$\dot{V} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -\theta g + \frac{f_{wx}}{m} \\ \phi g + \frac{f_{wy}}{m} \\ \frac{-U1}{m} + g + \frac{i\dot{V}_z}{m} + \frac{f_{wz}}{m} \end{bmatrix} \quad (2.56)$$

#### B Angular accelerations

$$\dot{\omega} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \dot{e} \\ \dot{s} \\ \dot{t} \end{bmatrix} = \begin{bmatrix} f_{wx} + U2/I_{xx} \\ f_{wy} + U3/I_{yy} \\ f_{wz} + U4/I_{zz} \end{bmatrix} \quad (2.57)$$

#### C Control inputs

$$U1 = T = u_z(Z_d - Z)m = kf(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (2.58)$$

$$U2 = T_\phi = u_\phi(\phi_d - \phi)I_{xx} = l * kf(-\Omega_1^2 + \Omega_3^2) \quad (2.59)$$

$$U3 = T_\theta = u_\theta(\theta_d - \theta)I_{yy} = l * kf(-\Omega_2^2 + \Omega_4^2) \quad (2.60)$$

$$U4 = T_\psi = u_\psi(\psi_d - \psi)I_{zz} = b(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (2.61)$$

### 2.6.2 State Space Representation

The state space is in the form:

$$\dot{X}(t) = A(t)X(t) + B(t)u(t)$$

$$Y(t) = C(t)X(t) + D(t)u(t)$$

$$Inputs[U1 \ U2 \ U3 \ U4]$$

$$\text{Outputs}[x \ y \ z \ \phi \ \theta \ \psi \ V_x \ V_y \ V_z]$$

$$\text{Disturbances}[f_{wx} \ f_{wy} \ f_{wz} \ T_{wx} \ T_{wy} \ T_{wz}]$$

### A State variables

$$x_1 = x \tag{2.62a}$$

$$x_2 = y \tag{2.62b}$$

$$x_3 = z \tag{2.62c}$$

$$x_4 = V_x \tag{2.62d}$$

$$x_5 = V_y \tag{2.62e}$$

$$x_6 = V_z \tag{2.62f}$$

$$x_7 = \phi \tag{2.62g}$$

$$x_8 = \theta \tag{2.62h}$$

$$x_9 = \psi \tag{2.62i}$$

$$x_{10} = \omega_x \tag{2.62j}$$

$$x_{11} = \omega_y \tag{2.62k}$$

$$x_{12} = \omega_z \tag{2.62l}$$

### B States equations

$$\dot{x}_1 = \dot{x} = V_x = x_4 \tag{2.63a}$$

$$\dot{x}_2 = \dot{y} = V_y = x_5 \tag{2.63b}$$

$$\dot{x}_3 = \dot{z} = V_z = x_6 \tag{2.63c}$$

$$\dot{x}_4 = \dot{V}_x = -\theta g + \frac{f_{wx}}{m} \tag{2.63d}$$

$$\dot{x}_5 = \dot{V}_y = \phi g + \frac{f_{wy}}{m} \tag{2.63e}$$

$$\dot{x}_6 = \dot{V}_z = \frac{-U1}{m} + g + \frac{i\dot{V}_z}{m} + \frac{f_{wz}}{m} \tag{2.63f}$$

$$\dot{x}_7 = \omega_x = x_{10} \quad (2.63g)$$

$$\dot{x}_8 = \omega_y = x_{11} \quad (2.63h)$$

$$\dot{x}_9 = \omega_z = x_{12} \quad (2.63i)$$

$$\dot{x}_{10} = \dot{\omega}_x = f_{wx} + U2/I_{xx} \quad (2.63j)$$

$$\dot{x}_{11} = \dot{\omega}_y = f_{wy} + U3/I_{yy} \quad (2.63k)$$

$$\dot{x}_{12} = \dot{\omega}_z = f_{wz} + U4/I_{zz} \quad (2.63l)$$

### C Output equations

$$y_1 = x = x_1 \quad (2.64a)$$

$$y_2 = y = x_2 \quad (2.64b)$$

$$y_3 = z = x_3 \quad (2.64c)$$

$$y_4 = V_x = x_4 \quad (2.64d)$$

$$y_5 = V_y = x_5 \quad (2.64e)$$

$$y_6 = V_z = x_6 \quad (2.64f)$$

$$y_7 = \phi = x_7 \quad (2.64g)$$

$$y_8 = \theta = x_8 \quad (2.64h)$$

$$y_9 = \psi = X_9 \quad (2.64i)$$

**D State matrices**

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.65a)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \quad (2.65b)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.65c)$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/m & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \quad (2.65d)$$

The model was checked for controllability using MATLAB. The system is of full rank 12 which is equal to the order of the system. Hence, the system is controllable.

# Chapter 3

## Linear Control (PID)

In this chapter a PID controller is applied on the quadcopter system to achieve stability for position and attitude given the desired values. The main nested control loops for position and attitude control are shown in Figure 3.1.

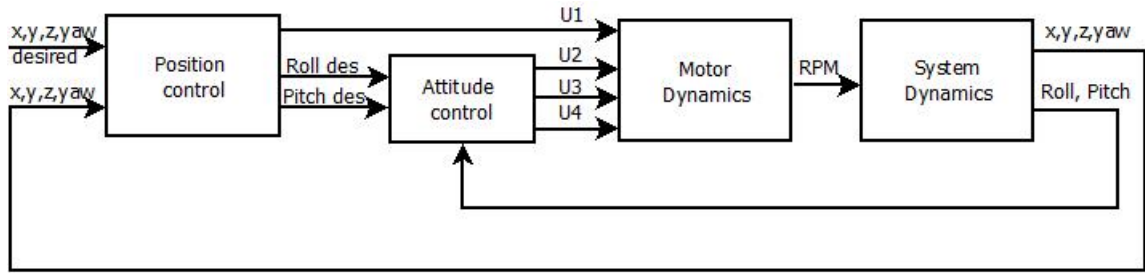


Figure 3.1: PID main control scheme

### 3.1 PID Control

PID controllers are the most used controllers in the industry. That is mainly due to:

- Simple structure
- Tuning without having a precise model.
- Good performance.
- Various manual method for determining the gains as well as various software for automatic tuning such as Simulink.

The PID controller structure is shown in Figure 3.2 and Equation (3.1)

$$u(t) = K_P e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt} \quad (3.1)$$

$$u(s) = K_P e(s) + K_I \frac{1}{s} e(s) - K_D s e(s)$$



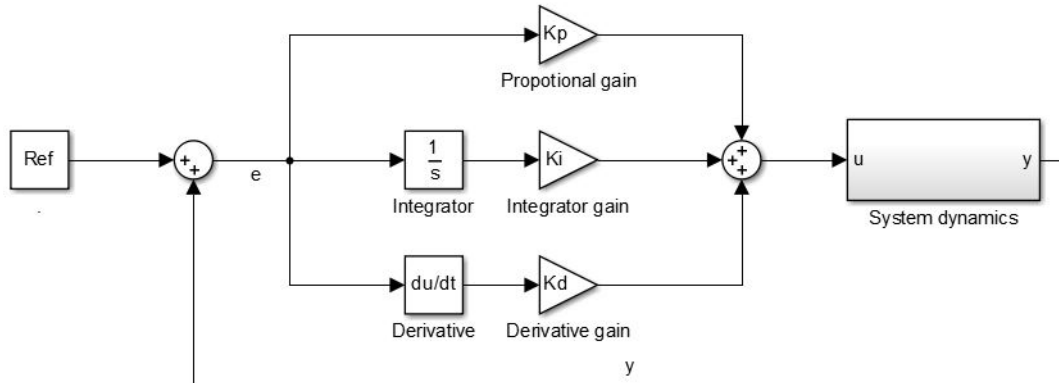


Figure 3.2: Traditional PID controller

The controller consists of three parts.

- Proportional term.

It produces an output value which is proportional to the error value. The proportional response can be adjusted by changing the proportional gain.

- Integral term

The integral term varies according to the integral of the error. It is responsible for eliminating the steady state error. Its response can be adjusted by changing the integral gain. However, since the integral term varies to the accumulated error, it can cause the present value to overshoot the desired value especially in the case of large errors.

- Derivative term

The derivative term varies according to the derivative of the error. It is responsible for decreasing the overshoot and the settling time. However, differentiation of a signal amplifies noise and can cause a process to become unstable. In other words the derivative term acts as a high pass filter [27].

As a result for a system with uncertainties such as the quadcopter. A method is needed

to overcome the problems that are mentioned in the derivative and integral terms.

## 3.2 Effect of Nonlinearities and Noise

The quadcopter system does not account for noise and nonlinearities effects. This affects the derivative and integral term which can be avoided using the following techniques.

### 3.2.1 Derivative filtering for PID

When applying the derivative term two problems could arise:

1. If the reference input has sudden large changes this would lead to a large derivative term, this would result in a high control signal.
2. Usually there is noise produced by the measurement sensors with high frequency which would lead to a large derivative term.

Since switching off the derivative term is not a good solution as it is the main contributor to the transient response another solution is to change the form of the derivative term by introducing a filter which will overcome the noise and disturbance by attenuating high frequency. The resulting PID controller when adding the filter to the derivative term is:

$$u(s) = K_P e(s) + K_I \frac{1}{s} e(s) + K_D \frac{Ns}{N+s} e(s) \quad (3.2)$$

### 3.2.2 Anti integral windup

An integral action within a controller combined with an actuator that becomes saturated gives undesirable effect. This is taking place when the control error is very large and accordingly the integral saturates the actuator and hence the feedback path breaks. Such situation is called integrator windup and leads to a large overshoot and high setting time. There are several ways to avoid integrator windup. The first method is by stopping to update the integral when the actuator is saturated. The second method is by introducing a feedback which measures the actuator output to form an error signal between the actuator

and the controller force and feeding this error to the integral through a gain. Thus, when the actuator saturates the feedback tries to make the error signal zero which means the integrator is reset [27] [28].

### 3.2.3 Modified PID controller structure

After applying the two techniques the PID structure becomes as shown in Figure 3.3

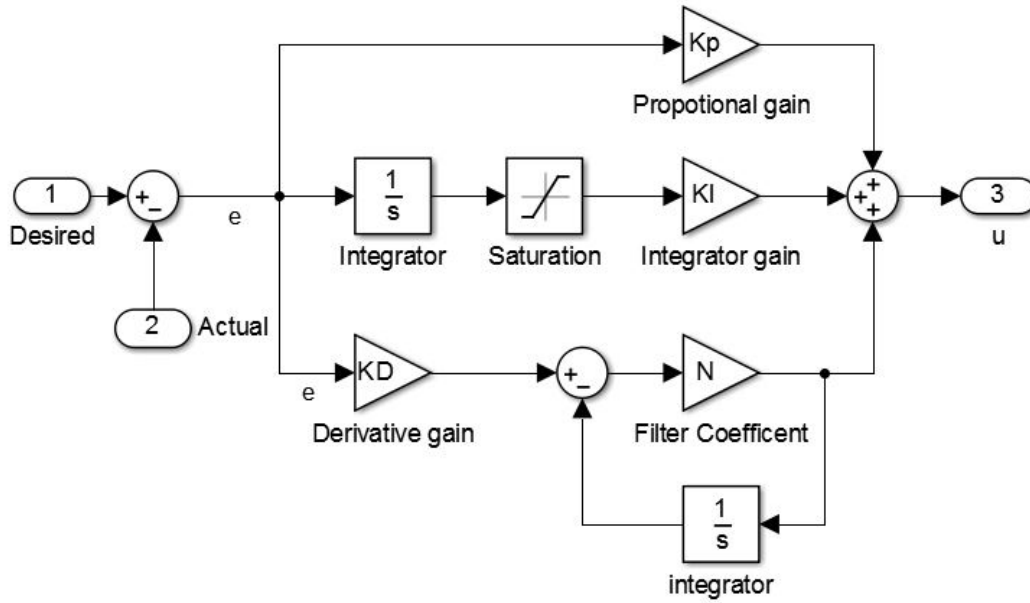


Figure 3.3: Modified PID controller

## 3.3 PID Stability Controller

The main closed loop control structure depicted to the quadcopter system is shown in figure 3.4. Six individual PID controllers are applied for each degree of freedom to achieve stability at the desired values.

### 3.3.1 Attitude and heading control

The Roll ( $\phi$ ) and Pitch ( $\theta$ ) angles are responsible for the attitude respectively while the Yaw ( $\psi$ ) angle is responsible for the heading. For each one, a PID controller is applied to

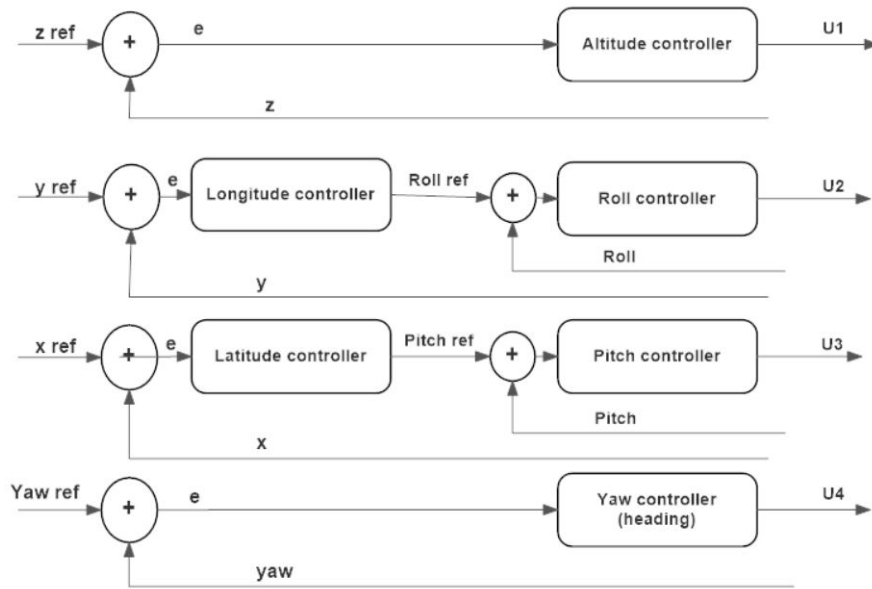


Figure 3.4: Main controller structure

produce  $[U2 \ U3 \ U4]^T$  respectively.

### A Roll control

The controller is shown in Figure 3.5. The inputs are the Roll ( $\phi$ ) actual and desired while the output is  $U2$ .

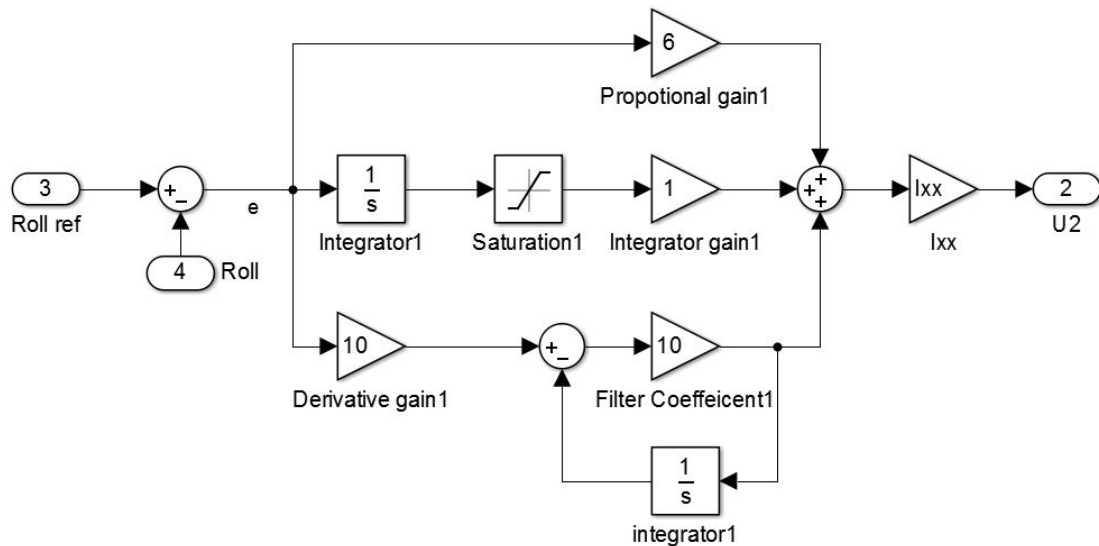


Figure 3.5: Roll PID control

### B Pitch control

The controller is shown in Figure 3.6. The inputs are the Pitch ( $\theta$ ) actual and desired while the output is  $U_3$ .

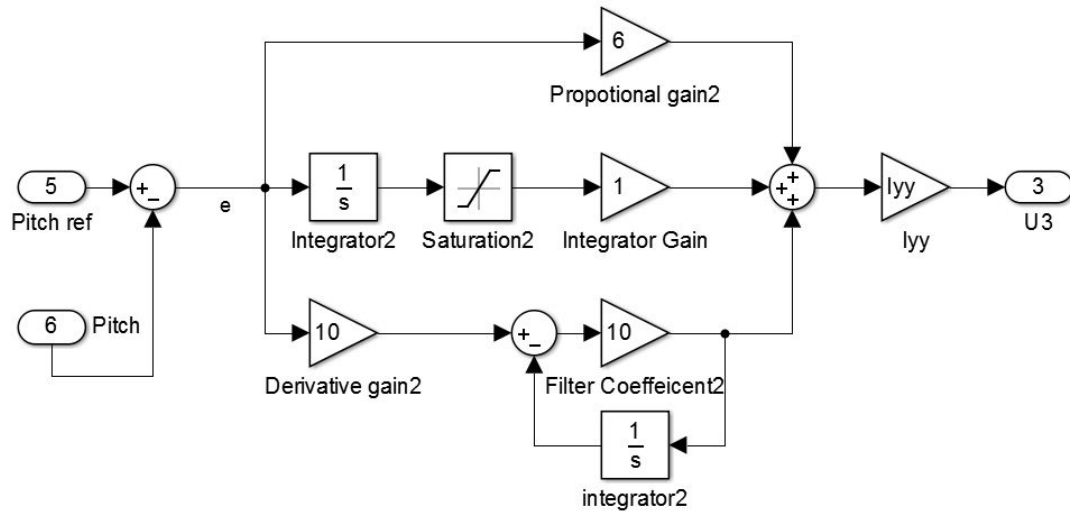


Figure 3.6: Pitch PID control

### C Yaw control

The controller is shown in Figure 3.7. The inputs are the Yaw ( $\psi$ ) actual and desired while the output is  $U_4$ .

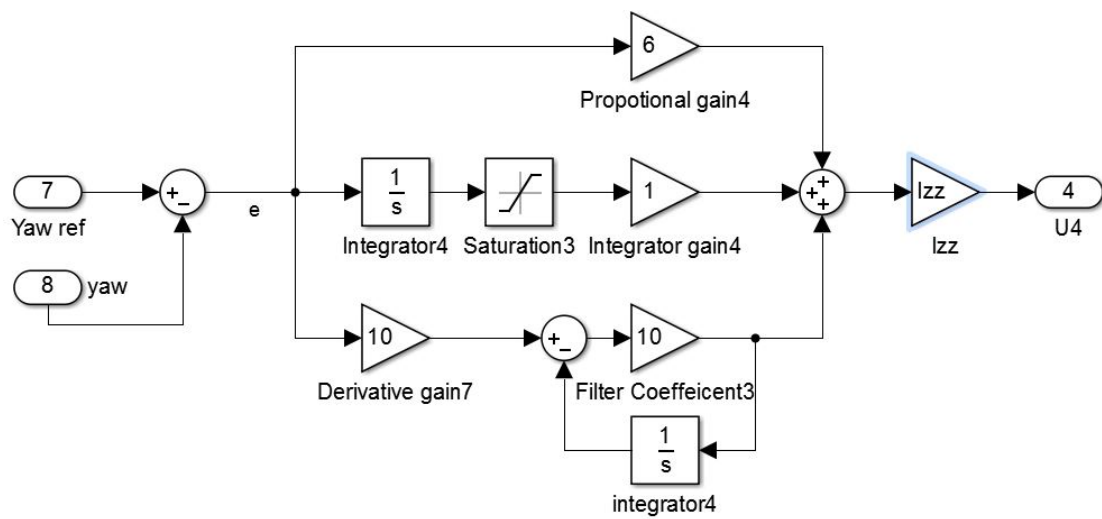


Figure 3.7: Yaw PID control

### 3.3.2 Position control

The position of the quadcopter is determined by the longitude (x direction), latitude (y direction) and the altitude (z direction). For each one a PID controller is applied to produce  $[U1 \ \theta_{des} \ \phi_{des}]^T$  respectively.

#### A Altitude control

The controller is shown in Figure 3.8. The inputs are z actual and desired while the output is U1.

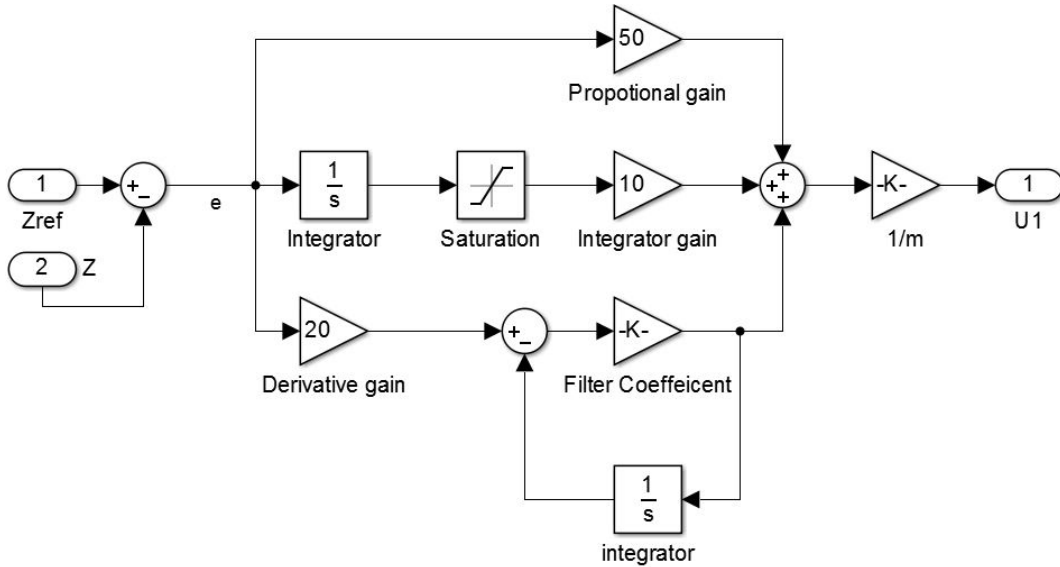


Figure 3.8: Altitude PID control

#### B Longitude control

The controller is shown in Figure 3.9. The inputs are x actual and desired while the output is the Pitch ( $\theta$ ).

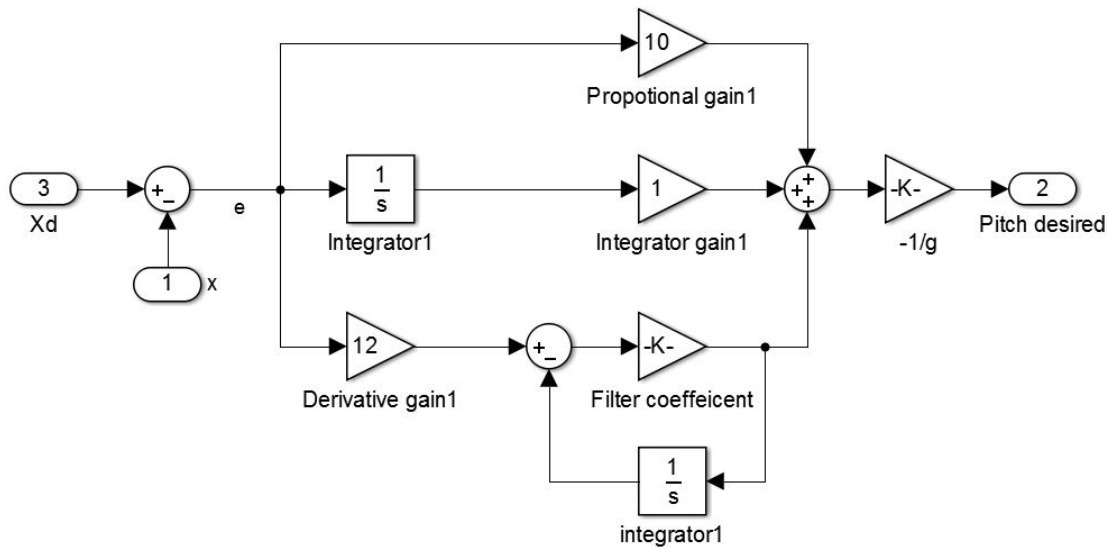


Figure 3.9: Longitude PID control

### C Latitude control

The controller is shown in Figure 3.10. The inputs are  $y$  actual and desired while the output is Roll ( $\phi$ ).

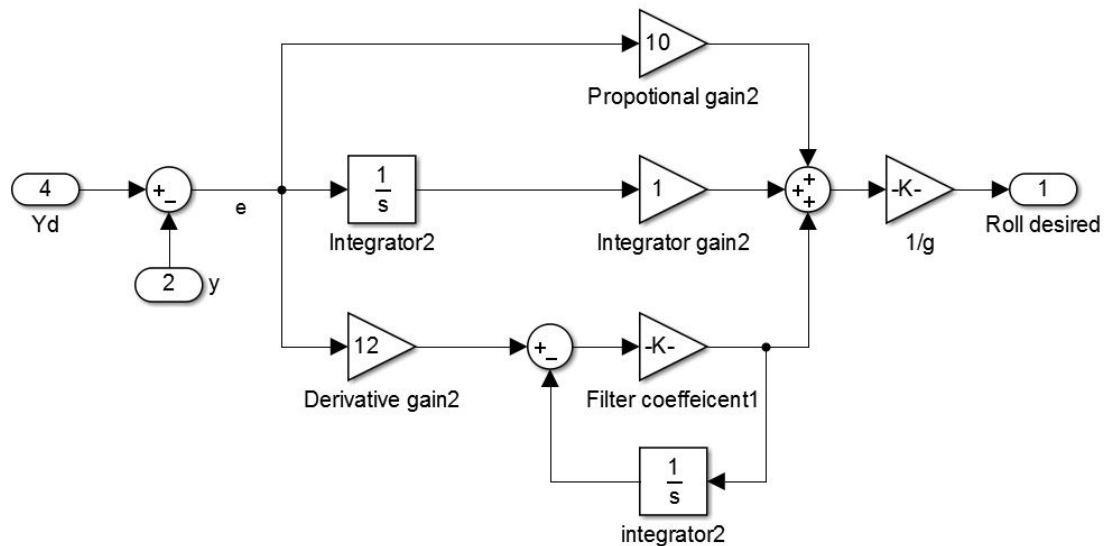


Figure 3.10: Latitude PID control

### 3.3.3 Position and Attitude control tuning

The gains for each controller are tuned by three means. The first one is using the tune feature in the Simulink. The second one is by trial and error and testing using the Simulink constructed simulator and finally based on literature. The general effect of each of the proportional, integral and derivative gains are stated in Table 3.1

Table 3.1: PID gains effect

Coeffeicent	Effect	Large value	Small value
$K_P$	Incresed acceleration and stability	Reduce Stability	Slower response, steady error
$K_I$	Decrease steady error	Reduce stability, oscillatory response	Large settling time
$K_D$	Increase stability, reduce oscillation (small settling time)	Increase stability, in crease turbulence effect	Could make the system unstable



## Chapter 4

### Fuzzy Gain Scheduled PID Controller

The PID controller adopted in the previous chapter has static gains which were determined on several simulations and many trials. However, wind disturbance and sensors measurement noise influence the PID controller performance. This is mainly due to the fact that the static PID gains need to be real-time tuned to adapt with wind disturbance and measurements noise. Thus, a proposed hybrid Fuzzy gain scheduled PID control (FPID) is applied on the quadcopter system. The fuzzy control will be responsible for actively choosing the gains for the PIDs that renders the quadcopter stable taking into consideration the circumstances it is flying in. The suggested control approach is shown in Figure 4.1

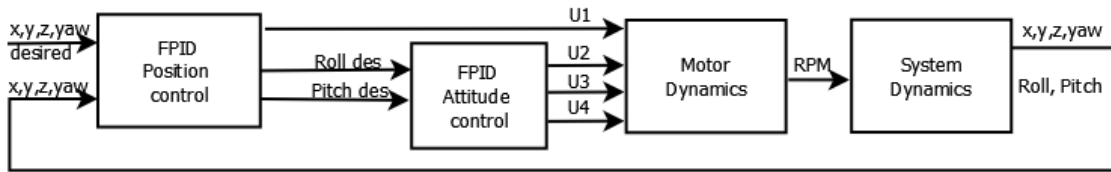


Figure 4.1: Fuzzy PID control

Two well known methods are adopted to define the output for the fuzzy control.

- The first method is the mamdani which treats the output as the inputs and set a predetermined range for the output. *if* input  $x = small$  and input  $y = large$  then output  $z = verylarge$
- The second method is Takagi-sugeno (TS) which defines the output as a function *if* input  $x = small$  and input  $y = large$  then output  $z = ax + by + c$

Each method has its advantage and disadvantage. The Mamdani method requires a large memory requirement. Thus, a powerful on board controller should be chosen. The TS method on the other hand requires less memory requirement but more complicated to

formulate. Hence, Mamdani approach is adopted where the rule base is based on linguistic directions rather than mathematical functions. To construct a fuzzy control based on the Mamdani approach certain steps must be carried out,

1. First different fuzzy linguistic variables should be set for each input and output, then a fuzzy rule is set which converts fuzzy input variables into fuzzy sets of variable outputs. The rules are in the form of IF/THEN statements. The rule is set according to human reasoning and past observations. The basic operators of fuzzy logic are (AND), (OR) and (NOT).
2. After setting the variables and the fuzzy rule fuzzification stage starts where each input variable is converted into the relevant fuzzy variable.
3. Fuzzy implication is implemented using the rules developed to map the input variables to the output membership functions.
4. Fuzzy aggregation then takes place where the output sets are combined to form a simplified definitions.
5. Final stage is the defuzzification where the output sets are translated to real values.

## 4.1 Developing of the Fuzzy Controller

Two inputs and three outputs are chosen for the control. The inputs are the error  $e$  and rate of change in error  $ed$  while the outputs are  $K_p$ ,  $K_I$  and  $K_D$ . The general structure of the Fuzzy control is shown in Figure 4.2 and the fuzzy linguistic variables are shown in Table 4.1

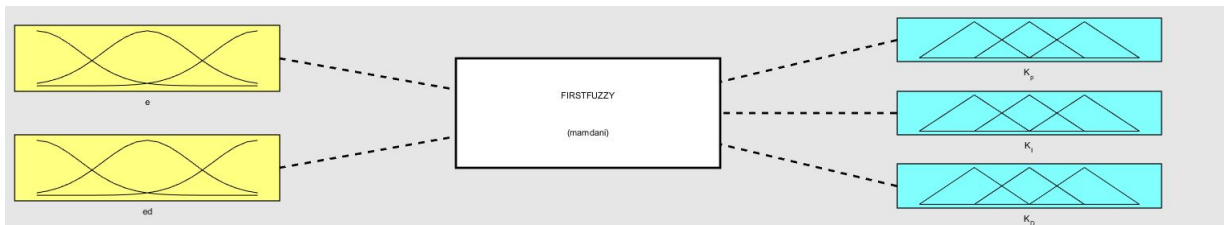


Figure 4.2: General fuzzy control structure

Table 4.1: Linguistic variables

L	Large
M	Medium
S	Small
NL	Negative Large
NM	Negative Medium
NS	Negative Small
Z	Zero
PS	Positive Small
PM	Positive Medium
PL	Positive Large

#### 4.1.1 Membership functions

The membership functions are chosen to be the same for all the inputs and outputs. They are chosen to be triangular functions. For all inputs and outputs minimum and maximum values are chosen based on testing and literature review [3][14][15].

Before proceeding the input values must be normalized using Equation (4.1).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

The outputs will be mapped into their real values using Equation (4.2)

$$x = x'(x_{max} - x_{min}) + x_{min} \quad (4.2)$$

After normalization the membership functions are shown in Figures 4.3 and 4.4.

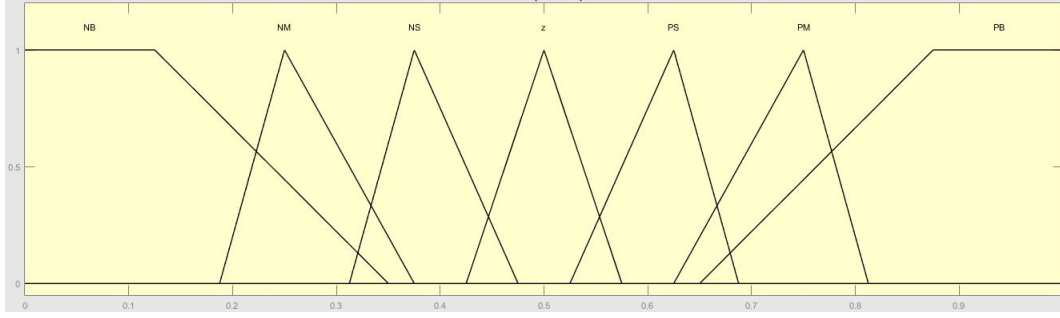


Figure 4.3: Inputs membership functions (e and ed )

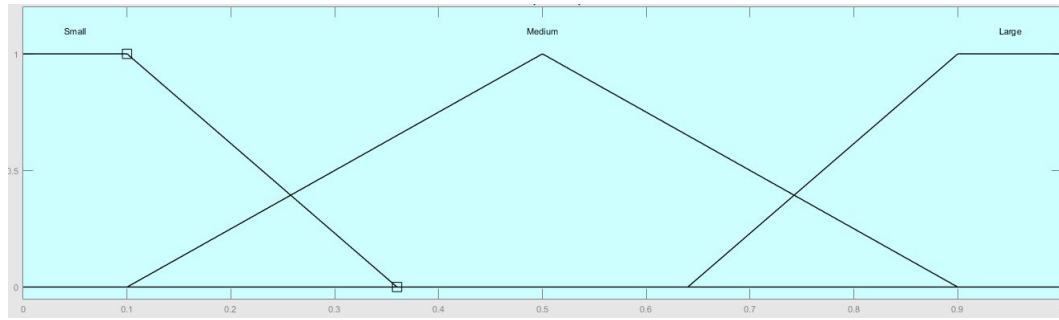


Figure 4.4: Outputs membership functions (Gains)

### 4.1.2 Fuzzy rules

The rules set for the Propotional, Integral and Derivative gains are set as follows [3][14][15]:

- IF  $e = NL$  AND  $ed = NL$  THEN  $K_P = M$ .
- IF  $e = NL$  AND  $ed = NM$  THEN  $K_P = L$ .
- IF  $e = NL$  AND  $ed = NS$  THEN  $K_P = L$ .
- IF  $e = NL$  AND  $ed = Z$  THEN  $K_P = L$ .
- IF  $e = NL$  AND  $ed = PS$  THEN  $K_P = L$ .
- IF  $e = NL$  AND  $ed = PM$  THEN  $K_P = L$ .
- IF  $e = NL$  AND  $ed = PL$  THEN  $K_P = M$ .
- IF  $e = NM$  AND  $ed = NL$  THEN  $K_P = S$ .

- IF  $e = \text{NM}$  AND  $ed = \text{NM}$  THEN  $K_P = \text{M}$ .
- IF  $e = \text{NM}$  AND  $ed = \text{NS}$  THEN  $K_P = \text{L}$ .
- IF  $e = \text{NM}$  AND  $ed = \text{Z}$  THEN  $K_P = \text{L}$ .
- IF  $e = \text{NM}$  AND  $ed = \text{PS}$  THEN  $K_P = \text{L}$ .
- IF  $e = \text{NM}$  AND  $ed = \text{PM}$  THEN  $K_P = \text{M}$ .
- IF  $e = \text{NM}$  AND  $ed = \text{PL}$  THEN  $K_P = \text{S}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{NL}$  THEN  $K_P = \text{S}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{NM}$  THEN  $K_P = \text{M}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{NS}$  THEN  $K_P = \text{L}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{Z}$  THEN  $K_P = \text{L}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{PS}$  THEN  $K_P = \text{L}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{PM}$  THEN  $K_P = \text{M}$ .
- IF  $e = \text{NS}$  AND  $ed = \text{PL}$  THEN  $K_P = \text{S}$ . And so goes on.

All the Fuzzy rules are shown in Tables 4.2 and Figure 4.3.

### 4.1.3 Defuzzification

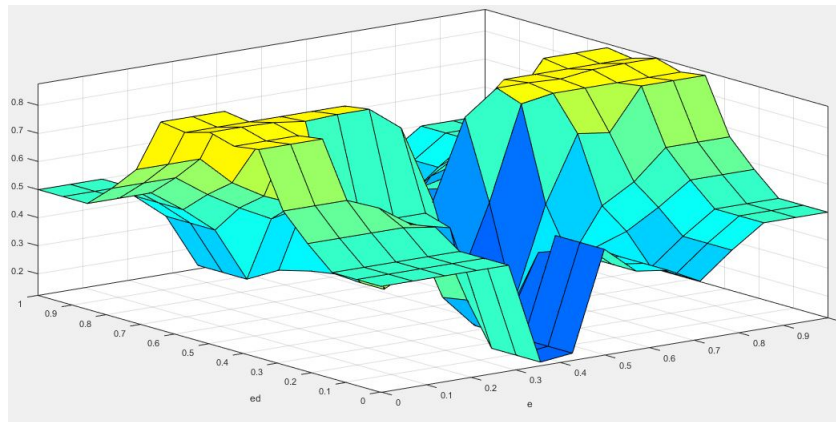
After setting the rules the fuzzy outputs cannot be directly provided to the system. Thus, Defuzzification process is required to convert the linguistic outputs into numbers. Any change in the inputs should not produce a large change in the outputs. Inorder to achieve this. The defuzzification method is chosen as centre average or centroid. The control surfaces for  $K_P, K_I$  and  $K_D$  are shown in Figures 4.5 and 4.6.

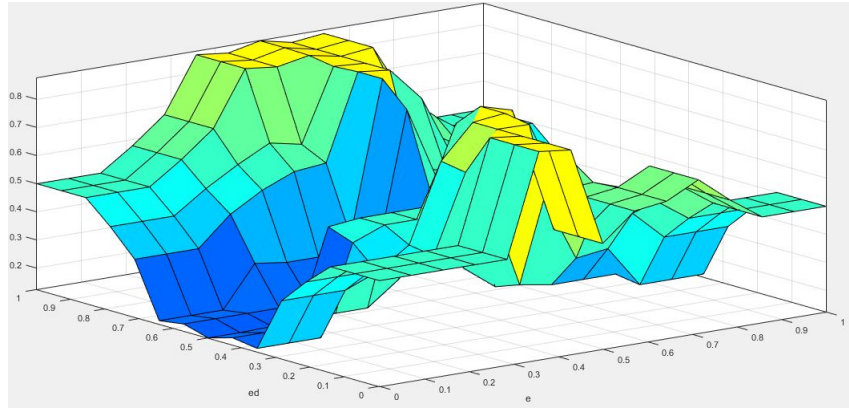
Table 4.2: Fuzzy rules for  $K_P$  and  $K_I$ 

e/ed	NL	NM	NS	Z	PS	PM	PL
NL	M	S	S	S	S	S	M
NM	L	M	S	S	S	M	L
NS	L	L	M	S	M	L	L
Z	L	L	L	M	L	L	L
PS	L	L	M	S	M	L	L
PM	L	M	S	S	S	M	L
PL	M	S	S	S	S	S	M

Table 4.3: Fuzzy rules for  $K_D$ 

e/ed	NL	NM	NS	Z	PS	PM	PL
NL	M	L	L	L	L	L	M
NM	S	M	L	L	L	M	S
NS	S	S	M	L	M	S	S
Z	S	S	S	M	S	S	S
PS	S	S	M	L	M	S	S
PM	S	M	L	L	L	M	S
PL	M	L	L	L	L	L	M

Figure 4.5: Control surfaces for  $K_P$  and  $K_I$

Figure 4.6: Control surfaces for  $K_D$ 

## 4.2 Implementing the Fuzzy Controller on the Quadcopter

The previous PID models are combined with the developed fuzzy control to be able to get the PID gains online. The fuzzy control is added to the PID controllers as shown in Figure 4.7.

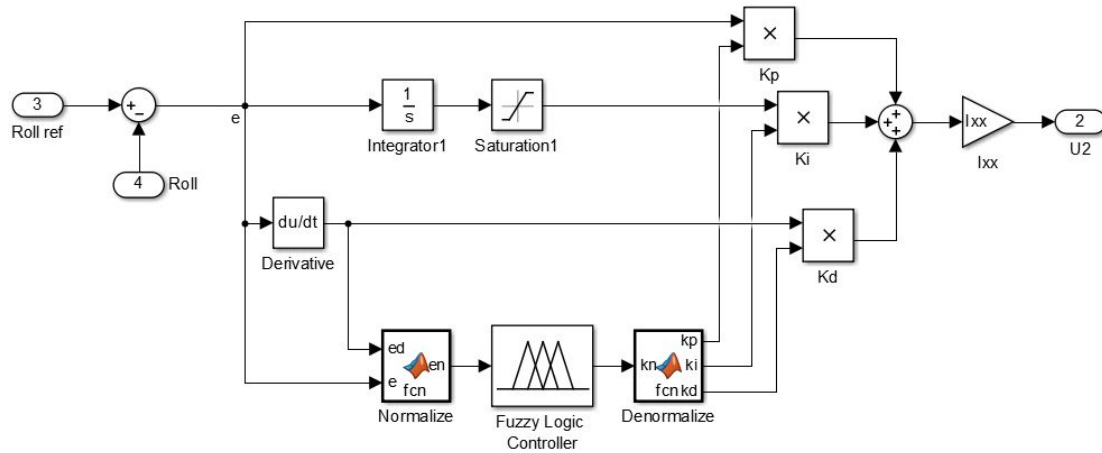


Figure 4.7: Fuzzy gain scheduled PID control

# Chapter 5

## Hybrid Control Strategy

In this chapter a hybrid control strategy is presented that constitutes three control techniques. These control techniques:

- FPID with state coordinates transformation
- State feedback control, and
- Sliding mode control.

The configuration of the hybrid control strategy is arranged into two control loops. The inner control loop aims to control the quadcopter's attitude and altitude while the outer control loop aims to control the quadcopter's position. Two configurations were developed to configure the developed hybrid control strategy:

1. A sliding mode control is used for the outer control loop while for the inner control loop two control techniques are used to realize it: FPID with state coordinates transformation and a state feedback control.
2. FPID control is used for the outer control loop while for the inner control loop two control techniques are used to realize it using the same formation as in the first configuration of the hybrid control strategy.

### 5.1 Hybrid Control Strategy: Configuration 1

The first configuration of the hybrid control strategy is shown in Figure 5.1. The configuration adheres to:

- For the outer loop, The position controller is based on a sliding mode controller. The trajectory generator and the position controller delivers the desired attitude and altitude  $[z_d \ \phi_d \ \theta_d \ \psi_d]$  to the inner control loop.
- In the inner control loop:



- The attitude and altitude controller consists of FPID controller which produces the linear control forces  $[v1 \ v2 \ v3 \ v4]$  using the system states and the desired attitude and altitude provided by the outer loop.
- Then the linear control forces are transformed into the nonlinear control inputs  $[U1 \ U2 \ U3 \ U4]$  via the state feedback control law.

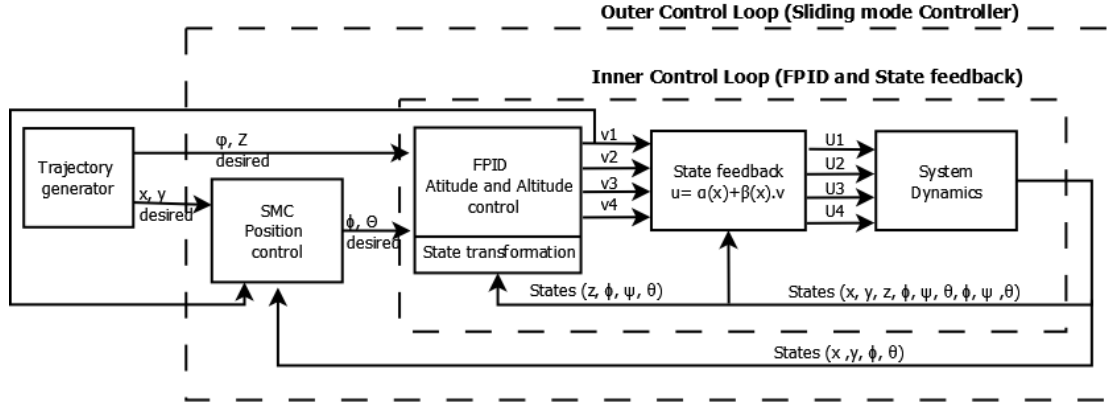


Figure 5.1: Configuration 1 of the hybrid control strategy scheme using FPID with state transformation, state feedback and SMC

### 5.1.1 Inner control loop: FPID with state coordinates transformation and a state feedback control

As previously stated a quadcopter system is considered as a coupled under-actuated system. Thus, if applying a full feedback linearization the decoupling matrix  $\Delta$  that represents the LIE derivatives of the states will be singular[29] [30]. Two methods were suggested for solving this problem,

1. The first method is called "dynamic extension" [28][31][29] in which the whole system is represented as one block and a new states are introduced to it. This is achieved by differentiating the output responsible for the system being coupled until the other inputs show up. This method is able to solve the problem however it results in very high derivative terms which are very complicated to compute.

2. The second method is called "Feedback linearization using partial states and stabilization of the internal dynamics". It is carried out by dividing the system into two subsystems (inner and outer control loops).

- The inner control loop is controlled using a state feedback control technique alone or as a combination with other linear controllers, such as, LQG and  $H_\infty$ .
- The outer control loop is considered as an internal dynamics for position control, which is stabilized using one of the relevant controllers [32] [29] [30].

The second method is much simpler than the first one since the derivatives of the outputs are of lower terms. However, the internal dynamics must be stable for the overall stability of the system to be achieved. Hence, feedback linearization using partial states linearization and stabilization of the internal dynamics is developed.

Before proceeding the system's Equations of motion derived in Chapter 2 are shown in Equation 5.1.

$$\ddot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \cdot \frac{U1}{m} \quad (5.1a)$$

$$\ddot{y} = (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \cdot \frac{U1}{m} \quad (5.1b)$$

$$\ddot{z} = -g + (\cos \phi \cos \theta) \cdot \frac{U1}{m} \quad (5.1c)$$

$$\ddot{\phi} = \frac{U2}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \quad (5.1d)$$

$$\ddot{\theta} = \frac{U3}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} \quad (5.1e)$$

$$\ddot{\psi} = \frac{U4}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \quad (5.1f)$$

The states for the inner control loop  $[z \ \phi \ \theta \ \psi]$  are considered as outputs and the control forces  $[U1 \ U2 \ U3 \ U4]$  as the control inputs. The nonlinear subsystem is described in the form of:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x), \end{aligned} \quad (5.2)$$

From Equations 5.1 and 5.2, we have

$$\dot{x} = \begin{bmatrix} \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} -g \\ \frac{I_{yy}-I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} \\ \frac{I_{zz}-I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} \\ \frac{I_{xx}-I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} \end{bmatrix} + \begin{bmatrix} (\cos \phi \cos \theta) \cdot \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \end{bmatrix} \quad (5.3)$$

Where,

$$f(x) = \begin{bmatrix} -g \\ \frac{I_{yy}-I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} \\ \frac{I_{zz}-I_{xx}}{I_{yy}}\dot{\phi}\dot{\psi} \\ \frac{I_{xx}-I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta} \end{bmatrix} \quad (5.4)$$

$$g(x) = \begin{bmatrix} (\cos \phi \cos \theta) \cdot \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \quad (5.5)$$

$$y = h(x) = \begin{bmatrix} z \\ \phi \\ \theta \\ \psi \end{bmatrix} \quad (5.6)$$

Assuming all system states are measurable. A state feedback control law can be used in the form of:

$$u = \alpha(x) + \beta(x).v \quad (5.7)$$

Where  $v = [v1 \ v2 \ v2 \ v2]$  is the linear control force vector which is fed to the state feedback control. From the literature, the formulation of  $\alpha(x)$  and  $\beta(x)$  is represented as, [33] [34]

$$\begin{aligned}\alpha(x) &= -\Delta^{-1}.b(x) \\ \beta(x) &= \Delta^{-1}\end{aligned}\tag{5.8}$$

Where  $\Delta$  is the decoupling matrix which must be nonsingular and  $b(x)$  is the Brunovsky normal form.  $b(x)$  and  $\Delta$  is computed as expressed in Equations 5.11 and 5.12 respectively.

We focus now on a single-input, single output system. More details on this technique is in appendix A. The derivative of the output  $y$  can be expressed as [29][30]:

$$\begin{bmatrix} y^{(r_1)} \\ y^{(r_2)} \\ y^{(r_3)} \\ y^{(r_4)} \end{bmatrix} = b(x) + \Delta(x).u\tag{5.9}$$

Where  $r_i$  is the relative degree that represents the number of times the output  $y$  is differentiated until the input shows up. To be able to transform the nonlinear system into a linear one via state feedback, the total relative degree  $r$  must be equal the total order of the system. Thus, differentiating the output vector till  $u$  shows up yields,[33] [29][30] [33] [34]

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \\ \ddot{y}_4 \end{bmatrix} = \begin{bmatrix} \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = b(x) + \Delta(x).u\tag{5.10}$$

The total order of the whole quadcopter system including the internal dynamics is equal to 12. However, the total order of the quadcopter inner control loop system under consideration is  $n = 8$  and the relative degree is  $r_1 = r_2 = r_3 = r_4 = 2$ . Therefore the total relative degree  $r = 8$ . . Thus, the total relative degree and the order of the system are equal  $r = n$ . Hence, The system can be transformed via a state feedback into a new system which is fully linear and controllable.

$b(x)$  and  $\Delta(x)$  is formulated as,

$$b(x) = \begin{bmatrix} L_f^{r_1} h_1(x) \\ L_f^{r_2} h_2(x) \\ L_f^{r_3} h_3(x) \\ L_f^{r_4} h_4(x) \end{bmatrix} \quad (5.11)$$

$$\Delta(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & L_{g_2} L_f^{r_1-1} h_1(x) & L_{g_2} L_f^{r_1-1} h_1(x) & L_{g_2} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & L_{g_2} L_f^{r_2-1} h_2(x) & L_{g_2} L_f^{r_2-1} h_2(x) & L_{g_2} L_f^{r_2-1} h_2(x) \\ L_{g_1} L_f^{r_3-1} h_3(x) & L_{g_2} L_f^{r_3-1} h_3(x) & L_{g_2} L_f^{r_3-1} h_3(x) & L_{g_2} L_f^{r_3-1} h_3(x) \\ L_{g_1} L_f^{r_4-1} h_4(x) & L_{g_2} L_f^{r_4-1} h_4(x) & L_{g_2} L_f^{r_4-1} h_4(x) & L_{g_2} L_f^{r_4-1} h_4(x) \end{bmatrix} \quad (5.12)$$

The input-output decoupling problem is solvable if and only if the matrix  $\Delta$  is non-singular. Accordingly, the static state feedback control law can be applied.

Substituting Equation 5.4, 5.5 and 5.6 in Equations 5.11 and 5.12 yields,

$$b(x) = \begin{bmatrix} -g \\ \frac{I_{yy}-I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \\ \frac{I_{zz}-I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} \end{bmatrix} \quad (5.13)$$

$$\Delta(x) = \begin{bmatrix} (\cos \phi \cos \theta) \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \quad (5.14)$$

As a result since  $r = n$  and  $\Delta$  is nonsingular, the input-output decoupling problem has been solved using a state feedback control law.

As for the controller responsible for the linear control force vector  $v = [v_1 \ v_2 \ v_2 \ v_2]$  it consists of:

- State coordinate transformation, and
- FPID control with the same formulation as in chapter 4.

The state coordinates transformation aims to transform the nonlinear states into linear ones. The control scheme for the inner control loop is shown in Figure 5.2.

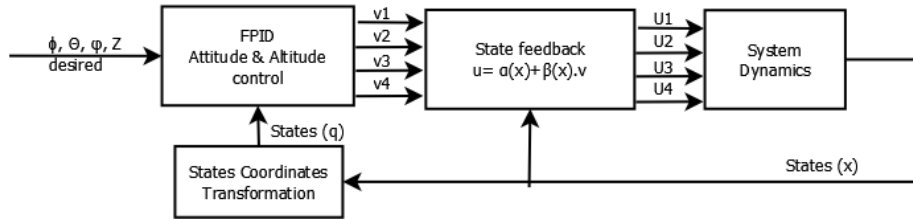


Figure 5.2: Inner control loop for the hybrid control strategy

### 5.1.2 Outer control loop: Stability of the internal dynamics using SMC

A sliding mode controller is developed to insure stability of the internal dynamics.

#### A Sliding mode control design

A Sliding Mode Control is a Variable Structure Control. The purpose of the SMC is to drive the nonlinear system's states onto a prespecified surface in finite time and to maintain the system's states trajectory on this surface. The surface is called a switching surface or sliding surface. Once the sliding surface is intercepted, the control maintains the system's states on the sliding surface.[35][22].

Consider the system to be controlled is described by:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x),\end{aligned}\tag{5.15}$$

Where  $x(t)$  is the vector of state variables.  $f(x)$  and  $g(x)$  are both nonlinear function.  $u$  is the control input.  $y$  is a vector of the system outputs.

The design of the sliding mode control consists of two steps. The choice of the sliding surface, and the design of the control law.

- Step 1: propose the general form for the sliding surface[35]:

$$S(x) = \sum_{i=1}^{i=n} \Lambda_i e_i = e_n + \sum_{i=1}^{n-1} \Lambda_i e_i\tag{5.16}$$

Where  $\Lambda_i$  (Lambda) is the plant coefficient and  $e$  is the error.

Generally the sliding surface is given by the following function:

$$S(x) = e + \Lambda \dot{e} \quad (5.17)$$

Where  $\Lambda$  is a constant positive value and  $\dot{e}$  is the derivative of the error.

A suitable nonlinear control force  $u$  has to be found so as to track the error on the sliding surface. To achieve this purpose, a positive Lyapunov function  $V$  is defined as:

$$V = \frac{1}{2} S^2 \quad (5.18)$$

To insure the system is stable the following condition should be valid [35]:

$$\dot{V} = \frac{1}{2} \dot{S}^2 = S \dot{S} = -k \cdot |S| \quad (5.19)$$

Where  $k$  is a positive definite constant.

- Step 2: The Choice of the Sliding Surface.

The sliding mode control consists of two terms; the equivalent control term and the switching control term:

$$u = u_{eq} + u_s \quad (5.20)$$

Where  $u_{eq}$  is the equivalent part of the control system which consists of the system dynamics. and  $u_s$  is the switching term that is responsible for sliding along the chosen surface.

$$u_s = -k \text{sign}(S) \quad (5.21)$$

Where,

$$\text{sign}(s) = \begin{cases} -1 & \text{if } S < 0; \\ 1 & \text{if } S > 0. \end{cases}$$

## B Sliding mode control implementation

The outer loop is considered to be the internal dynamics of the quadcopter system which were not included in the inner control loop. The internal states are responsible for providing  $\phi$  and  $\theta$  desired to the inner control loop subsystem. As for the outer control loop the equation of motion are particularly hard to deal with due to the coupling complexity. In order to decouple the equations of motion the  $\psi$  angle is assumed that it does not affect the position  $x$  and  $y$  [33] [29] [30]. Substituting  $\psi = 0$  in equation 5.1a and 5.1b yields:

$$\ddot{x} = \cos(\phi) \sin(\theta) \frac{U1}{m} \quad (5.22)$$

$$\ddot{y} = -\sin(\phi) \frac{U1}{m} \quad (5.23)$$

From Equation 5.1c we have:

$$U1 = \frac{m(v_1 + g)}{\cos \phi \sin \theta} \quad (5.24)$$

Where  $v_1$  is the linear control force input provided by the FPID controller for attitude and altitude.

From Equations 5.22 and 5.23  $\phi_d$  and  $\theta_d$  can be expressed as:

$$\phi_d = \arctan\left(\frac{-u_y \cos \theta}{v_1 + g}\right) \quad (5.25)$$

$$\theta_d = \arctan\left(\frac{u_x}{v_1 + g}\right) \quad (5.26)$$

Where  $u_x$  and  $u_y$  are the Sliding mode control forces. Let us define the state error  $e = (x_d - x, y_d - y)$  and the sliding mode error as:

$$S = \dot{e} + \Lambda e \quad (5.27)$$

Note that  $\dot{e} + \Lambda e = 0$  defines a stable sliding mode surface. The function of the controller to be designed is to force the system onto this surface by making  $S$  small. The parameter  $\Lambda$  is selected for a satisfactory sliding mode response.

A proposed controller to keep  $S$  small and  $S\dot{S} < 0$  is shown in Equations 5.28 and 5.29 respectively,

$$u_x = -k_{x1} \text{sign}(S_x) + \ddot{x}_d + k_{x2} e_x \quad (5.28)$$



$$u_y = -k_{y1} \text{sign}(S_y) + \ddot{y}_d + k_{y2} e_y \quad (5.29)$$

Where  $k_x$  and  $k_y$  are positive definite constants. Moreover, selection of those constants effect the chattering phenomenon and the quality of tracking. Hence, proper balance should be considered.

To ensure the proposed control law stabilizes the internal dynamics a proof must be carried out using lyapunov method.

The lyapunov function is defined as:

$$V(S_x) = \frac{1}{2} S^2 \quad (5.30)$$

If  $\dot{V}(S_x) < 0$  then  $S\dot{S} = -k \cdot |S| < 0$ . This condition is achieved by choosing  $k$  as a positive definite constant. Thus, the necessary conditioned is fulfilled where the system's state converges to the sliding surface and the stability of the internal dynamics is guaranteed.

As a result, the controller used to stabilize the outer loop is going to have the following form:

$$\begin{aligned} \phi_d &= \arctan\left(\frac{-u_y \cos \theta}{v_1 + g}\right) = \arctan\left(\frac{-(k_{y1} \text{sign}(S_y) + \ddot{y}_d + k_{y2} e_y) \cos \theta}{v_1 + g}\right) \\ \theta_d &= \arctan\left(\frac{u_x}{v_1 + g}\right) = \arctan\left(\frac{(k_{x1} \text{sign}(S_x) + \ddot{x}_d + k_{x2} e_x)}{v_1 + g}\right) \end{aligned} \quad (5.31)$$

### C Chattering effect

After testing the developed sliding mode controller and simulating the quadcopter model. It was observed that there is a chattering effect. Where, chattering happens because the control signal exhibits high frequency oscillations after the system state reaches the sliding surface. This is due to the switching nature of the sliding mode controller. The controller is always trying to force the value of  $S$  to zero. However, due to the delay between the control action and the change of the  $s$  sign the trajectory passes the surface  $s$  and so goes on. This causes the chattering effect.

There are several approaches to implement SMC with reduced chattering. one is the boundary layer solution [35]. Another approach is the use of High Order Sliding Mode

(HOSM) algorithms[36][37].

With introducing of the saturator the the discontinuous control law is replaced by a saturation function which approximates the sign(s) term in a boundary layer of the sliding manifold. This makes the SMC formulated as,

$$u_s = -ksat(S). \quad (5.32)$$

The use of HOSM is considered the simplest alternative to obtain continuous and smooth control signals. For a smooth control signal the relative degree of the designed sliding mode surface. Which can be achieved by lowering the term  $\dot{S}$ . [35]. Thus the sliding mode control coefficients were changed to the values shown in Table 7.9

## 5.2 Hybrid Control Strategy: Configuration 2

The second configuration of the hybrid control strategy is shown in Figure 5.3. The inner control loop is exactly the same as the first configuration which is controlling the attitude and altitude using a FPID with state tranformation and state feedback. However, for the stability of the internal dynamics in the outer control loop, this configuration uses an FPID similar to the one used in chapter 4.

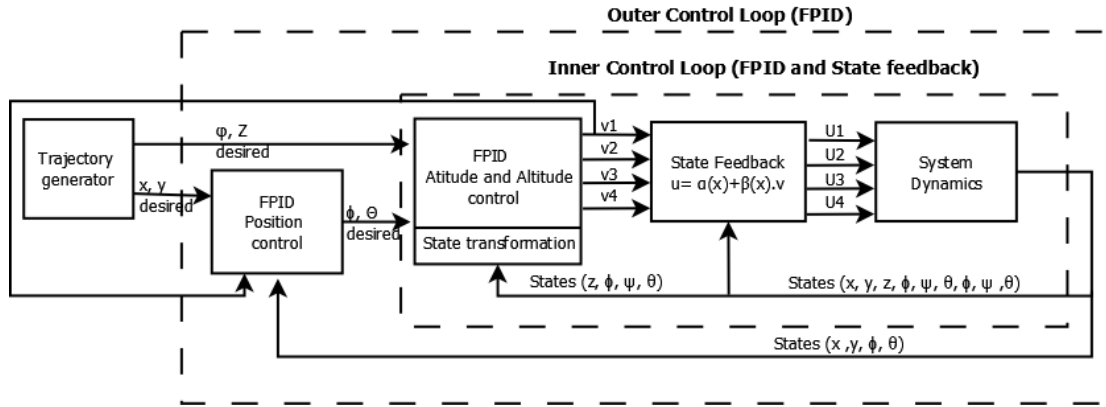


Figure 5.3: Configuration 2 of the hybrid strategy scheme using FPID, with state transformation, state feedback and FPID

### 5.2.1 Outer control loop: Stability of the internal dynamics using FPID

The controller used to stabilize the outer control loop is the FPID. The gains for the controller are scheduled using a fuzzy controller with same rules developed in chapter 4 and the relevant Equation of motion are linearized as in chapter 2.

Thus,  $\phi_d$  and  $\theta_d$  could be expressed as:

$$\theta_d = -\frac{v_x}{g} = -\frac{K_{Px}e(s) + K_{Ix}\frac{1}{s}e(s) + K_{Dx}\frac{Ns}{N+s}e(s)}{g} \quad (5.33)$$

$$\phi_d = \frac{v_y}{g} = \frac{K_{Py}e(s) + K_{Iy}\frac{1}{s}e(s) + K_{Dy}\frac{Ns}{N+s}e(s)}{g} \quad (5.34)$$

Where  $v_x$  and  $v_y$  are the linear control inputs provided by the FPID position controller to ensure the stability of the whole system.

# Chapter 6

## Trajectory Planning

The main aim of this chapter is to provide the quadcopter with a feasible trajectory in a such a way a quadcopter would accelerate and decelerate in a smooth path. Thus, a technique is required to generate a smooth trajectory given any number of waypoints.

In [38] and [39] it was suggested that smoothness of a trajectory could be achieved using jerk, which is the time derivative of acceleration or the third time derivative of position. Thus it is defined as:

$$\ddot{x}(t) = \frac{d^3 x(t)}{dt^3} \quad (6.1)$$

In order to move the quadcopter from one point to another smoothly, it should minimize the sum of the squared jerk along its trajectory. Hence, smoothness can be calculated by a jerk cost:

$$\int_{t_i}^{t_f} \ddot{x}(t)^2 dt \quad (6.2)$$

Where  $t_i$  is the start time,  $t_f$  is the end time and  $x(t)$  is a particular trajectory.

In [34] it was suggested that there is a function  $x(t)$  that smoothly connects a starting point to a target point in a given amount of time. This function  $x(t)$  has the minimum jerk cost compared to any other possible functions. To prove this let us say that we want the quadcopter to move 10 m in the the  $x$  direction in 1 second. The quadcopter will have zero velocity at the start and end point. Now a jerk cost should be assigned to each possible trajectory and then find the trajectory with the least cost [38][39]. Thus, the following function should be minimized.

$$H(x(t)) = \frac{1}{2} \int_{t_i=0}^{t_f=1} \ddot{x}(t)^2 dt \quad (6.3)$$

To find the minimum of this function a technique called calculus of variations was used [38]. The main idea is to find the derivative of the function with respect to a small perturbation and when that derivative equals zero then a minimum is found. The variation is a function that is called  $\eta(t)$ . After finding the derivative of the function with respect to the variation and using integration by parts, the final form is [38]:

$$\frac{dH(x + e\eta)}{e} \Big|_{e=0} = \int_{t_i=0}^{t_f=1} \eta x^6 dt = 0 \quad (6.4)$$

The above property is true for any function  $\eta(t)$  and therefore it could be reached that:

$$x^{(6)} = 0$$

This means that some function  $x(t)$  that have its sixth derivative equal to zero will minimize the jerk function and gives a smooth trajectory. Moreover, the differential Equation  $x^{(6)}$  which will give us a smooth trajectory has the general solution of:

$$x_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (6.5)$$

differentiating 6.5 twice yields [38][30]:

$$\dot{x}_d(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \quad (6.6)$$

$$\ddot{x}_d(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \quad (6.7)$$

Since the initial and final velocities and acceleration are equal zero, it can be concluded from Equations 6.5, 6.6, 6.7 m[38].

$$x_d = x_i + \left( d_{x1} 10t^3 - d_{x2} 15t^4 + d_{x3} 6t^5 \right) \quad (6.8)$$

Where,

$$\begin{aligned}
d_{x1} &= \frac{x_f - x_i}{(t_f - t_i)^3} \\
d_{x2} &= \frac{x_f - x_i}{(t_f - t_i)^4} \\
d_{x3} &= \frac{x_f - x_i}{(t_f - t_i)^5}
\end{aligned} \tag{6.9}$$

Where  $d_x$  is a constant which depends on initial and final conditions.

This can be extended to three positions  $[xyz]$  rather than only  $[x]$  [39] where:

$$\begin{aligned}
x_d &= x_i + \left( d_{x1}10t^3 - d_{x2}15t^4 + d_{x3}6t^5 \right) \\
y_d &= y_i + \left( d_{y1}10t^3 - d_{y2}15t^4 + d_{y3}6t^5 \right) \\
z_d &= z_i + \left( d_{z1}10t^3 - d_{z2}15t^4 + d_{z3}6t^5 \right)
\end{aligned} \tag{6.10}$$

Where  $x_i$ ,  $y_i$  and  $z_i$  are the initial positions.

An example for the method will be carried out. Let us assume that  $t_i = 0$ ,  $t_f = 10$ ,  $x_{di} = 0$  and  $x_{df} = 30$ . Thus,

$$\begin{aligned}
d_{x1} &= \frac{30 - 0}{(10 - 0)^3} = 0.03 \\
d_{x2} &= \frac{30 - 0}{(10 - 0)^4} = 0.003 \\
d_{x3} &= \frac{30 - 0}{(10 - 0)^5} = 0.0003
\end{aligned} \tag{6.11}$$

Substituting 6.13 in 6.10:

$$\begin{aligned}
x_d &= 0 + \left( 0.03 \times 10t^3 - 0.003 \times 15t^4 + 0.0003 \times 6t^5 \right) \\
x_d &= \left( 0.3t^3 - 0.045t^4 + 0.0018t^5 \right)
\end{aligned} \tag{6.12}$$

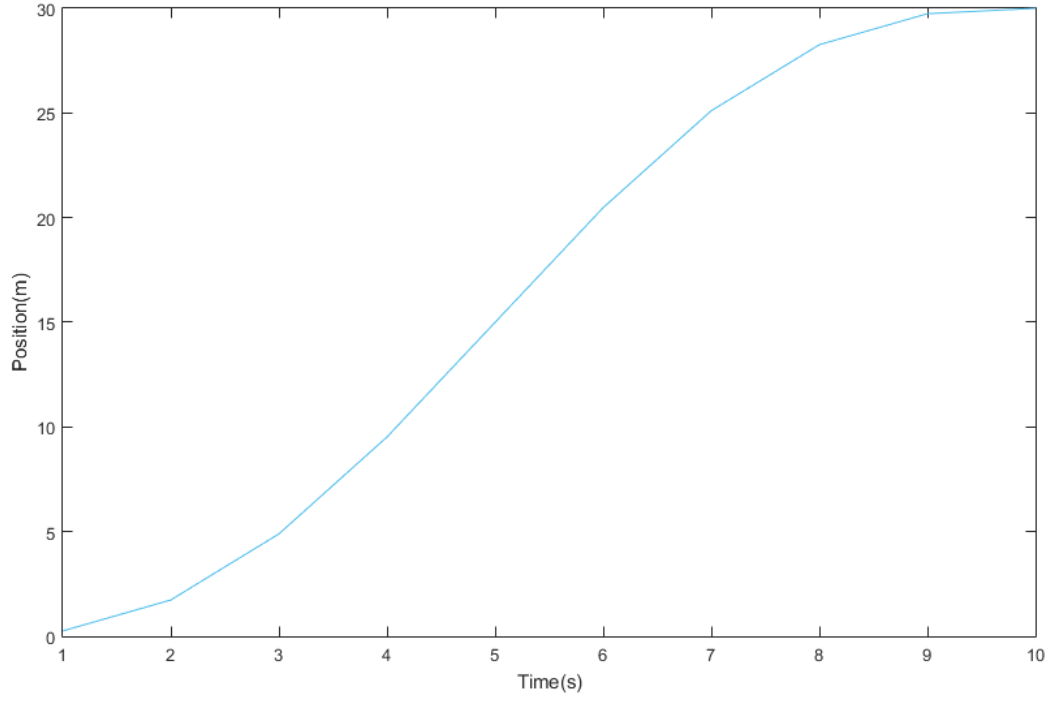


Figure 6.1: Trajectory generation for Equation 6.12

Lets introduce another example with  $t_i = 10$ ,  $t_f = 15$ ,  $x_{di} = 30$  and  $x_{df} = 40$ . Thus,

$$d_{x1} = \frac{40 - 30}{(15 - 10)3} = 0.08$$

$$d_{x2} = \frac{40 - 30}{(15 - 10)4} = 0.016$$

$$d_{x3} = \frac{40 - 30}{(15 - 10)5} = 0.0032$$
(6.13)

$$x_d = \left( 0.8t^3 - 0.24t^4 + 0.0192t^5 \right)$$
(6.14)

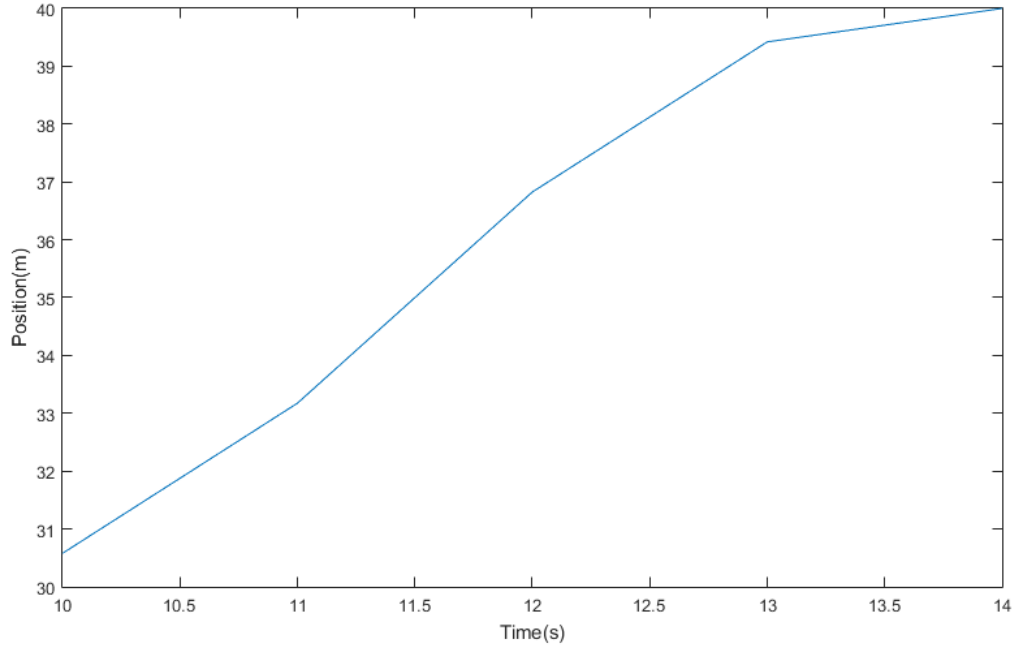


Figure 6.2: Trajectory generation for Equation 6.14

Now adding both desired positions represent the total trajectory with  $t_i = 0$ ,  $t_f = 15$ ,  $x_{di} = 0$  and  $x_{df} = 40$  with an intermediate way point with desired distance of 30 meters at a desired time of 10 seconds.

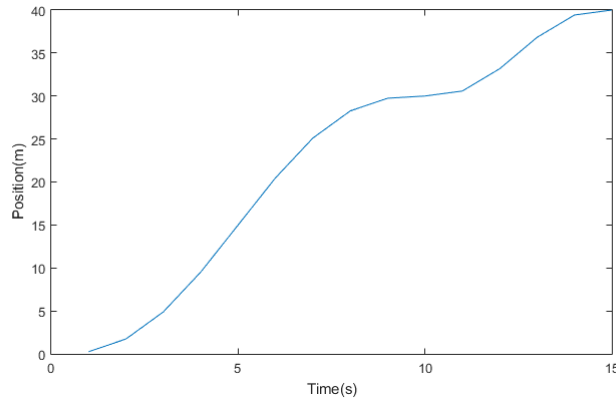


Figure 6.3: Trajectory generation with a way point

This can be extended to our quadcopter system. Where a desired initial point and final point are given with desired way points if needed. In order to implement this an algorithm



will be made to achieve the desired trajectory. Matlab was used to create the function used to plan a trajectory to achieve any desired final positions with desired way points.

To test the algorithm an example is given below with an intermediate point.

Table 6.1: Desired way points

Position/Time	0 sec	10 sec	20 sec	30 sec	40 sec	50 sec
X (m)	0	30	35	40	25	20
Y (m)	0	10	15	30	20	5
Z (m)	0	30	35	40	25	25

Given the desired way points for x, y and z in Table 6.1. Equation 6.10 to plan the quadcopter trajectory. Thus, for the first way point

At  $t = 10\text{sec}$

$$x_d = \left(0.3t^3 - 0.045t^4 + 0.0018t^5\right) \quad (6.15)$$

$$y_d = \left(0.1t^3 - 0.015t^4 + (3/5000)t^5\right) \quad (6.16)$$

$$z_d = \left(0.3t^3 - 0.045t^4 + 0.0018t^5\right) \quad (6.17)$$

At  $t = 20\text{sec}$

$$x_d = 30 + \left((1/20)t^3 - (3/400)t^4 + (3/10000)t^5\right) \quad (6.18)$$

$$y_d = 10 + \left((1/20)t^3 - (3/400)t^4 + (3/10000)t^5\right) \quad (6.19)$$

$$z_d = 30 + \left((1/20)t^3 - (3/400)t^4 + (3/10000)t^5\right) \quad (6.20)$$

At  $t = 30\text{sec}$

$$x_d = 35 + \left((1/20)t^3 - (3/400)t^4 + (3/10000)t^5\right) \quad (6.21)$$

$$y_d = 15 + \left(0.15t^3 - 0.0225t^4 + (9/10000)t^5\right) \quad (6.22)$$

$$z_d = 35 + \left((1/20)t^3 - (3/400)t^4 + (3/10000)t^5\right) \quad (6.23)$$

At  $t = 40\text{sec}$

$$x_d = 40 + \left( - (1/20)t^3 - (-3/400)t^4 + (-3/10000)t^5 \right) \quad (6.24)$$

$$y_d = 30 + \left( - (1/10)t^3 - (-0.015)t^4 + (-3/5000)t^5 \right) \quad (6.25)$$

$$z_d = 40 + \left( - (1/20)t^3 - (-3/400)t^4 + (-3/10000)t^5 \right) \quad (6.26)$$

At  $t = 50\text{sec}$

$$x_d = 25 + \left( (-1/20)t^3 - (-3/400)t^4 + (-3/10000)t^5 \right) \quad (6.27)$$

$$y_d = 20 + \left( - 0.15)t^3 - (-0.0225)t^4 + (-9/10000)t^5 \right) \quad (6.28)$$

$$z_d = 25 \quad (6.29)$$

Each planned trajectory is shown in Figures 6.4, 6.5 and 6.6.

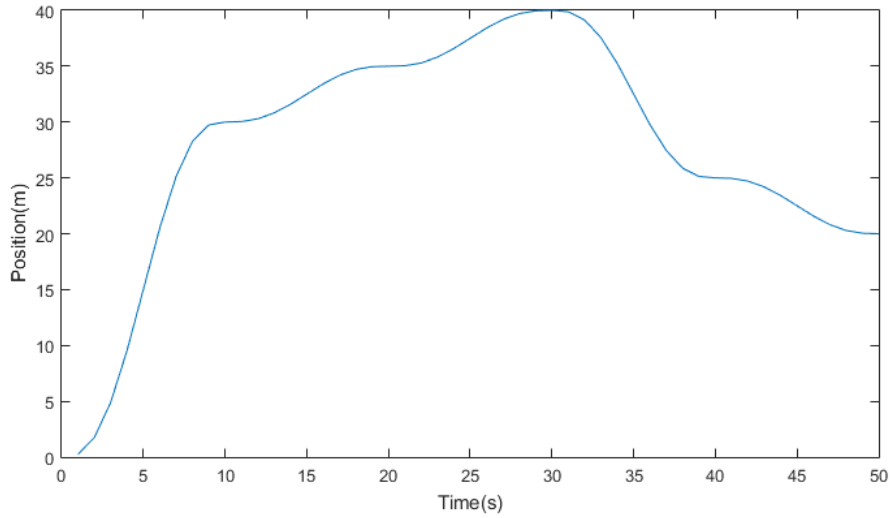


Figure 6.4: X trajectory planning

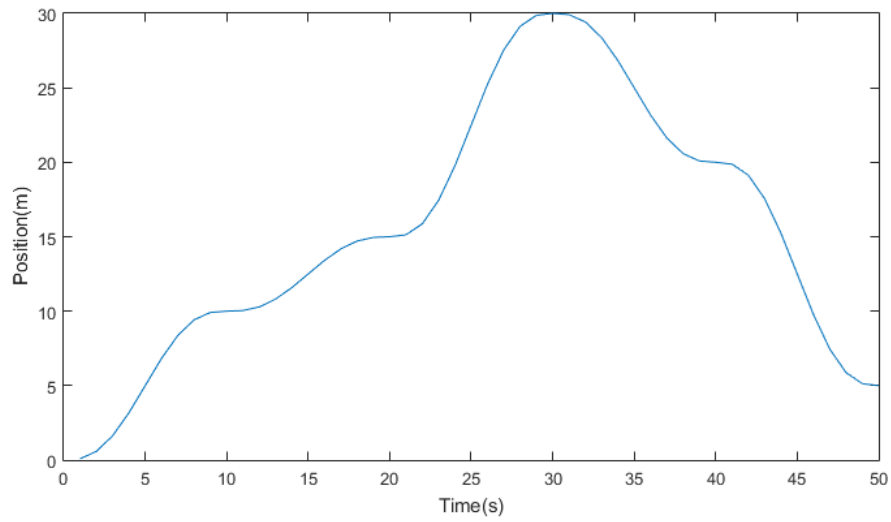


Figure 6.5: Y trajectory planning

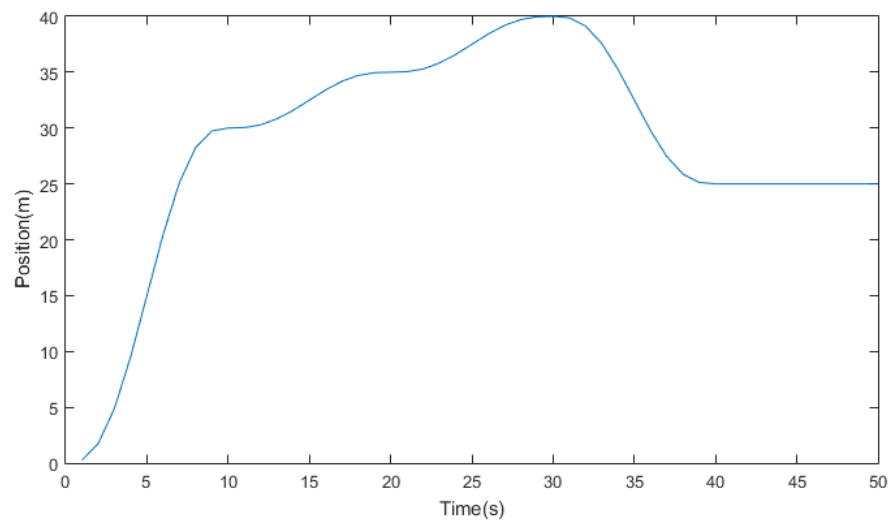


Figure 6.6: Z trajectory planning

In Figure 6.7 the three planned trajectory are plotted together. Note that this plot is not against time.

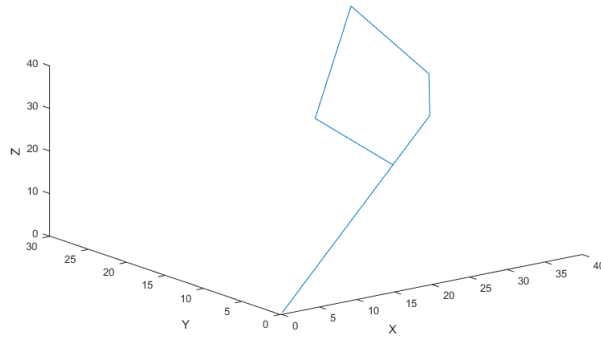


Figure 6.7: 3D trajectory generation

After developing the technique for trajectory planning, it is used to generate a trajectories for the quadcopter. The implementation of the trajectory planner with the quadcopter control system is shown in Figure 6.8.

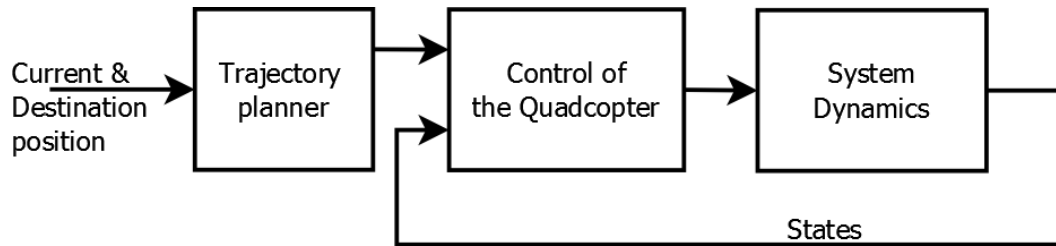


Figure 6.8: Trajectory planning added to the system

# Chapter 7

## Simulation and Results

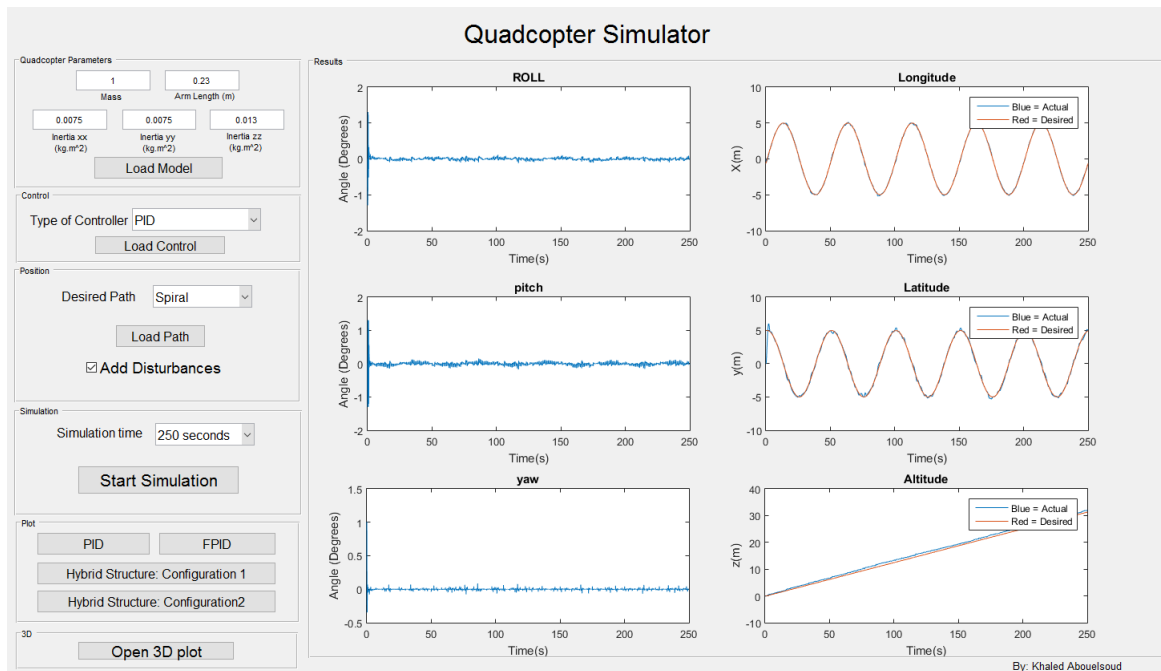
Once the control techniques were developed, simulation is carried out to validate the controllers and compare the difference between each one. Simulation was carried out under two conditions. The first one without introducing sensor measurement noise and wind disturbances while the second one with the presence of sensor measurement noise and wind disturbances. Furthermore, to make it easier to interact with and test the quadcopter's controllers a graphical user interface is developed. Moreover, for more advanced users it is possible to add on their own control techniques with ease using Simulink based model.

### 7.1 GUI

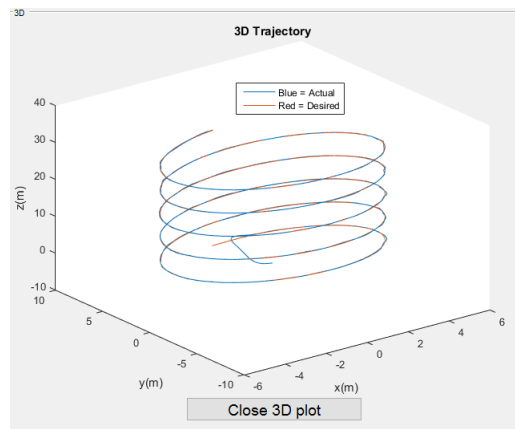
The GUI is developed using MATLAB. Each of the developed controller techniques were modeled using Simulink and integrated with the developed GUI. The GUI is user friendly and enables users to graphically interact to realize the following:

- Adjust the parameters of the quadcopter.
- Select among the available controllers to run and test.
- Select among the available predetermined paths.
- Add measurement noise and wind disturbance as necessary for the test.
- Use the default or change the simulation running time.
- Plot the attitude, altitude, position and 3D trajectory under the execution of each control technique.

The interactive layout of the developed GUI is shown in Figure 7.1.



(a) Main Interface



(b) 3D trajectory view

Figure 7.1: Graphical User Interface

## 7.2 Parameters

The parameters used are from the hardware implementation chapter 8 shown in Table 7.1, where the quadcopter parameters were identified and calculated.

Table 7.1: Parameters

Definition	Coefficient	Value	Unit
Mass	$m$	1.2	Kg
Inertia around x	$I_x x$	$2.344 * 10^{-2}$	$\text{Kg.m}^2$
Inertia around y	$I_y y$	$2.344 * 10^{-3}$	$\text{Kg.m}^2$
Inertia around z	$I_z z$	$3.333 * 10^{-2}$	$\text{Kg.m}^2$
Propeller's radius	$r$	0.10	m
Arm's length	$l$	0.3	m
Rotor's inertia	$j_r$	$6 * 10^{-5}$	$\text{Kg.m}^2$
Thrust coefficient	$C_t$	$3.13 * 10^{-5}$	$\text{Ns}^2$
Drag coefficient	$b$	$7.4 * 10^{-7}$	$\text{Nm.s}^2$
Cross section of propeller	$A_r$	$\pi r^2$	$\text{m}^2$
Density of air	$p$	1.2754	$\text{Kg.m}^3$

## 7.3 Simulation Results:

### Case 1 without including disturbance or noise

#### 7.3.1 Altitude and attitude stabilization

The stability of the quadcopter has been tested without any disturbance or noise. All the developed control techniques are given initial and desired values to test whether the quadcopter can stabilize itself. Table 8.1 lists the initial and desired values of the attitude and altitude. The simulation time was set to 10 seconds.

Table 7.2: Attitude and Altitude initial and desired values

	Initial	Desired
Attitude ( $\phi, \theta, \psi$ )	$57^\circ$	$0^\circ$
Altitude ( $z$ )	0 m	10 m

The following is the testing results in association with each of the developed control techniques.

### A PID control technique

Using the developed mathematical formulation of the PID control technique with the PID gains listed in Table 7.3. The results of the altitude ( $Z$ ) and the attitude ( $\phi, \theta, \psi$ ) stability tests are shown in Figures 7.2 and 7.3 respectively.

Table 7.3: Gains and Filter coefficient Parameters

State	$K_P$	$K_I$	$K_D$	N
Roll ( $\phi$ )	9	1	15	10
Pitch ( $\theta$ )	9	1	15	10
Yaw ( $\psi$ )	8	1	7	10
Altitude (z)	10	1	8	10
Longitude (x)	5	1	6	10
Lattitude (y)	5	1	6	10

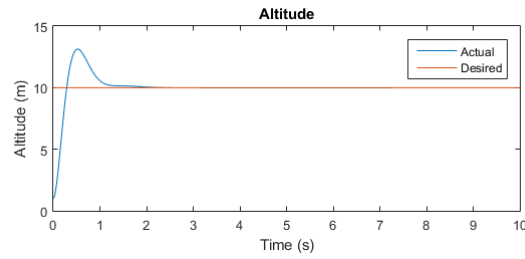


Figure 7.2: Altitude stability response using PID control



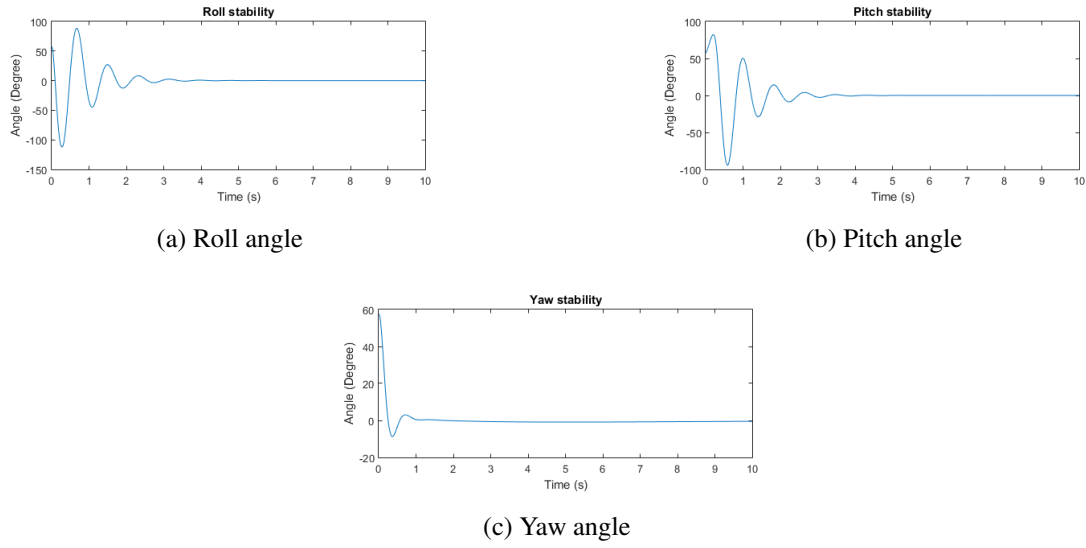


Figure 7.3: Attitude stability response using PID control

## B FPID control technique

Using the developed mathematical formulation of the FPID control technique with the insertion of PID range of output gains listed in Table 7.4 and the range of input gains listed in Table 7.5. The results of the altitude ( $Z$ ) and the attitude ( $\phi$ ,  $\theta$ ,  $\psi$ ) stability tests are shown in Figures 7.4 and 7.5 respectively.

Table 7.4: Outputs gain range

	Minimum	Maximum
$K_P$	5.5	11
$K_I$	0.7	1.5
$K_D$	7	15

Table 7.5: Inputs gain range

	Minimum	Maximum
e	-10	10
ed	-1	1

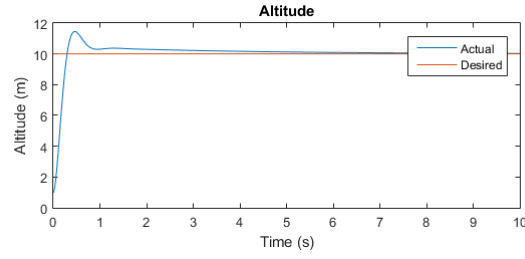
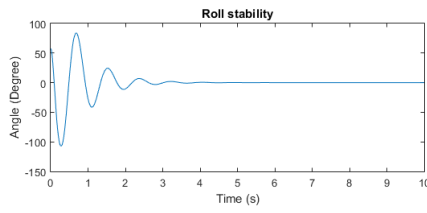
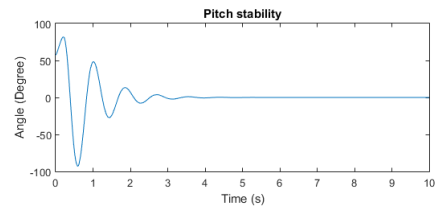


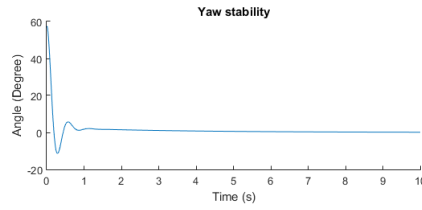
Figure 7.4: Altitude stability response using FPID control



(a) Roll angle



(b) Pitch angle



(c) Yaw angle

Figure 7.5: Attitude stability response using FPID control

### C Hybrid control strategy: Configuration 1

The testing of the first hybrid control configuration requires the setting of both FPID inputs and outputs gain range as listed in Tables 7.6 and 7.7, and setting the SMC control parameters.

Table 7.6: Outputs gain range

	Minimum	Maximum
$K_P$	8	15
$K_I$	0.7	1.5
$K_D$	18	35

Table 7.7: Inputs gain range

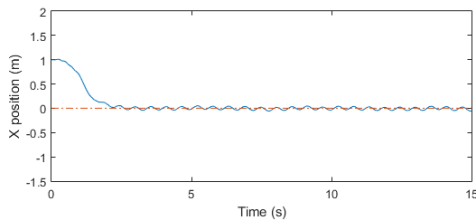
	Minimum	Maximum
e	-10	10
ed	-1	1

However, the SMC is associated with chattering effect and this effect is sensitive to the selection of its control parameters ( $k_x$  and  $k_y$ ). Hence, it is necessary to study this effect properly and find out the suitable values of these parameters while maintaining effective trajectory tracking performance. Thus, in order to illustrate chattering effect within configuration 1, the selected SMC control parameters are listed in Table 7.8. Accordingly, position  $[x \ y]$ , roll desired ( $\phi_d$ ) and pitch desired ( $\theta_d$ ) signals are tested using the parameters listed in Tables 7.6, 7.7, and 7.8 respectively.

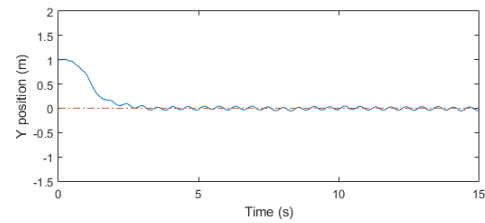
Table 7.8: Sliding Mode control simulation parameters

State	$\Lambda$	$k_1$	$k_2$
$x$	1.1	5	0.8
$y$	0.8	7	0.7

Figures 7.6 and 7.7 are the results which shows the positions  $[x, \ y]$  converging to zero with a noticeable chattering. While, roll desired ( $\phi_d$ ) and pitch desired ( $\theta_d$ ) angles shows severe chattering..



(a) Longitude (X)



(b) Latitude (Y)

Figure 7.6:  $x$  and  $y$  position with chattering effect.

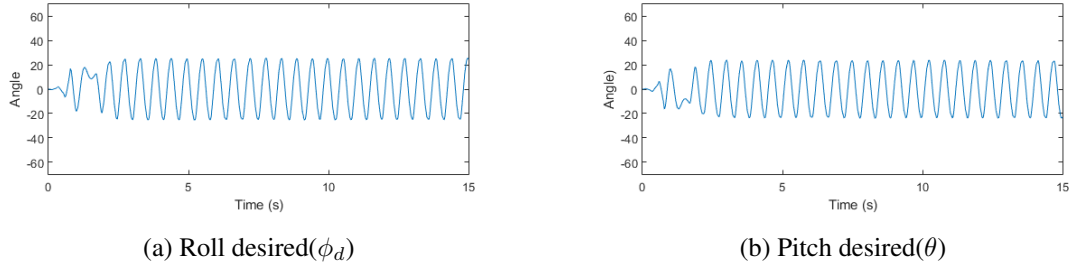


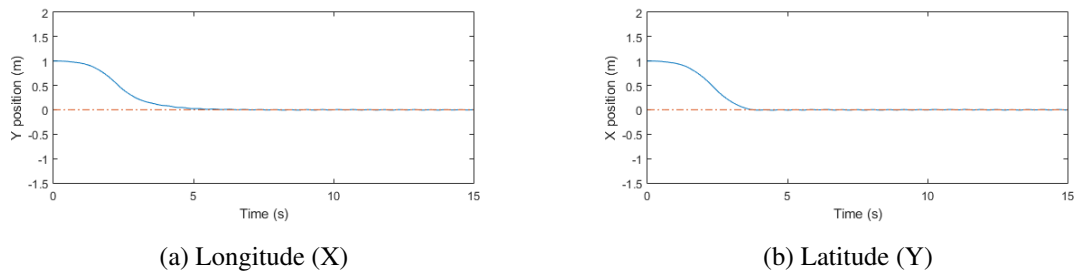
Figure 7.7: Roll and pitch with chattering effect.

Now, SMC control parameters were tuned to reduce the effect of chattering and set it as listed in Table 7.9. The same test was conducted with the parameters listed in Tables 7.6, 7.7, and 7.9 respectively.

Table 7.9: Sliding Mode control refined simulation parameters

State	$\Lambda$	$k_1$	$k_2$
$x$	1.1	0.7	0.002
$y$	0.8	0.5	0.01

The results are presented in Figures 7.8 and 7.9 respectively. It shows a smoother signal for the positions  $[x, y]$  with reduced chattering for roll desired( $\phi_d$ ) and pitch desired( $\theta_d$ ) angles.

Figure 7.8:  $x$  and  $y$  position with reduced chattering effect.

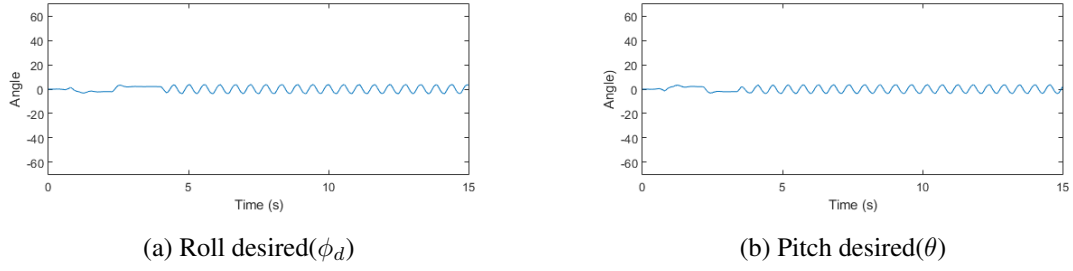


Figure 7.9: Roll and pitch with reduced chattering effect.

After concluding the required setting parameters listed in Tables 7.6, 7.7, and 7.9 for FPID and SMC within the first configuration of the hybrid control strategy. The results of the altitude ( $z$ ) and the attitude ( $\phi, \theta, \psi$ ) stability tests are shown in Figures 7.10 and 7.11 respectively.

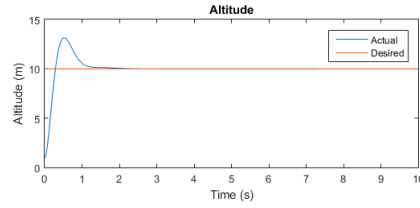


Figure 7.10: Altitude stability response using first hybrid control configuration

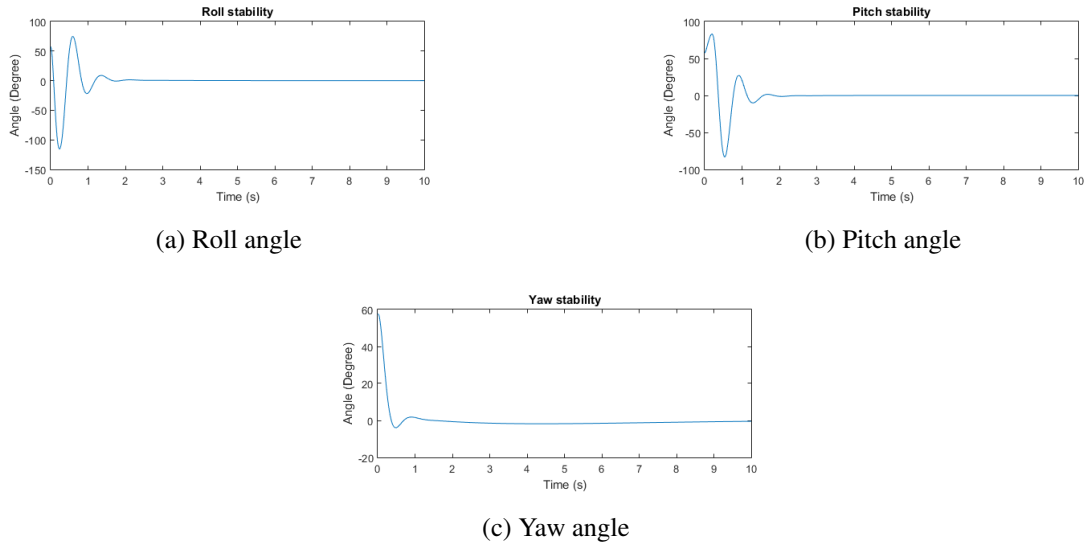


Figure 7.11: Attitude stability response using first hybrid control configuration

Moreover, to illustrate the chattering effect, the same test was carried out using the parameters listed in Tables 7.6, 7.7, and 7.8. The results of the attitude and altitude stability test with the chattering effect are shown in Figures 7.12 and 7.13 respectively.

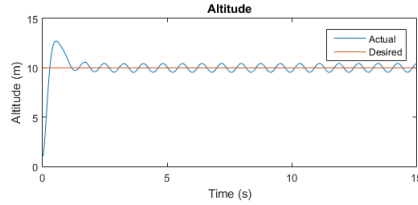
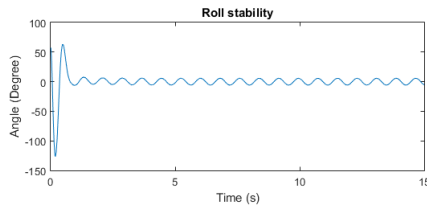
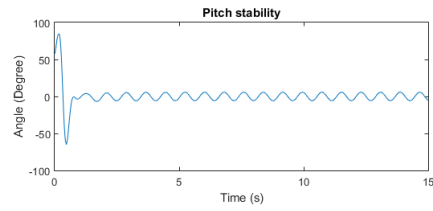


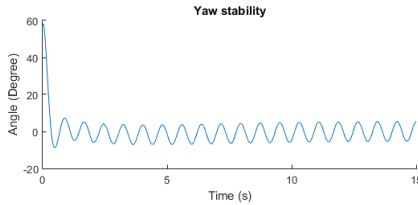
Figure 7.12: Altitude stability response using first hybrid control configuration with chattering



(a) Roll angle



(b) Pitch angle



(c) Yaw angle

Figure 7.13: Attitude stability response using first hybrid control configuration with chattering

## D Hybrid control strategy: Configuration 2

Using the developed mathematical formulation of the second configuration of the hybrid control strategy and using the parameters listed in Tables 7.6 and 7.7 for the inner control loop FPID while using the parameters in Tables 7.4 and 7.5 for the outer control loop

FPID. The results of the altitude ( $Z$ ) and the attitude ( $\phi, \theta, \psi$ ) stability tests are shown in Figures 7.14 and 7.15 respectively.

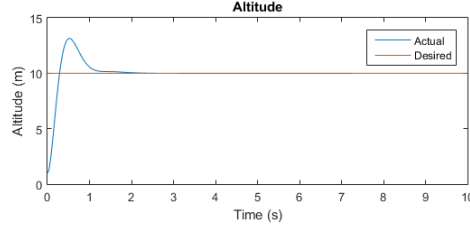


Figure 7.14: Altitude stability response using second hybrid control configuration

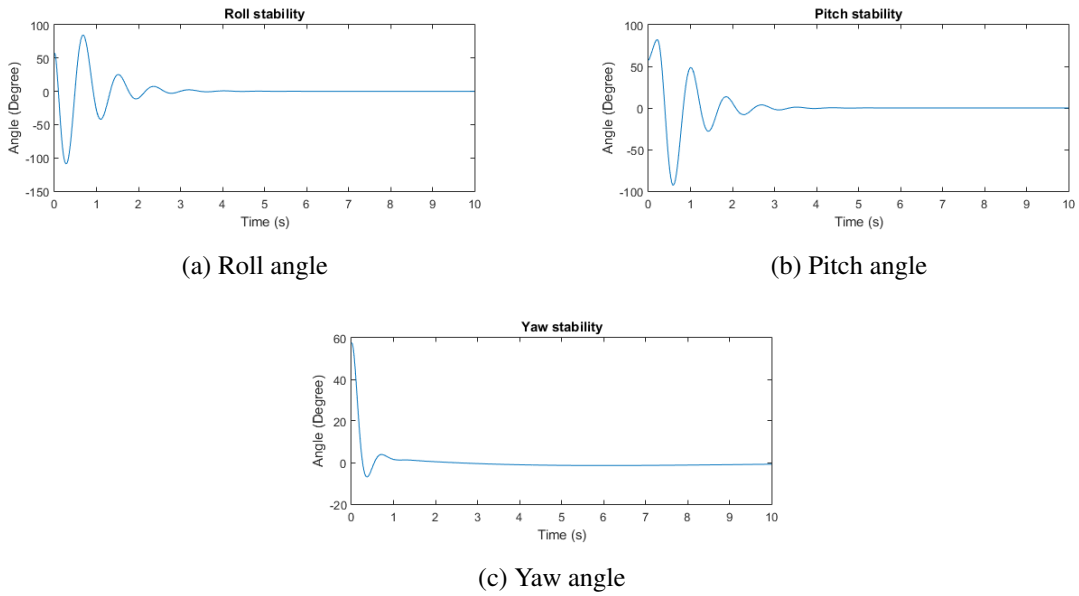


Figure 7.15: Attitude stability response using second hybrid control configuration

### E Comparison between all developed control techniques: Altitude and attitude

A comparison is carried out in Figures 7.16 and 7.17 between the four developed control techniques. The response of the quadcopter using four control techniques shows good stability with very small steady state error. This is mainly because no uncertainties are added to the simulation.

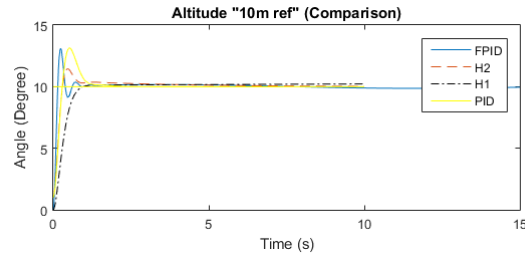
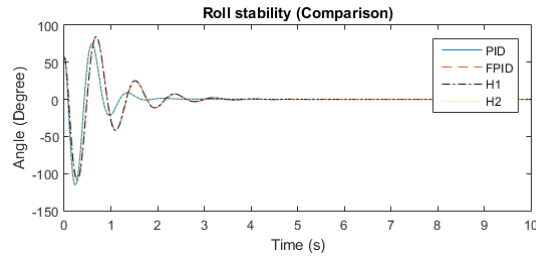
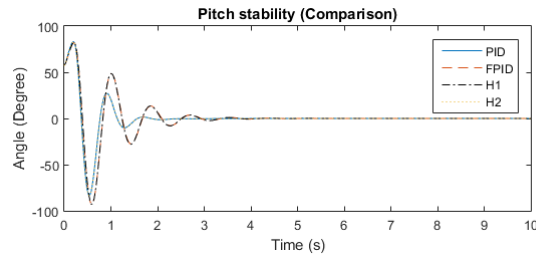


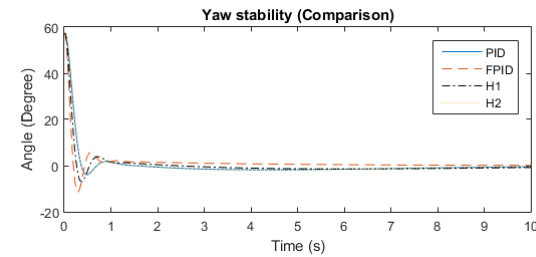
Figure 7.16: Altitude stability comparison without disturbances



(a) Roll angle



(b) Pitch angle



(c) Yaw angle

Figure 7.17: Attitude stability comparison

### 7.3.2 Position tracking

The ability of the quadcopter for position tracking has been tested without any disturbance or noise. All the developed control techniques were given a position trajectory to follow.

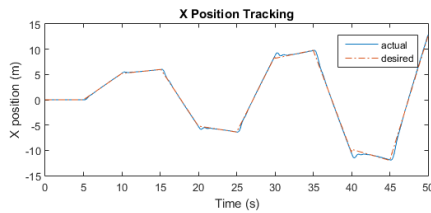


The simulation time was set to 50 seconds.

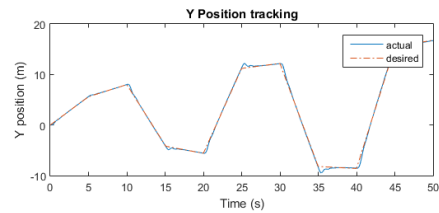
### A PID control technique

The given PID grains are listed in Table 7.3 and the test results of longitude ( $x$ ) and latitude ( $y$ ) tracking are shown in Figure 7.18.

Using the developed mathematical formulation of the PID control technique with the PID parameters listed in Table 7.3. The results of longitude ( $x$ ) and latitude ( $y$ ) trajectory tracking tests is shown in Figure 7.18.



(a) Longitude (X)

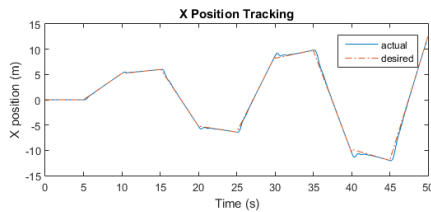


(b) Latitude (Y)

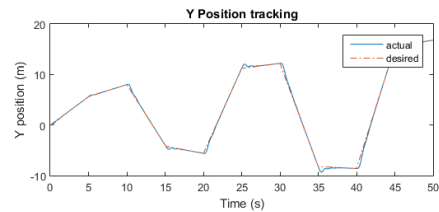
Figure 7.18: Longitude and latitude tracking using PID control

### B FPID control technique

The listed range of PID output gains and the range of input gains are listed in Tables 7.4 and 7.5. The test results of longitude ( $x$ ) and latitude ( $y$ ) tracking are shown in Figure 7.19.



(a) Longitude (X)



(b) Latitude (Y)

Figure 7.19: Longitude and latitude tracking using FPID control

### C Hybrid control strategy: Configuration 1

The developed hybrid control strategy with first configuration was tested using parameters listed in Tables 7.6, 7.7 and 7.9 and the test results of longitude ( $x$ ) and latitude ( $y$ ) tracking tests are shown in Figure 7.20.

7.6, 7.7 and 7.9, the results of longitude ( $x$ ) and latitude ( $y$ ) trajectory tracking tests is shown in Figure 7.20.

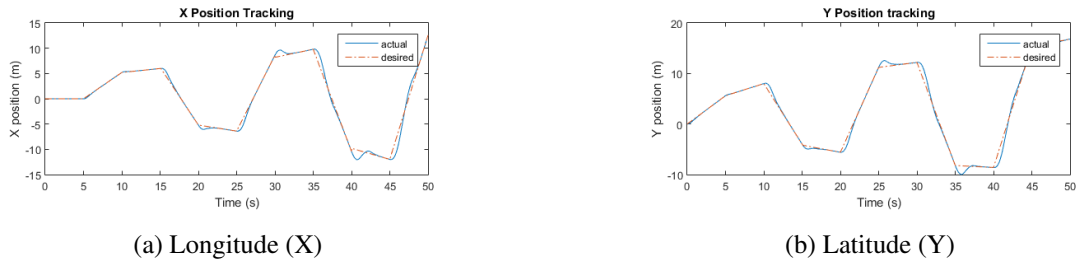


Figure 7.20: Longitude and latitude tracking using first control configuration

### D Hybrid control strategy: Configuration 2

The developed hybrid control strategy with second configuration was tested using the parameters listed in Tables 7.6, 7.7 for the inner FPID and Tables 7.4 and 7.5 for the outer control loop. The test results of longitude ( $x$ ) and latitude ( $y$ ) tracking tests are shown in Figure 7.21.

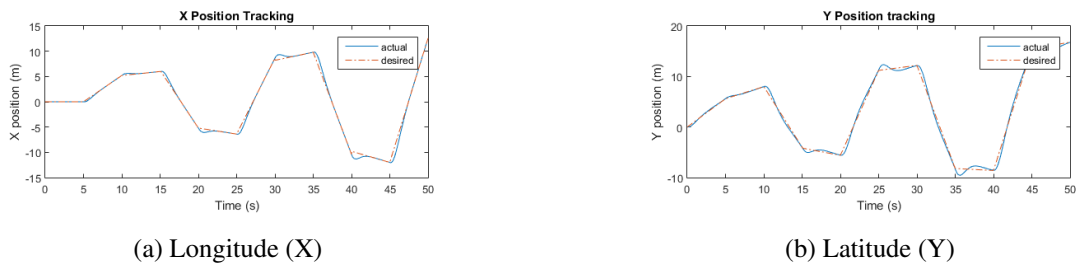


Figure 7.21: Longitude and latitude tracking using second hybrid control configuration

### E Comparison between all developed control techniques: Position tracking

Comparison between the four developed control techniques are shown in Figure 7.22. The four control techniques shows good results in terms of the position x and y tracking with almost the same values.

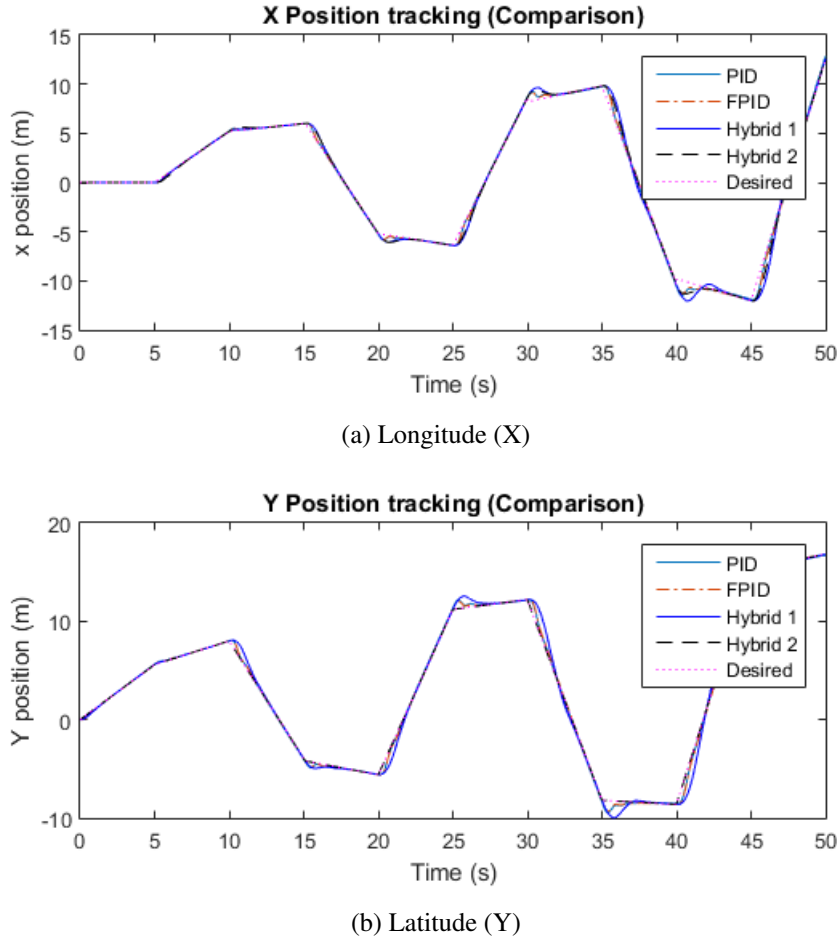


Figure 7.22: Longitude and latitude tracking response (comparison)

### 7.3.3 3D trajectory tracking

This simulation test aims to evaluate the ability of each controller to follow a desired 3D trajectory. Two 3D trajectories are tested: spiral trajectory and a random trajectory. The four control techniques shows good results in terms of the following the 3D trajectory with almost the same values.

## A Spiral trajectory

A spiral trajectory is generated and it is given the desired trajectory. The simulation run-time was set to 250 seconds. Figure 7.23 shows the ability of each of the developed control techniques to track the desired 3D trajectory while Figure 7.24 shows a comparison between the tracking ability of all the control techniques.

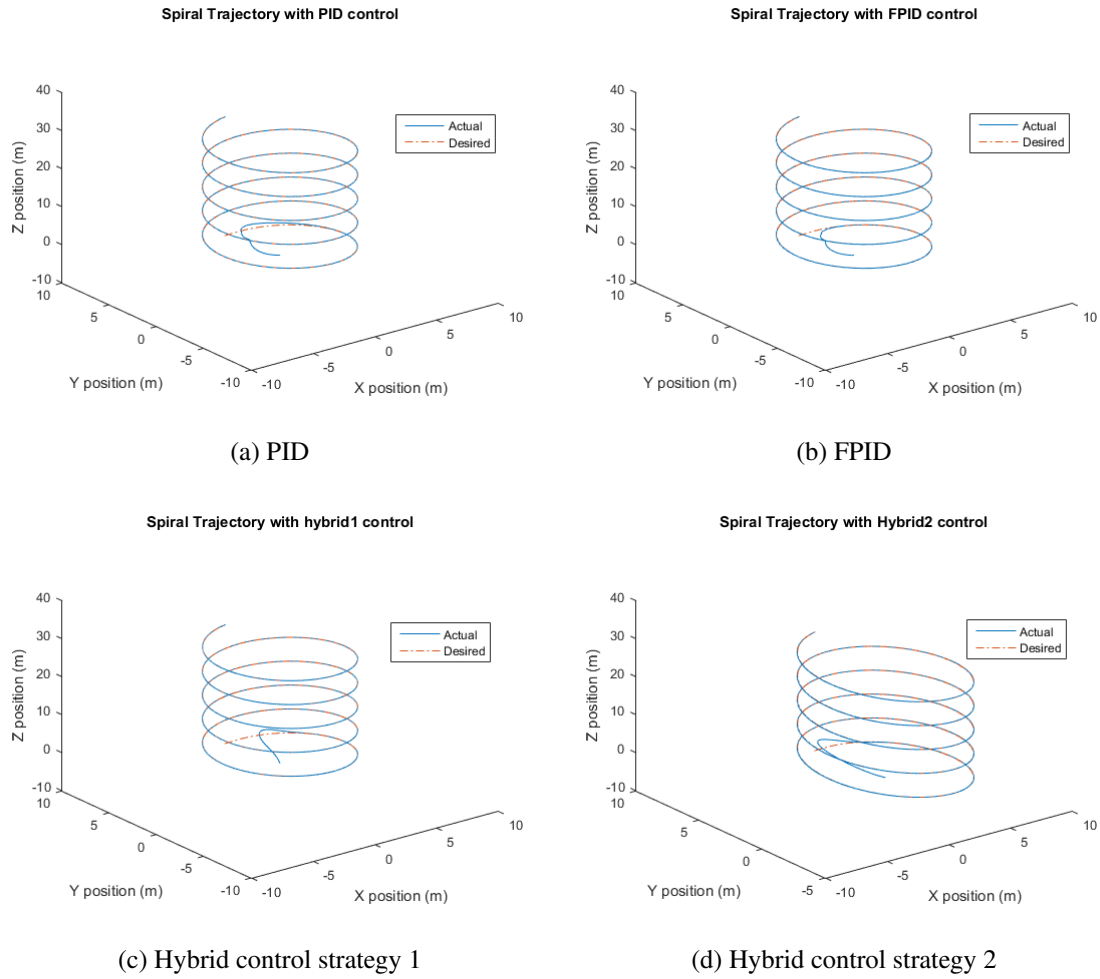


Figure 7.23: Spiral trajectory tracking response with each of the developed control techniques

### Spiral Trajectory Comparison

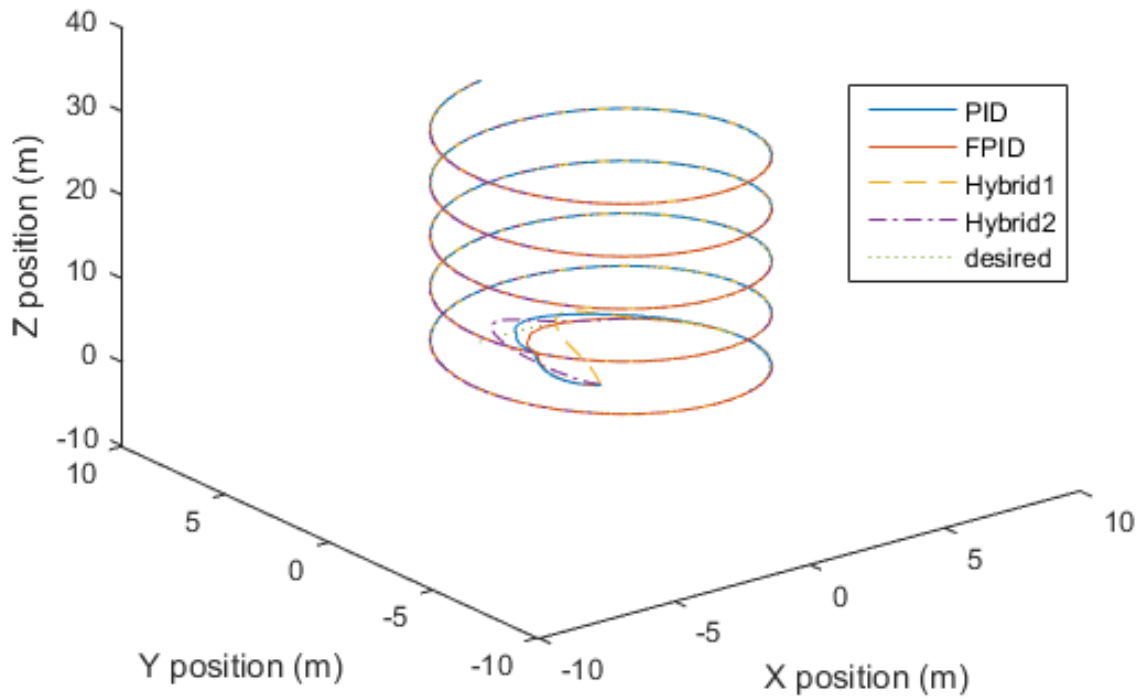


Figure 7.24: Spiral trajectory tracking response comparison between all the control techniques

### B Random trajectory

A random trajectory is generated and it is given to the system as the trajectory. The simulation run time was set to 50 seconds. Figure 7.25 shows each control ability to track the desired trajectory while Figure 7.26 shows a comparison between all the control techniques.

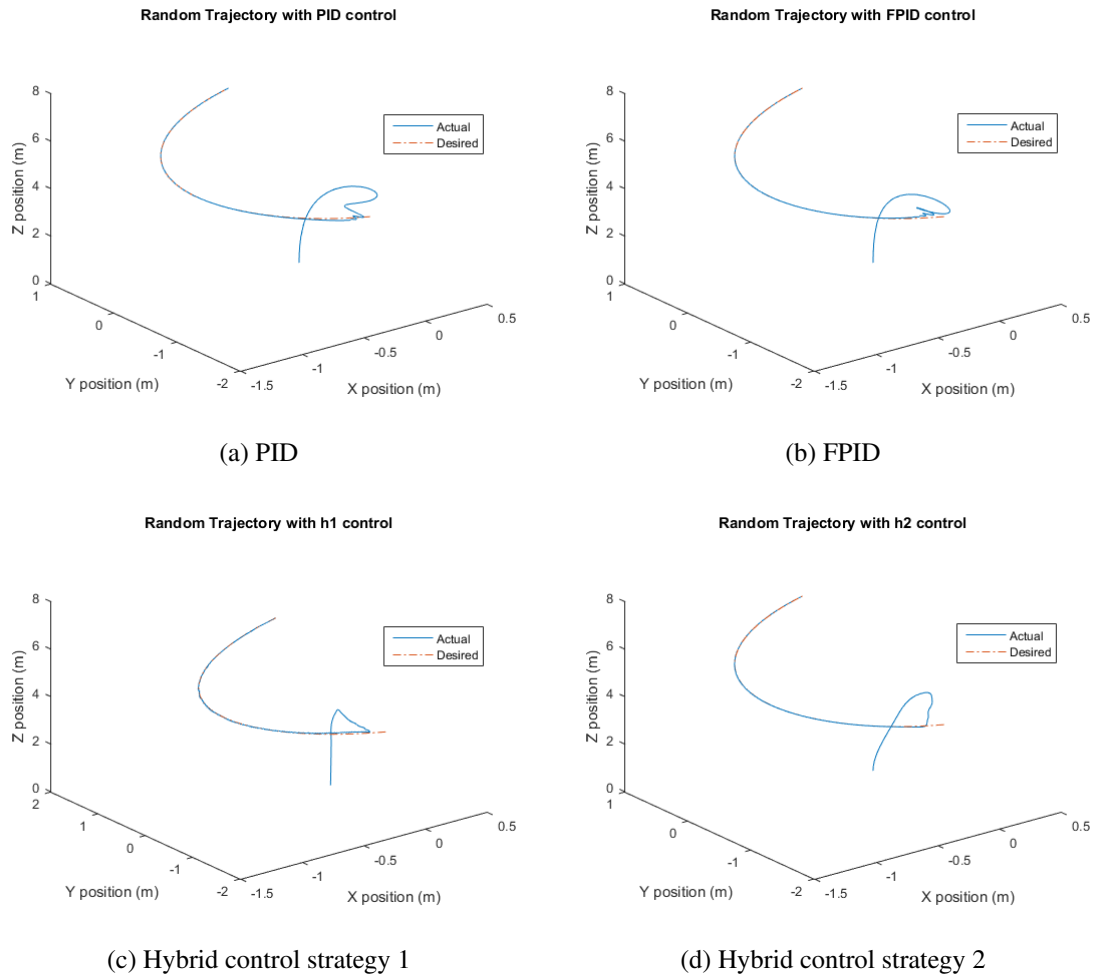


Figure 7.25: Random trajectory tracking with each of the developed control techniques.

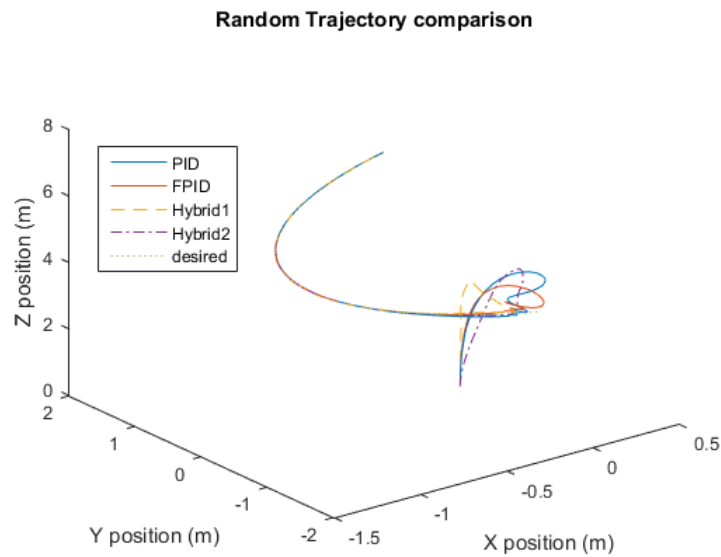


Figure 7.26: Random trajectory tracking comparison between all the control techniques

## 7.4 Simulation Results:

### Case 2 with the presence of disturbance and noise

In this section the simulation is performed to verify the robustness of each control technique against wind disturbances and sensor measurement noise. The wind disturbance was introduced as a frequency of  $\cos(2t)$  while the measurement noise was introduced to the system as uniform noise with zero-mean and standard deviation of 0.05. [22][30][31]

#### 7.4.1 Altitude and attitude stabilization with noise and disturbance

All the developed control techniques are given initial and desired values to test whether the quadcopter can stabilize itself. Table 8.1 lists the initial and desired values. The simulation time was set to 10 seconds. The following is the testing results in association with each control technique.

##### A PID control technique

The PID control technique with the parameters listed in Table 7.3 was tested and the test results of altitude ( $Z$ ) and attitude ( $\phi \theta \psi$ ) stability are shown in Figures 7.27 and 7.28 respectively.

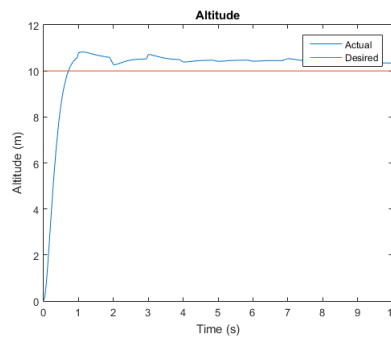


Figure 7.27: Altitude stability response using PID control

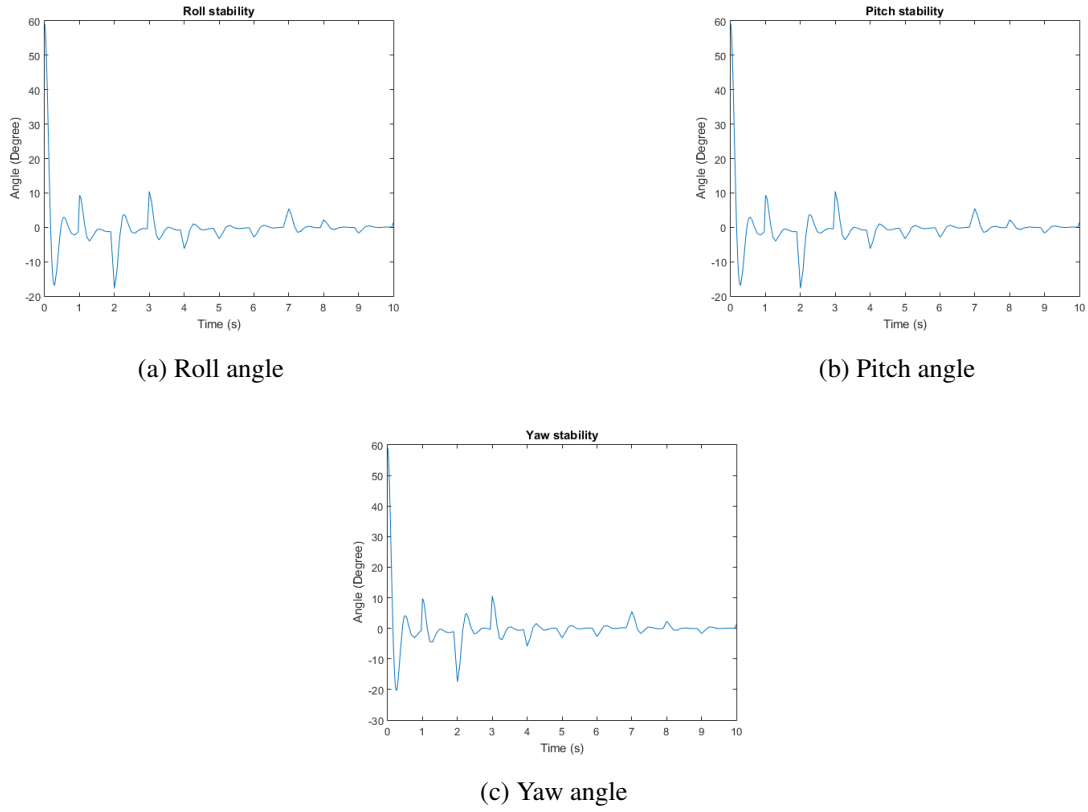


Figure 7.28: Attitude stability response using PID control

## B FPID control technique

The FPID control technique with the PID parameters range listed in Tables 7.4 and 7.5 was tested and the results of altitude ( $Z$ ) and attitude ( $\phi$ ,  $\theta$ ,  $\psi$ ) stability are shown in Figures 7.29 and 7.30 respectively.

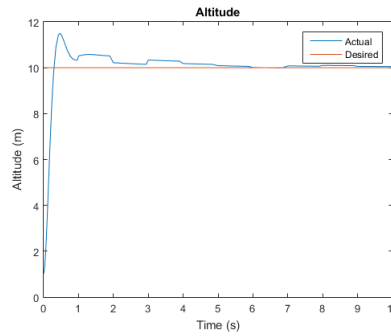


Figure 7.29: Altitude stability response using FPID control



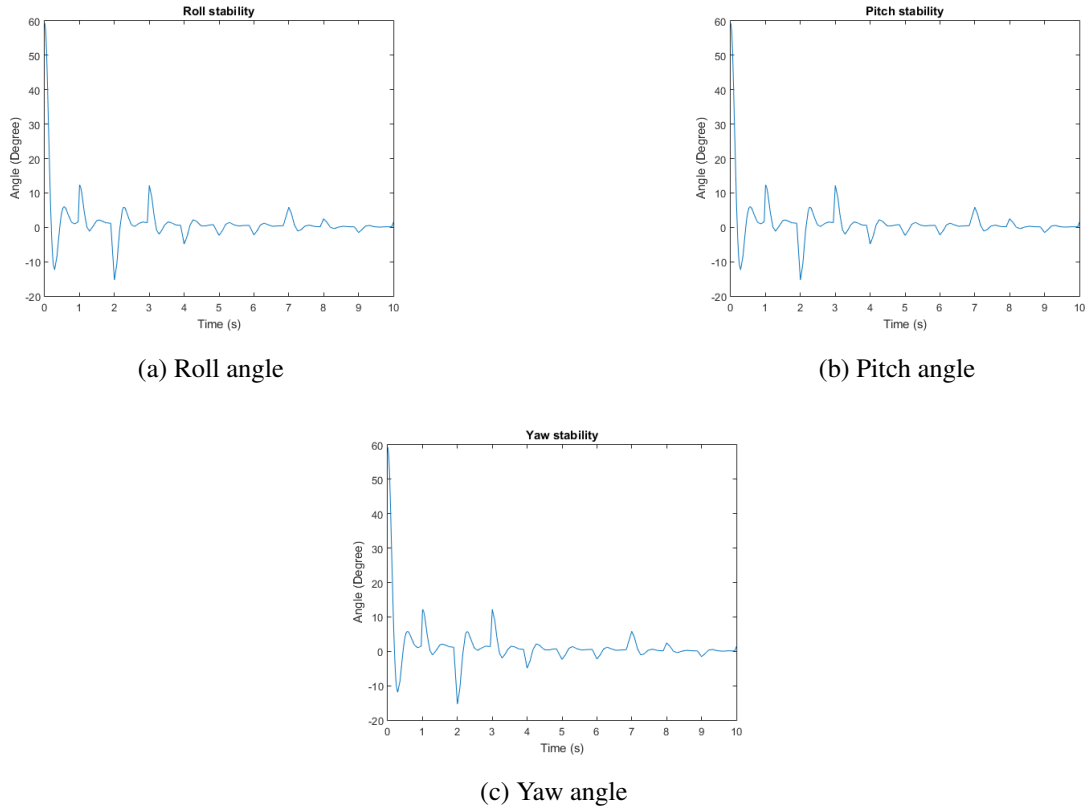


Figure 7.30: Attitude stability response using FPID control

### C Hybrid control strategy: Configuration 1

The first configuration of the hybrid control strategy with the parameters listed in Tables 7.6, 7.7 and 7.9 was tested and the test results of the altitude ( $Z$ ) and the attitude ( $\phi, \theta, \psi$ ) stability are shown in Figures 7.31 and 7.32 respectively.

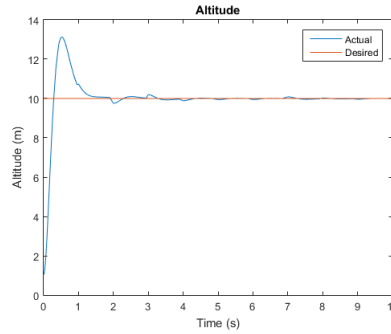


Figure 7.31: Altitude stability response using first hybrid control configuration

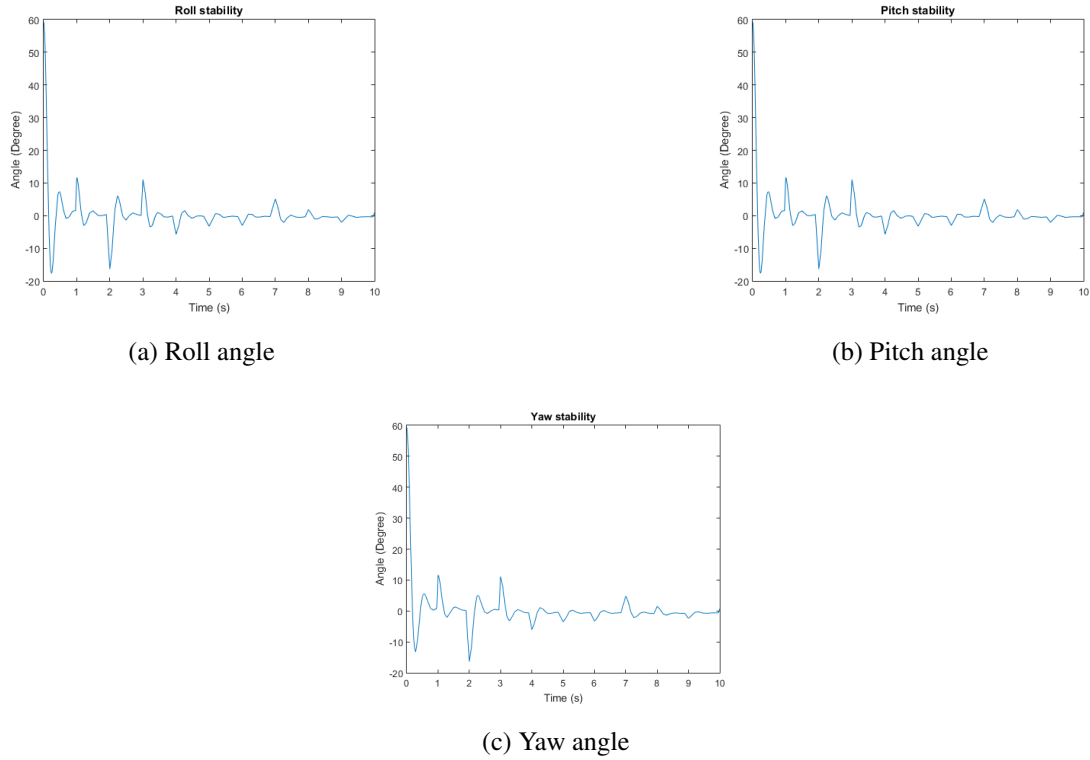


Figure 7.32: Attitude stability response using first hybrid control configuration

#### D Hybrid control strategy: Configuration 2

The second configuration of the hybrid control strategy using the parameters listed in Tables 7.6 and 7.7 for the inner control loop FPID while using the parameters in Tables 7.4 and 7.5 for the outer control loop was tested and the results of altitude ( $Z$ ) and attitude ( $\phi, \theta, \psi$ ) stability are shown in Figures 7.33 and 7.34 respectively.

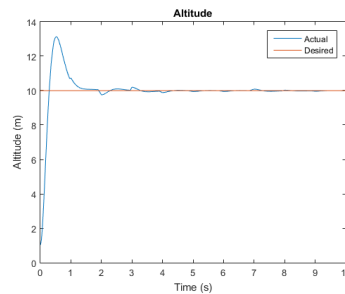


Figure 7.33: Altitude stability response using second hybrid control configuration

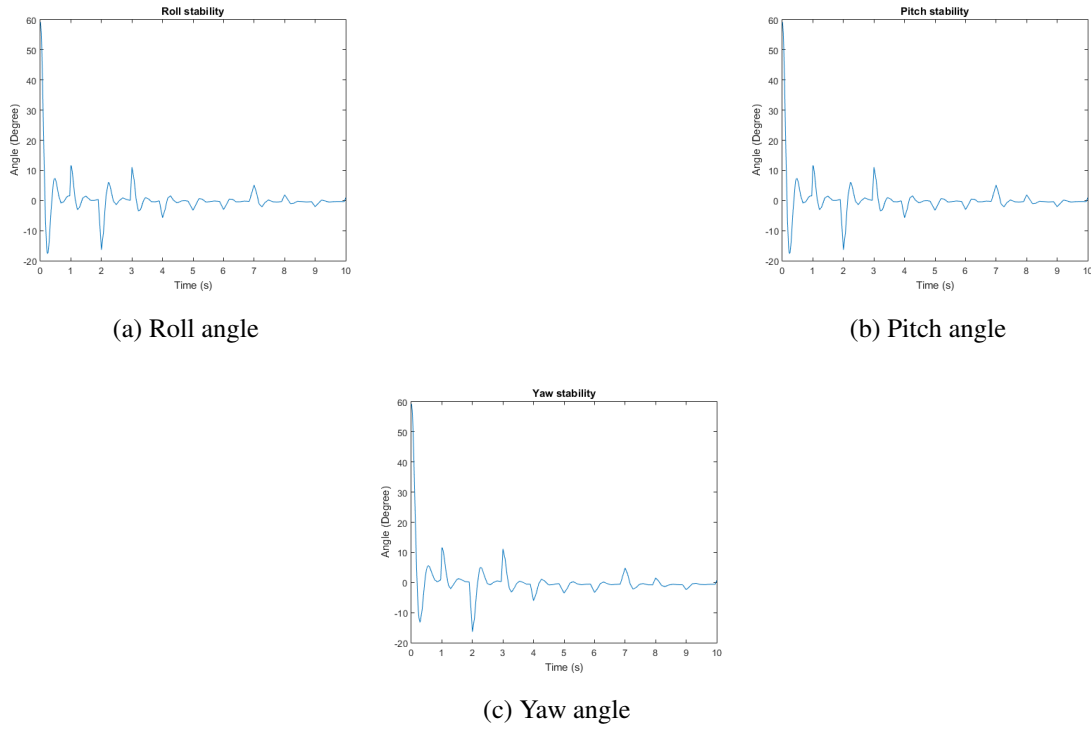


Figure 7.34: Attitude stability response using second hybrid control configuration

### E Comparison between all developed control techniques: Altitude and attitude

A comparison is made in Figures 7.36 and 7.35 between the four control techniques where the first configuration in the hybrid control strategy shows robust results compared to the other three techniques.

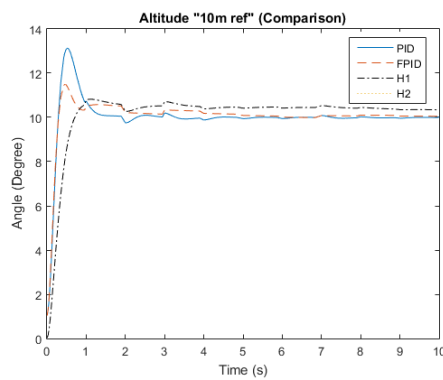


Figure 7.35: Altitude stability comparison between all the control techniques

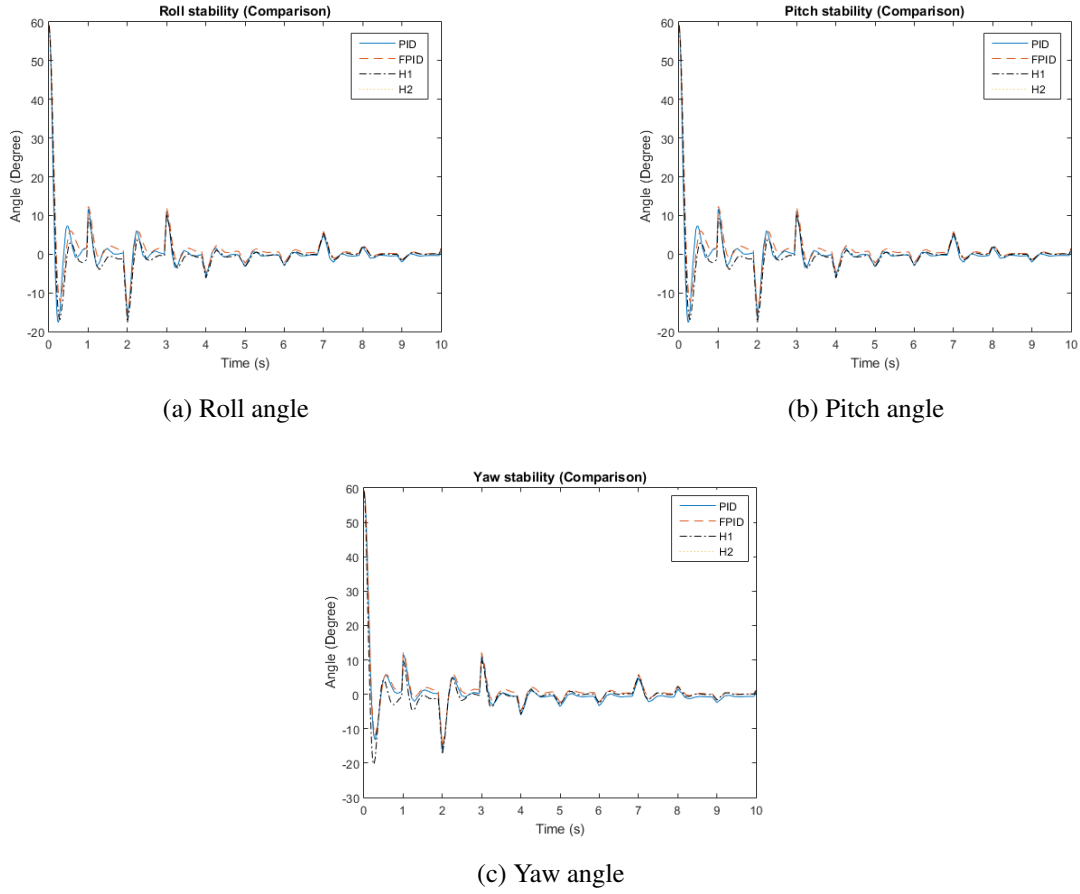


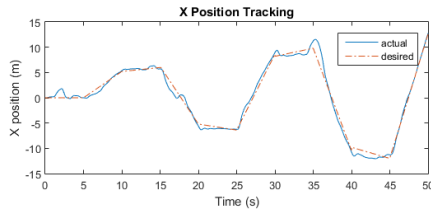
Figure 7.36: Attitude stability comparison between all the control techniques

## 7.4.2 Position tracking with noise and disturbance

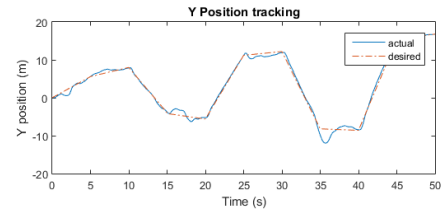
The ability of the quadcopter to track position has been tested with the presence of disturbance and noise. All the developed control techniques are given a position trajectory to follow. The simulation time was set to 50 seconds.

### A PID control technique

The PID control technique with the PID parameters listed in Table 7.3 was tested and the results of longitude ( $x$ ) and latitude ( $y$ ) position tracking are shown in Figure 7.37.



(a) Longitude (X)

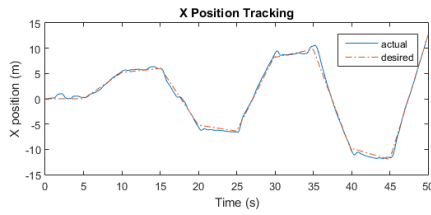


(b) Latitude (Y)

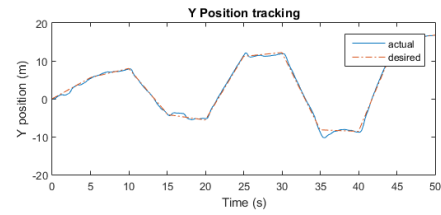
Figure 7.37: Longitude and latitude tracking using PID control

## B FPID control technique

The FPID control technique with the output gain range listed in Tables 7.4 and 7.5. The test results of longitude ( $x$ ) and latitude ( $y$ ) position tracking tests are shown in Figure 7.38.



(a) Longitude (X)



(b) Latitude (Y)

Figure 7.38: Longitude and latitude tracking using FPID control

## C Hybrid control structure: Configuration 1

The first configuration of the hybrid control strategy with the parameters listed in Tables 7.6, 7.7 and 7.9 was tested and the test results of longitude ( $x$ ) and latitude ( $y$ ) position tracking are shown in Figure 7.39.

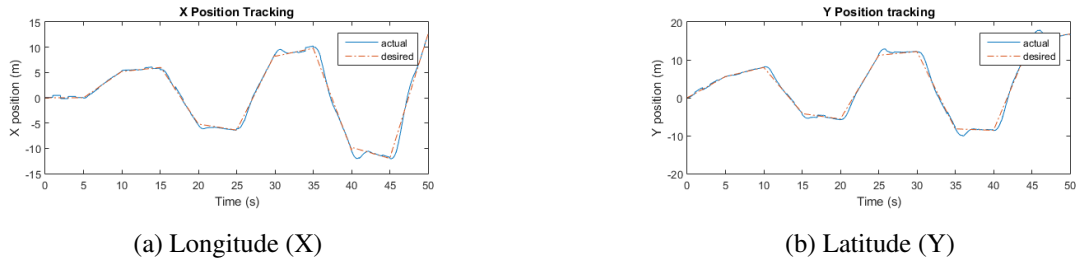


Figure 7.39: Longitude and latitude tracking using first hybrid control configuration

#### D Hybrid control strategy: Configuration 2

The second configuration of the hybrid control strategy with the parameters listed in Tables 7.6 and 7.7 for the inner control loop FPID while using the parameters in Tables 7.4 and 7.5 for the outer control loop. The test results of longitude ( $x$ ) and latitude ( $y$ ) position tracking are shown in Figure 7.40.

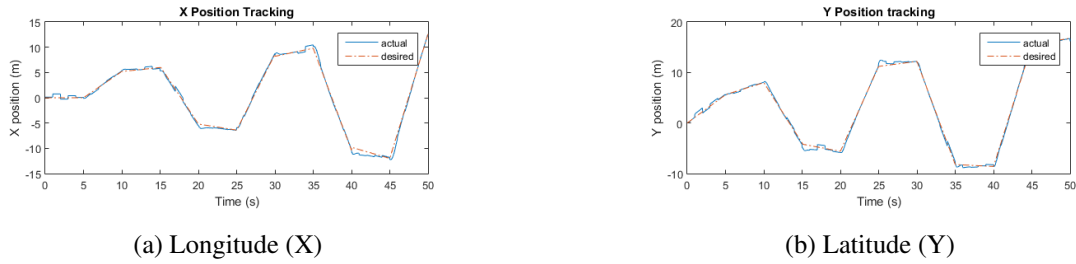
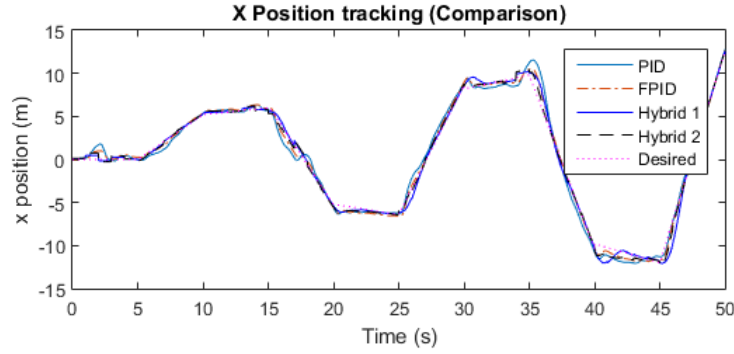


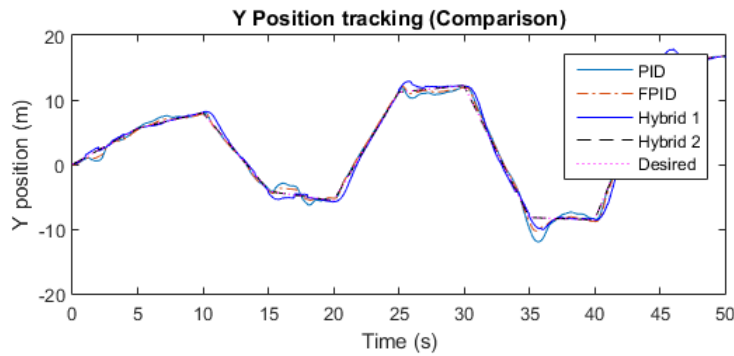
Figure 7.40: Longitude and latitude tracking using second hybrid control configuration

#### E Comparison between all developed control techniques: Position tracking

Comparison between the four developed control techniques are shown in Figure 7.41 where the first configuration in the hybrid control strategy shows robust results compared to the other three techniques. The second configuration and the FPID shows small perturbations and the PID control shows large perturbations.



(a) Longitude (X)



(b) Latitude (Y)

Figure 7.41: Longitude and latitude tracking comparison between all the control techniques

### 7.4.3 3D Trajectory tracking with noise and disturbance

The final simulation is to test the ability of each of the developed control techniques to follow a desired 3D trajectory under the presence of disturbance and noise. Two 3D trajectories were tested: a spiral trajectory and a random trajectory where the first configuration in the hybrid control strategy shows robust results compared to the other three techniques. The second configuration and the FPID control shows small perturbations and the PID control shows large perturbations.

#### A Spiral trajectory

A spiral trajectory is generated and it is given to the system as the desired trajectory. The simulation runtime was set to 250 seconds. Figure 7.42 shows each of the developed

control techniques ability to track the desired 3D trajectory while Figure 7.43 shows a comparison between all the developed control techniques.

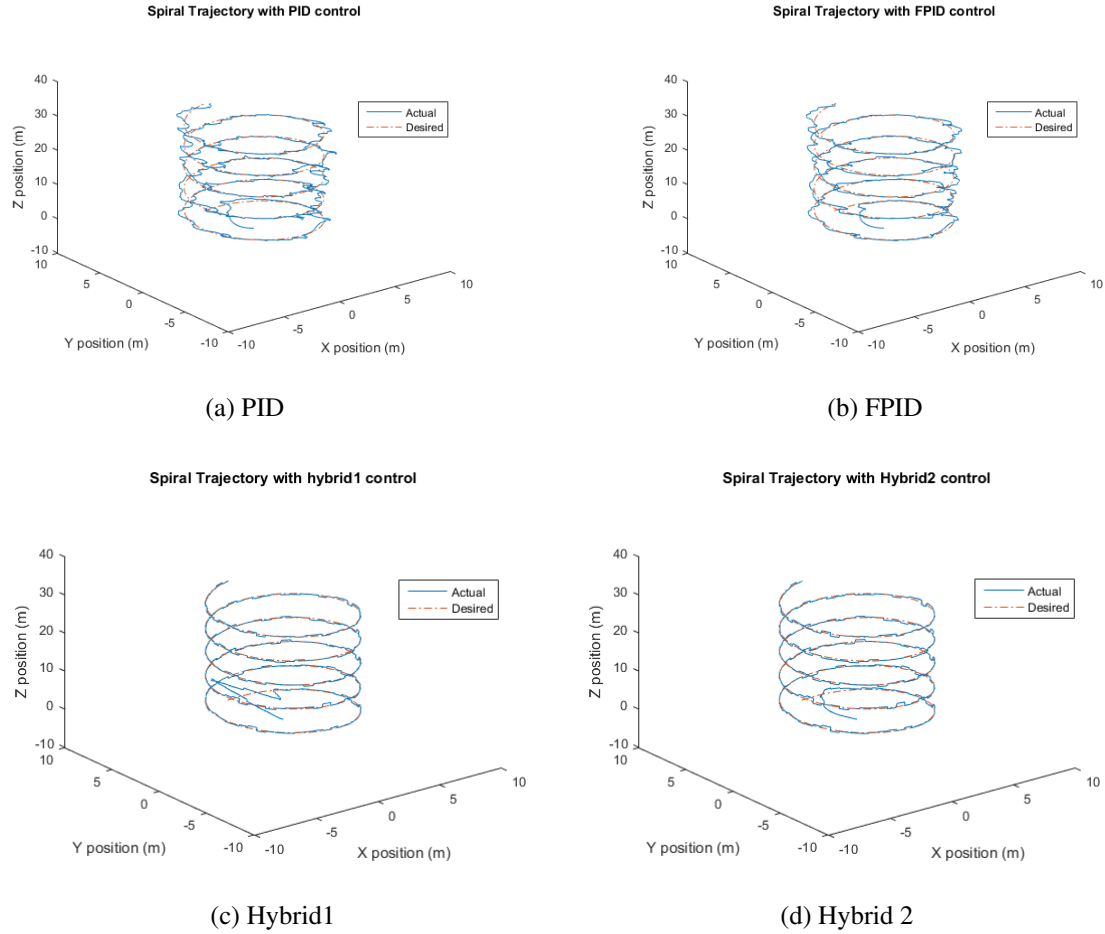


Figure 7.42: Spiral trajectory tracking with each control technique.



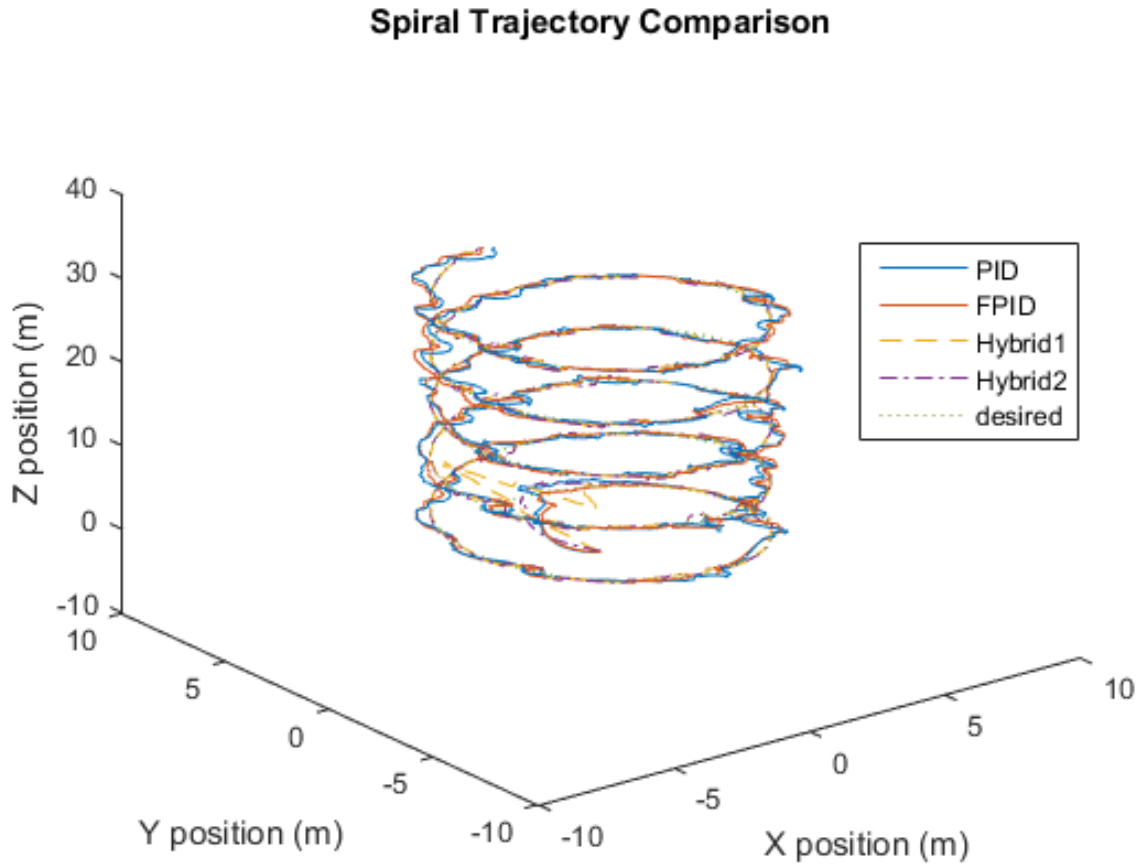


Figure 7.43: Spiral trajectory tracking comparison between all the control techniques

### B Random trajectory

A random trajectory is generated and it is given to the system as the desired trajectory. The simulation run time was set to 50 seconds. Figure 7.44 shows each of the developed control techniques ability to track the desired trajectory while Figure 7.45 shows a comparison between all the developed control techniques.

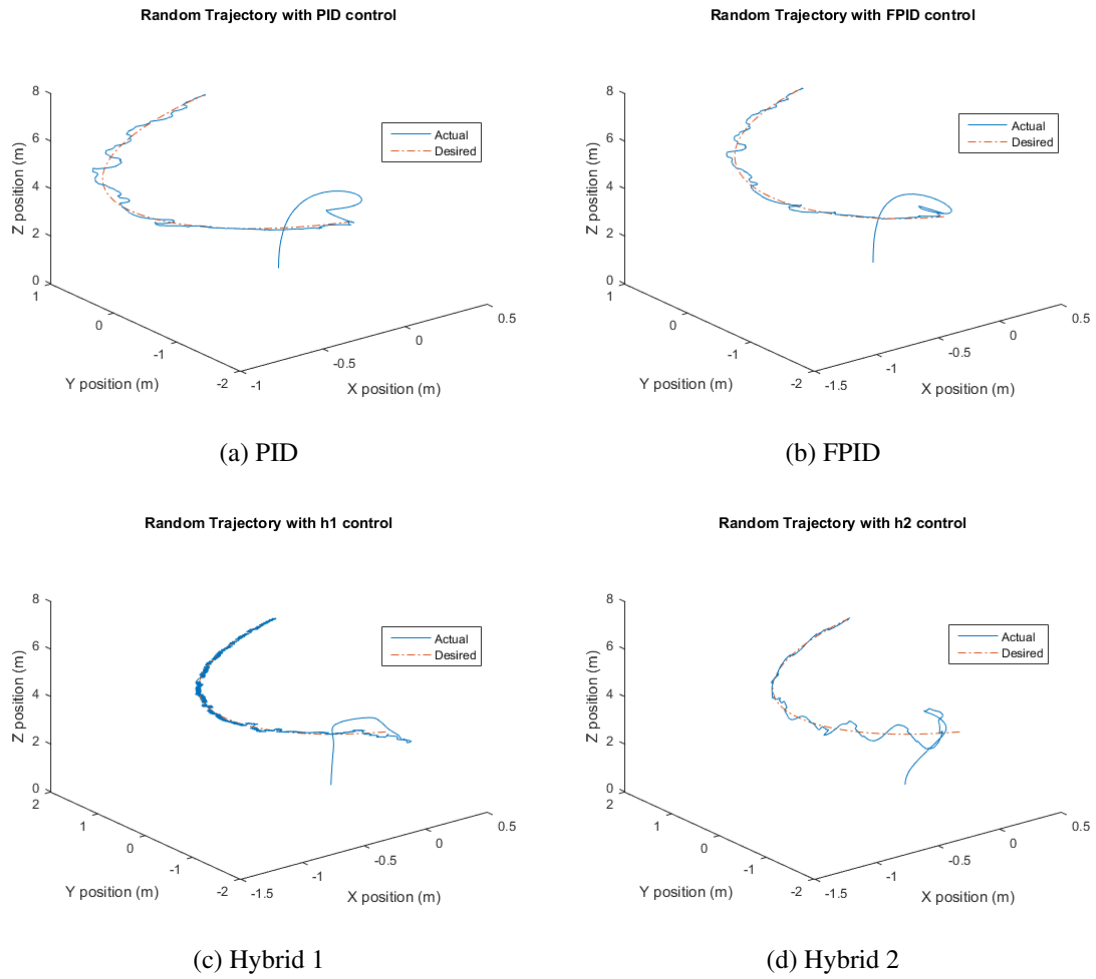


Figure 7.44: Random trajectory tracking with each control technique)

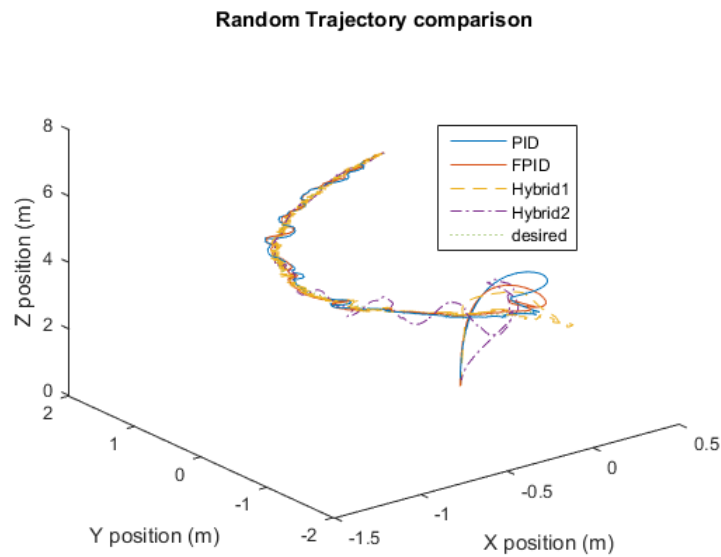


Figure 7.45: Random trajectory tracking comparison between all the control techniques

# Chapter 8

## Hardware Development and Implementation

A quadcopter was designed and implemented for the purpose of validating the results acquired by the simulation. This chapter discusses the developed hardware of the quadcopter. Available on shelf hardware modules are used in this research to develop the quadcopter. Steps of the physical integration of the modules to form the system are discussed in this chapter. Furthermore, the chapter introduces the calculation of the quadcopter's main parameters that are used to support the simulation and the control techniques development along with the real experimental testing.

### 8.1 Quadcopter Design and Requirements

The quadcopter was designed to fulfill the principle of self-contained flying operation that supports two modes of operational functions: autonomous and remote flying.

Before designing and developing the required quadcopter, it is necessary first to list the design constraints then, list the functional and specification requirements to help conclude the type and the technical specification of the required hardware. Accordingly, the following design constraints were considered,

- Lift a minimum of 1 kg payload,
- Robust against winds up to 15 km/h,
- Maximum flight speed speed of 15 m/s and maximum climbing rate of 10 m/s,
- $\psi$  is assumed to be zero to the coupling and complexity of the position equations of motions, and
- Sustain self contained flying for at least 10 minutes.

The required functional and technical requirements that support flying and navigation of

the quadcopter should cover:

- Attitude and Altitude control,
- Position control,
- Trajectory generation and tracking.
- Maneuverability and agility,
- Remote and autonomous modes of operation,
- Sensing Capabilities to measure altitude, attitude, position, orientation and speed, and
- Wireless communication between the quacopter and a ground station.

Accordingly, in order to fulfill such requirements, the developed quadcopter was designed to include the following hardware modules:

- Quadcopter frame structure.
- Brushless motor module:
  - Propeller,
  - Brushless DC motor, and
  - Electronic speed controller.
- Sensor module:
  - Global Positioning System (GPS) and Compass, and
  - Telemetry.
- Wireless Transceiver module.
- Quadcopter controller module.
- Battery module.

Figure 8.1 shows the layout of the designed quadcopter with the main hardware modules while Figure 8.2 shows the developed quadcopter with all hardware modules.

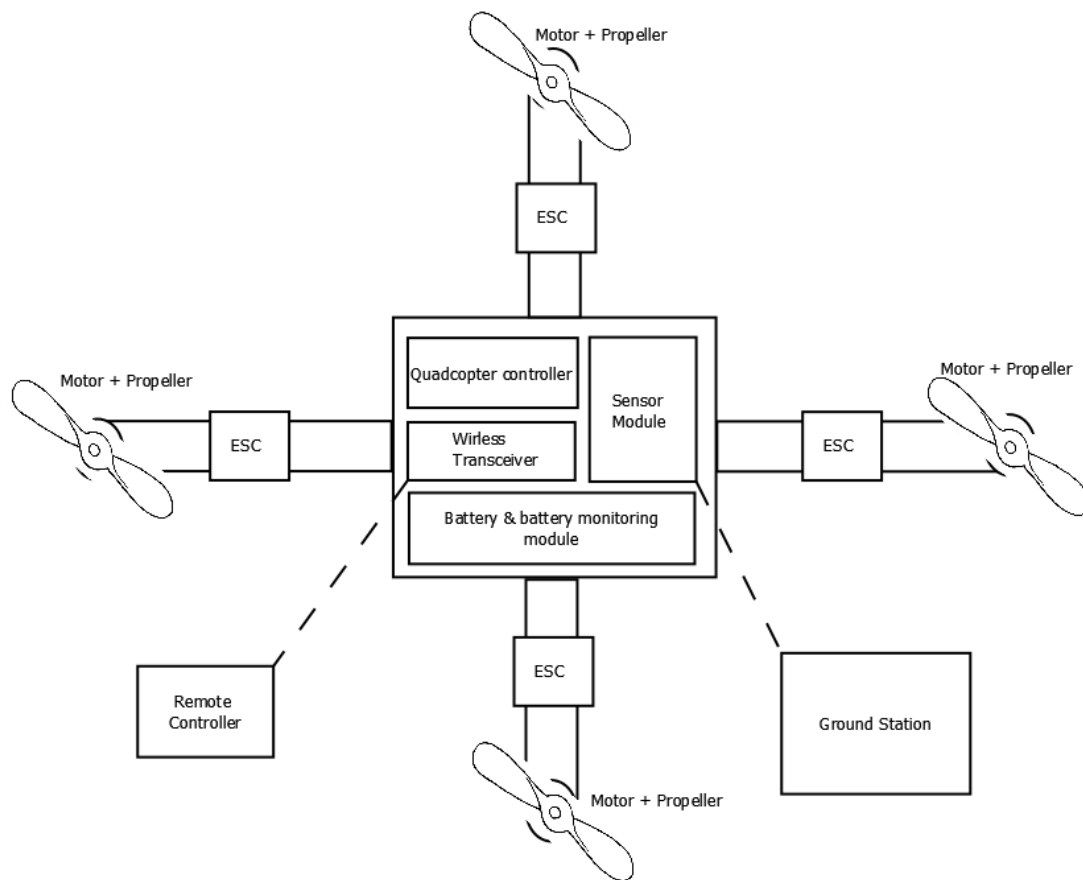


Figure 8.1: The layout of the quadcopter system.



Figure 8.2: The developed quadcopter.

## 8.2 Hardware Modules

### 8.2.1 Quadcopter frame structure.

The frame structure of the commercial quadrotor named Tarot FY650 Ironman model is used to form the body skeleton of the developed quadcopter. The frame structure is composed of four carbon fiber tubes that are connected to a central carbon fiber hub forming a plus shape. At each of the other ends of the carbon tubes there is a flight seat motor mount to host a motor-propeller assembly. The hub is made out of two carbon fiber plates of 163 mm x 145 mm each and a clearance distance between them of 25 mm. The specifications of the frame structure are listed in Table 8.1:

Table 8.1: Frame Specification

Parameter	Specification
Distance between the center of opposite corner motors	650 mm
Arm length	300 mm
Total weight of the frame structure (without motors)	476 g
Height from ground to lower plate of the hub	180 mm
Height from ground to the upper plate of the hub	220 mm
Arm tube diameter	16 mm
Arm tube weight (including the flight seat motor mount)	99 g per arm
Hub weight	80g

The frame structure after assembly is shown in Figure 8.3.



Figure 8.3: Frame specifications.

### 8.2.2 Brushless Motor Module

The brushless motor module consists of three main components that are shown below:

- Propeller,
- Brushless DC motor, and
- Electronic speed controller (ESC).

#### A Propeller

The propeller combined with the motor are responsible for achieving the desired lift force. The relation for calculating the lift force per motor with a propeller is,

$$f_{lift} = C_T \rho A_r r^2 (\Omega^2)$$

Where  $\Omega$  is the motor's RPM.  $\rho$ ,  $C_T$ ,  $A_r$  and  $r$  are the air density, the propellers lift coefficient, cross sectional area of the propeller's rotation and propellers diameter respectively. To achieve the required design constraints the propeller used along with the motor should produce the minimum desired lift force.

## B Brushless DC motor

The quadcopter's motor were chosen to fulfill the required payload of at least 1 kg and to achieve agility and maneuverability. The agility and maneuverability aims to generate a smooth flight and this demands that the total thrust of the motors to be at least 50% more than the total weight of the quadcopter.[40]

The quadcopter's total weight with all hardware modules is 1.2 kg, hence, to achieve this purpose the total thrust of the motors should cover,

- 50% more thrust than the total weight of the quadcopter which is equal to  $1200 + 600 = 1800g$ ,
- 1000 g of payload

Thus, the minimum total thrust the motor should produce is  $1800 + 1000 = 2800g$ . This means that each motor should produce at least  $2800/4 = 700g$ . Accordingly, The selected motor is the Emax BL2220/07 that is going to produce a thrust of 1300g in association with a propeller of size 10" diameter by 4.7" pitch. As a result, The total thrust of the selected motor is  $1300 * 4 = 5200g$  which is more than the required total thrust by 85%.

The specifications of the selected motor is shown in Table 8.2 and its picture is listed in Figure 8.4.

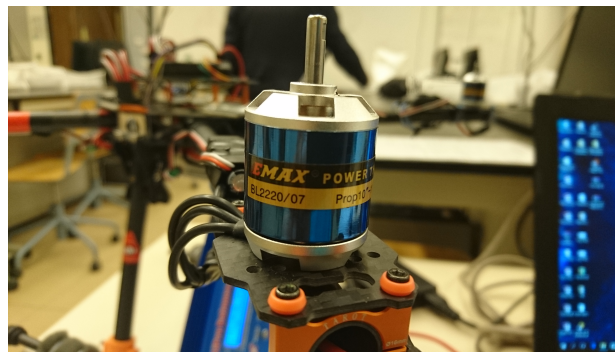


Figure 8.4: Emax BL2220 Motor.

## C Electronic Speed Controller (ESC)

The purpose of the ESC is to regulate the voltage for the brushless motor. The main requirement in the selection of the ESC is to deliver sufficient current for the DC brushless



Table 8.2: Motors Specification

Parameter	Specifications
Propeller size	10"x47"
Revolutions per minute per one volt (RPM/V)	1200 RPM
Maximum revolutions per minute (RPM max)	8560 RPM
Maximum current	25 A
Thrust (With a propeller of 10"*45")	1300 g
Dimensions of the motor size	22 mm height / 20 mm diameter
Motor shaft diameter	4 mm
Weight of one motor	85 g

motor to work between the minimum and maximum speed. The maximum current of the chosen motor is 25A to operate at 8560 RPM. Accordingly, the selected ESC is Turnigy K force series 40A is shown in Figure 8.5. The specification are listed in Table 8.3.

Table 8.3: Turnigy ESC specification

Parameter	Value
Weight	73 g
Size (length x width x height)	150 mm x 14 mm x 112 mm)
Current	40 A
Burst current	60A
BEC output (Battery Eliminator Circuit)	5.25V or 6V, 3A



Figure 8.5: Turnigy ESC.

### 8.2.3 Sensor module

Several types of sensors have been chosen to support navigation and control requirements.

These sensors include:

- GPS and Compass, and
- Telemetry.

#### A GPS and compass

The GPS is used to measure the longitude and latitude of the quadcopter. While a compass is used to measure the orientation of the quadcopter. The GPS module chosen is the Quanum GPS module shown in Figure 8.6. It has the Ubox LEA-6H GPS receiver and incorporates the HMC5883L digital compass.

The features of the selected GPS and compass are,

- Ublox LEA-6H module
- 5 Hz update rate
- Size: 27.5 x 27.5 x 7mm
- Rechargeable 3V lithium backup battery
- Low noise 3.3V regulator
- ArduPilot Mega (APM) and HKPilot Mega compatible 5-pin Molex connector
- Weight: 38g with case

- Configuration: Baud rate 38400, 5Hz



Figure 8.6: Quantum GPS module

## B Telemetry

The telemetry should be able to send the sensors data such as IMU, gyroscope and accelerometer to the controller and the ground station. Hence, The telemetry used is 3DR 915 Mhz shown in Figure 8.7. Having a dimension of 25.5 mm x 53 mm x 11 mm and a weight of 11.5 g



Figure 8.7: 3DR telemetry

### 8.2.4 Wireless transceiver

Based on the required design constraints, the wireless transceiver should have at least six channels. Four channels for the attitude and altitude inputs, and two channels for pre-determined flight modes. As a result, the RC chosen is the FlySky FS-i6-M2 shown in Figure 8.8, and its main features are:

- Six channels,
- RF range: 2.40-2.48Ghz, and
- Bandwidth: 500Khz.



Figure 8.8: FlySky FS-i6 RC.

### 8.2.5 Quadcopter controller

The controller of the quadcopter should be able to:

- Supervise all hardware modules,
- Gather sensory information,
- Execute the developed control techniques,

- Record flight data, and
- Communicate with the ground station.

Based on the aforementioned and the available controllers along with the required technical specifications asserted by other hardware modules and programming environment, the selected controller module is the ArduPilot Mega 2.6 (APM) shown in Figures 8.9 and 8.10 respectively. The ArduPilot Mega 2.6 is a complete open source autopilot system, it allows the user to turn any quadcopter capable of performing programmed missions with waypoints. The OS / Firmware for the APM 2.6 for MultiRotors is Arducopter.



Figure 8.9: APM 2.6.

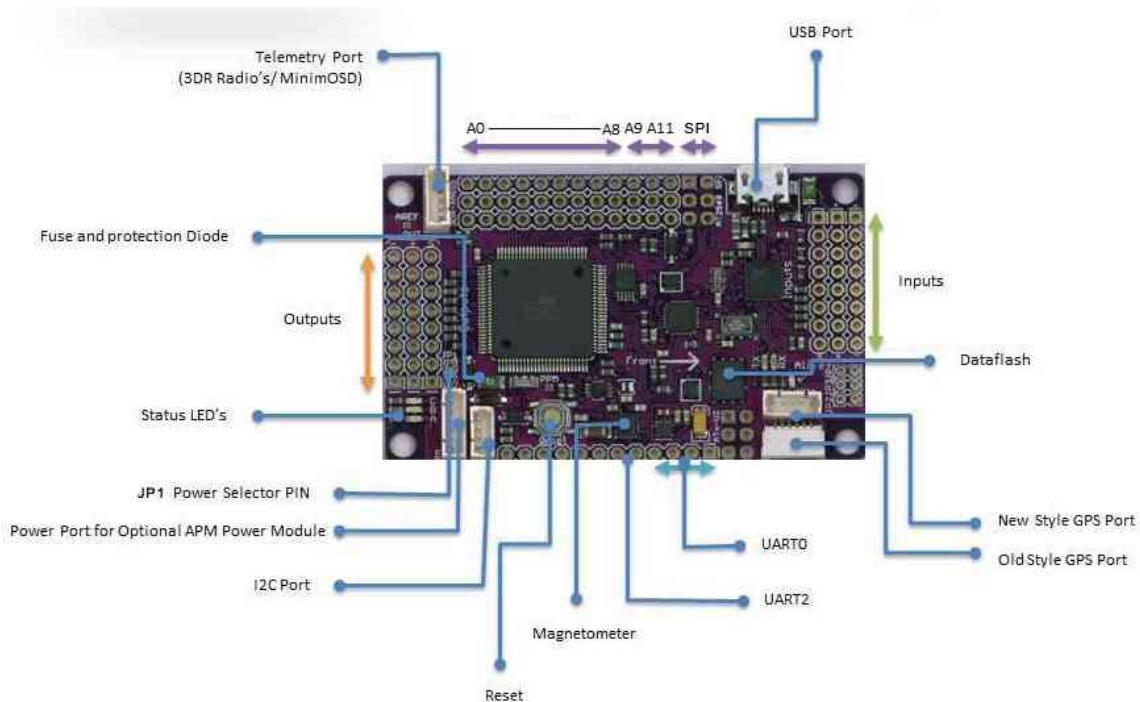


Figure 8.10: APM 2.6 board layout.

The list of features describing APM controller is:

- Arduino Compatible,
- 3-axis gyro, accelerometer and magnetometer, along with a high-performance barometer,
- Onboard 4 MP Dataflash chip for automatic data logging,
- InvenSense's 6 DoF Accelerometer/Gyro MPU-6000,
- Barometric pressure sensor, MS5611-01BA03, from Measurement Specialties, and
- Atmel's ATMEGA2560 and ATMEGA32U-2 chips for processing and USB functions.

The APM board specifications is shown in Table 8.4

Table 8.4: APM 2.6 Specification

Parameter	Value
Dimension	70.5x45x13.5mm
Weight	31g
Program Memory Type	Flash
Program memory	256
CPU speed	16 MIPS
Ram	8192 bytes
Data EEPROM	4096 bytes
Digital communication peripherals	4-UART, 5-SPI, 1-I2C
Capture/Compare/PWM Peripherals	4 Input Capture, 4 CCP, 16PWM
Timers	2 x 8-bit, 4 x 16-bit
Comparators	1
Temperature Range	-40 to 85 C
Temperature Range	-40 to 85 C
Operating Voltage Range	1.8 TO 5.5 v
Pin Count	100

### 8.2.6 Battery module

The battery module consists of:

- Battery,
- Battery monitoring unit, and
- Power distribution board.

#### A Battery

The battery is selected to fulfill,

- Sufficient discharge rate for the quadcopter with all the hardware modules, and
- Self contained flight of at least 10 minutes.

The motors are the main components that the battery will power, In addition all the other hardware modules. The maximum current for the motor is 25A. Thus, calculation for the discharge rate and flight time to identify the minimum requirements for the battery in terms of discharge rate and flight time is carried out respectively, The battery specifications is assumed to carry out the calculations of the charge rate and the flying time. The specifications assumed to have a battery capacity of 2200 mAh and maximum burst of 60c.

- Discharge rate

$$\text{Battery capacity} * \text{maximum burst} = \text{discharge rate} = 2200A * 60c = 132A \text{ for 10 seconds.}$$

- Flight time with an average constant current draw of 10A

$$\frac{2200mAh}{1000} / 10A = 0.22hours = 0.22 * 60 = 13.2 \text{ minutes.}$$

Thus a battery with the assumed specs is sufficient for the requirements. As a result, the battery chosen is a Wild Scorpion Li-Po Battery shown in Figure 8.11. The main features of the selected battery are:

- Capacity: 2200mAh
- Maximum burst (Discharge rating): 60 c
- Battery section: 3S (11.1V)
- Cell unit: 3 Cells
- Weight: 190 g
- Dimensions: 23.5 x 35 x 68 mm



Figure 8.11: Lithium Polymer Battery.

## B Battery monitoring unit

The battery monitoring unit is used to monitor the voltage level of the battery and to ensure that the quadcopter has sufficient amount of supplied power to support its flight. When the voltage is below the set value, the buzzer will beep and the LED light will flash. The battery monitor used to is shown in Figure 8.12.



Figure 8.12: Battery monitor module.



### C Power distribution board

This power distribution board shown in Figure 8.13 is ready wired and assembled. It is used with the quadcopter frame to distribute power from the flight battery to all relevant hardware modules.

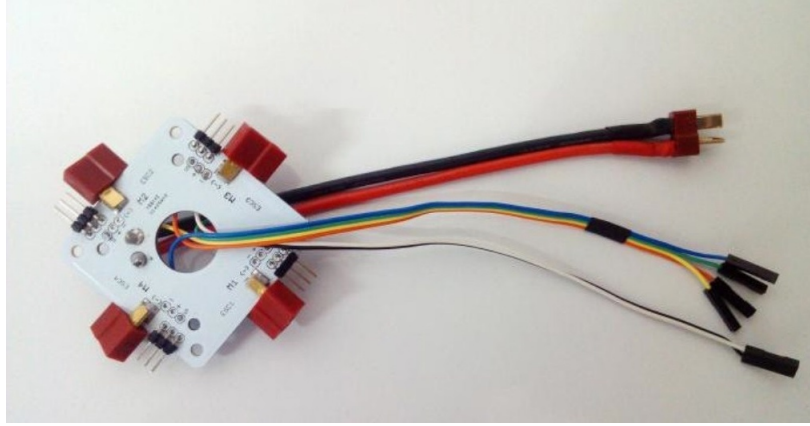


Figure 8.13: Power Distribution Board.

## 8.3 Quadcopter's Parameters Identification

### 8.3.1 Airframe

The total mass of the quadcopter is weighted, resulting in a total approximate mass of  $m = 1.2kg$ . The weight of each motor is  $m_r = 0.085kg$ . The moment of inertia of the quadcopter is calculated using the measured weights and dimensions of the assembled frame. For simplification the quadcopter is regarded as two perpendicular rods with one point mass on each edge representing the rotor's mass and a solid sphere in the middle. The arm length measured is  $l_a = 0.3m$ . All the quadcopter's mass excluding the rotor's is assumed to be homogeneously distributed inside the sphere of radius  $R = 7.5$  cm, centered at the origin of the axes. Knowing that the moment of inertia of a solid sphere around an axis is given by  $I_s = (2/5)m_s R^2$ . Where as for a point-mass distant  $l_a$  from the rotation axis is given by  $I_r = m_r l_a^2$ , and having the sphere mass as  $m_s = m - 4m_r = 1.16kg$ .

Hence, the moment of inertia around axes x and y can be calculated as:

$$I_{xx} = I_{yy} = 2(m_r l_a^2) + \frac{2}{5} m_s R^2 = 2.344 * 10^{-2} kgm^2 \quad (8.1)$$

For the Z-axis, the four rotors need to be considered, thus

$$I_{zz} = \sum_{j=1}^4 (m_r l_a^2 + \frac{2}{5} m_s R^2) = 3.333 * 10^{-2} kgm^2 \quad (8.2)$$

Table 8.5 summarizes the identified parameters for the quadrotor's airframe.

Table 8.5: Identified quadcopter airframe parameters.

Symbol	Value	Description
m	1.2 kg	Total quadcopter mass
$l_a$	0.3 m	rotor lever length to quadcopter's CG
$I_{xx} = I_{yy}$	$2.344 * 10^{-2} kgm^2$	quadcopter's moment of inertia around X and Y axes
$I_{zz}$	$3.333 * 10^{-2} kgm^2$	aircraft's moment of inertia around Z axis

### 8.3.2 Propellers

The propellers installed on the quadcopter have dimensions of  $10'' \times 4.7''$  in diameter and pitch respectively. A test stand experiment setup that rotates the propeller at a specified speed and measuring the thrust force generated is used to determine the propellers thrust coefficient,  $C_T$  and torque coefficient  $C_P$ .

The University of Illinois at Urbana-Champaign (UIUC) had performed a series of experiments as described in [41] and [42] to determine the performance of different small-scale propellers at low Reynolds number. The experimental results for the propeller of dimensions  $10'' \times 4.7''$  were summarized in Table 8.6, while the plots for the thrust and power coefficients against the propeller speed are plotted in Figures 8.14 and 8.15.

Table 8.6: (10 x 4.7) Propeller Thrust &amp; Power Coefficients at Different Speed.[41]

Propeller Rotational Speed (RPM)	Propeller Rotational Speed (rad/s)	Thrust Coefficient, $C_T$	Power Coefficient, $C_P$
2377	248.9189	0.1059	0.0431
2676	280.2301	0.1079	0.0437
2947	308.6091	0.1079	0.0437
3234	338.6637	0.1104	0.0444
3494	365.8908	0.1117	0.0450
3762	393.9557	0.1143	0.0460
4029	421.9159	0.1158	0.0466
4319	452.2846	0.1177	0.0474
4590	480.6637	0.1200	0.0484
4880	511.0324	0.1223	0.0494
5147	538.9926	0.1237	0.0500
5417	567.2669	0.1252	0.0508
5715	598.4734	0.1263	0.0513
5960	624.1297	0.1278	0.0520
6226	651.9852	0.1286	0.0524
6226	651.9852	0.1286	0.0531

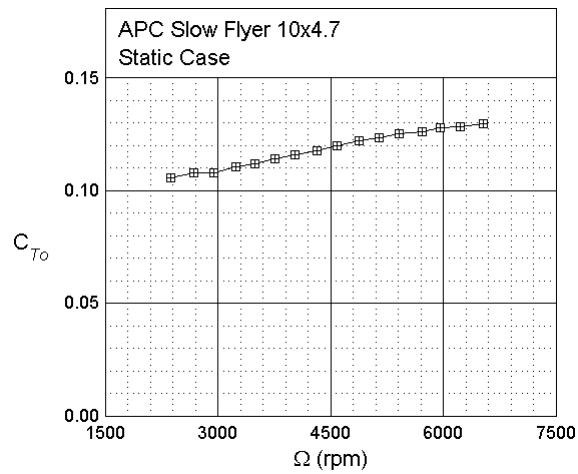


Figure 8.14: Thrust Coefficient vs Propeller Speed Plot.[41]

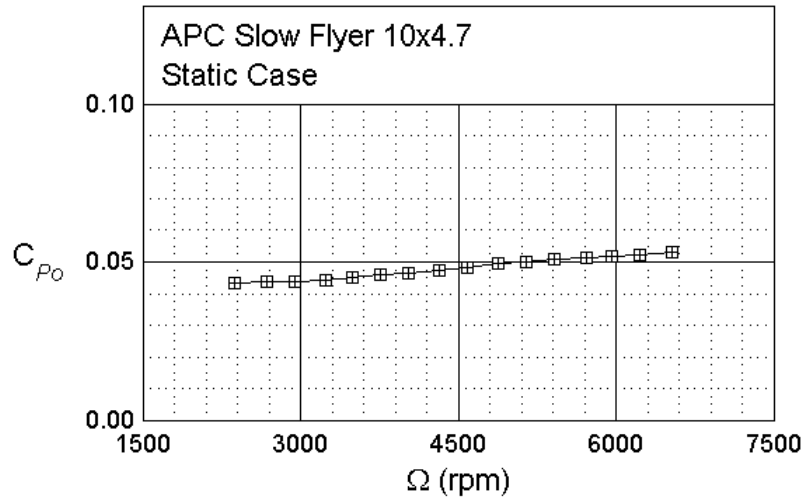


Figure 8.15: Power Coefficient vs Propeller Speed Plot. [41]

### 8.3.3 Motors

The data sheet provided by the motor's manufacturer were adequate to interpret all the parameters needed for the motor dynamics. The parameters are listed in Table 8.7.

Table 8.7: Motor Parameters

Symbol	Value	Description
$m$	0.075 kg	Motor mass
RPM/V	1200 RPM	Revolution per minute per volt
RPM	8560 RPM	Revolution per minute
Thrust	1300 g	Thrust
$R_a$	63 mohms	Armature current
$i_{amax}$	28 A	Maximum armature current
$h * l$	20 * 22 mm	Dimensions
$j_r$	$6 * 10^{-5} kg.m^2$	Motor's inertia

## 8.4 Real Time Control Implementation

A free open source application is used to work specifically with the selected quadcopter controller called mission planner[43]. The mission Planner is a ground control station for the quadcopter. It is compatible with Windows only and it can be used as a configuration utility for the quadcopter to support:

- Load the firmware into the APM that controls the quadcopter,
- Setup, configure, and tune the quadcopter for optimum performance,
- Plan, save and load autonomous missions into the controller on Google maps,
- Download and analyze mission logs created by the APM,
- Monitor the quadcopter status while in operation,
- Record telemetry logs, and
- View and analyze the telemetry logs.

However, the mission planner application uses PID as the control technique for the quadcopter and cannot be changed, only the controller gains could be tuned. Thus, it won't be possible to apply the control techniques developed early on for the quadcopter. As a result, another mean is used to implement the developed control techniques using Simulink and MATLAB. A feature called Run On Target Hardware (ROTH) is used to achieve all the mission planner features and at the same time be able to upload any control technique. as long as the available controller memory can accommodate it.

### 8.4.1 Run on target hardware

The ROTH feature converts the model built on Simulink using a compiler into embedded code and then upload it to the APM. However in order to be able to build a model. Simulink blocks for all the components available in the APM board should be available.

Fortunately, [44] made a library specifically for the APM board named Ardupilot 2.0 Simulink blockset. The library consists of the following blocks:

- Sensor Blocks
  - Inertial Measurement Unit (IMU),
  - Barometric Pressure Sensor,
  - Magnetometer,
  - GPS, and
  - Pilot Probe.
- Receiver Input / Servo Output Blocks
  - RC Read, and
  - RC Write.
- Timing Blocks
  - Clock, and
  - Real Time Monitor.
- Data Output Blocks
  - Telemetry
  - USB Output, and
  - Flash Memory Chip.
- LED Blocks

APM Simulink library is installed on the available MATLAB while setting all the necessary compilers and configuration. A Simulink model is built with the help of the APM Multicopter Development Kit [45]. The model developed consists of three subsystems. the first subsystem is for the hardware inputs, the second subsystem is for the control technique model built on Simulink in the previous chapters and the last subsystem consists of

the hardware outputs. The main scheme is shown in Figure 8.16. The hardware subsystems are shown in Figures 8.17 and 8.19 respectively. Moreover, the control subsystem is shown in Figure 8.18 which consists of a control block that hosts the control technique. Using the model developed, it is possible to replace the control block with one control technique with another block that includes the desired control technique. This would make it easy for any other user to test different control techniques on the quadcopter. However, after compiling the code the APM board cannot run it if it is computationally expensive or if the size of the compiled code is larger than the available memory storage of the APM board.

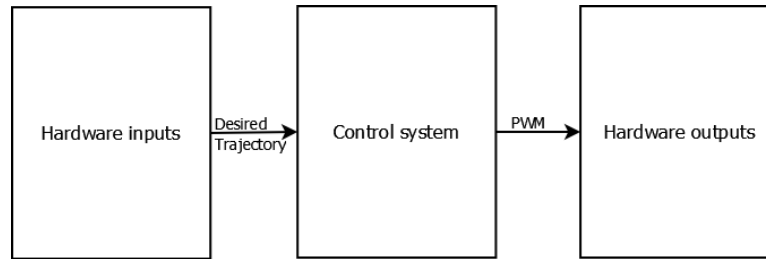


Figure 8.16: Main simulink model layout for hardware implementation.

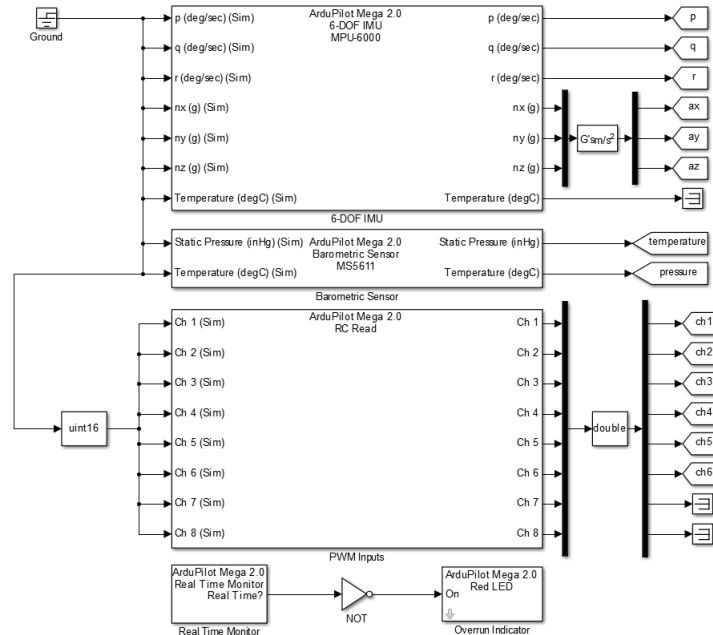


Figure 8.17: Hardware inputs subsystem.

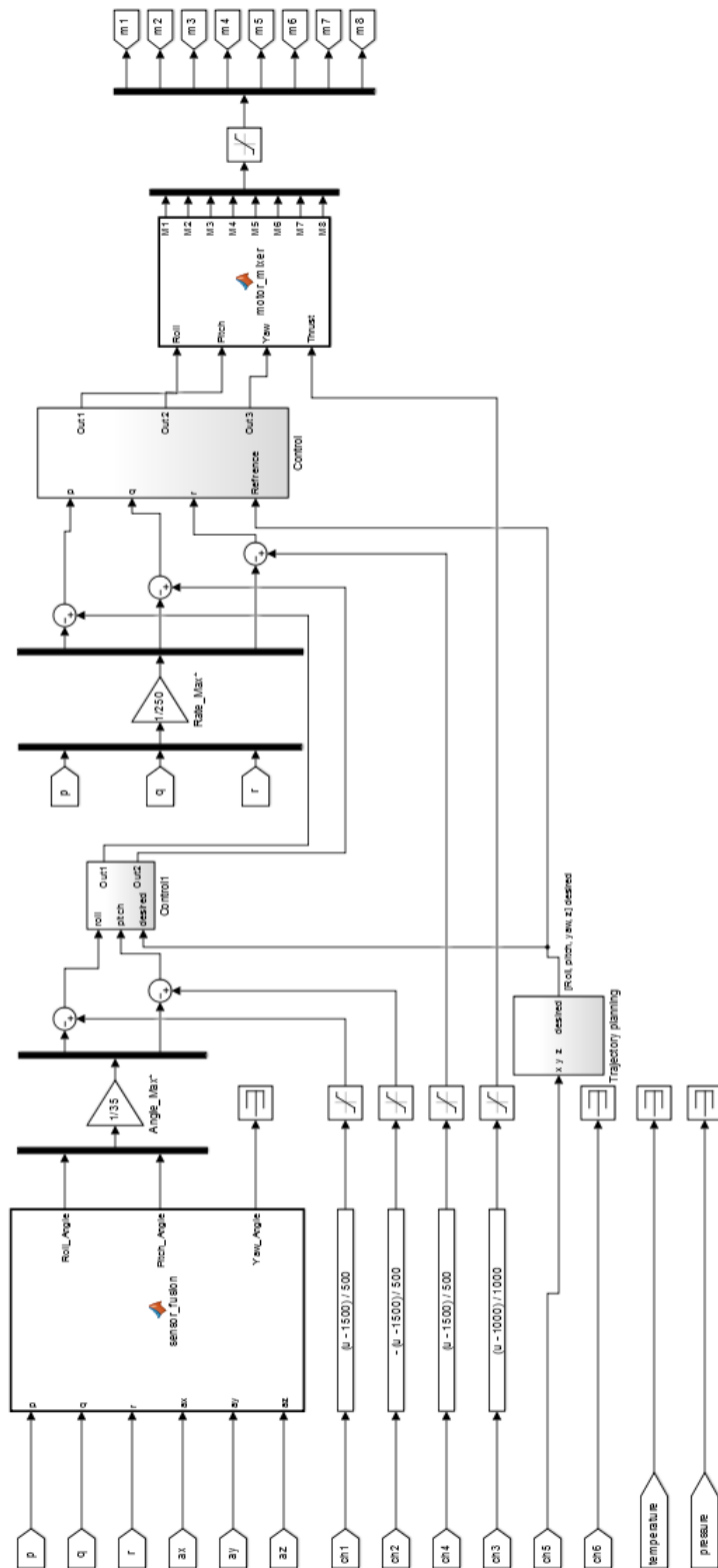


Figure 8.18: Control subsystem.



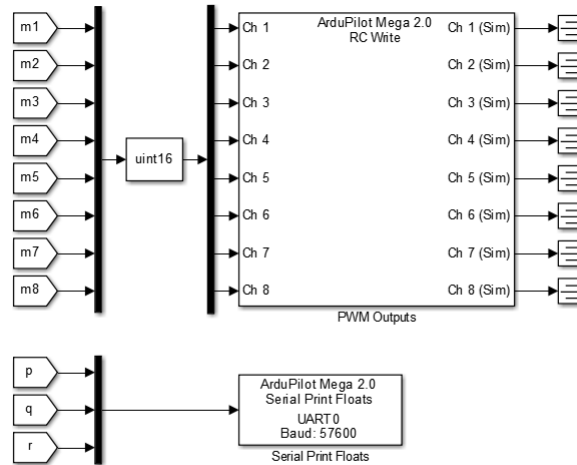


Figure 8.19: Hardware outputs subsystem.

## 8.5 Quadcopter Flight Test

After downloading the Simulink based controller application successfully on the APM, a flight test was set up to assess the performance of the quadcopter. The flight test was carried out at the American University in Cairo Campus. Figure 8.20 shows a photo of the quadcopter test.

### 8.5.1 Flight test procedure

The main objective of the flight test is to analyze the transient and steady-state performance of the three attitude controllers developed while holding a desired altitude. The steps for the flight test are further elaborated below:

#### A Step 1

Set up the ground computer to record the flight test data by connecting the telemetry radio.



Figure 8.20: Quadcopter flight test.

## **B Step 2**

Once the ground computer is setup to record the flight data, the motors on the quadcopter are armed and the copter flies using the RC while holding the altitude. During the flight of the quadcopter, roll, pitch and yaw commands are transmitted from the RC transmitter to obtain the attitude controllers response.

## **C Step 3**

When the quadcopter is landed, The motors are disarmed. A MATLAB log file is then generated for data analysis using the flight data recorded. The MATLAB file is generated using the mission planner software.

### **8.5.2 Flight test results: Calm weather**

A series of flight tests were conducted for each inner control loop of each on-board control technique to determine the performance of the quadcopter.

### A Roll angle

The plots of the flight data for the actual roll angle and the desired roll angle for a duration of 35 seconds and for each of developed controller are shown in Figures 8.21, 8.22 and 8.23 respectively. The dashed line represents the desired roll magnitude command transmitted by the RC transmitter and the solid line represents the actual magnitude of roll measured by the APM IMU.

- PID

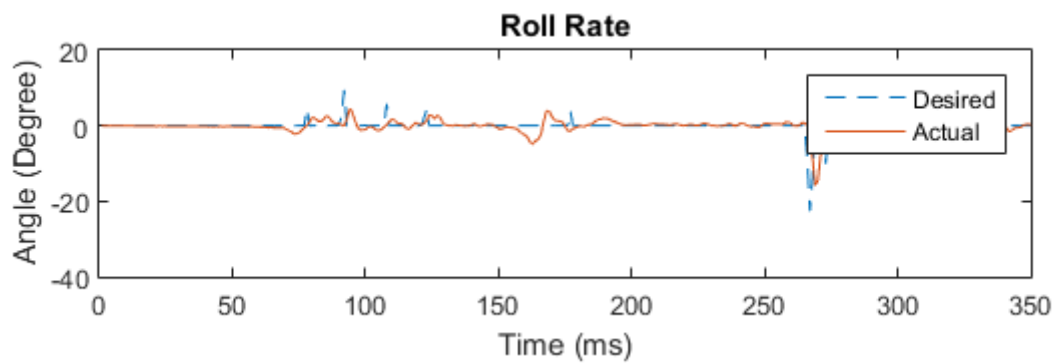


Figure 8.21: Roll plot (PID)

- FPID

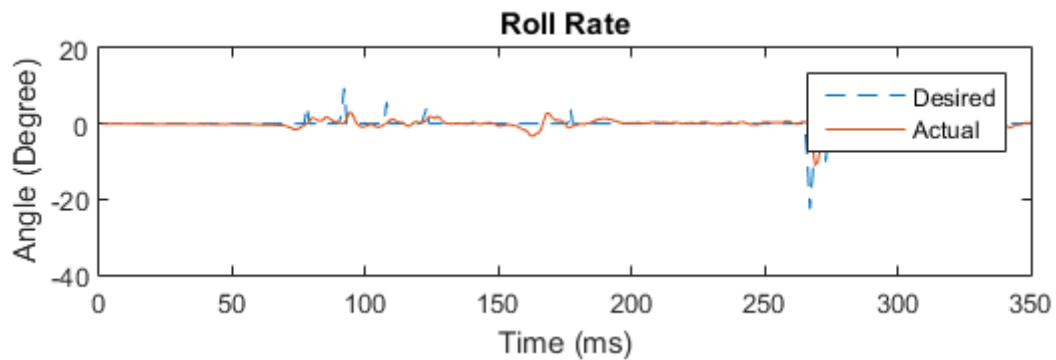


Figure 8.22: Roll plot (FPID)

- Hybrid control strategy

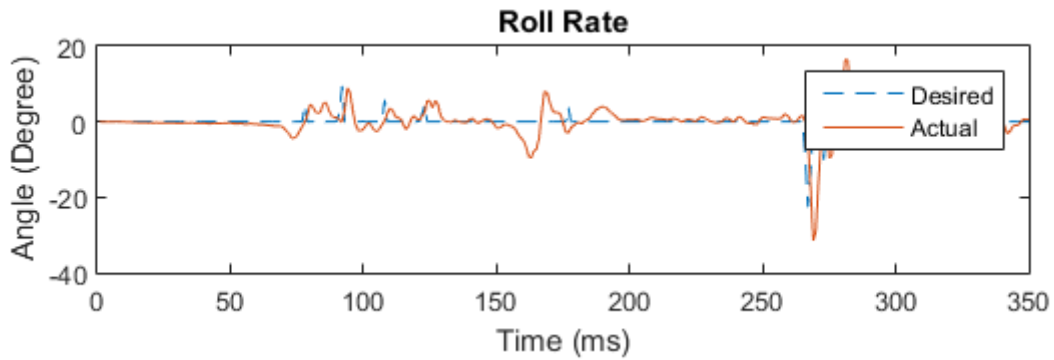


Figure 8.23: Roll plot (FL)

### B Pitch angle

The plots of the flight data for the actual pitch angle and the desired pitch angle for a duration of 35 seconds and for each of the developed control techniques are shown in Figures 8.24, 8.25 and 8.26 respectively. The dashed line represents the desired pitch magnitude command transmitted by the RC transmitter and the solid line represents the actual magnitude of pitch measured by the APM IMU.

- PID

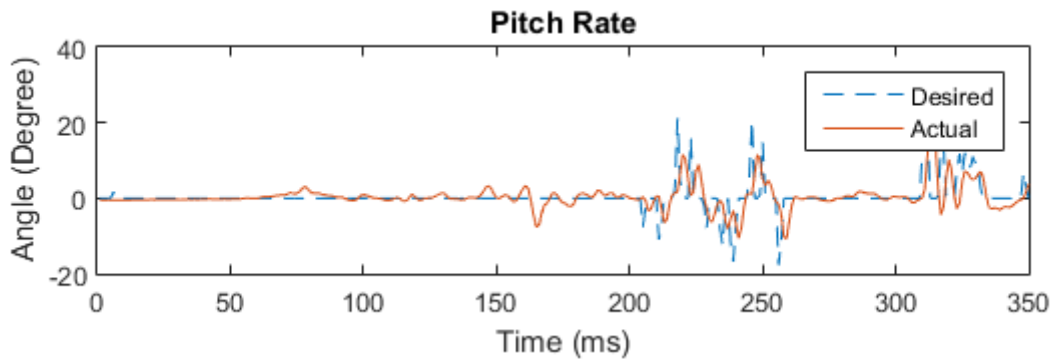


Figure 8.24: Pitch plot (PID)

- FPID

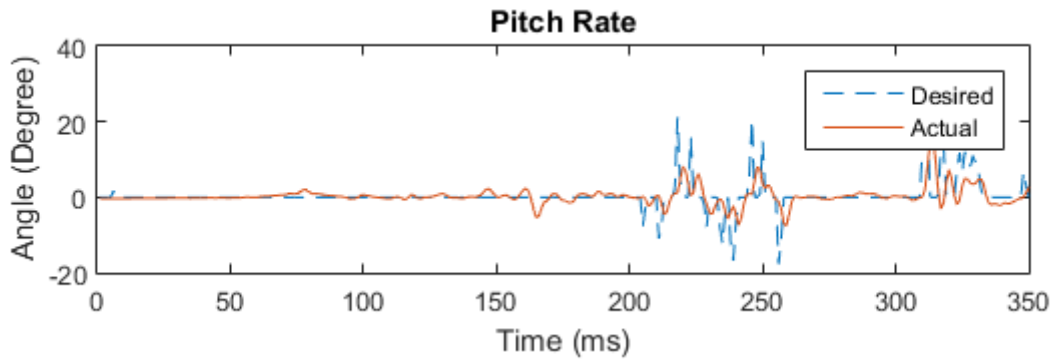


Figure 8.25: Pitch plot (FPID)

- Hybrid control strategy

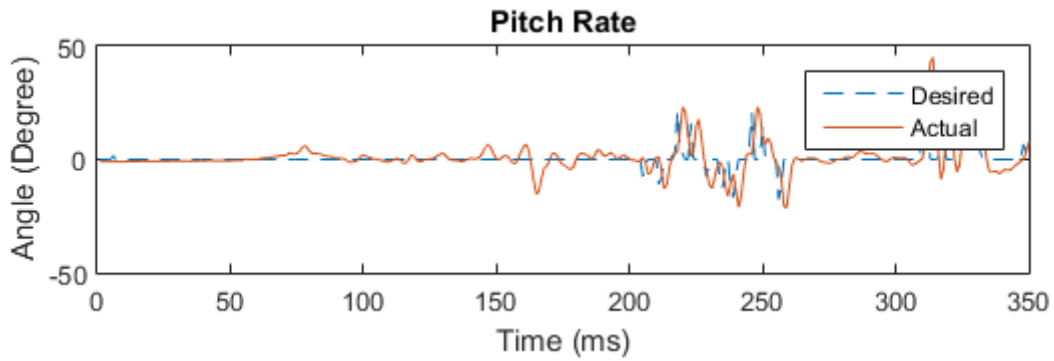


Figure 8.26: Pitch plot (FL)

### C Yaw angle

The plots of the flight data for the actual yaw angle and the desired yaw angle for a duration of 35 seconds and for each of the developed control techniques are shown in Figures 8.27, 8.28 and 8.29 respectively. The dashed line represents the desired yaw magnitude command transmitted by the RC transmitter and the solid line represents the actual magnitude of yaw measured by the APM IMU.

- PID

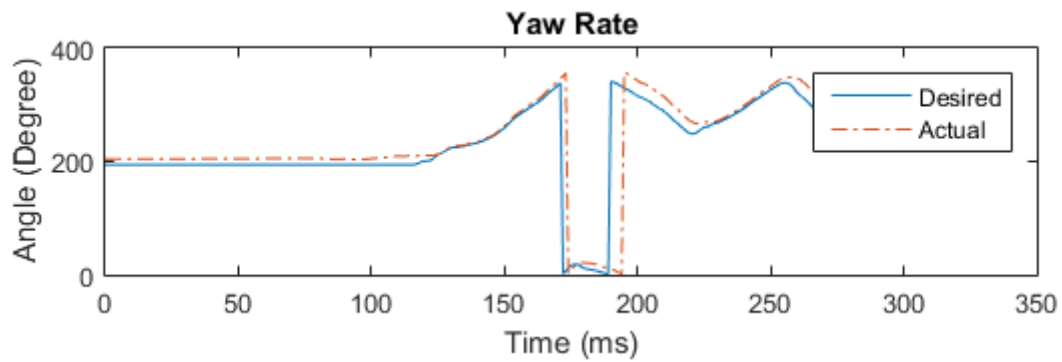


Figure 8.27: Yaw plot (PID)

- FPID

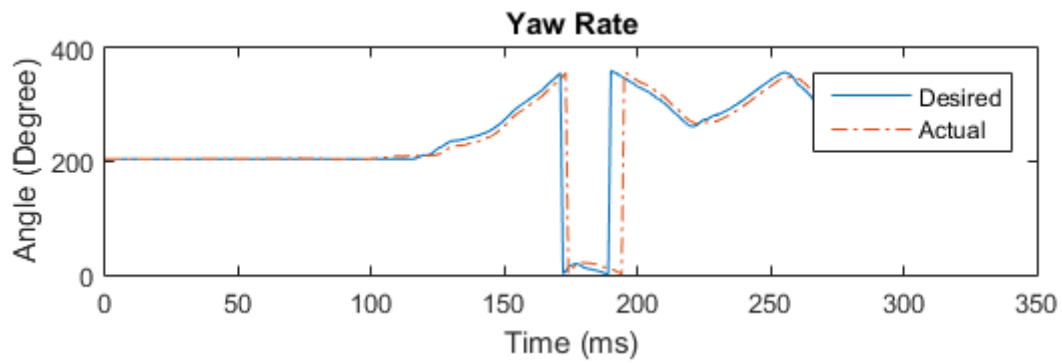


Figure 8.28: Yaw plot (FPID)

- Hybrid control strategy

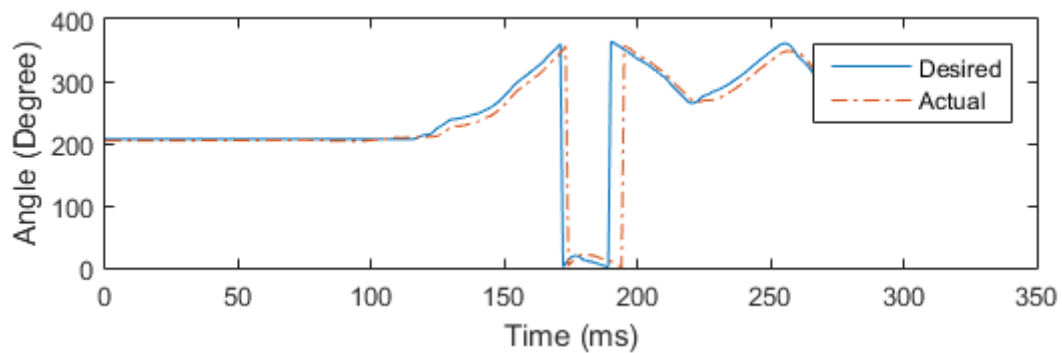


Figure 8.29: Yaw plot (FL)

The controllers shows the ability of tracking the desired inputs. However, some delay

is shown. The delay is because of the APM control board slow performance, this delay was random in each flight test.

### 8.5.3 Flight test results: Windy weather

A series of flight tests were conducted for each inner control loop of each on-board control technique to determine the performance of the quadcopter in windy weather, similar to case two in the simulation and testing chapter.

#### A Roll angle

The plot of the flight data for the actual roll angle and the desired roll angle for a duration of 35 seconds for each of developed controller are shown in Figure 8.30. Where the hybrid control strategy and the FPID controller shows robust results, while the PID controller shows very large perturbations.

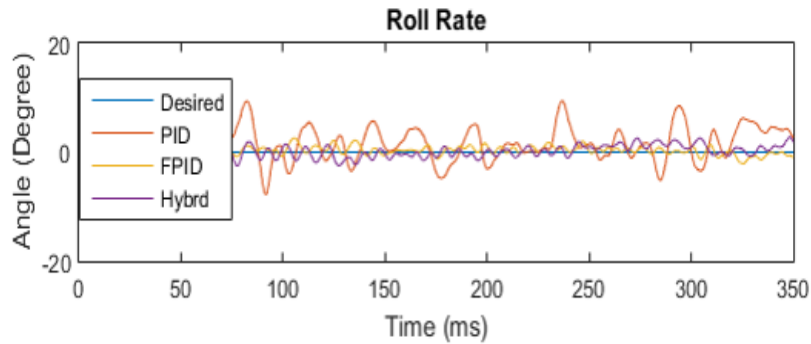


Figure 8.30: Roll plot (comparison)

#### B Pitch angle

The plots of the flight data for the actual pitch angle and the desired pitch angle for a duration of 35 seconds for each of the developed control techniques are shown in Figure 8.31.

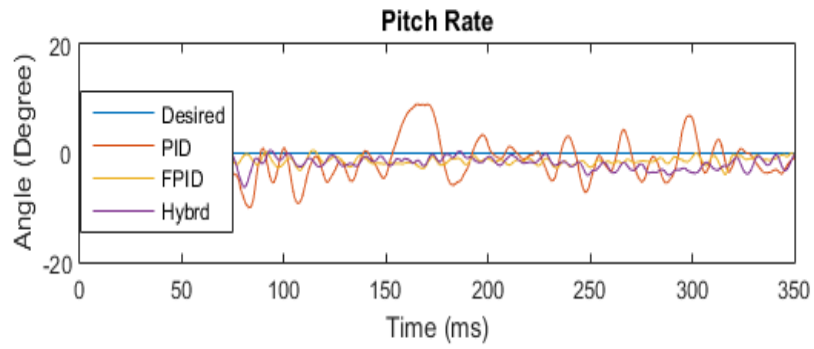


Figure 8.31: Pitch plot (comparison)

### C Yaw angle

The plots of the flight data for the actual yaw angle and the desired yaw angle for a duration of 35 seconds for each of the developed control techniques are shown in Figure 8.32

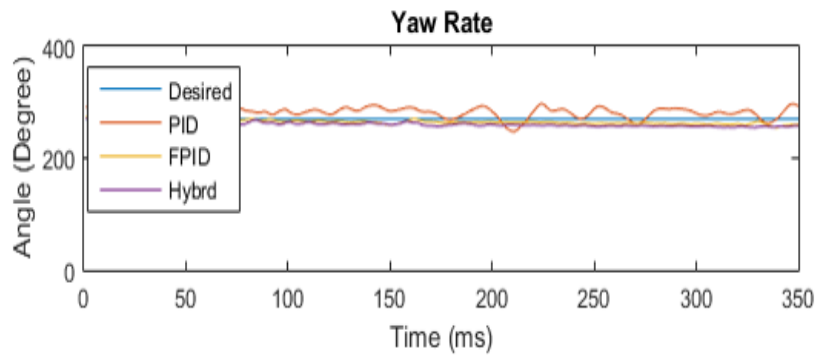


Figure 8.32: Yaw plot (comparison)



# Chapter 9

## Conclusion and Future Work

The content of this thesis shows that it successfully achieved the stated objectives to a high extent. A mathematical model representing the dynamics of a quadcopter has been derived in association with the stated assumptions. The developed mathematical model enables the user to gain understanding of how a quadcopter is behaving in real time. In addition, it has been shown that the derived mathematical model of quadcopter's flight dynamics can be represented using Simulink. Moreover, the same approach has been used to represent a designed controller that helps to stabilize and control a quadcopter's flight.

Furthermore, in order to ensure a feasible desired trajectory before tracking it, a trajectory planning algorithm has been developed and tested successfully. Subsequently, a simulation environment with a friendly GUI has been developed to simulate the dynamics of a quadcopter's mathematical model and then to use it as a test bed to validate the developed control techniques with and without the effect of wind disturbance and measurement noise.

Three control techniques were developed and tested. These control techniques are:

- PID control,
- FPID control, and
- Hybrid control strategy. This includes two configurations:
  - (a) A sliding mode control is used for the outer control loop while for the inner control loop two control techniques are used to realize it: FPID control with state coordinates transformation and a state feedback control.
  - (b) FPID control is used for the outer control loop while for the inner control loop

two control techniques were developed and implemented to realize it using the same formulation as in (a) above.

The quadcopter with each control technique has been tested using the simulation environment under different operational conditions. The results in terms of tracking a desired trajectory shows the robustness of the first configuration of the control techniques within the hybrid control strategy under the presence of wind disturbance and measurement noise compared to all the other techniques developed. Then, the second configuration of the control techniques came second in terms of results quality. The third and fourth results in the sequence shown by the fuzzy scheduled PID and the standard PID respectively.

Finally, the thesis demonstrated the successful implementation of a flight controller developed using Simulink on board the APM board by using the Simulink build function, while eliminating the need to be proficient in high level programming language. The controllers developed were tested in term of the attitude stability and compared the result against the simulation results. Finally, validating the simulation results on a real system, a quadcopter has been successfully designed, implemented and tested. The developed control techniques were tested using the implemented quadcopter and the results were demonstrated and compared with the simulation results.

The methods outlined in this thesis provide control engineering students with an enabling tool to implement their flight controller design that was developed in Simulink directly onto an APM board. The flight controller design can subsequently be put through an actual flight test and the performance of the flight controller can be readily analyzed with the ground control station.

As a continuation for this thesis, the following areas can be considered for future research works:

- Instead of the FPID controller used with the state feedback linearization technique, alternatives to the controller design (e.g. LQR,  $H_\infty$ ) can be implemented on the

quadcopter.

- APM autopilot board was used as the controller for the quadcopter in this thesis. A more reliable and faster controller could be used such as Pixhawk autopilot.
- The GPS signal detected by the quadcopter within the indoor facility was not always in good condition. Therefore, an image motion capture system could be integrated to provide the quadcopter with accurate position and attitude indoor and outdoor.
- The APM board uses barometric sensor for measuring the altitude which is not always reliable since the air pressure changes due to many factors such as wind and propellers thrust. Thus, a range finder could be implemented on the quadcopter such as a laser sensor or sonar which would make the altitude measurement much more accurate.

# **Appendices**

# Appendix A

## Feedback Linearization

The feedback linearization is based on two operations: [33] [34]

- nonlinear change of coordinates;
- After the feedback linearization, the input-output model is linear in the new set of coordinates.

Most feedback linearization approaches are based on input-output linearization or state-space linearization. In the input-output linearization approach, the objective is to render a linear input-output map from the new input  $v$  to the actual output  $y$ . A controller is then designed for the linearized input-output model. A linear controller is then synthesized for the linear input-state model. However, this approach may lead to a complex controller design task because the map between the transformed inputs and the original outputs  $y$  is generally nonlinear. Feedback linearization produces a linear model by the use of nonlinear coordinate transformations and nonlinear state feedback. In some applications, the control objectives can be achieved with a nonlinear static feedback control law of the form,

$$u = \alpha(x) + \beta(x).v \tag{A.1}$$

where  $\alpha$  is an  $m$ -dimensional vector of nonlinear functions and  $\beta$  is an  $m \times m$  matrix of nonlinear functions. For some processes, it is not possible to satisfy the control objective with a static control and hence a dynamic state feedback control law must be implied.

The suggested feedback linearization technique in this thesis is based on the input-output linearization explained below.

Consider a system with state  $x \in \mathbb{R}^n$ , input  $u \in \mathbb{R}$  and output  $y \in \mathbb{R}$  whose dynamics are given by: [33] [34]

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x),\end{aligned}\tag{A.2}$$

We focus now on a single-input, single output system, i.e.,  $u, y \in \mathbb{R}$ . The derivative of the output  $y$  can be expressed as:

$$\dot{y} = \frac{\partial h}{\partial x}[f(x) + g(x)u]\tag{A.3}$$

The derivative of  $h$  along the trajectory of the state  $x$  is known as the Lie Derivate and is denoted as:

$$\dot{y} = \frac{\partial h}{\partial x}[f(x) + g(x)u] = L_f h(x) + L_g h(x)u\tag{A.4}$$

If on the first derivative  $L_g h(x) = 0$ , we have

$$\dot{y} = y^1 = L_f h(x)\tag{A.5}$$

Note that, in this case, the output  $y$  remains independent of input  $u$ . However, further higher order derivatives can be considered, specifically,

$$y^{(2)} = L_f^2 h(x) + L_g L_f h(x)u,\tag{A.6}$$

$$y^{(3)} = L_f^3 h(x) + L_g L_f^2 h(x)u,\tag{A.7}$$

$$y^{(i)} = L_f^i h(x) + L_g L_f^{i-1} h(x)u,\tag{A.8}$$

and if for a certain  $L_g L_f^{i-1} h(x)u \neq 0$ , then the Equation A.8 can be linearized with full state feedback by,

$$u = \frac{1}{L_g L_f^{i-1} h(x)}(-L_f^i h(x) + v)\tag{A.9}$$

In the state region where the inverse  $\frac{1}{L_g L_f^{i-1} h(x)}$  exists the feedback linearized model becomes,

$$y^{(i)} = v \quad (\text{A.10})$$

The value  $i$  is defined to be the relative degree of the system. The resulting linear dynamic system defined in Equation A.10 can be stabilized by a standard linear control technique and consists of a set of  $i - 1$  integrators up to the required output  $y$ . Moreover, with this linearization a linear controller can be designed such that the overall system can be proven to be exponentially stable [33][34]. The concepts used for SISO systems can be also extended to MIMO systems. In the MIMO case, we consider square systems (that is systems with the same number of inputs and outputs) of the form,

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x), \end{aligned} \quad (\text{A.11})$$

where  $x \in \mathbb{R}^n$  is a state vector,  $u \in \mathbb{R}^m$  is a control input vector (of components  $u_i$ ),  $y \in \mathbb{R}^m$  is a vector of system output (of components  $y_i$ ),  $f, h$  are smooth vector fields, and  $G$  in an  $n \times m$  matrix whose columns are smooth vector fields  $g_i$ . Input-output linearization of MIMO systems is obtained similarly to the SISO case, by differentiating the outputs  $y_i$  until the inputs appear. Assume that  $r_i$  is the smallest integer such that at least one of the inputs appears in  $y^{(r_i)}$  then,

$$y^{(r_i)} = L_f^{r_i} h_i(x) + \sum_{j=1}^m L_{g_j} L_f^{r_i-1} h_i(x) u_j, \quad (\text{A.12})$$

with  $L_{g_j} L_f^{r_i-1} h_i(x) \neq 0$  for some  $x$ . Performing the above procedure for each output  $y_i$  yields

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} = L_f^{r_1} h_1(x) \\ \vdots \\ = L_f^{r_m} h_m(x) \end{bmatrix} + E(x)u, \quad (\text{A.13})$$

yields  $m$  equations of the simple form

$$y_i^{(r_i)} = v_i \quad (\text{A.14})$$

Since the input  $v_i$  only affects the output  $y_i$ , Equation A.13 is called a decoupling control law, and the invertible matrix  $E(x)$  is called decoupling matrix of the system. The system in Equation A.11, is then said to have relative degree  $(r_1, \dots, r_m)$ , and the scalar  $r = r_1 + \dots + r_m$  is called the total relative degree of the system. An interesting case corresponds to the total relative degree being  $n$ . In this case, there are no internal dynamics. With the control law in the form of Equation A.13, we thus obtain an input-state linearization of the original nonlinear system. With the equivalent inputs  $v_i$  designed as in the SISO case, both stabilization and tracking can then be achieved for the system without any worry about the stability of the internal dynamics.



# Bibliography

- [1] A. Juniper. "The Complete Guide to Drones. Hachette UK." (2015).
- [2] S. Bouabdallah. "Design and Control of Quadrotors with Application to Autonomous Flying" (2007).
- [3] E. Abbasi and M. Mahjoob. "Controlling of quadrotor UAV using a fuzzy system for tuning the PID gains in hovering mode" (2015).
- [4] S. Bouabdallah A. Noth and R. Siegwart. "PID vs LQ control techniques applied to an indoor micro quadrotor. Presented at Intelligent Robots and Systems" (2004).
- [5] K. Runcharoon and V. Srichatrapimuk. "Sliding mode control of quadrotor" (2013).
- [6] T. Luukkonen. "Modelling and control of quadcopter. Independent Research Project in Applied Mathematics" (2011).
- [7] H. Bouadi S. S. Cunha A. Drouin and F. Mora-Camino. "Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking" (2011).
- [8] I. Palunko and R. Fierro. "Adaptive control of a quadrotor with dynamic changes in the center of gravity" (2011).
- [9] I.D. Cowling O.A. Yakimenko J.F. Whidborne and A.K. Cooke. "A Prototype of an Autonomous Controller for a Quadrotor UAV. European Control Conference" (2007).
- [10] R. Xu and U. Ozguner. "Sliding mode control of a quadrotor helicopter. In Decision and Control, 2006 45th IEEE Conference on (pp. 4957-4962). IEEE." (2006).
- [11] R. Ashton and M. Maggiore. "Sliding mode control of quadrotor" (2013).
- [12] M. Tahar K. M. Zemalache and A. Omari. "Control of an under-actuated X4-flyer using integral backstepping controller" (2011).
- [13] X.Huo M. Huo and H.R. Karimi. "Attitude Stabilization Control of a Quadrotor UAV by Using Backstepping Approach. Mathematical Problems in Engineering, 2014, 1-9.," (2014).

- [14] S. A. Raza and W. Gueaieb. "Intelligent Flight Control of an Autonomous Quadrotor" (2010).
- [15] M. S. Shaikh. "Quadrocopter fuzzy flight controller" (2011).
- [16] M. Santos V. Lopez and F. Morata. "Intelligent Fuzzy Controller of a Quadrotor. Proceedings of the 2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Hangzhou," (2010).
- [17] H. Kong R. Hercus and K. Ho. "Control of an unmanned aerial vehicle using a neuronal network. Presented at Computational Intelligence, Cognitive Algorithms, Mind, and Brain" (2013).
- [18] H. Boudjedir F. Yacef O. Bouhali and N. Rizoug. "Adaptive neural network for a quadrotor unmanned aerial vehicler" (2012).
- [19] T. Dierks and S. Jagannathan. "Output Feedback Control of a Quadrotor UAV Using Neural Networks. IEEE Transactions on Neural Networks, 21, 50-66. <http://dx.doi.org/10.1109/TNN.2010.2010.2010> (2010).
- [20] T. Madani and A. Benallegue. "Adaptive control via backstepping technique and neural networks of a quadrotor helicopter" (2008).
- [21] S. Zeghlache D. Saigaa K. Kara A. Harrag and A. Bouguerra. "Backstepping Sliding Mode Controller Improved with Fuzzy Logic: Application to the Quadrotor Helicopter. Archives of Control Sciences, 22, 315-342." (2012).
- [22] A. Das F.L. Lewis and K. Subbarao. "Sliding mode approach to control quadrotor using dynamic inversion. In Challenges and Paradigms in Applied Robust Control. InTech." (2011).
- [23] T. Bresciani. "Modelling, Identification and Control of a Quadrotor Helicopter" (2008).
- [24] S. Bouabdallah P. Murrieri and R. Siegwart. "Design and control of an indoor micro quadrotor." (2004).
- [25] <http://documentation.quest3d.com/index.php?Quaternions> (2012).

- [26] V. James Diebel. "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors" (2006).
- [27] K. J. Astrom and B. Wittenmark. "(2013) Computer-controlled systems: theory and design. Courier Corporation." ().
- [28] C. Bohn and D.P. Atherton. "An analysis package comparing PID anti-windup strategies. IEEE Control Systems Magazine, 15(2), 34-40." (1995).
- [29] H. Martine G. Fernandez-Anaya G. Ferreira D.Flores-Godoy and A. Lopez-Gonzalez. "Trajectory tracking of a quadcopter uav with optimal translational control. In 11th IFAC Symposium on Robot Control (pp. 228-233)." (2015).
- [30] A. Das K.Subbarao and F. Lewis. "Dynamic inversion with zero-dynamics stabilisation for quadrotor control. IET control theory & applications, 3(3), 303-314." (2009).
- [31] A. Benallegue A. Mokhtari and L. Fridman. "Highorder slidingmode observer for a quadrotor UAV. International journal of robust and nonlinear control 18(45), 427-440." (2008).
- [32] Altug E. Ostrowski and R. Mahony. "Control of a quadrotor helicopter using visual feedback. In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on (Vol. 1, pp. 72-77). IEEE." (2002).
- [33] Slotine. J. J. E. and Li W. (1991). "Applied nonlinear control (Vol. 199 No. 1)." (Englewood Cliffs, NJ: prentice-Hall.).
- [34] H. Khalil. "Nonlinear Systems. Prentice-Hall, New Jersey." (1996).
- [35] Slotine J. J. E. and W. Li. "Applied nonlinear control (Vol. 199, No. 1). Englewood Cliffs, NJ: Prentice hall." (1991).
- [36] Young K. V. I. Utkin and U. Ozguner. "A control engineer's guide to sliding mode control. In Variable Structure Systems, 1996. VSS'96. Proceedings., 1996 IEEE International Workshop on (pp. 1-14). IEEE." (1996).
- [37] Lee H. and Utkin V. I. "Chattering suppression methods in sliding mode control systems. Annual Reviews in control, 31(2), 179-188.." (2007).

- [38] N. Hogan. "Adaptive control of mechanical impedance by coactivation of antagonist muscles. *IEEE Transactions on Automatic Control*, 29(8), 681-690." (1984).
- [39] T. Flash and N. Hogan. "The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7), 1688-1703." (1985).
- [40] L. Chovaneca A. Chovancov T. Ficoa and P. Hubinska. "Mathematical Modelling and Parameter Identification of Quadrotor (a survey)" (2014).
- [41] J. B. Brandt and M. S. Selig. "Propeller performance data at low Reynolds numbers, in 49th AIAA Aerospace Sciences Meeting, Orlando, pp. 2011-1255." (2011).
- [42] R. W. Deters G. K. Ananda and M. S. Selig. "Reynolds number effects on the performance of small-scale propellers, in 32nd AIAA Applied Aerodynamics Conf., Atlanta, GA, pp. 2014-2151." (2014).
- [43] Michael Osborne. "<http://ardupilot.org/planner/>" (2010).
- [44] R. F. Hartley F. Hugon P. Anderson and H. Moncayo. "Development and Flight Testing of a Model Based Autopilot Library for a Low Cost Unmanned Aerial System. In AIAA Guidance, navigation and control conference." (2013, August.).
- [45] A. Polak. "PX4 development kit for Simulink. [Online]." (2014, Jan. 21.).
- [46] D. Lee H. J. Kim and S. Sastry. "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter" (2009).
- [47] A. I. Alshbatat A. Khamaisa and M. Khreisat. "Adaptive control system for an autonomous quadrotor unmanned aerial vehicle" (2012).
- [48] E. Reyes-Valeria. "LQR Control for a Quadrotor using Unit Quaternions: Modeling and Simulation" (2013).
- [49] E. Fresk and G. Nikolakopoulos. "Full Quaternion Based Attitude Control for a Quadrotor" (2013).
- [50] Jack B. Kuipers. "Quaternions and rotation sequences" (1999).
- [51] Yan-Bin Jia. "Quaternions and Rotations" (2008).

- [52] V. Mistler A. Benallegue and N. K. M'sirdi. "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In Robot and Human Interactive Communication 2001. Proceedings. 10th IEEE International Workshop on (pp. 586-593). IEEE" (2001).
- [53] A. Mokhtari and A. Benallegue. "Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on (Vol. 3, pp. 2359-2366). IEEE." (2004).
- [54] A. Benallegue A. Mokhtari and L. Fridman. "Highorder slidingmode observer for a quadrotor UAV. International journal of robust and nonlinear control 18(45), 427-440." (2008).